



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Entrenamiento de modelos de Aprendizaje Automático para
generación de mapas de Uso y Cobertura del Suelo
(LULC) utilizando datos de satélite

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Esteve Molner, Borja

Tutor/a: Monserrat Aranda, Carlos

Cotutor/a externo: JAUME CATANY, RAFAEL

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Entrenamiento de modelos de Aprendizaje
Automático para generación de mapas de
Uso y Cobertura del Suelo (LULC)
utilizando datos de satélite

Trabajo Fin de Grado

Grado en Ciencia de Datos

Autor: Borja Esteve

Tutor: Carlos Montserrat

Cotutores: Noelia Abascal y Rafael Catany

2022-2023

Resumen

En el ámbito de la teledetección y el procesamiento de imágenes satelitales, este trabajo de fin de grado se centra en la creación de un modelo avanzado de segmentación semántica. El objetivo principal es desarrollar un enfoque de aprendizaje automático que permita generar mapas precisos de Uso y Cobertura del Suelo (LULC de sus siglas en inglés) a partir de datos capturados por satélites equipados con sensores ópticos [1], como Sentinel 2. Un aspecto clave del proyecto es la implementación de soluciones de Deep Learning, específicamente la arquitectura VGG16 Unet [2], la Attention Unet [3] y el Swin Transformer [4].

Lo que buscamos es crear un algoritmo de aprendizaje profundo que genere mapas LULC con capacidad para adaptarse a nuevas zonas geográficas sin requerir un proceso completo de reentrenamiento. Esta característica es esencial para lograr una aplicación eficiente en diversas ubicaciones y contextos. La colaboración con Albavalor, una *start up* del ámbito de explotación de datos de satélite para consultoría medio ambiental y climática, agrega un componente práctico y aplicado al proyecto. Este trabajo se alinea con iniciativas anteriores, como el proyecto Coastal Erosion [5] realizado por ARGANS, una empresa asociada con Albavalor. La experiencia previa en el ámbito de la erosión costera complementa la creación de modelos LULC con datos de teledetección y promueve la utilidad real de los resultados.

En resumen, este estudio pretende crear un modelo de segmentación semántica basado en Deep Learning que puede generar mapas de Uso y Cobertura del Suelo a partir de datos satelitales. La innovación radica en la implementación de técnica de Machine Learning para mejorar la metodología de la generación de LULC clásica y ampliar su adaptabilidad a nuevas áreas geográficas, reduciendo la necesidad de reentrenamiento completo. Colaborando con Albavalor y aprovechando la experiencia adquirida en proyectos anteriores, este trabajo se enmarca en la intersección entre la investigación avanzada y la aplicación práctica para abordar desafíos importantes en la teledetección y la monitorización del entorno terrestre y marítimo costero.

Palabras clave: LULC, segmentación semántica, Deep Learning, teledetección, redes convolucionales, redes neuronales, mecanismos de atención, Transformers.

Abstract

In the field of remote sensing and satellite image processing, this undergraduate thesis focuses on the creation of an advanced semantic segmentation model. The main objective is to develop a machine learning approach that allows for the generation of accurate Land Use and Land Cover (LULC) maps from data captured by Sentinel 2 satellites equipped with optical sensors [1]. A key aspect of the project is the implementation of Deep Learning solutions, specifically the VGG16 Unet [2] architecture, Attention Unet [3], and the Swin Transformer [4].

What we aim to achieve is the creation of a deep learning algorithm that can generate LULC maps capable of adapting to new geographical areas without requiring a complete retraining process. This feature is essential for achieving efficient application in various locations and contexts. Collaboration with Albavalor, a startup in the field of satellite data exploitation for environmental and climate consultancy, adds a practical and applied component to the project. This work aligns with previous initiatives, such as the Coastal Erosion project [5] carried out by ARGANS, a company associated with Albavalor. Previous experience in the field of coastal erosion complements the creation of LULC models with remote sensing data and promotes the real-world utility of the results.

In summary, this study aims to create a semantic segmentation model based on Deep Learning that can generate Land Use and Land Cover (LULC) maps from satellite data. The innovation lies in the implementation of machine learning techniques to enhance the traditional LULC generation methodology and expand its adaptability to new geographical areas, reducing the need for complete retraining. By collaborating with Albavalor and leveraging the experience gained in previous projects, this work falls at the intersection of advanced research and practical application to address important challenges in remote sensing and the monitoring of terrestrial and coastal maritime environments.

Keywords : LULC, semantic segmentation, Deep Learning, remote sensing, convolutional networks, neural networks, attention mechanisms, Transformers.

Índice General

1. Introducción.....	6
1.1. Motivación.....	7
1.1.1. Proyecto 'Coastal Erosion from Space'	7
1.1.2. LULC.....	8
1.1.3. IOTA.....	9
1.1.4. Limitaciones de IOTA	10
1.2. Objetivos	11
1.3. Impacto esperado	12
1.4. Metodología	13
1.5. Estructura de la memoria.....	15
2. Estado del arte	17
2.1. Clasificación LULC.....	17
2.2. Crítica al estado del arte.....	17
2.3. Propuesta.....	18
3. Contexto	19
3.1. Teledetección.....	19
3.1.1. Introducción	19
3.1.2. Principios fundamentales	20
3.1.3. Preprocesamiento de datos en teledetección.....	26
3.1.4. Desafíos y limitaciones.....	26
3.2. Deep Learning	26
3.2.1. Redes neuronales	27
3.2.2. Función discriminante lineal	28
3.2.3. Perceptrón multicapa.....	28
3.2.4. Función de activación.....	29
3.2.5. Estimación de parámetros.....	29
3.2.6. Redes convolucionales	31
3.2.7. Attention Gate.....	34
3.2.8. Transformers	36
3.2.9. Self Supervised Learning	40
3.3. Segmentación semántica	41

4.	Análisis del problema	43
4.1.	Análisis de los datos	43
4.1.1.	Conjunto de datos IOTA, CLC y Sentinel 2.	43
4.1.2.	Conjunto de datos Open Sentinel Map	47
4.2.	Análisis del marco legal y ético	49
4.3.	Solución propuesta	50
5.	Descripción de modelos.....	54
5.1.	VGG16 Unet	54
5.2.	Attention Unet	56
5.3.	Swin Transformer.....	57
6.	Recursos utilizados	59
6.1.	Herramientas tecnológicas	59
6.1.1.	Software.....	59
6.1.2.	Hardware	60
7.	Experimentación y resultados.....	61
7.1.	Métricas utilizadas	61
7.2.	Selección de hiperparámetros.....	63
7.2.1.	Vgg16.....	63
7.2.2.	Attention.....	64
7.2.3.	Swin Transformer	65
7.3.	Estrategia, entrenamiento y validación	65
7.4.	Resultados	66
8.	Conclusiones.....	78
8.1.	Conclusiones	78
8.2.	Reproducibilidad	79
8.3.	Relación con los estudios cursados	79
8.4.	Trabajos futuros	80
	Bibliografía.....	81
	Anexos	96



1. Introducción

En un mundo en constante evolución, impulsado por avances tecnológicos y científicos, la teledetección y el procesamiento de imágenes satelitales han emergido como herramientas fundamentales para comprender y monitorear nuestro entorno terrestre. En este contexto, este trabajo de fin de grado se adentra en el ámbito de la teledetección y se enfoca en la creación de un modelo avanzado de segmentación semántica, que busca mejorar la forma en que se obtienen mapas de Uso y Cobertura del Suelo (LULC, siglas en inglés Land Use and Land Cover) a partir de datos capturados por satélites.

El objetivo central de esta investigación radica en la creación de un enfoque de aprendizaje automático que tenga la capacidad de generar mapas de Uso y Cobertura del Suelo con una notable flexibilidad para adaptarse a áreas geográficas previamente no abordadas por el modelo. Esta característica es esencial ya que reduce significativamente la necesidad de llevar a cabo un proceso completo de reentrenamiento del sistema, lo que a su vez agiliza y simplifica la aplicación de la metodología en diversas regiones geográficas, sin importar cuán nuevas o desconocidas sean para el modelo. Para lograr este objetivo, se emplean soluciones de Deep Learning, específicamente las arquitecturas VGG16 Unet [2], Attention Unet [3] y Swin Transformer [4]. Lo que distingue este enfoque de otros trabajos [6][7][8] es que va más allá de su capacidad para generar mapas precisos. Se destaca por su capacidad de adaptarse a nuevas zonas geográficas sin necesidad de someterse a un proceso completo de reentrenamiento. Esta característica es crucial para lograr una aplicación eficiente y efectiva en diversos contextos y ubicaciones geográficas.

Un aspecto destacado que amplía aún más la relevancia y aplicabilidad de este trabajo es la colaboración con Albavalor, una empresa española de reciente creación en el campo de la teledetección. Esta asociación aporta un componente práctico y aplicado al proyecto, conectando la investigación avanzada con las demandas y desafíos reales de la industria. Además, esta iniciativa encuentra resonancia en proyectos anteriores, como el proyecto Coastal Erosion [9] llevado a cabo por ARGANS, una empresa asociada con Albavalor. La experiencia acumulada en el estudio de la erosión costera complementa la creación de modelos LULC y amplía el potencial de aplicación de los resultados obtenidos.

En resumen, este trabajo de fin de grado persigue la creación de un modelo de segmentación semántica basado en Deep Learning que trasciende los límites de la precisión convencional en la generación de mapas de Uso y Cobertura del Suelo a partir de datos satelitales. La capacidad de adaptarse a nuevas áreas geográficas sin requerir un proceso exhaustivo de reentrenamiento añade un nivel de versatilidad que lo distingue. Al colaborar con Albavalor y aprovechar la experiencia acumulada en proyectos previos, este trabajo se sitúa en la intersección entre la investigación avanzada y la aplicación práctica, abordando desafíos fundamentales en la teledetección y la monitorización del entorno terrestre.

1.1. Motivación

Este Trabajo de Fin de Grado se origina en un profundo interés por abordar un desafío crítico a través de la ciencia de datos y la teledetección. La elección de esta temática se centra en la necesidad apremiante de encontrar soluciones efectivas para problemas ambientales y urbanos en la actualidad.

El proyecto se enfoca en la creación de un modelo avanzado de segmentación semántica, capaz de generar mapas detallados de Uso y Cobertura del Suelo. Este desafío técnico es apasionante y conlleva la responsabilidad de transformar datos de imágenes satelitales en información valiosa.

Hoy en día, enfrentamos desafíos globales de gran envergadura, como el cambio climático y la gestión de recursos naturales limitados. La teledetección se alza como una herramienta esencial para abordar estos problemas. Este proyecto no solo representa un reto técnico, sino también una oportunidad concreta para contribuir a la toma de decisiones informadas en la gestión territorial y la conservación de recursos naturales.

La colaboración con Albavalor, una entidad líder en teledetección, añade un nivel adicional de estímulo a este trabajo. La conexión entre el ámbito académico y las necesidades del mundo real resalta aún más la importancia de resolver estos desafíos contemporáneos a través de nuestro proyecto.

En resumen, este proyecto es el núcleo de nuestra búsqueda por abordar problemas ambientales y urbanos cruciales en la sociedad actual. La creación de un modelo de segmentación semántica es el pilar central de esta iniciativa, aprovechando la ciencia de datos y la teledetección para generar soluciones efectivas y contribuir al bienestar de nuestra sociedad y el planeta.

1.1.1. Proyecto ‘Coastal Erosion from Space’

ARGANS Ltd, una empresa líder a nivel mundial en la monitorización de cambios costeros, está desarrollando un servicio global para la vigilancia de la erosión costera, evaluación del riesgo ambiental e investigación sobre el impacto potencial del cambio climático en la costa. Este proyecto tiene como objetivo principal proporcionar información valiosa sobre la evolución de la línea costera, el riesgo ambiental y los efectos del cambio climático en las áreas costeras, contribuyendo a la toma de decisiones informadas y a la planificación sostenible.

En colaboración con socios como el British Geological Survey (BGS), el Geological Survey Ireland (GSI), el Instituto de Hidráulica Ambiental de Cantabria y Arctus, así como la Universidad de Quebec, Rimouski (UQAR), como parte de un consorcio de Cambio Costero, ARGANS Ltd ha liderado un proyecto encargado por la Agencia Espacial Europea (ESA) para evaluar los beneficios para el usuario final de un servicio de este tipo.

El enfoque del proyecto se centra en la recopilación y análisis de datos satelitales durante un período de 25 años. Para ello, se emplean datos provenientes de los

sensores Sentinel-1, Sentinel-2 de Copernicus, así como datos Landsat del USGS, combinados con datos comerciales proporcionados por el programa ESA Third Party Mission. Estos datos permiten llevar a cabo una evaluación continua y detallada de la evolución de la línea costera en más de 1000 km de costa, abarcando 16 sitios de estudio en Canadá, Irlanda, España y el Reino Unido, con condiciones ambientales diversas.

El proyecto ha desarrollado una serie de productos y servicios que abordan aspectos clave de la monitorización de la erosión costera y el cambio ambiental. La herramienta Waterline utiliza imágenes satelitales para detectar la línea de agua, permitiendo una mayor precisión espacial mediante la corrección de desplazamientos pixelares. Además, se emplea la técnica de batimetría derivada por satélite (SDB) para monitorear cambios en el lecho. La clasificación de características emplea algoritmos de aprendizaje supervisado para identificar áreas de tierra, zona litoral y océano, ofreciendo una comprensión detallada de la costa.

Este proyecto no solo presenta avances técnicos en la utilización de datos satelitales y técnicas de análisis, sino que también demuestra la relevancia y aplicabilidad de estas soluciones en la gestión del entorno costero. Al colaborar con expertos en geomorfología y trabajar en estrecha colaboración con la comunidad de usuarios finales, el proyecto evoluciona de un servicio de monitoreo a una herramienta de evaluación de riesgos y predicción, abordando desafíos críticos en el ámbito de la erosión costera y el cambio climático.

1.1.2. LULC

El concepto de Uso y Cobertura del Suelo se refiere a cómo la tierra es utilizada por actividades humanas y cómo está cubierta por elementos naturales [10]. Las aplicaciones derivadas de la categorización LULC tienen un impacto significativo tanto en la sociedad como en el medio ambiente, ya que abordan una variedad de desafíos y necesidades cruciales. En el contexto específico del proyecto de *Coastal Erosion from Space* [5], el LULC se convierte en una herramienta crucial para la extracción de información de las áreas de interés de zonas costeras. El LULC se utiliza en este proyecto para la clasificación de características mediante aprendizaje supervisado. El LULC desempeña un papel fundamental al proporcionar información detallada sobre cómo se utilizan y cambian las áreas costeras a lo largo del período de 25 años de datos recopilados de diversas fuentes satelitales, lo que contribuye a la evaluación precisa de la erosión costera y la gestión de riesgos ambientales en un contexto global. Esta información es valiosa tanto para los científicos involucrados en el proyecto como para la comunidad de usuarios finales, ya que respalda la toma de decisiones informadas sobre la gestión de la costa y la adaptación al cambio climático en todo el mundo.

Además de lo mencionado, existen una gran diversidad de aplicaciones. Una de las áreas en las que estas aplicaciones son de gran relevancia es en la planificación urbana y el control del crecimiento desorganizado de las áreas urbanas [10]. También, la clasificación y seguimiento de la cobertura del suelo juegan un papel crucial en la conservación del hábitat natural [11]. Esta información permite identificar y proteger áreas críticas para la biodiversidad. En el ámbito de la gestión de desastres naturales

[12], la tecnología LULC es esencial para el monitoreo y la respuesta rápida. Otra aplicación importante es en la salud pública al poderse identificar patrones de uso del suelo que pueden favorecer la propagación de enfermedades transmitidas por vectores [13]. Por último, el análisis de LULC puede ayudar a identificar lugares adecuados para la instalación de sistemas de energías renovables [13]. Esto es crucial para impulsar la transición hacia fuentes de energía más limpias y sostenibles, reduciendo así el impacto ambiental y las emisiones de gases de efecto invernadero.

En resumen, las aplicaciones basadas en la categorización de Uso y Cobertura del Suelo tienen un impacto profundo en la sociedad y el medio ambiente al abordar desafíos como el control de la erosión costera, el crecimiento urbano desordenado, la conservación de la biodiversidad, la gestión de desastres, la salud pública y la transición energética. Estas aplicaciones demuestran cómo la tecnología puede ser una herramienta poderosa para promover un desarrollo sostenible y una mejor calidad de vida para las personas.

1.1.3. IOTA

El modelo *iota2* [14], usado por la empresa Argans para la producción de mapas LULC en el proyecto Coastal Erosion from Space, es una herramienta diseñada originalmente para la cartografía de cobertura terrestre a gran escala. Este es un modelo basado en Random Forest perteneciente a la categoría de los *Transformation Based Models* según la clasificación realizada en [15]. Aunque su enfoque principal es la clasificación, sus componentes y estructura también son aplicables a otras tareas de generación de datos, como la regresión y la extracción de características. *iota2* se basa en tres conceptos esenciales: Tareas, Pasos y Grupos.

Tareas: Una tarea representa un paso de procesamiento específico aplicado a datos de entrada, lo que resulta en datos de salida que se pueden utilizar en tareas posteriores. Las tareas pueden incluir preprocesamiento de datos, entrenamiento de modelos, cálculo de métricas y más. Las tareas pueden ejecutarse varias veces con diferentes parámetros o entradas.

Pasos: Un paso es un contenedor que gestiona un conjunto de tareas relacionadas que se pueden ejecutar de manera secuencial. Los pasos se utilizan para organizar las tareas y facilitar la gestión de las dependencias entre ellas. Los pasos pueden ayudar a optimizar el flujo de trabajo al permitir que las tareas compartan datos en memoria cuando sea posible.

Grupos: Un grupo contiene varios pasos y se utiliza con fines de programación. Los grupos proporcionan una forma de etiquetar y organizar conjuntos de pasos dentro del flujo de trabajo. Los usuarios pueden configurar el flujo de trabajo para ejecutar grupos específicos o combinaciones de grupos según sea necesario.

A continuación se muestra un ejemplo simplificado del uso de las tareas, los pasos y los grupos en un flujo de trabajo de clasificación:

- Grupo: Inicio

- Tarea: Calcular la máscara común entre todas las imágenes para cada azulejo
- Tarea: Calcular una máscara binaria para nubes, saturaciones y bordes
- Tarea: Calcular la proporción de clases entre aprendizaje y validación
- Tarea: Extraer las muestras para el aprendizaje
- Tarea: Entrenar el clasificador
- Tarea: Clasificar las imágenes
- Tarea: Fusionar todas las clasificaciones para producir un único mapa

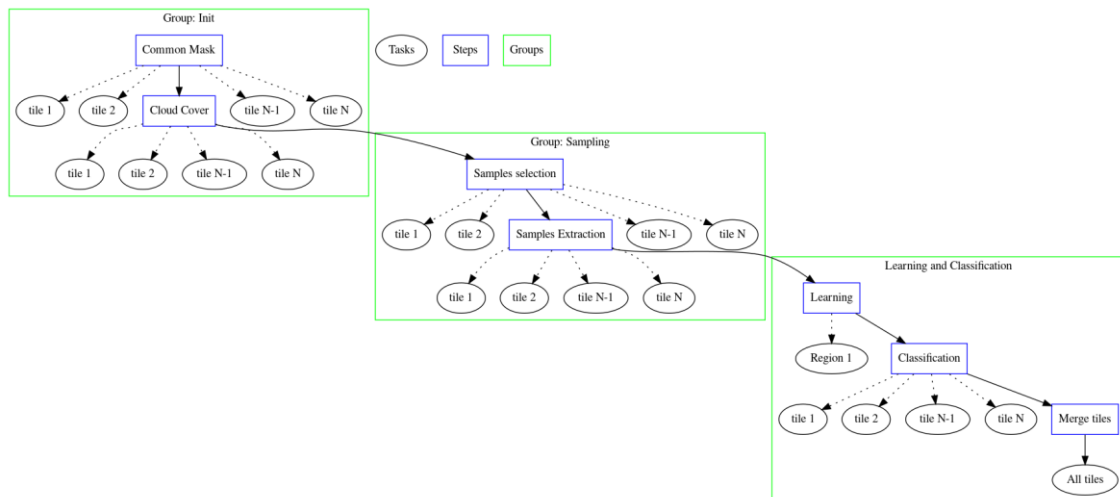


Figura 1.1: Flujo de trabajo de IOTA2 para clasificación. Fuente: [14]

La Figura 1.1 ilustra cómo se organizan las tareas en pasos dentro del grupo "Inicio". Cada tarea representa una operación específica en el flujo de trabajo, y los pasos ayudan a gestionar el orden de ejecución y las dependencias entre las tareas.

1.1.4. Limitaciones de IOTA

La debilidad principal del modelo expuesto en el punto anterior radica en su necesidad de ser reentrenado en cada instancia de uso. Esta limitación se traduce en largos tiempos de entrenamiento de este modelo. Cada adaptación de IOTA y la producción de series temporales de 25 años en una nueva área geográfica implica entre 6 y 8 meses de trabajo. No existe la capacidad de reutilizar el conocimiento adquirido durante un entrenamiento para aplicarlo en diferentes zonas geográficas. Esta limitación se traduce en una ineficiencia significativa cuando se busca aplicar el modelo a nuevas imágenes satelitales. La falta de transferencia de conocimiento entre distintas ubicaciones geográficas resulta en una pérdida de tiempo y recursos al tener que volver a entrenar el modelo desde cero para cada nueva aplicación.

Además, pueden detectarse algunas otras debilidades de entre las que destacan:

1. Dependencia en el Número de Segmentos por Mosaico: El modelo es dependiente del conocimiento del número de segmentos por mosaico antes de la clasificación, lo que limita su aplicabilidad en situaciones

donde la segmentación es dinámica y no se conocen de antemano el número de segmentos.

2. Gestión de Errores y Reanudación: Si un error ocurre en un paso específico, es necesario reiniciar todo el flujo de trabajo desde el principio, en lugar de poder reanudar la ejecución desde el punto de error.
3. Complejidad de la Configuración: La definición de pasos, tareas y flujos requiere una configuración detallada, lo que aumenta la curva de aprendizaje y la posibilidad de cometer errores de configuración.

1.2. Objetivos

El propósito del presente Trabajo de Fin de Grado se articula en torno a una serie de objetivos clave que impulsan la investigación y desarrollo de un modelo avanzado de segmentación semántica basado en Deep Learning. Estos objetivos reflejan la ambición de transformar datos satelitales en mapas detallados de Uso y Cobertura del Suelo (LULC) con alta precisión y versatilidad. Los objetivos se estructuran de la siguiente manera:

1. Desarrollar un Modelo de Segmentación Semántica Preciso: consiste en diseñar y construir un modelo de segmentación semántica empleando arquitecturas de Deep Learning, como *VGG16 Unet*, *Attention Unet* y *Swin Transformer*. Se busca alcanzar una alta precisión en la generación de mapas de Uso y Cobertura del Suelo a partir de datos satelitales, permitiendo una comprensión detallada de los patrones geoespaciales.
2. Adaptabilidad a Nuevas Zonas Geográficas: Un aspecto distintivo del modelo es su capacidad para adaptarse eficientemente a nuevas áreas geográficas sin requerir un proceso exhaustivo de reentrenamiento. Este objetivo amplía la aplicabilidad del modelo y lo convierte en una herramienta versátil para diversas ubicaciones y contextos, agilizando su implementación en diferentes regiones.
3. Contribución a la Monitorización del Entorno Terrestre: El trabajo aspira a contribuir a la comprensión y monitoreo del entorno terrestre, abordando desafíos contemporáneos relacionados con la gestión territorial, recursos naturales y toma de decisiones informada. El modelo desarrollado busca ser una herramienta valiosa para generar información relevante en la gestión ambiental y urbana.
4. Crear un conjunto de datos coherente y eficiente a partir de 3 conjuntos de datos distintos.
5. Colaboración Práctica con Albavalor: La colaboración con la empresa privada enriquece el proyecto al proporcionar una dimensión aplicada y práctica. El objetivo es integrar los avances de la investigación con las necesidades y desafíos reales de la industria de la teledetección. Esta colaboración brinda una

oportunidad única para validar y ajustar el modelo en función de las demandas del mundo real.

En conjunto, estos objetivos guían el desarrollo del modelo de segmentación semántica y establecen la base para una investigación significativa y un impacto aplicado en el ámbito de la teledetección y la monitorización del entorno terrestre.

1.3. Impacto esperado

El desarrollo y la implementación exitosa del modelo de segmentación semántica basado en Deep Learning propuesto en este trabajo de fin de grado tienen el potencial de generar un impacto significativo en varios niveles y para diversos actores involucrados. A continuación, se detalla el impacto esperado en diferentes ámbitos:

1. **Gestión Territorial y Planificación Urbana:**

El modelo permitirá a los planificadores urbanos, gobiernos locales y autoridades ambientales acceder a mapas de Uso y Cobertura del Suelo detallados y actualizados. Esto facilitará la toma de decisiones informada para el desarrollo urbano sostenible (la mejora duradera y a largo plazo de las condiciones sociales, económicas y ambientales de un área urbana [16]), la conservación de áreas naturales y la gestión de recursos en función de la comprensión precisa de la distribución espacial de diferentes tipos de terrenos. Asimismo, ayudará a prever y mitigar los efectos del cambio climático y la urbanización descontrolada.

2. **Mejora del servicio medioambiental para la detección de erosión costera y el diseño de planes de uso del suelo**

La colaboración con Albavalor y la aplicación práctica del modelo en la industria de la teledetección abrirán nuevas oportunidades de negocio y servicios. La capacidad de generar mapas LULC con alta precisión y adaptabilidad a nuevas zonas geográficas puede impulsar la oferta de productos y soluciones para la monitorización ambiental, evaluación de impacto ambiental, seguimiento de cultivos y mucho más. Esto potencialmente expandirá el mercado y mejorará la eficiencia de las soluciones existentes. Por ejemplo, la implementación de este nuevo modelo tiene el potencial de disminuir los costes relacionados con el estudio de la erosión costera para la Comunidad Valenciana y también se podría aplicar a todo el territorio nacional.

3. **Desarrollo de Políticas Sostenibles:**

Los responsables de la formulación de políticas y las organizaciones gubernamentales podrán utilizar los resultados del modelo para respaldar la planificación y ejecución de estrategias sostenibles. La información precisa sobre la distribución del uso del suelo y la cobertura será esencial para cumplir con los Objetivos de Desarrollo Sostenible (ODS) [17] (son un conjunto de 17 metas globales adoptadas por la Asamblea General de las Naciones Unidas en 2015 para abordar desafíos mundiales, como la pobreza, la igualdad de género y el cambio climático, con el objetivo de

lograr un desarrollo sostenible para 2030 [18]) y establecer políticas que aborden desafíos ambientales y sociales.

4. Sociedad en General:

El impacto se extiende a la sociedad en su conjunto, ya que una gestión efectiva del entorno terrestre tiene un efecto directo en la calidad de vida de las personas. La capacidad de tomar decisiones basadas en datos precisos y actualizados contribuirá a un entorno más saludable, a la conservación de recursos naturales y a la mitigación de desastres naturales.

En el contexto de los Objetivos de Desarrollo Sostenible (ODS), este trabajo contribuye directamente a varios de ellos, incluyendo el ODS 11 (Ciudades y comunidades sostenibles) [19] al permitir una planificación urbana informada, y el ODS 15 (Vida de ecosistemas terrestres) [20] al facilitar la monitorización y gestión de los recursos terrestres de manera sostenible. También podemos encontrar relación con otros Objetivos de Desarrollo Sostenible como el ODS 13 (Acción por el clima) [21] que busca adoptar medidas urgentes para combatir el cambio climático y sus efectos.

En resumen, el impacto esperado de este trabajo de fin de grado se extiende desde la toma de decisiones informada en la gestión territorial hasta la promoción de prácticas sostenibles y la mejora general de la calidad de vida. La aplicación práctica en la industria y la colaboración con Albavalor añaden un matiz concreto y aplicado, asegurando que los resultados de la investigación tengan un valor real y significativo en el mundo actual.

1.4. Metodología

En este proyecto, la metodología juega un papel crucial al abordar el desafío de desarrollar un modelo avanzado de segmentación semántica para la generación de mapas de Uso y Cobertura del Suelo a partir de datos satelitales. La metodología seleccionada en este proyecto asegura que el proceso de investigación sea estructurado y que las decisiones tomadas estén respaldadas por un enfoque sólido y bien fundamentado.

Para llevar a cabo este trabajo, se utilizará la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), que proporciona un marco bien definido para la planificación, implementación y evaluación de proyectos de minería de datos. Como se argumenta en [22], CRIPS-DM es una buena metodología en proyectos dirigidos por objetivos y por tanto, este modelo en nuestro caso es totalmente válido.

La metodología CRIPS-DM consta de 6 fases. Estas son: (I) comprensión del proyecto, (II) comprensión de los datos, (III) preparación de los datos, (IV) modelado, (V) evaluación y (VI) despliegue.

Estas fases son fundamentales en el proyecto y cada una tiene un papel crucial para completar el trabajo. En cada una de estas se realiza lo siguiente:

- I. Comprensión del proyecto: Es la primera fase de esta metodología. El objetivo de esta fase es obtener una comprensión profunda del problema a tratar, siendo capaz de identificar objetivos y necesidades del proyecto.

Las próximas cuatro fases han sido consolidadas en un pipeline que abarca desde la captura inicial de las imágenes hasta la generación de predicciones para las máscaras de segmentación. En esta sección, se presentará de manera conceptual cada una de estas fases. Los aspectos técnicos y la solución propuesta serán detallados en la sección 4.4.

- II. Comprensión de los datos: En esta fase los datos se recopilan y se exploran.
- III. Preparación de los datos: Esto incluye todas las operaciones y transformaciones necesarias para construir un conjunto de datos final.
- IV. Modelado: Esta fase consiste en seleccionar las técnicas adecuadas para nuestro problema así como ajustar sus parámetros de forma óptima.
- V. Evaluación: Una vez creado el modelo, pasamos a evaluar estos para comprobar si cumplen con nuestras expectativas. En el caso de que los resultados sean satisfactorios, pasaríamos a la siguiente fase. En caso contrario, volveremos a las fases anteriores con el fin de encontrar una mejora en los resultados.
- VI. Despliegue: Esta fase consiste en recoger el conocimiento generado por los modelos y transmitirlo. Puede ser en forma de aplicación, informe...

La Figura 1.2 muestra gráficamente el proceso descrito:

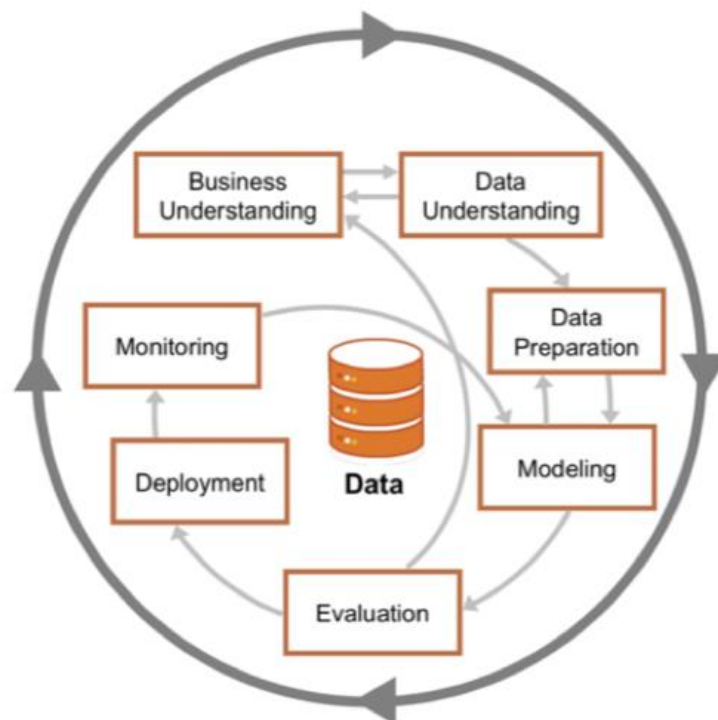


Figura 1.2: Metodología CRIPS-DM. Fuente: [24]

1.5. Estructura de la memoria

A continuación vamos a hacer una breve explicación de los distintos capítulos que forman la memoria del proyecto. Estos capítulos son:

1. Introducción

En esta sección, se presenta una visión general del proyecto 'Coastal Erosion from Space'. Se abordan la motivación detrás de este proyecto, los objetivos que se buscan alcanzar, el impacto esperado de los resultados, la metodología utilizada en el estudio y una breve descripción de la estructura de la memoria.

2. Estado del arte

En el capítulo sobre el estado del arte, se realiza una revisión exhaustiva de la clasificación de Land Use Land Cover. Además, se analiza críticamente el estado actual de las técnicas de análisis y se propone un enfoque innovador para el proyecto.

3. Contexto

Este capítulo abarca dos temas fundamentales: teledetección y Deep Learning. Se explora en profundidad la teledetección, desde sus principios fundamentales hasta el preprocesamiento de datos. También se discuten conceptos clave en Deep Learning, como redes neuronales, CNN y segmentación semántica en el contexto del análisis de datos.

4. Análisis del problema

Aquí se lleva a cabo un análisis detallado de los datos utilizados en el proyecto, que incluyen conjuntos de datos como IOTA, CLC y Sentinel 2. Se abordan las consideraciones legales y éticas relacionadas con la investigación y se presenta una solución propuesta para el problema en cuestión.

5. Descripción de modelos

Este capítulo se dedica a describir en detalle los modelos específicos empleados en el proyecto. Se presentan modelos como VGG16 Unet, Attention Unet y Swin Transformer, explicando sus características y aplicaciones.

6. Recursos utilizados

Aquí se proporciona una visión general de las herramientas tecnológicas utilizadas en el proyecto, tanto en términos de software como de hardware.

7. Experimentación y resultados

En esta sección se detalla la metodología experimental, incluyendo las métricas utilizadas para evaluar los resultados, la selección de hiperparámetros para los diferentes modelos, estrategias de entrenamiento y validación. Además, se presentan los resultados obtenidos durante la investigación.

8. Conclusiones

Entrenamiento de modelos de Aprendizaje Automático para generación de mapas de Uso y Cobertura del Suelo (LULC) utilizando datos de satélite

En el último capítulo, se resumen las conclusiones generales del proyecto, se discute la reproducibilidad de los resultados, se analiza la relación del trabajo con los estudios previos y se proponen posibles líneas de trabajo futuro.

2. Estado del arte

2.1. Clasificación LULC

El objetivo de la clasificación de uso del suelo y cobertura del suelo es proporcionar automáticamente etiquetas que describan el tipo de terreno físico representado y su uso [6]. En [23] podemos ver como se dividen las técnicas de clasificación LULC en dos grupos: técnicas de *Machine Learning* y técnicas de *Deep Learning*.

Dentro de las técnicas de *Machine Learning* destacan el método de Máxima Verosimilitud [25][26], *Random Forest* [27] y SVM (*Support Vector Machine*) [28]. Sin embargo, los enfoques basados en redes neuronales profundas, como las *Convolutional Neural Networks* (CNN), superan los resultados de estas técnicas gracias a su capacidad para aprender automáticamente características de alto nivel [29].

Las CNN son ampliamente utilizadas en aplicaciones de clasificación LULC y han demostrado un excelente rendimiento en diversos tipos de datos y áreas geográficas [30][31]. Estudios han demostrado que las CNN superan significativamente a las *Artificial Neural Networks* (ANN) y a los *Random Forest* [32]. Además, se han desarrollado variaciones de las CNN con mejoras significativas, como la incorporación de Mecanismos de Atención [3].

En la actualidad, los *Vision Transformers* (ViT) están ganando popularidad y han demostrado ser superiores a las CNN en ciertos contextos específicos [4]. Otra aproximación interesante es el *Semi-Supervised Learning*, que permite entrenar modelos con un buen rendimiento utilizando menos datos etiquetados [4][33].

2.2. Crítica al estado del arte

El estado actual de los sistemas de clasificación de Uso de Suelo y Cobertura del Suelo presentan una limitación fundamental que afecta significativamente su utilidad y eficiencia. Esta limitación se relaciona con su capacidad limitada para generalizar y adaptarse a nuevas áreas geográficas. Estos modelos se entrenan utilizando imágenes satelitales de ubicaciones específicas, lo que dificulta enormemente la transferencia de conocimientos a regiones desconocidas. Este aspecto limitante está presente en el proyecto de Coastal Erosion, que emplea un modelo de *Machine Learning* como *Random Forest*. Estos modelos requieren un proceso completo de reentrenamiento al utilizarse en nuevas zonas, lo que impide aprovechar el conocimiento previamente adquirido en áreas ya mapeadas. Esta falta de capacidad de generalización y la necesidad de reentrenamiento prolongado generan un tiempo de entrenamiento considerable, como se evidencia en el modelo *iota2*, donde se pierden entre 6-8 meses por cada nuevo módulo. Esta ineficiencia en la adaptación y la falta de aprovechamiento de la experiencia previa son desafíos críticos que requieren soluciones para avanzar en la precisión y la eficiencia de los sistemas de clasificación en este campo.

2.3. Propuesta

Como solución a los desafíos mencionados anteriormente, se sugiere la implementación de modelos de Deep Learning en lugar de modelos de Machine Learning como el Random Forest de *iota2* para capitalizar la transferencia de conocimiento inherente a estos enfoques. En concreto, se proponen tres arquitecturas de modelos:

- Un modelo VGG16-Unet que se beneficiará del preentrenamiento utilizando el conjunto de datos Imagenet, seguido de un afinamiento de sus capas finales.
- Un modelo Attention Unet diseñado para aprovechar el poderoso mecanismo de atención en su estructura, lo que potenciará la capacidad de generalización y la captura de características relevantes.
- Un modelo preentrenado basado en ViT (Transformers de Visión) con aprendizaje Auto Supervisado, como se detalla en el artículo [4], que promete una capacidad excepcional para capturar patrones y características en datos geoespaciales.

Estas propuestas se orientan hacia la mejora de la eficiencia y la precisión en la clasificación de Uso de Suelo y Cobertura del Suelo, abordando así los desafíos de generalización y transferencia de conocimiento en proyectos como el de Coastal Erosion. Para realizar el estudio, utilizaremos datos provenientes de Open Sentinel Map propuestos en [34] que abarca diversas zonas de todo el mundo.

3. Contexto

En este capítulo, abordaremos los conceptos clave necesarios para comprender el contexto de este trabajo de fin de grado. Nos centraremos en dos pilares fundamentales: la teledetección y el Deep Learning. Estos dos campos convergen para desempeñar un papel crucial en la interpretación de datos geospaciales y la resolución de problemas relacionados con la percepción y análisis de la Tierra desde el espacio. A lo largo de este capítulo, exploraremos las bases teóricas de la teledetección y el aprendizaje profundo, así como su relevancia en el marco de este estudio.

3.1. Teledetección

La Teledetección desempeña un papel crucial en este proyecto, al proporcionar la base teórica y práctica para la adquisición de datos satelitales necesarios en la generación de mapas LULC. Comprender los principios fundamentales de la Teledetección, el preprocesamiento de datos y los desafíos asociados es esencial para entender el contexto de este trabajo. Esta sección proporcionará la base teórica necesaria para comprender cómo los datos de Sentinel 2 y otros recursos que son utilizados en la creación de mapas de usos del suelo y cobertura terrestre.

3.1.1. Introducción

La teledetección es el método científico de recopilar datos sobre la superficie terrestre sin contacto físico, mediante la detección y registro de la energía reflejada o emitida [11].

Este concepto se puede ejemplificar mediante la utilización de sistemas de imágenes que involucran los siguientes siete elementos (Figura 3.1): Fuente de energía o iluminación (A), La radiación y la atmósfera (B), Interacción con el objetivo (C), Registro de la energía por el sensor (D), Transmisión, recepción y procesamiento (E), Interpretación y análisis (F) y Aplicación (G).

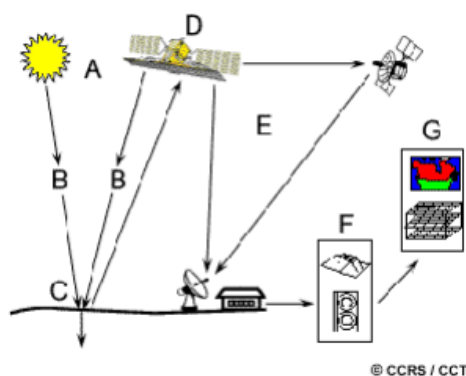


Figura 3.1: Sistema de Teledetección. Fuente:[11]

Los datos recopilados a través de la teledetección poseen una amplia gama de aplicaciones prácticas. Entre estas aplicaciones se encuentran la detección de plagas en la agricultura [35], la vigilancia y seguimiento de la erosión del suelo [36], así como la capacidad de pronosticar el deshielo tanto a corto como a largo plazo [37]. En el próximo apartado, se presentarán y explicarán los conceptos clave relacionados con esta tecnología.

3.1.2. Principios fundamentales

3.1.2.1. Tipos de sensores y plataformas de teledetección

Existen dos tipos de sensores según la energía utilizada para captar la información. Estos tipos son sensores pasivos y activos [38].

Los dispositivos de teledetección que registran la energía natural disponible se denominan sensores pasivos [39]. Estos dispositivos son efectivos únicamente cuando la energía natural está presente, y esta energía emitida naturalmente es detectable tanto de día como de noche. Ejemplos de este tipo de sensores es el sensor infrarrojo térmico, que capturan la radiación infrarroja emitida por los objetos en función de su temperatura [40], el espectrómetro, que detecta, mide y analiza el contenido espectral de la radiación electromagnética incidente [41] o el espectrorradiómetro que determina la potencia de la radiación en varios rangos de banda [41]. Este tipo de sensores no transmite energía propia al objeto de interés dependiendo de la energía natural que rebote en el objetivo y esto tiene como consecuencia la limitación de uso según las condiciones atmosféricas [41]. Por otro lado, los sensores activos generan su propia fuente de energía, y son capaces de emitir ondas en la región del espectro de las microondas y recibir la señal que rebota de la superficie Terrestre [39]. Los sensores LIDAR (que utilizan láser para medir distancias y crear nubes de puntos tridimensionales) y los sensores RADAR (que utilizan ondas de radio para detectar objetos y medir distancias) [40] son ejemplos de este tipo de sensor. También existen sensores como el altímetro, que mide la elevación junto al Lidar o la sonda que estudia condiciones meteorológicas [42]. Una ventaja de este tipo de sensores con respecto al anterior es que este funciona en cualquier momento del día ya que no requiere luz solar y es relativamente independiente de las dispersiones atmosféricas [42].

Un caso especial consiste en la ‘teledetección por microondas’, que mezcla tanto métodos pasivos como activos, transmitiendo y recibiendo señales o simplemente recibéndolas. La longitud de onda que utiliza tiene un espectro muy grande (varía entre 1 cm y 1 m) y esto provoca que a diferencia de longitudes de ondas más cortas esta no tenga ningún problema con las condiciones meteorológicas [42].

En cuanto a las plataformas de teledetección, en [40], proponen una clasificación que abarca diversas categorías. En primer lugar, se encuentran las plataformas satelitales, como ejemplos destacados Landsat-1 y Sentinel-2. A continuación, se incluyen las plataformas aéreas, que engloban sistemas de aeronaves no tripuladas. Por otro lado, las plataformas móviles se refieren a sensores instalados en vehículos en movimiento. Finalmente, se mencionan las plataformas estáticas, que abarcan cámaras

de lapso de tiempo instaladas en ubicaciones fijas. Esta clasificación resulta fundamental para comprender de manera integral los diversos enfoques y aplicaciones de la teledetección. El interés principal de nuestro proyecto se centra en el caso de Sentinel-2, ya que esta plataforma servirá como fuente principal de imágenes para nuestro trabajo. Es importante destacar que Sentinel-2 está equipado con un sensor óptico multiespectral pasivo, lo que implica que la calidad de las imágenes capturadas puede verse afectada por condiciones meteorológicas, como la presencia de nubes, el impacto de la luminosidad, los cambios en la vegetación durante diferentes estaciones del año (i.e. variabilidad cubierta vegetal de invierno a verano). Abordaremos esta limitación en detalle en la sección 4.1.1 de nuestro proyecto para encontrar soluciones adecuadas.

Espectro Electromagnético y bandas espectrales

Como se ha comentado, el primer requisito en un sistema de teledetección es tener una fuente de energía con la que iluminar el objetivo. Esta energía viene en forma de radiación electromagnética.

Toda la radiación electromagnética se comporta según la teoría de Ondas, por lo que su comportamiento es predecible [11]. La radiación electromagnética se compone de un campo eléctrico y uno magnético.

La radiación electromagnética tiene dos características principales: la longitud de onda, que podemos definirla como la distancia entre dos crestas sucesivas y la frecuencia, que se refiere al número de ciclos de una onda que pasan por un punto fijo por unidad de tiempo.

Por otro lado, podemos definir el espectro electromagnético como el conjunto de longitudes de onda de todas las radiaciones electromagnéticas [43]. Por ejemplo, la parte del espectro UV (espectro ultravioleta) es la más corta y una de las más usadas en teledetección ya que algunos materiales terrestres emiten luz al ser iluminados por radiación UV [11]. También existe el espectro visible que abarca aproximadamente de 0.4 a 0.7 μm , siendo la única parte del espectro que podemos relacionar con colores [11]. La región infrarroja (IR) abarca un amplio rango de longitudes de onda, desde 0.7 μm hasta 100 μm , por lo que es 100 veces más amplio que la porción visible [11]. Se divide en dos categorías: el IR reflejado y el IR térmico, que es la radiación de calor emitida desde la Tierra. La región de microondas, que abarca desde aproximadamente 1 mm hasta 1 m, cubre longitudes de onda muy largas. Las longitudes de onda más cortas se comportan de manera similar a la región de infrarrojo térmico, mientras que las más largas se acercan a las usadas en las emisiones de radio (Figura 3.2).

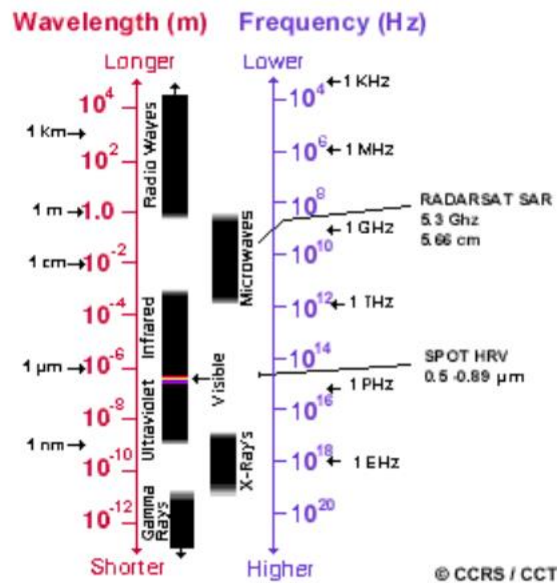


Figura 3.2: Espectro Electromagnético. Fuente: [11]

Los materiales reflejan y absorben ondas de radiación electromagnética de diferentes longitudes. Observando las distintas longitudes de onda de un sensor podemos determinar el tipo de material por el cual fueron reflejadas. Esto es la 'firma espectral' mostrada en la Figura 3.3, donde podemos ver la relación que existe entre el porcentaje de reflectancia y las longitudes de onda reflectivas de algunos componentes terrestres.

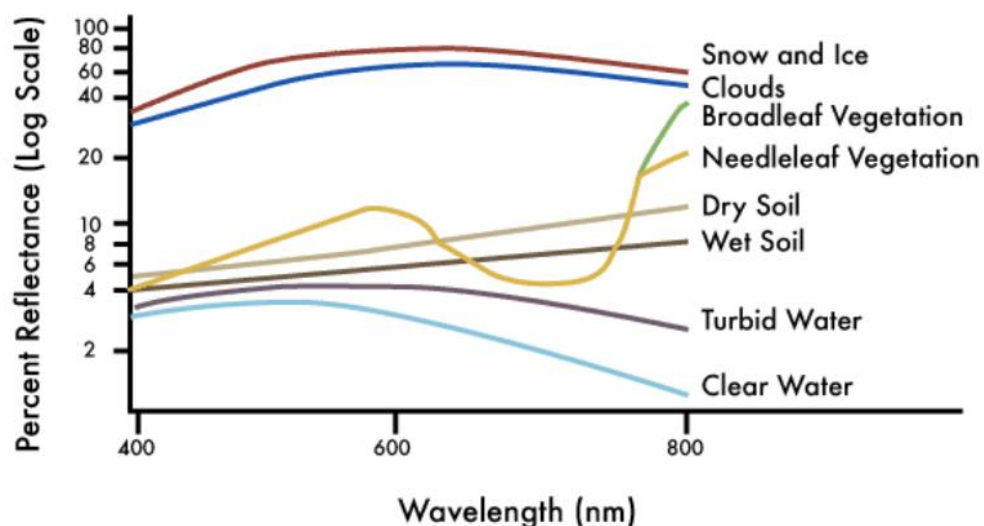


Figura 3-1.3: Firma espectral de objetos terrestres. Fuente: [147]

Es habitual realizar combinaciones de diferentes bandas espectrales según la aplicación del proyecto. Algunos ejemplos de estas combinaciones son:

Infrarrojo

Esta combinación tiene buena sensibilidad al color verde y aparecerá representado en color rojo, debido a la alta reflectividad en el infrarrojo y la baja en el visible [44]. Se crea combinando la banda NIR (Near Infrared por sus siglas en inglés), la banda roja y la banda verde. Un ejemplo de esta combinación puede verse en Figura 3.4:



Figura 3.4:Índice Infrarrojo. Fuente: [44]

Índice de Vegetación de Diferencia Normalizada (NDVI)

El NDVI es un índice de vegetación que se utiliza para estimar la cantidad, calidad y desarrollo de la vegetación con base a la medición de la intensidad de la radiación de ciertas bandas del espectro electromagnético que la vegetación emite o refleja [44]. Se consigue realizando la siguiente operación con las bandas: $(\text{NIR} - \text{Rojo}) / (\text{NIR} + \text{Rojo})$ y se consigue una imagen como la Figura 3.5:

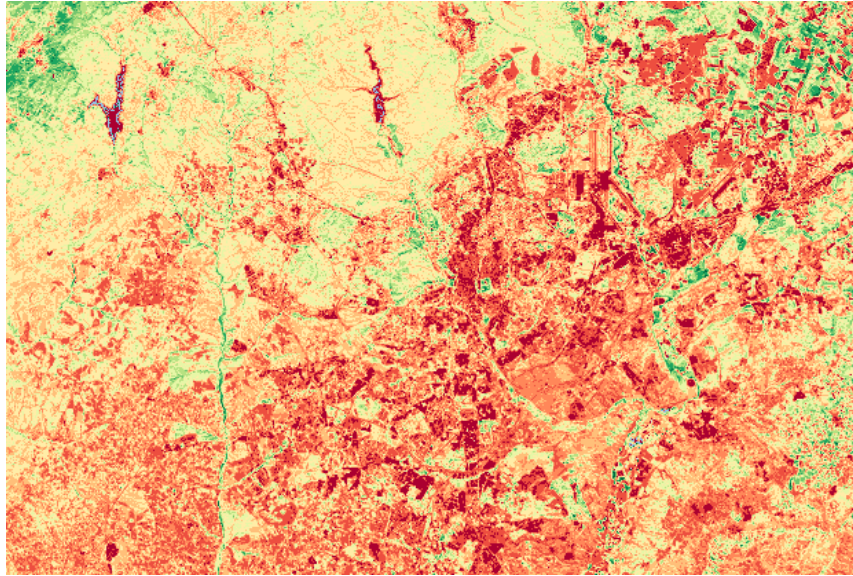


Figura 3.3-2: Índice NVDI. Fuente: [44]

Falso color para detección de zonas urbanas

Las áreas urbanas aparecen en tonos magentas mientras que las praderas o pastos se representan en tonos verdes claros. De verde oliva a verde brillante indica áreas forestales [44]. Se consigue mediante la combinación de las bandas SWIR2, SWIR1 y Roja y se muestra en la Figura 3.6:



Figura 3.3-3: Índice Falso color para detección de zonas urbanas. Fuente: [44]

3.1.2.2. Resolución Espacial, Espectral, Radiométrica y Temporal.

La calidad de los datos de satélite se determina mediante cuatro tipos de resolución:

Resolución Espacial, que se refiere a la capacidad de discernir detalles en una imagen [38].

Resolución Espectral, relacionada con la capacidad de un sensor para distinguir diferentes longitudes de onda en el espectro electromagnético [38]. Esta resolución se divide a su vez en datos pancrómicos (una sola banda espectral amplia diseñada para capturar el rango completo visible), multispectrales (varias bandas de onda al mismo tiempo) e hiperspectrales (cientos de bandas) [45]. Como se menciona en [46], la cantidad de bandas es un aspecto significativo a la hora de capturar detalles en las características de la superficie terrestre, lo que significa que los datos hiperspectrales pueden distinguir entre diferentes tipos de tierra y cobertura con mayor precisión que los datos multispectrales, que tienen menos bandas espectrales. Cuanta mayor sea la resolución espectral, más detalles pueden capturarse. Esto es crucial para identificar con precisión diferentes tipos de tierra y cobertura, ya que los objetos en la Tierra pueden tener firmas espectrales únicas en diferentes longitudes de onda.

Resolución Radiométrica, que describe la capacidad de un sensor para detectar pequeñas diferencias en la cantidad de energía reflejada o emitida por un objeto [38].

Resolución Temporal, que se refiere a la frecuencia con la que se pueden obtener imágenes de la misma área [11].

Estos aspectos de resolución son fundamentales para comprender y utilizar datos de detección remota de manera efectiva en diversas aplicaciones, desde la detección de cambios en la vegetación hasta la monitorización de fenómenos naturales y actividades humanas [11]. La Figura 3.7 muestra las distintas resoluciones:

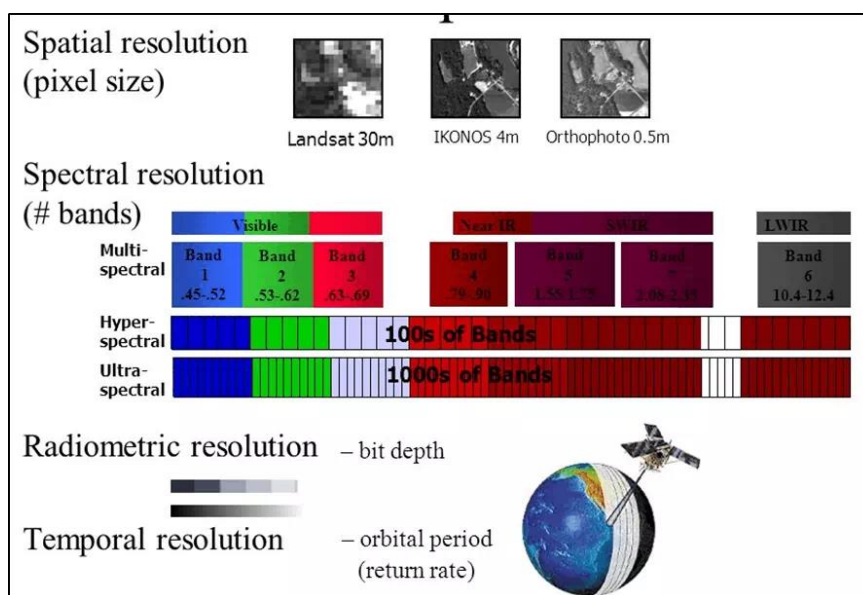


Figura 3.7: Diferentes resoluciones en Remote Sensing. Fuente: [47]

3.1.3. Preprocesamiento de datos en teledetección

La necesidad del preprocesamiento de imágenes en el contexto de la teledetección y el análisis de imágenes digitales es innegable en el mundo actual de la tecnología avanzada. Esta necesidad surge por ejemplo de ciertas limitaciones presentadas en el punto 3.1.2 donde se explicaban las desventajas de los sensores pasivos. Es por eso que en muchas ocasiones, es necesario realizar una corrección de errores y distorsiones de las imágenes capturadas. También hay que tener en cuenta que no todas las imágenes de un mismo lugar se capturan en el mismo instante de tiempo. Es por eso que existe una necesidad de comparabilidad, asegura que todas las imágenes estén en la misma escala radiométrica y que los efectos atmosféricos y topográficos se tengan en cuenta [48]. También los efectos del ruido aleatorio, la iluminación solar o los efectos topográficos deben ser corregidos [48]. Con el objetivo de mejorar la calidad visual, existen 3 tipos de correcciones usadas en teledetección: la corrección radiométrica que restablece los valores de la imagen y elimina anomalías en la radiancia del objetivo causadas por la atmósfera o defectos del sensor, la corrección geométrica que ajusta la imagen a un sistema de coordenadas y proyección cartográfica para corregir distorsiones causadas por movimientos en la plataforma, cambios en la altitud de la plataforma, rotación de la Tierra y relieve del terreno y la corrección atmosférica que elimina el efecto de los aerosoles y la radiancia intrínseca en la imagen [49].

3.1.4. Desafíos y limitaciones

Entre los principales desafíos en el ámbito de la teledetección, se destacan varios aspectos. La complejidad de los datos, caracterizados por su diversidad y alta dimensionalidad [50], plantea un reto significativo. La necesidad de recopilar datos con alta frecuencia y garantizar su alta calidad también constituye un desafío fundamental [51][52]. Además, los costos asociados a la implementación de sistemas de teledetección pueden ser un obstáculo importante [51]. Aunque el problema de la resolución espacial se ha abordado en parte gracias a la misión Sentinel-2, sigue siendo relevante [52]. La utilización de datos fusionados de diversas fuentes agrega otra capa de complejidad [52].

3.2. Deep Learning

En este capítulo se presentarán algunos aspectos clave del aprendizaje profundo que son fundamentales para comprender el funcionamiento de esta tecnología, la cual ha captado gran atención en los sistemas más avanzados para la segmentación de imágenes satelitales.

El Deep Learning (en español Aprendizaje Profundo), es una rama del aprendizaje automático que intenta modelar abstracciones de alto nivel de los datos utilizando múltiples capas de neuronas que consisten en estructuras complejas o transformaciones no lineales [53].

Los modelos de Deep Learning están basados en redes neuronales artificiales. Las redes neuronales artificiales (ANN, siglas en inglés de Artificial Neural Networks),

son modelos de Machine Learning que evolucionaron a partir de la idea de simular el cerebro humano [54].

Las ANNs son capaces de analizar grandes volúmenes de datos para acelerar la toma de decisiones. Algunas de las aplicaciones del aprendizaje profundo incluyen la detección de correos no deseados, la clasificación y reconocimiento de imágenes, la predicción de precios de acciones, el reconocimiento de entidades nombradas en textos e incluso la implementación de vehículos autónomos [55].

En algunas de estas aplicaciones, se ha comprobado que el Deep Learning ya consigue superar el rendimiento de los humanos [56] y por lo tanto, superando también los sistemas clásicos.

Pese a que el Deep Learning goza de una gran popularidad y eficacia en muchos ámbitos, también presenta una serie de debilidades. La necesidad de grandes conjuntos de datos, la no explicabilidad de los sistemas, la carencia de sentido común [57] o sesgos como los sesgos raciales o sesgos de género (producidos sobretudo en sistemas de procesamiento del lenguaje natural) [58] son algunas de las debilidades que presentan estos sistemas.

En la siguiente sección, se llevará a cabo una detallada exploración de los fundamentos del Deep Learning que han sido empleados a lo largo de este trabajo. A través de una minuciosa explicación, se desentrañarán los diversos conceptos clave del Aprendizaje Profundo, proporcionando una base sólida para comprender y contextualizar las aplicaciones y métodos presentados en el presente estudio. Desde los cimientos de las redes neuronales hasta las complejidades de la optimización y la arquitectura de modelos, esta sección se propone ofrecer una visión completa de las herramientas y técnicas fundamentales que han impulsado el desarrollo de este trabajo de investigación.

3.2.1. Redes neuronales

Podemos entender las redes neuronales como una función matemática la cual es provista por una entrada y produce a partir de esta una salida. Estas redes están compuestas por nodos interconectados llamados neuronas. Estas neuronas trabajan conjuntamente para resolver así problemas complejos con datos de todo tipo, desde reconocimiento de voz hasta la predicción de la estructura secundaria de proteínas [59].

El origen de las redes neuronales surge con el trabajo en [60]. A partir de aquí, el uso de las redes neuronales hasta llegado el auge del deep learning ha pasado por 3 etapas de máximo interés:

- 1950: Frank Rosenblatt propone la primera red neuronal: el perceptrón [61].
- 1980: Se crea el algoritmo de retropropagación [62] y Yan Le Cun y sus compañeros proponen la primera red neuronal para reconocer dígitos escritos: la red neuronal convolucional [63].
- 2000: Actualmente estamos viviendo el mayor auge del aprendizaje profundo. El uso de las GPU en el entrenamiento de estas redes ha posibilitado el entrenamiento de redes muy profundas y de rendimiento notable.

3.2.2. Función discriminante lineal

Las redes neuronales se basan en una idea muy simple: la función discriminante lineal. Se trata del producto escalar entre un vector de entrada x y un vector de pesos w (los parámetros de la red), Además, se agrega un término independiente b al producto escalar, llamado sesgo, quedando como resultado la siguiente fórmula:

$$f(x) = \sum_{i=0}^d w_i^t x_i + w_0$$

Estas funciones son de carácter no lineal lo cual supone un problema cuando modelamos situaciones complejas (por ejemplo, la clasificación de imágenes). Estas funciones están limitadas a aproximar relaciones lineales, por lo que para aproximar este tipo de situaciones, pueden usarse funciones discriminantes no lineales:

$$F(f(x)) = F(w^t x + w_0)$$

3.2.3. Perceptrón multicapa

El perceptrón multicapa está compuesto por tres tipos de capas: capas de entrada, salida y ocultas. La capa de entrada es la que recibe los datos. Por otro lado, la capa de salida es la encargada de proporcionar el resultado final. Las capas ocultas son las responsables de aprender las relaciones entre los datos de entrada y de salida. Cada capa oculta está formada por diversas funciones discriminantes. La salida de cada una de estas funciones es una combinación lineal de las entradas. A esto, le sigue una función de activación no lineal. Así podemos introducir no linealidad, consiguiendo modelar relaciones más complejas.

Los pesos de la capa k que conectan las neuronas entre si, pueden ser definidos de la siguiente forma:

$$W^k = (w_{1,0}^k, \dots, w_{M_k, M_{k-1}}^k) : w_{M_k, M_{k-1}}^k \in R, 1 \leq k \leq L$$

Donde L es el número de capas y M_k es el número de nodos en la capa k . Por ejemplo, en una red con una única capa oculta, las salidas y la capa oculta se definen como:

$$\begin{aligned} s_j^1 &= F(f_{1j}) = F\left(\sum_i w_{j,i}^1 x_i\right), & 1 \leq j \leq M_1 \\ s_j^2 &= F(f_{2j}) = F\left(\sum_i w_{j,i}^2 x_i\right), & 1 \leq j \leq M_2 \end{aligned}$$

donde s_{1j} denota la salida de la j -ésima neurona en la capa oculta, s_{2j} denota la salida de la j -ésima neurona en la capa de salida, M_1 y M_2 denotan el número de neuronas en las capas oculta y de salida respectivamente y F es la función de activación

3.2.4. Función de activación

Las funciones de activación son funciones no lineales usadas en las redes neuronales para definir el *output* de una neurona. Estas funciones tienen suma importancia en la calidad del modelo, ya que “La precisión de predicción de una red neuronal depende del número de capas utilizadas y, lo que es más importante, del tipo de función de activación utilizada” [64].

Existen diferentes funciones de activación, cada una con sus pros y contras. Una interesante tabla con un resumen de las características de estas funciones se recoge en el artículo [65] y se puede observar en la Tabla 3.1:

TABLE I
FREQUENT PROPERTIES OF ACTIVATION FUNCTIONS

Property	Description	Problems	Examples
derivative	f'	> 1 exploding gradient (e) < 1 vanishing (v)	Sigmoid, Hyperbolic Tangent
zero-centered	range centered around zero?	if not, slower learning	Hyperbolic Tangent
saturating	finite limits	vanishing gradient in the limit	Hyperbolic Tangent, ReLU
monotonicity	$x > y \implies f(x) \geq f(y)$	unclear	exceptions: Swish

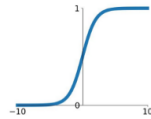
Tabla 1.1: Propiedades de las funciones de activación. Fuente: [65]

En la Figura 3.8, pueden observarse la forma de las diferentes funciones más comúnmente usadas:

Activation Functions

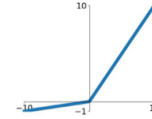
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



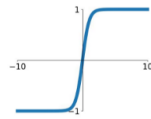
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

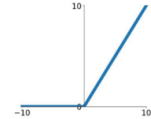


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

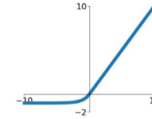


Figura 3.8: Funciones de activación más comunes. Fuente: [66]

3.2.5. Estimación de parámetros

Las redes neuronales son entrenadas mediante un método conocido como retropropagación o *Backpropagation* [62].

Este algoritmo aprovecha la regla de la cadena para calcular gradientes en relación a los parámetros de la red con respecto a una función de pérdida definida, que medirá la diferencia entre la salida de la red y la salida real. Después, se usa un

algoritmo de optimización como por ejemplo el descenso por gradiente, que va ajustando los parámetros en la dirección opuesta al gradiente con el objetivo de minimizar la función de pérdida.

La Figura 3.9 ilustra el algoritmo:

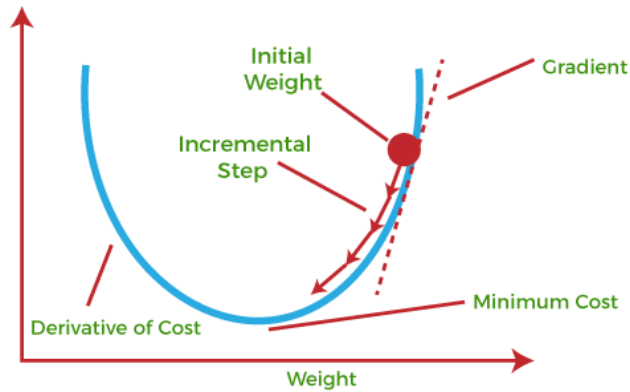


Figura 3.9: Algoritmo de Backpropagation. Fuente: [67].

Durante el proceso de entrenamiento, una vez que la propagación hacia adelante se completa, la salida de la red se coteja con la salida real, empleando la función de pérdida. Luego, se procede a calcular los gradientes de la función de pérdida en relación a los parámetros de la red a través de la técnica de retropropagación. Estos gradientes se utilizan para ajustar los parámetros mediante el mecanismo de descenso de gradiente. Este ciclo se repite en múltiples iteraciones, también conocidas como épocas, hasta que el desempeño de la red en un conjunto de validación alcance un nivel satisfactorio.

Además del descenso por gradiente, existen otros algoritmos para el cálculo de los parámetros de una red, de entre los que destacan algoritmos como *Stochastic Gradient Descent (SGD)* [68] o *Adaptive Moment Estimation (Adam)* [69], que son los dos algoritmos comúnmente usados en el campo del Deep Learning.

El algoritmo de descenso por gradiente estocástico [68] es una variante del descenso por gradiente clásico. *SGD* realiza una actualización de pesos por cada ejemplo de entrenamiento y etiqueta.

Mini-batch Gradient Descent [70] por otro lado combina lo mejor del descenso por gradiente y de *SGD* y realiza una actualización para cada lote pequeño de n ejemplos de entrenamiento.

Esto tiene algunas consecuencias como: reducir la variabilidad de las actualizaciones de parámetros, lo que puede conducir a una convergencia más estable y aprovechar las optimizaciones matriciales altamente optimizadas comunes en las bibliotecas de aprendizaje profundo de última generación, lo que hace que el cálculo del gradiente con respecto a un mini-lote sea muy eficiente [71].

Por otro lado, el algoritmo *Adam* [69] es un método que combina las ideas de dos optimizadores: *SGD* y *RMSprop* [72]. *Adam*, calcula tasas de aprendizaje adaptativas para cada uno de los parámetros. Además, almacena un promedio de los gradientes cuadrados pasados que decrece de forma exponencial. *Adam* también incluye una corrección de sesgo. En las primeras iteraciones del entrenamiento, los momentos estimados de los gradientes están lejos de su verdadero valor. Esto

introduce un sesgo que es corregido por *Adam*. Como se comenta en [73] “*Adam se comporta como una bola pesada con fricción, que interactúa con la superficie de error (la función a optimizar)*”. El optimizador *Adam* ha demostrado un rendimiento experimental superior a todos los demás optimizadores en Deep Learning [74]

Durante la fase de *Backpropagation*, los gradientes suelen enfrentar dos típicos problemas:

1. *Vanishing gradients*: Esto pasa cuando el gradiente se hace muy pequeño. Este problema puede provocar que los pesos y los sesgos no cambien su valor lo que en el peor de los casos puede provocar que la red deje de entrenarse [75]. Los gradientes desvanecientes suelen venir causados por el uso de funciones de activación con derivadas muy pequeñas cuando sus entradas están lejos de cero [76].
2. *Exploding Gradients*: Es el caso contrario al expuesto anteriormente y ocurre cuando los gradientes son excesivamente grandes, lo que produce cambios radicales en los pesos y sesgos de la red, lo que puede llevar a una inestabilidad en el proceso de entrenamiento de la red, pasando a un proceso de divergencia en lugar de convergencia [77].

3.2.6. Redes convolucionales

Los fundamentos de las redes convolucionales fueron creados por Kunihiro Fukushima con el Neocognitron [78]. Más tarde, estas redes fueron mejoradas por Yann Le Cun y su equipo en [63] al introducir a los fundamentos de Fukushima un método de aprendizaje basado en *backpropagation* [62] para entrenar el sistema de forma correcta. En 2012, gracias al uso de GPU [79] se consiguieron refinar los resultados obtenidos con anterioridad.

Las redes convolucionales son utilizadas en una amplia gama de aplicaciones: clasificación de objetos y vídeos, detección de objetos, segmentación de imágenes...[80].

La arquitectura de la red cambia ligeramente según la aplicación, por lo que a continuación, daremos una visión general de la arquitectura general.

3.2.6.1. Arquitectura de la red

Como se puede observar en la Figura 3.10, en general, una red convolucional, está compuesta por dos bloques principales: el bloque convolucional (feature extraction) y el bloque de clasificación (classification).

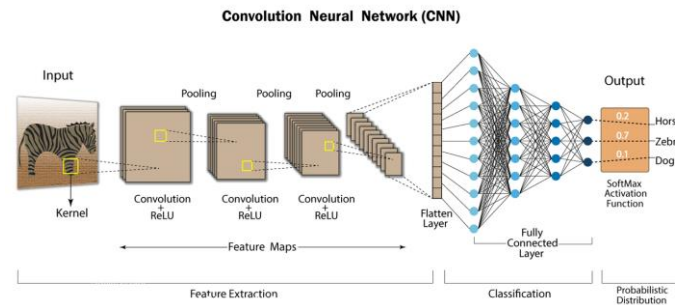


Figure 3.10: Arquitectura común de una CNN. Fuente: [29]

Para comprender esta arquitectura, comenzaremos por presentar la operación de convolución, que constituye el núcleo central de este tipo de redes.

3.2.6.2. Convolución

La convolución puede conceptualizarse como el producto entre dos funciones: f (la función original) y g (la función base), cuyo resultado es una tercera función que describe la magnitud de su superposición [81].

Considerando dos funciones, $f(x)$ y $g(x)$. En este contexto, la convolución se expresa a través de una integral que cuantifica cómo la función g se superpone a medida que se desplaza sobre la función f [82]. La función resultante de la convolución, denotada como $(f * g)(x)$, se define como:

$$(f * g)(x) = f(x) * g(x) = \int_{-\infty}^{+\infty} f(x')g(x - x')dx'$$

donde x' es una variable de integración.

En el contexto de redes neuronales que operan con imágenes, una de las funciones correspondería a la matriz de una imagen. Por otro lado, la otra función representa el filtro o kernel que se deslizará a lo largo de esa imagen. Estos filtros, que son esencialmente matrices, serán explorados en mayor en la sección 3.2.6.3.

3.2.6.3. Bloque convolucional

Este bloque tiene la función de analizar todos los detalles representativos de las imágenes que se utilizan como entradas. Es esencial distinguir entre los distintos conceptos que conforman este bloque:

1. Capas convolucionales: Esta capa es la encargada de convolucionar la matriz de píxeles generada para la imagen dada, produciendo así un mapa de características para esta misma imagen extrayendo así las características de la imagen en este mapa [83]. La Figura 3.11 muestra un proceso de convolución:

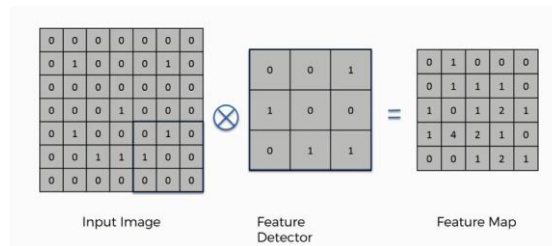


Figura 3.11: Capas de convolución. Fuente: [84]

2. **Pooling:** El pooling es una etapa importante para reducir las dimensiones del mapa de activación, manteniendo las características esenciales y mejorando la invarianza espacial [85]. Esto disminuye las características aprendibles del modelo y aborda el sobreajuste [86]. Los tipos incluyen max pooling, average pooling, stochastic pooling y spatial pyramid pooling, siendo el max pooling el más popular. La Figura 3.12 muestra un ejemplo del funcionamiento de 'max pooling':

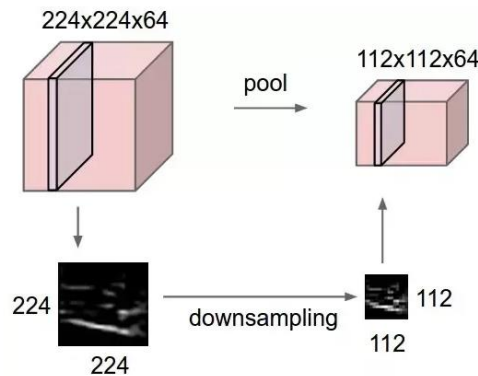


Figura 3.12: Ejemplo de Max Pooling. Fuente: [87]

3. **Funciones de activación:** Tal como se mencionó previamente, son componentes esenciales en una red neuronal. Su tarea consiste en determinar la salida en cada neurona, que a su vez se utiliza como entrada para la siguiente. Existe una variedad de funciones de activación, cada una elegida según su propósito específico. La más utilizada en redes convolucionales es la ReLu [88], ya que esta convierte todos los elementos negativos en cero.

Para controlar el proceso de aprendizaje, se hace uso de una serie de hiperparámetros, dependiendo la efectividad del modelo en gran medida de la elección de estos.

Entre estos hiperparámetros, destacamos: tamaño del filtro, número de filtros, tamaño del pooling y longitud del stride [89]:

1. **Kernel Size o Tamaño del Filtro:** Es crucial establecer la dimensión en ancho y alto de los diversos kernels utilizados en la convolución. Diferentes tamaños de kernels capturan variados patrones en la imagen. Por ejemplo, los kernels más pequeños tienen la tendencia a detectar detalles más sutiles de la imagen, tales como bordes o elementos pequeños [90].

2. **Número de filtros:** Este valor corresponde a la cantidad de kernels necesarios para que la red pueda aprender a identificar patrones en la imagen.

3. *Stride* o Salto: Se refiere a la cantidad de desplazamiento que el kernel realiza después de cada convolución. A medida que este valor aumenta, el volumen resultante de la convolución se reduce. Por ejemplo, si el stride es de 2, el kernel saltará dos columnas de píxeles en cada convolución.

4. *Padding*: Después de cada operación de convolución, debido a la naturaleza misma de esta operación, la imagen sufre una disminución en su tamaño [83]. Para contrarrestar este efecto, se implementa una operación que añade ceros en el borde de la imagen de entrada, preservando así el tamaño en la salida de la convolución. Esta medida es fundamental, ya que la reducción de dimensiones durante la convolución podría ocasionar la pérdida de información crucial y, por ende, una disminución en el rendimiento.

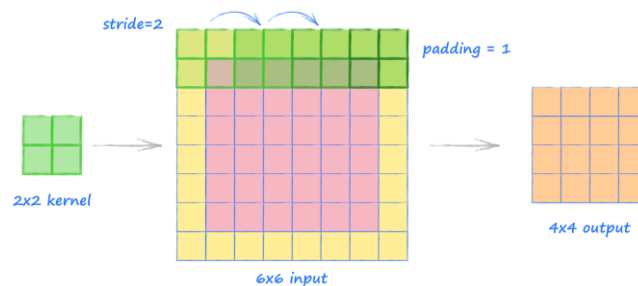


Figura 3.13: Hiperparámetros de una CNN. Fuente: 91

La Figura 3.13 resume lo explicado: se presenta un kernel de tamaño 2x2 que se aplicará a una matriz (imagen). Se establece un valor de paso (stride) de dos, lo que significa que el kernel se desplazará dos píxeles en cada paso. Además, se ha implementado un relleno unitario, visualizado como los píxeles coloreados en amarillo, que representan los píxeles añadidos con valor 0.

3.2.6.4. Bloque de clasificación

Esta capa no siempre está presente en todas las redes convolucionales. Por ejemplo, en tareas de segmentación, se prescinde de esta, teniendo redes completamente convolucionales, como por ejemplo, arquitecturas Unet [3].

Una vez acabado el proceso correspondiente al bloque de convolución, los datos deben ser aplanados y pasan a este último bloque (bloque de clasificación). El bloque de clasificación usa capas densamente conectadas (*fully connected*). En una capa densa, las neuronas están completamente conectadas con las de la siguiente capa. Estas capas se utilizan para clasificar las imágenes generadas por las capas convolucionales.

3.2.7. Attention Gate

El mecanismo de atención fue propuesto por primera vez en [92] con el objetivo de mejorar las técnicas existentes como [93][94] en el campo de la traducción automática. Con el paso de los años, esta técnica se ha ido adaptando a diferentes ámbitos, llegando a tener aplicación en *image caption generation* [95], clasificación de textos [96] o incluso en recomendadores [97].

Los mecanismos de atención se basan en los sistemas biológicos de los humanos, que tienden a enfocarse en las partes más distintivas al procesar grandes cantidades de información [98]. En otras palabras, los mecanismos de atención tratan de resaltar las partes importantes y de ignorar la información que no es relevante.

Podemos dividir los mecanismos de atención en diversos grupos [99]. En esta explicación nos centraremos en el mecanismo de atención propuesto en [100], que es el tipo de mecanismo de atención que adaptaremos a nuestro problema. Esencialmente, se utilizan dos tipos de mecanismo de atención: espacial y de canal. La atención espacial se enfoca en dónde se encuentra una parte informativa, lo cual es complementario a la atención de canal [101]. El autor, en [100], propone un novedoso mecanismo de atención espacial llamado *Attention Gate*. Este tipo de mecanismo se centra en regiones específicas de los datos y suprime las características irrelevantes. Teóricamente con esto conseguimos realizar una mejora en la capacidad de representación del modelo sin aumentar los costos computacionales [100].

En el método de la "*Attention Gate*", se utiliza una señal de compuerta recopilada a un nivel muy general que contiene información contextual sobre el mapa de características de entrada [100].

Esta señal de compuerta se utiliza en un proceso que implica atención aditiva (se asigna un peso a diferentes partes de los datos de entrada mediante la suma ponderada de características relevantes) para obtener el coeficiente de compuerta.

El mapa de características de entrada y la señal de compuerta se asignan primero linealmente a un espacio de n dimensiones, y luego la salida se comprime en el dominio del canal para producir un mapa de pesos de atención espacial [100]. El proceso general se puede escribir como:

$$S = \sigma \left(\varphi \left(\delta (\varphi x(X) + \varphi g(G)) \right) \right)$$

$$Y = SX$$

Donde σ , φx , φg y δ son transformaciones lineales implementadas como convoluciones de 1×1 .

El método de la puerta de atención genera un mapa de pesos de atención espacial, dirigiendo el enfoque del modelo hacia las áreas clave de la entrada y reduciendo la activación de características en regiones menos significativas.

La Figura 3.14 ilustra la arquitectura del mecanismo de Atención descrito:

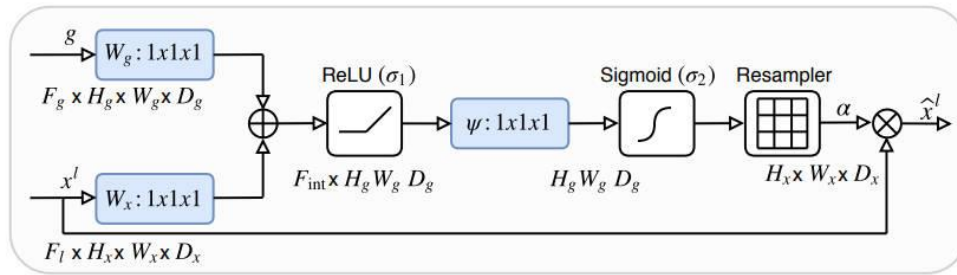


Figura 3.14: Puerta de Atención. Fuente: [100]

Y la Figura 3.15 los mapas de atención que acaba generando:

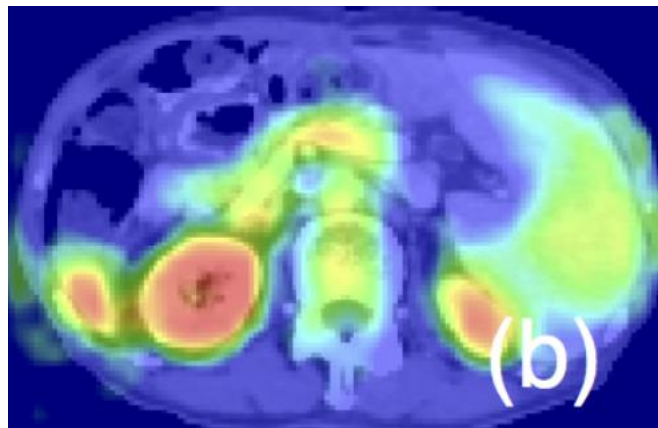


Figura 3.15: Mapas de atención. Fuente:[100]

3.2.8. Transformers

Un Transformer [102] es una arquitectura de modelo de aprendizaje automático desarrollada para procesar y generar secuencias de datos, como texto, imágenes o cualquier otro tipo de información que se pueda representar en forma de secuencia gracias al conocido 'mecanismo de atención'. Fue introducida en [102] en 2017 y marcó un hito en el campo del procesamiento del lenguaje natural y otras tareas relacionadas. Actualmente, son el estado del arte en muchas tareas de NLP [103].

A raíz de los acertados resultados arrojados en el campo del procesamiento del lenguaje natural, se propuso una adaptación de esta arquitectura para tareas de visión por computador [104]. Los ViT [104] (siglas de Vision Transformers en inglés), son la adaptación de la arquitectura de Transformer usada en el procesamiento de lenguaje natural al dominio de la visión por computador. Actualmente, es principalmente usada en tareas de segmentación y clasificación y es el estado del arte por encima de las redes convolucionales [105].

Usage Over Time

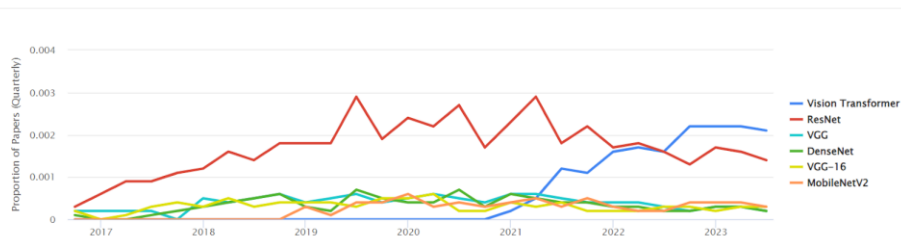


Figura 3.16: Proporción de Papers por modelo. Fuente: [105]

La Figura 3.17 muestra la arquitectura de un ViT:

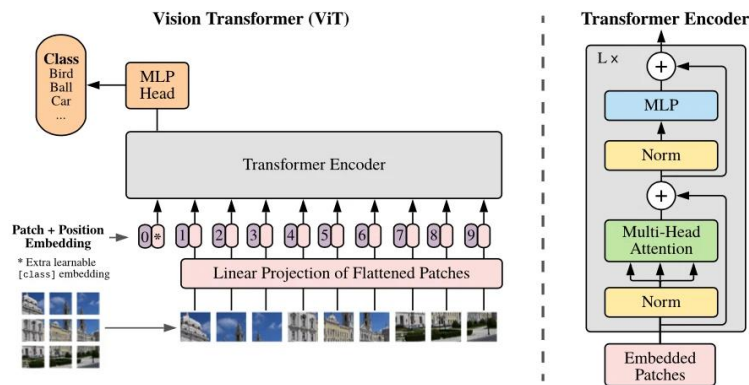


Figura 3.17: Arquitectura de un ViT. Fuente: [106]

A continuación, vamos a tratar de dar una idea intuitiva de su arquitectura y el funcionamiento, dividiendo el modelo en tres partes: *embedding*, *transformer encoder* y *MLP head*.

3.2.8.1. Embedding

En este paso, fragmentamos la imagen de entrada en parches de tamaño fijo de tamaño $[P, P]$ y los aplanamos linealmente, incluyendo los canales si están presentes. Por ejemplo, un parche de tamaño $[P, P, C]$ se transforma en tamaño $[P \cdot P \cdot C, 1]$. Luego, este parche aplanado se introduce en una capa *feedforward* con una función de activación lineal. Esto resulta en una proyección lineal del parche con dimensiones $[D, 1]$, donde D es la dimensión de incrustación, un hiperparámetro utilizado en todo el modelo Transformer.

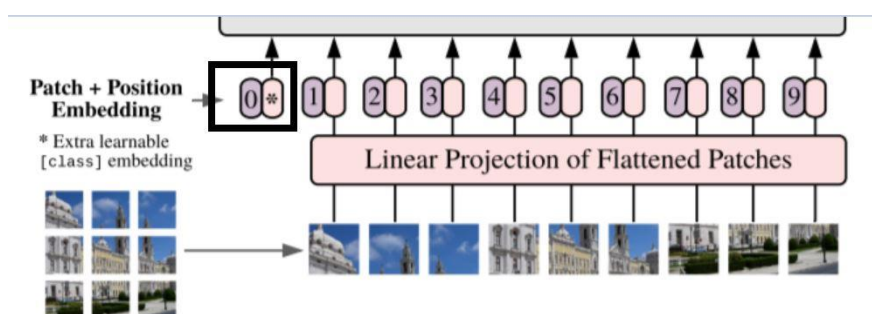


Figura 3.18: Fase de Embedding. Fuente: [106]

Se agrega la incrustación comentada anteriormente y un token de clase a las proyecciones de parches para la clasificación. Este token de clase representa la clase de cada *patch* y se une a los tokens de imagen, capturando información global. El modelo aprende esta agregación global a medida que avanza a través de las capas de atención. Además, se introduce una incrustación posicional 1D en los parches lineales para establecer un orden en los parches de entrada.

Este paso es crucial dentro de la arquitectura ya que los Transformers no son capaces de recordar el orden de entrada [104]. Es decir, si los fragmentos de las imágenes se reorganizasen de una forma distinta, se perdería el significado de la imagen original. Es por ello que se añade una incrustación posicional a los fragmentos de imagen embebidos con el fin de llevar un registro de la secuencia.

3.2.8.2. Transformer Encoder

La estructura del *Transformer Encoder* se compone de diversos bloques de codificación. Cada uno de estos bloques cuenta con una unidad de Atención Multi-Cabeza y un MLP. Además, después de cada capa se aplica una capa de normalización. También, en cada una de estas capas se usan conexiones residuales. Una vez el vector de entrada es procesado por el *Transformer Encoder*, este devuelve un vector llamado vector de contexto. La Figura 3.19 muestra la arquitectura de un *Transformer Encoder*, donde podemos ver las distintas partes comentadas:

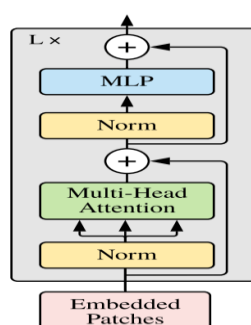


Figura 3.19: Estructura del Transformer Encoder. Fuente: [106].

A continuación se profundizará en el concepto de *Multi-Head-Attention*.

3.2.8.2.1. Atención Multicabeza

El mecanismo de atención es un componente clave en el Transformer utilizado en Vision Transformers (ViT). En ViT, el mecanismo de atención permite a la red enfocarse en partes relevantes de la imagen al asignar diferentes pesos a las relaciones entre los parches de la imagen.

El núcleo principal de una unidad de Atención Multi-Cabeza es la capa de *Scaled Dot Product Attention* [107]. En un inicio, el vector de entrada Z se clona en tres copias y se combina con los pesos W_q , W_k y W_v , para crear las Consultas, Claves y Valores respectivamente. Luego, las Consultas se multiplican con las Claves, y el resultado se ajusta dividiéndolo por la raíz cuadrada de la dimensión D . Esto se hace para evitar que la salida sea demasiado grande o demasiado pequeña, lo que evita el ‘*vanishing gradient problem*’ [108]. La matriz resultante pasa a través de una capa de *softmax* y se combina con los Valores para generar la salida final, conocida como Cabeza H . El proceso completo del *Scaled Dot Product Attention* con la especificación de las dimensiones de cada matriz se describe en la Figura 3.20:

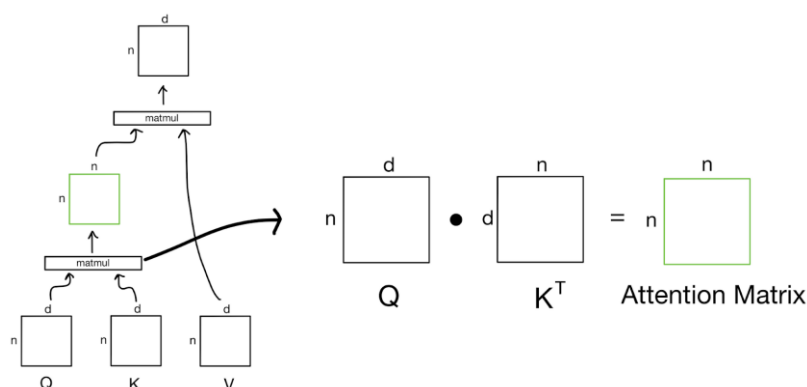


Figura 3.20: Proceso de Scaled Dot Product. Fuente: [109]

El *Scale Dot Product Attention* se aplica n veces, por lo que al final del proceso, se obtienen n cabezas de atención. Cada una de estas cabezas se concatenan y se pasan a través de una capa densa para obtener el vector final de dimensión incrustada D , llamado Z . Finalmente el vector Z pasa a través de varios bloques de codificación para proporcionarnos el vector de contexto final C .

La arquitectura general de todo el mecanismo de atención se ilustra en la Figura 3.21:

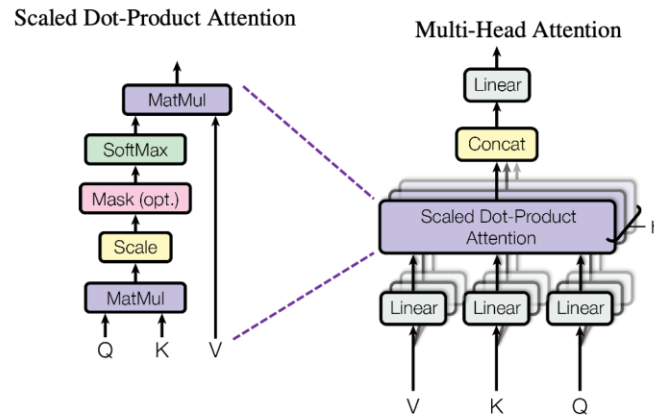


Figura 3.21: Arquitectura del Mecanismo de Atención. Fuente: [106]

3.2.8.3. MLP Head

Una vez obtengamos nuestro vector de contexto C, nos enfocaremos únicamente en el token de contexto c0 para fines de clasificación. Este c0 se introduce en una cabeza MLP. Esta cabeza MLP tiene una capa oculta con función de activación tangente hiperbólica durante la etapa de pre-entrenamiento y una sola capa lineal durante la etapa de ajuste fino. Esta MLP nos proporcionará el vector de probabilidad final para predecir la clase.

3.2.9. Self Supervised Learning

El *Self Supervised Learning* (SSL por sus siglas en inglés) es una técnica de aprendizaje automático en la que los datos de entrenamiento se etiquetan de forma autónoma o automática, sin necesidad de etiquetado manual humano [110]. Estos modelos suelen estar basados en redes neuronales profundas o en otros modelos como *40ecisión lists* [111]. El modelo aprende en dos pasos: En primer lugar, se aborda la tarea mediante una tarea auxiliar o de clasificación de pretexto que emplea pseudoetiquetas para configurar los parámetros iniciales del modelo [112]. Luego, la tarea principal se lleva a cabo utilizando métodos de aprendizaje supervisado o no supervisado [113].

Existen dos enfoques o tipos de aprendizaje autosupervisado que se explicarán a continuación.

3.2.9.1. Contrastive Learning

Este tipo de aprendizaje autosupervisado usa muestras positivas (relacionadas con el objetivo) y negativas (no relacionadas con el objetivo) [114]. Este enfoque tiene como objetivo estructurar el espacio latente de manera que las incrustaciones de

muestras similares estén cerca entre sí mientras que las de muestras diferentes estén lejos [4].

Gracias a este enfoque se han desarrollado varias técnicas como la propuesta en [115] y también se ha podido demostrar que mejora el rendimiento de algunos enfoques supervisados en conjuntos como Imagenet [116].

3.2.9.2. *Non Contrastive Learning*

El enfoque no contrastivo utiliza únicamente ejemplos positivos [114] y aunque esto pueda parecer contraintuitivo al solo observar datos de una clase, se demostró en [117] que podía aprender representaciones efectivas independientemente de la falta de ejemplos negativos.

El enfoque no contrastivo utiliza un predictor adicional y una operación de stop-gradient. El método no contrastivo SimSiam [118], ha demostrado la necesidad del predictor y la operación stop-gradient para evitar un colapso representacional en el modelo.

3.3. Segmentación semántica

La segmentación de imágenes es el proceso de dividir una imagen en regiones con el objetivo de identificar y separar áreas específicas de interés en la imagen. Estas áreas pueden representar objetos, bordes, texturas u otras características visuales distintas.

Entre las aplicaciones más importantes de esta técnica, podemos destacar: análisis de imágenes médicas, vehículos autónomos, vigilancia por vídeo y realidad aumentada [119].

Dentro de la segmentación, existen distintos tipos, de entre las que destacan:

1. Segmentación semántica: Se asigna una etiqueta a cada píxel de una imagen.
2. Segmentación en instancias: Identifica clases de objetos y también distingue instancias individuales de esos objetos en una imagen.
3. Segmentación panóptica: Combina la segmentación semántica y la segmentación en instancias.

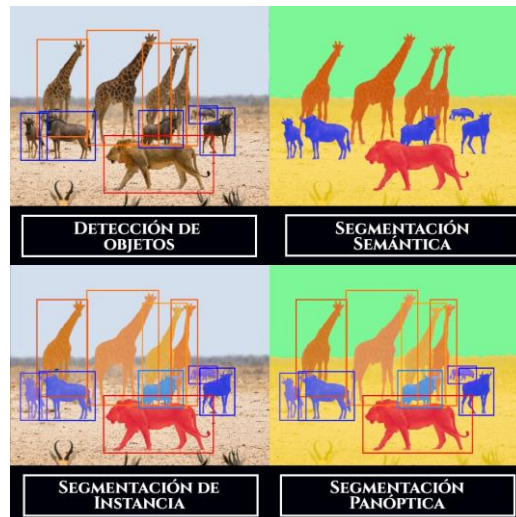


Figure 3.22: Tipos de segmentación. Fuente: [120]

Por otro lado, existen numerosas técnicas para realizar tareas de segmentación. En un principio, esta tarea se realizaba mediante técnicas de *thresholding* [121]. Más tarde, se enfocaron en utilizar técnicas basadas en los histogramas [122], por crecimiento de regiones [123] o basados en técnicas de Clustering, más específicamente en *k-means* [124]. Después, se incluyeron algoritmos más novedosos, donde podemos destacar la popularidad de los métodos basados en campos aleatorios condicionales y de Markov [125][126] o '*sparsity based*' [127].

Actualmente, los métodos que dominan las técnicas de segmentación son métodos basados en *Deep Learning*. Entre estos, podemos destacar los *Fully Convolutional Network* [128], *Attention Based Models* [129] o *Encoder-decoder Based Models* [130].

En cuanto al rendimiento de estos algoritmos, existen diversas métricas con las que podemos medir el desempeño de la segmentación, entre las que destacamos:

1. *Pixel Accuracy*: Proporción de píxeles correctamente clasificados dividida entre el número total de píxeles.
2. *Intersection Over Union (IoU)*: También conocida como índice de Jaccard, se define como el área de intersección entre la segmentación predicha y la segmentación verdadera. Esta intersección se divide entre el área de unión de las dos segmentaciones. Este índice varía entre 0 y 1, siendo 0 ninguna superposición y siendo 1 total superposición.
3. *Mean IoU*: Se define como la media del IoU sobre todas las clases.
4. *Dice Coefficient*: Se puede definir como el doble del área de superposición entre los mapas predichos y de referencia, dividido entre el número total de píxeles.

4. Análisis del problema

En esta sección, destacamos la importancia de la solución elegida, presentando un análisis exhaustivo del problema en cuestión. Pondremos en relieve cada etapa que conforma nuestra solución, resaltando la selección cuidadosa entre múltiples alternativas disponibles. Además, profundizaremos en la naturaleza de los datos que hemos empleado, subrayando su relevancia para el éxito del proyecto. No pasaremos por alto la evaluación del contexto legal y ético asociado a nuestra propuesta, lo que contribuye a garantizar que nuestro enfoque sea sólido desde todos los aspectos, cumpliendo así con los más altos estándares de integridad y responsabilidad en la investigación.

4.1. Análisis de los datos

Inicialmente, se tenía la intención de emplear un conjunto de datos que comprendía las máscaras de segmentación de IOTA del proyecto Coastal Erosión from Space, los mapas LULC del proyecto Corine Land Cover [131] e imágenes Sentinel-2 [1]. Estas imágenes, que debían descargarse desde [132] debían coincidir con las áreas de interés delineadas por los mapas LULC de IOTA y Corine Land Cover. Los sensores pasivos, como los montados a bordo de Sentinel-2, realizan medidas solo en condiciones meteorológicas despejadas (sección 3.1.2). Para mitigar el efecto meteorológico sobre la calidad de las imágenes de Sentinel-2, este TFG utiliza imágenes correspondientes a varias épocas. No obstante, a lo largo del desarrollo de este trabajo, surgieron otros obstáculos (sección 4.1.1) que hicieron imposible utilizar estos datos tal como se había planeado.

Como resultado de estos desafíos y limitaciones, se tomó la decisión de cambiar la fuente de datos utilizada en el proyecto. En particular, se optó por emplear los datos proporcionados por el proyecto Open Sentinel Map [34] como una alternativa viable. Esta modificación en la elección de datos no solo requirió una adaptación en el enfoque del análisis, sino que también abrió nuevas oportunidades y desafíos que se explorarán detalladamente en esta sección. A lo largo de esta introducción, se destacará la importancia de entender el contexto y las razones detrás de esta transición en los datos, lo que proporcionará una base sólida para el análisis posterior y el logro de los objetivos del proyecto.

4.1.1. Conjunto de datos IOTA, CLC y Sentinel 2.

En este contexto, el objetivo principal era fusionar tres conjuntos de datos diferentes para crear una base de datos coherente. Estos tres conjuntos de datos incluían las máscaras de segmentación generadas por el modelo IOTA, los mapas de Uso y Cobertura del Suelo del proyecto Corine Land Cover y los mapas de Sentinel 2 del programa Copernicus.

Las máscaras de segmentación de IOTA provienen del modelo mencionado anteriormente en el proyecto de Erosión Costera. Este conjunto de datos abarca un

período de 25 años e incluye imágenes recopiladas de Sentinel 2. Estas máscaras representan las predicciones del modelo en más de 16 áreas de estudio distribuidas en Canadá, España, Irlanda y el Reino Unido, lo que representa una amplia variedad de condiciones ambientales. Cada una de estas áreas tiene sus propias etiquetas, como mar, áreas urbanas, acantilados, playas de arena, vegetación de playa, cultivos, entre otras. Cada mapa de segmentación corresponde a una mezcla de imágenes en las distintas estaciones del año con el objetivo de generar un solo mapa que comprenda los cambios estacionales y los valores de reflectancia de cada imagen. Es decir, cada mapa IOTA es la clasificación de la zona de estudio para todo el año. Esto nos permitirá a la hora de descargar imágenes Sentinel 2 poder usar todas las imágenes pertenecientes a un año, consiguiendo así mitigar las limitaciones anteriormente expuestas. La Figura 4.1 muestra un ejemplo de la máscara de segmentación correspondiente a la zona de Dublín:



Figura 4.1: Ejemplo de máscara de segmentación de IOTA. Fuente: Propia.

Las imágenes se presentan en formato TIFF, y es evidente que contienen una gran cantidad de fondo que carece de información relevante. Esta presencia innecesaria de fondo aumenta el tamaño de la imagen y, por ende, su consumo de recursos. Para abordar esta cuestión, se llevó a cabo un proceso de recorte de las imágenes utilizando el software QGIS [133] (un Sistema de Información Geográfica de software libre y de código abierto). El propósito de este procedimiento era eliminar el fondo que carecía de

información útil, permitiendo así conservar principalmente la porción de la imagen que representaba la mayor área de interés.

Por otro lado, tenemos las imágenes del proyecto Corine Land Cover (CLC). Este proyecto tiene como objetivo desarrollar una base de datos sobre la cobertura y uso del suelo en el territorio de la unión Europea. Estas imágenes son recogidas por los satélites LandSat, SPOT y desde 2018 también Sentinel 2. El registro de esta base de datos comenzó en 1985, actualizándose en los años 1990, 2000, 2006, 2012 y 2018. Categoriza la cobertura terrestre en 44 clases y utiliza una Unidad Mínima de Mapeo (UMM) de 25 hectáreas para fenómenos de área y un ancho mínimo de 100 metros para fenómenos lineales.

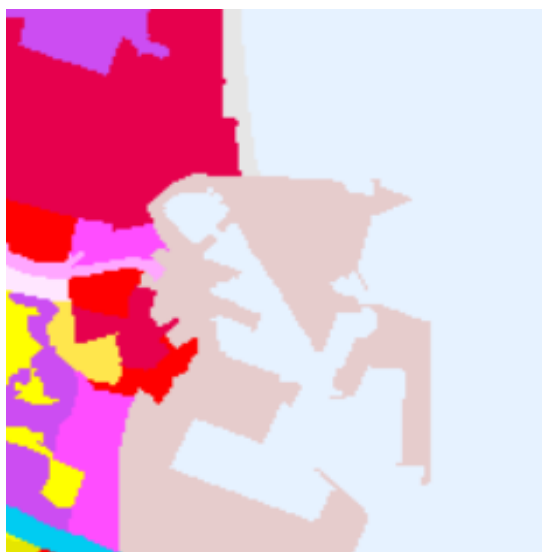


Figura 4.2: Ejemplo de una captura de CLC de la zona de Valencia. Fuente: [131].

El objetivo con respecto a los mapas CLC consistía en lograr una coherencia con las clases presentes en los mapas de segmentación de IOTA. Esto representaba un desafío significativo dado que los mapas CLC registran un total de 44 clases. Además, se planeaba realizar recortes de las áreas de interés en los mapas CLC y abordar las diferencias en la resolución espacial entre ambos conjuntos de datos. Mientras que los mapas CLC tienen una resolución espacial de 100 metros, los mapas de IOTA cuentan con resoluciones de 10 o 30 metros.

Por otra parte, el tercer componente fundamental de este conjunto de datos consistió en las imágenes obtenidas a través del programa Copernicus por medio del satélite Sentinel 2. La misión Sentinel 2 se compone de dos satélites, S2-A y S2-B, cuyo propósito principal es supervisar la variabilidad en las condiciones de la superficie terrestre, abarcando una extensa área de 290 km [134].

El instrumento principal de la misión SENTINEL-2 es el MultiSpectral Instrument (MSI), que adquiere datos de imágenes mientras el satélite se desplaza en su órbita. Este instrumento descompone la luz entrante en longitudes de onda individuales correspondientes a diversas bandas.

La resolución espacial se divide en tres opciones posibles (10, 20 y 60 metros), y los detalles de la resolución espectral se encuentran disponibles en la Tabla 4.1:

Spatial Resolution (m)	Band Number	S2A		S2B	
		Central Wavelength (nm)	Bandwidth (nm)	Central Wavelength (nm)	Bandwidth (nm)
10	2	492.4	66	492.1	66
	3	559.8	36	559.0	36
	4	664.6	31	664.9	31
	8	832.8	106	832.9	106
20	5	704.1	15	703.8	16
	6	740.5	15	739.1	15
	7	782.8	20	779.7	20
	8a	864.7	21	864.0	22
	11	1613.7	91	1610.4	94
	12	2202.4	175	2185.7	185
60	1	442.7	21	442.2	21
	9	945.1	20	943.2	21
	10	1373.5	31	1376.9	30

Tabla 4.1: Resolución espacial y espectral de Sentinel 2. Fuente: [134]

El proceso de descarga de estas imágenes se llevaba a cabo de la siguiente manera: se creaba un archivo .shp (shapefile) que representaba la zona de interés en el mapa de IOTA. Es decir, el mapa LULC generado por IOTA era el que designaba las zonas geográficas de las que descargar mapas de Sentinel-2. Este archivo se generaba recortando la región de interés del mapa de segmentación y luego convirtiéndolo a formato .shp, todo ello mediante el uso de QGIS. Posteriormente, se automatizaba la descarga de las imágenes Sentinel 2 correspondientes a esa área utilizando un script de Python. Este script requería el nombre de usuario y la contraseña del sitio web Copernicus Open Access Hub [132], el tipo de imagen a descargar, el rango de fechas deseado y el nivel máximo de cobertura de nubes, además del shapefile previamente creado. Un ejemplo de imagen de Sentinel-2 descargada se proporciona en la Figura 4.3:



Figura 4.3: Visualización de imagen Sentinel 2 con QGIS. Fuente: Propia

Este paso resultaba crítico en el proceso de trabajo con el conjunto de datos por diversas razones. En primer lugar, el script descargaba una gran cantidad de imágenes, algunas de las cuales no coincidían exactamente con la zona del mapa de IOTA. Además, la cantidad de información descargada era de un volumen de cientos de gigabytes de información, ya que cada imagen descargada estaba compuesta por 13 archivos .jp2 correspondientes a las diferentes bandas capturadas por el satélite.

El desafío de almacenar estas imágenes, dadas sus enormes dimensiones y la falta de precisión al alinear el mapa de IOTA con las imágenes de Sentinel 2, impulsaron la búsqueda de un enfoque y conjunto de datos alternativos.

4.1.2. Conjunto de datos Open Sentinel Map

El conjunto de datos "OpenSentinelMap", propuesto en [34], presenta una colección de imágenes satelitales multispectrales adquiridas por el satélite Sentinel-2. Cada imagen se encuentra asociada a una celda espacial no superpuesta de 1920 metros x 1920 metros, lo que equivale a un área de 3.7 km². Además de las imágenes, el conjunto de datos incluye etiquetas por píxel que representan 15 categorías de uso de la tierra y características geoespaciales, como carreteras, edificios, agua, entre otras. Estas etiquetas se derivan de datos públicos de OpenStreetMap (OSM de sus siglas en Inglés) [135], una fuente global de información geoespacial que etiqueta entidades geográficas, como carreteras, estacionamientos y contornos de edificios, con diversos tags. Las etiquetas en "OpenSentinelMap" se generan mediante ontologías que mapean estos tags de OSM a categorías de uso de la tierra. Cabe destacar que las etiquetas no son necesariamente mutuamente excluyentes, lo que permite una representación detallada de la información geoespacial en el conjunto de datos.

En el caso de este TFG, las imágenes fueron guardadas en un servidor *sftp* de ARGANS. De este servidor, se extrajeron 9534 imágenes con sus respectivas máscaras de segmentación. Estas imágenes tenían 3 tipos de resolución (10, 20 y 60 metros) con distintas bandas en cada una de estas resoluciones. Inicialmente este TFG pretendía utilizar una combinación de las 13 bandas, adaptando las diferentes resoluciones y ajustando la geometría y tamaño de píxel. Se hicieron varias pruebas con 13, 8 y 6 bandas mezcladas mediante interpolación bilineal (calcula el valor de cada píxel en la imagen de destino como una combinación lineal de los píxeles vecinos en la imagen de origen) a una resolución objetivo de 60 metros. El resultado de usar un mayor número de bandas produjo dos inconvenientes: mayor consumo de recursos tanto a (1) nivel de almacenamiento como a (2) nivel de computación. Esto provocó que el conjunto de datos tenía que ser recortado y se debía utilizar un menor número de imágenes. En su lugar, se decidió utilizar únicamente la combinación de las bandas 4, 3 y 2 correspondientes a las bandas Rojo, Verde y azul (RGB, de las siglas en inglés Red, Green and Blue) y una resolución espacial más alta de Sentinel-2 (10 m).

En cuanto a las etiquetas (Figura 4.4), utilizamos únicamente a la categoría *OSM Land Use*, que hacen un total de 13 etiquetas: (1) No etiquetado, (2) arbolado, (3) agrícola, (4) residencial, (5) industrial, (6) comercial, (7) recreativo, (8) aeropuerto, (9) cantera, (10) militar, (11) arena desértica, (12) roca de montaña y (13) otro. La distribución de estas etiquetas estaba desbalanceada, como puede observarse histograma de la Figura 4.4, donde se muestran que las clases 0, 1, 2 y 12 tenían cantidades de píxeles máximas (Cantidad > 1.6 1e7 píxeles) y el resto de las clases tenían cantidades variables pero siempre por debajo de 0.4 1e7 píxeles.

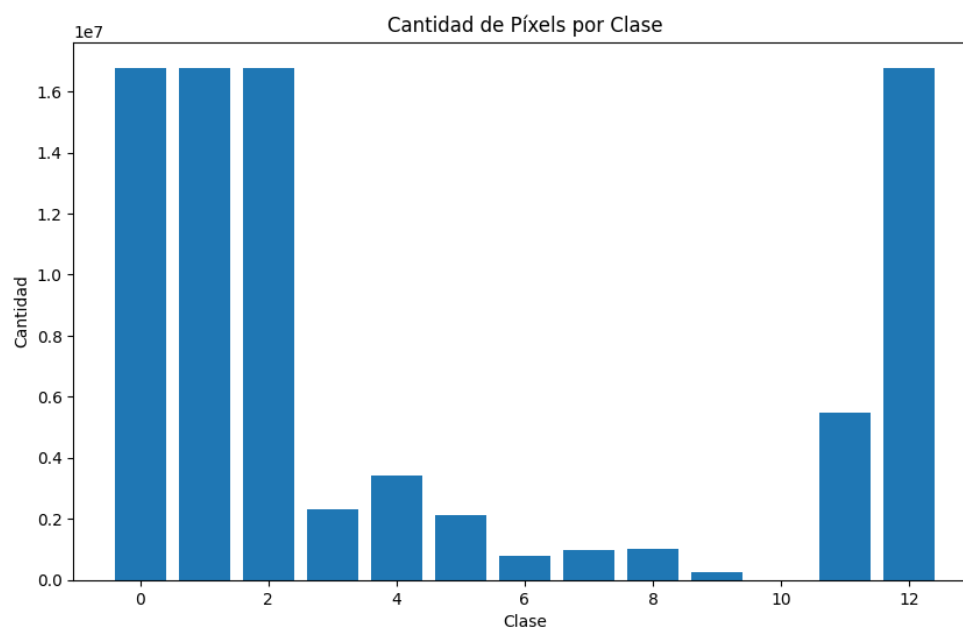


Figura 4.4: Cantidad de píxeles por clase en las imágenes. Fuente: Propia.

El desbalanceo puede dificultar que un modelo aprenda de manera equitativa todas las clases, lo que puede llevar a un rendimiento deficiente en las clases minoritarias y medidas de evaluación sesgadas. Este problema fue tratado y se explicará en la sección 4.3. La Figura 4.5 muestra un ejemplo de una imagen original y su máscara de segmentación:

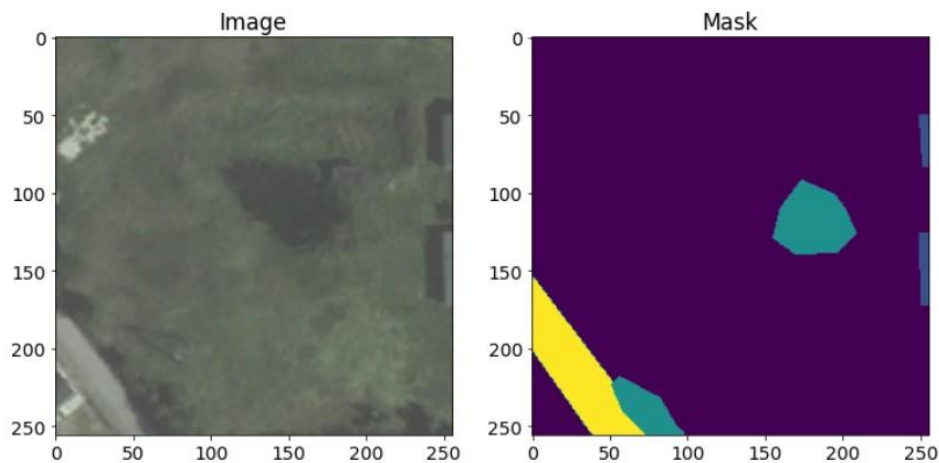


Figura 4.5: Imagen original y máscara de segmentación. Fuente: Propia.

4.2. Análisis del marco legal y ético

En el desarrollo de cualquier proyecto, es imperativo considerar y cumplir con un sólido marco legal y ético. Este análisis aborda diversos aspectos esenciales, como la protección de datos, la propiedad intelectual o la ética, que deben ser cuidadosamente evaluados en el contexto de nuestro proyecto, que se centra en la creación de un modelo avanzado de segmentación semántica de mapas de Uso y Cobertura del Suelo (LULC) a partir de imágenes satelitales obtenidas del conjunto de datos "OpenSentinelMap."

Protección de Datos:

El procesamiento de datos es fundamental en este proyecto, y debemos abordar cuestiones de protección de datos. Es crucial rastrear el origen de los datos, su almacenamiento y asegurarse de cumplir con todas las normativas pertinentes. Dado que trabajamos con imágenes satelitales, es fundamental garantizar que no se incluyan datos que puedan identificar a personas o propiedades privadas en nuestros resultados.

Propiedad Intelectual:

La cuestión de la propiedad intelectual debe ser considerada meticulosamente. Esto se refiere a la licencia que se aplicará a nuestro software y a cualquier otro elemento que forme parte del proyecto. Es fundamental respetar los derechos de autor y asegurarse de tener los permisos adecuados, especialmente cuando se utilizan imágenes o datos de terceros en el producto final.

Ética

Este proyecto de desarrollo de un modelo de segmentación semántica conlleva importantes consideraciones éticas. Entre los aspectos más relevantes se encuentran la posibilidad de discriminación inadvertida, la protección de la privacidad en imágenes satelitales, la transparencia y explicabilidad del modelo, la equidad en el acceso a sus

beneficios, la responsabilidad en caso de errores y el impacto ambiental y social de las decisiones basadas en el modelo. Además, es fundamental abordar la recopilación de datos de manera ética y cumplir con las regulaciones legales y éticas pertinentes en el campo de la teledetección y la inteligencia artificial.

4.3. Solución propuesta

A continuación se proporcionará una visión detallada y completa del sistema desarrollado para conseguir los objetivos descritos al principio de la memoria (sección 1.2). Para ello, se utilizará un enfoque visual a través de un diagrama gráfico (Figura 4.6) que ilustrará de manera concisa cada etapa del proceso. Después, se desglosarán los componentes clave de dicho esquema, detallando su funcionalidad y contribución a la consecución de los objetivos.

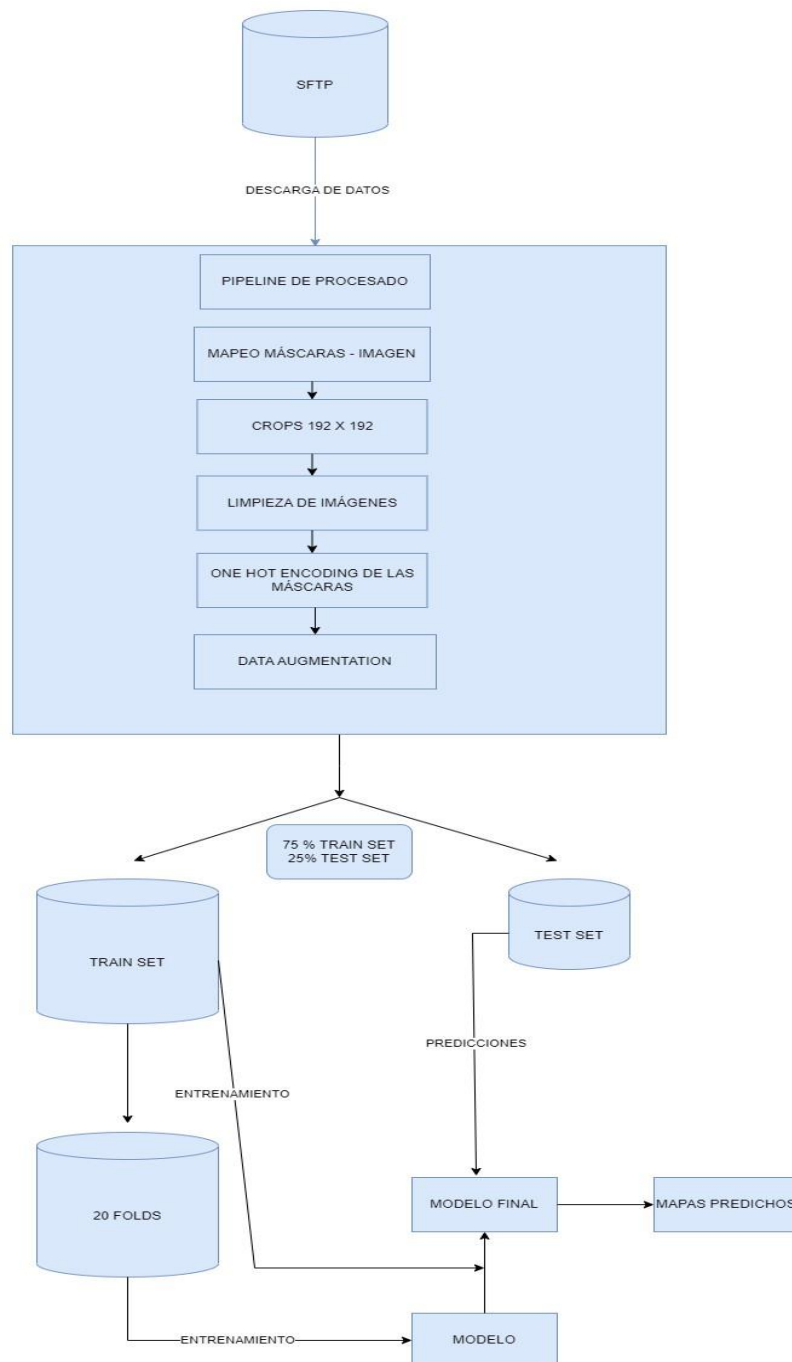


Figura 4.6: Solución propuesta. Fuente: Propia

Podemos descomponer la solución propuesta en tres fases claramente definidas y completamente relacionadas con la metodología explicada en la sección 1.4:

1. Primera Fase: Adquisición de Datos

En esta etapa inicial, llevamos a cabo la obtención de datos, que incluyen imágenes y máscaras de segmentación, desde el servidor SFTP. Este proceso se ejecuta mediante la herramienta FileZilla. La herramienta permite establecer una conexión con el servidor utilizando credenciales de usuario y contraseña para descargar las imágenes almacenadas en el servidor a nuestro propio servidor local. Esta fase

correspondería a la segunda fase de la metodología CRISP-DM: Entendimiento de los datos.

2. Segunda Fase: Procesamiento de Imágenes

La segunda fase se enfoca en un conjunto de operaciones de procesamiento de imágenes conocido como pipeline de preprocesado de imágenes. Este pipeline constituye la base sobre la cual se construye un proyecto de visión por computadora o procesamiento de imágenes. Su objetivo principal es transformar los datos visuales crudos en una forma óptima para la tarea en cuestión. Esto se logra asegurando que las imágenes estén libres de errores, sean coherentes y resalten las características relevantes. Toda esta fase correspondería con el tercer paso de CRIPS-DM: Preparación de datos.

En nuestro caso, las imágenes pasan por varios pasos durante este pipeline:

- a) Asociación Imagen-Máscara: Inicialmente, se realiza la asociación de cada imagen con su correspondiente máscara de segmentación. Para cada ubicación y fecha, existe un conjunto de imágenes junto con una sola máscara de segmentación. Es necesario que cada imagen esté asociada a una máscara para el posterior proceso de entrenamiento y ahí reside la necesidad de este paso.
- b) Recortes de Imágenes (Crops): Luego, procedemos a realizar recortes en las imágenes para reducir su tamaño a 192x192 píxeles. Esto se hace con el propósito de aumentar el conjunto de datos y reducir la carga computacional en el entrenamiento del modelo.
- c) Limpieza de Imágenes: Se identifica que la etiqueta 'desconocido' abarca una cantidad significativamente mayor de píxeles en comparación con otras etiquetas. Este desequilibrio en el dataset es problemático, ya que no aporta información útil. Para abordar este problema, se implementa una función que elimina los recortes de imágenes cuyas máscaras de segmentación contienen menos del 65% (varios *thresholds* fueron probados siendo este el que mejor resultado aportó) de información útil, donde la información útil se refiere a los píxeles con etiquetas distintas de 'desconocido'.
- d) Codificación One Hot: Una vez que las imágenes están limpias y se han eliminado las que carecen de información relevante, se procede a realizar una codificación One Hot. Esto implica representar cada clase como un vector binario, donde un único elemento es igual a 1 (indicando la clase) y todos los demás elementos son iguales a 0.
- e) Aumento de Datos: Finalmente, se lleva a cabo un aumento de datos en función de las imágenes existentes. Esto se logra mediante una función de la biblioteca Keras, que define un rango de rotación aleatoria de 20 grados, un rango de desplazamiento horizontal y vertical aleatorio de 0.1 (lo que equivale a un máximo de ± 19.2 píxeles), un rango de zoom aleatorio de 0.2 y un volteo horizontal aleatorio.

Después de este proceso de preprocesamiento, el conjunto total de imágenes se incrementa de 7,627 imágenes a 13,585 imágenes.

3. Tercera Fase: Modelado

La fase final involucra el modelado. Esta fase correspondería con la 4 y 5 de la metodología: Modelado y Evaluación. Comienza con la partición de los datos en diferentes conjuntos:

- a) Partición Inicial: En un primer paso, los datos se dividen aleatoriamente en un 75% para el conjunto de entrenamiento (*train*) y un 25% para el conjunto de prueba (*test*).
- b) 20 Fold Cross Validation: El conjunto de entrenamiento se divide posteriormente en 20 folds para llevar a cabo una validación cruzada de 20 iteraciones. Esta etapa se utiliza para entrenar un modelo inicial y probar los hiperparámetros, aprovechando el *Cross Validation* para la validación de los resultados.
- c) Una vez que se ha validado que el modelo con los hiperparámetros ajustados no sufre de sobreajuste (*overfitting*) o subajuste (*underfitting*), se procede a crear un modelo final. Este modelo se entrena utilizando todo el conjunto de entrenamiento de una sola vez.

Finalmente, una vez que este último modelo ha finalizado su entrenamiento, se utiliza el conjunto de prueba, que el modelo nunca ha visto previamente, para realizar predicciones, generar los mapas de segmentación correspondientes y evaluar su rendimiento en un conjunto de datos desconocido.



5. Descripción de modelos

En este capítulo, se detallarán las diversas componentes que conforman las arquitecturas utilizadas, todas diseñadas con el propósito de lograr el objetivo previamente expuesto: desarrollar una red neuronal de fácil reentrenamiento capaz de llevar a cabo la segmentación de imágenes satelitales en un conjunto predefinido de clases. Para cumplir este objetivo, este TFG ha sometido a prueba tres modelos diferentes que son: [1] VVG16 Unet, [2] Attention Unet y [3] Swin Transformer. A continuación, se describe cada uno de los modelos expuestos.

5.1. VGG16 Unet

Este modelo es propuesto en [1] donde los autores proponen la combinación de la arquitectura VGG16, propuesta por primera vez en [2] y la arquitectura Unet, propuesta en [3].

La arquitectura Unet consta de dos partes: *encoder* y *decoder*. La idea propuesta en el paper es la de utilizar la arquitectura VGG16 preentrenada mediante el conjunto de datos 'ImageNet' [4] en la parte del encoder, pretendiendo mejorar la extracción de características de la imagen. Esta arquitectura se muestra en la Figura 5.1:

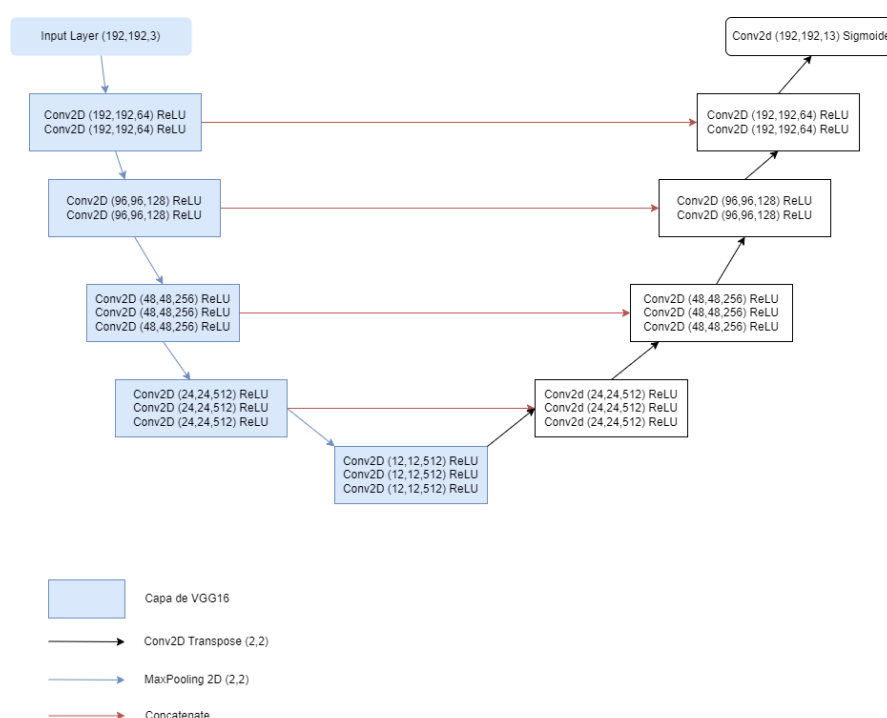


Figura 5.1: Arquitectura VGG16 Unet Modificada. Fuente: Propia.

Esta red ha experimentado evoluciones con respecto a su versión original. Se ha optimizado el código para que la red sea adaptable a imágenes de diversos

tamaños y con distintos números de clases. Así, lo que conseguimos, es una red general en la que simplemente especificando tamaño y número de clases puede ser usada para diferentes tipos de zonas geográficas con imágenes con diferentes características. Adicionalmente, la mayor parte de las capas del *encoder* han sido descongeladas con el propósito de entrenar algunas de estas capas preentrenadas con las imágenes del problema que estamos tratando. Con esto se pretende que el *encoder* aprenda a extraer características propias de las imágenes satelitales con mayor eficacia. Por lo tanto, nuestra VGG16 Unet modificada consta de la siguiente arquitectura:

1. *Input Layer*: Es la capa en la que introducimos las imágenes. Para nuestro caso concreto las dimensiones de la capa son de (192,192,3), que corresponden a imágenes RGB de tamaño 192 * 192.

2. *Encoder*: 4 capas convolucionales con la arquitectura propia del VGG16. Estas capas han sido descongeladas (se permite que sus pesos se ajusten durante el proceso de entrenamiento) y entrenadas con el conjunto de datos del problema con el propósito comentado anteriormente. Estas capas también están conectadas al *decoder* por conexiones residuales.

3. *Bridge*: Esta capa actúa como puente entre el *encoder* y el *decoder*. Corresponde también a la arquitectura de la VGG16 y también ha sido reentrenada con el conjunto de datos del problema.

4. *Decoder*: Se encarga de tomar la representación de características generada por el *encoder* y la expande de nuevo a las dimensiones originales de la imagen. El proceso de expansión se realiza a través de capas de convolución transpuesta o de upsampling, que aumentan gradualmente el tamaño de la representación. El decoder está compuesto por 4 capas convolucionales con 512,256,128 y 64 filtros respectivamente.

5. Capa de salida: Devuelve la máscara de segmentación predicha por el modelo. En nuestro caso, el tamaño que devuelve va en forma de (altura imagen, ancho imagen, número de clases) y este parámetro es adaptable a otro problemas.

Con todos los cambios mencionados, finalmente nuestra arquitectura modificada tiene un total de 19,855,437 parámetros entrenables.

La elección de este modelo, la VGG16-Unet, se basa en una serie de consideraciones clave. En primer lugar, decidimos probar un modelo preentrenado en un contexto diferente al de la teledetección, ya que esto nos permitiría explorar cómo un modelo entrenado en un dominio ajeno podría adaptarse y aprender patrones relevantes en nuestras imágenes satelitales. Esta transferencia de conocimientos es esencial, ya que puede ahorrar una cantidad significativa de tiempo y recursos de entrenamiento en comparación con iniciar un modelo desde cero.

En segundo lugar, la arquitectura de red VGG16 es conocida por su rendimiento en tareas de clasificación de imágenes. Esta red neuronal convolucional

ha demostrado ser eficaz en la extracción de características relevantes de las imágenes y en la obtención de representaciones de alto nivel que pueden ser útiles para la segmentación de imágenes. Al utilizar esta arquitectura, tenemos la ventaja de contar con capas preentrenadas que han aprendido a reconocer características comunes en imágenes, lo que puede ser beneficioso para nuestro proyecto.

Además, al aprovechar las capas más externas del modelo VGG16, podemos beneficiarnos de la capacidad de estas capas para detectar características generales, como bordes, formas y texturas. Esto es fundamental para la segmentación de imágenes, ya que nos permite identificar regiones de interés y separarlas del fondo de manera efectiva.

En última instancia, nuestro objetivo principal en este proyecto es lograr una segmentación precisa y generalizable de áreas geográficas diversas. La combinación de un modelo preentrenado con la arquitectura VGG16 nos brinda una sólida base para alcanzar este objetivo, al tiempo que acorta los tiempos de entrenamiento y mejora la adaptabilidad del modelo a diferentes contextos geográficos.

5.2. Attention Unet

La siguiente red empleada es una variante de la arquitectura U-Net que incorpora un mecanismo de atención, el cual fue previamente detallado en el apartado 3.2.7. La Figura 5.2 muestra esta arquitectura:

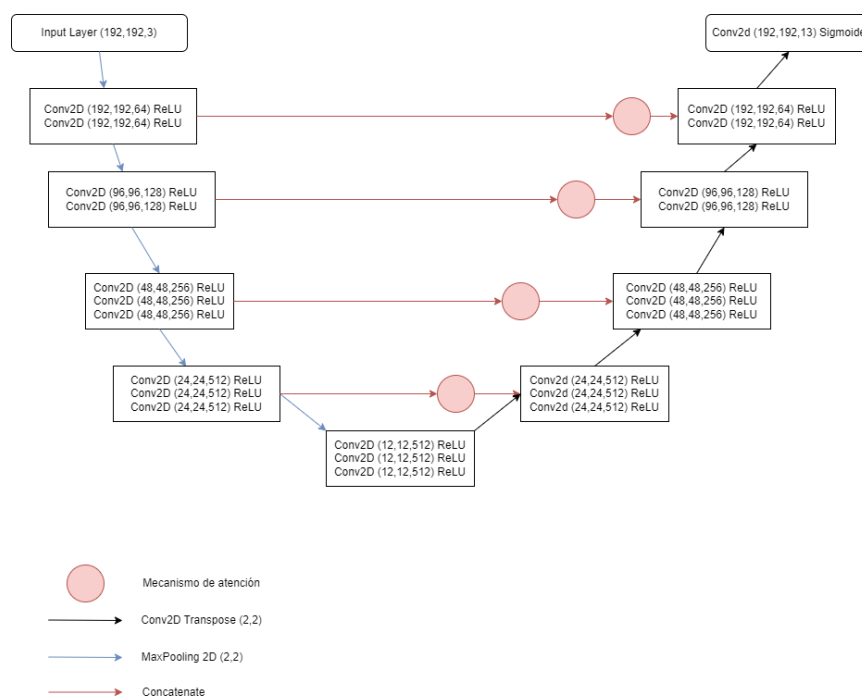


Figura 5.2: Arquitectura Attention Unet Modificada. Fuente: Propia.

Se trata de una red muy parecida a la anterior. Las diferencias residen en que ya no se utiliza un modelo preentrenado en la parte del *encoder* ni tampoco una arquitectura VGG16 para la extracción de características, si no que se utilizan capas

de convolución y reducción de tamaño que solamente se entrenan con las imágenes de nuestro problema. El *encoder* de esta red está compuesto por 4 bloques de convolución con un número de filtros de 64,128,256 y 512 respectivamente.

Podemos observar que la parte del puente o *bridge* tampoco corresponde ya a una capa de la VGG16. También se puede apreciar que en cada conexión residual entre *encoder* y *decoder* se ha añadido un mecanismo de atención. En cuanto a la parte del *decoder* y las capas de entrada y salida, son totalmente iguales en ambos modelos.

Este modelo consta de 12 millones de parámetros entrenables, una cantidad menor que el modelo anterior.

La elección de utilizar un modelo basado en Attention U-Net se fundamenta en su capacidad para abordar de manera efectiva problemas de segmentación de imágenes. Este enfoque incorpora un mecanismo de atención que permite que el modelo se centre en regiones específicas de la imagen, dando prioridad a las áreas más relevantes y significativas en el proceso de segmentación. La atención selectiva es especialmente valiosa en la teledetección, donde las imágenes suelen contener una gran cantidad de información, pero solo algunas áreas son cruciales para nuestros objetivos.

Con la incorporación de la Attention U-Net, esperamos que el modelo sea capaz de identificar y destacar características geográficas importantes, como cuerpos de agua, límites de terrenos, vegetación o áreas urbanas, mientras que descarta detalles menos relevantes, como ruido o áreas uniformes. Esto no solo puede mejorar la precisión de la segmentación, sino también acelerar el proceso de entrenamiento al reducir la carga de información innecesaria.

5.3. Swin Transformer

El siguiente modelo está basado en el modelo utilizado en [4].

Este artículo propone la combinación de una solución basada en aprendizaje profundo, específicamente en una arquitectura Transformer, con el aprendizaje auto-supervisado de tipo *Contrastive Learning*, con el fin de procesar datos de observación terrestre.

El paper propone pre-entrenar modelos con grandes conjuntos de datos sin etiquetar para después usar las representaciones aprendidas en tareas tanto de clasificación como segmentación de cobertura terrestre. Gracias a este enfoque auto-supervisado, se demuestra que no es necesario un enorme conjunto de datos para realizar '*fine-tuning*' lo que supone una gran ventaja a la hora de adaptar el modelo a una tarea propia. La arquitectura del modelo (Figura 5.3) del artículo es la siguiente:

Entrenamiento de modelos de Aprendizaje Automático para generación de mapas de Uso y Cobertura del Suelo (LULC) utilizando datos de satélite

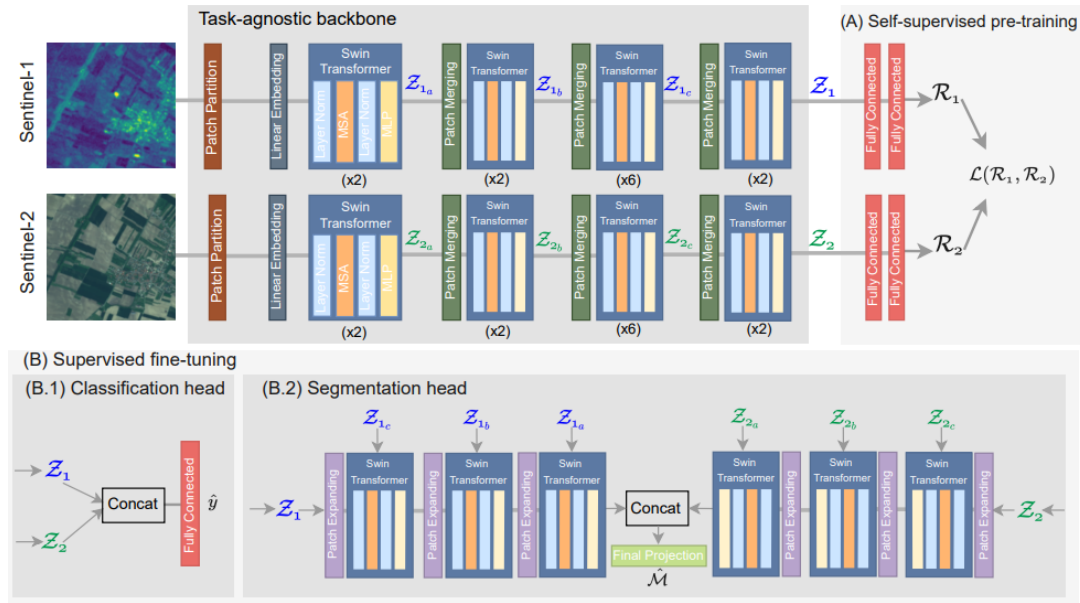


Figura 5.3: Arquitectura del Swin Transformer. Fuente: [4]

La arquitectura de esta red consta de 2 partes: una primera parte que recibe las entradas de las imágenes Sentinel 1 y Sentinel 2 y pasan por dos flujos de Swin Transformers (una adaptación de la arquitectura transformer explicada en el punto 3.2.8, que tiene como ventaja su enfoque de ventana desplazada, lo que reduce la complejidad computacional respecto al transformer estándar [4]) usando un enfoque de auto-aprendizaje contrastivo. Toda esta parte corresponde a la parte (A) de la imagen. Después, para el aprendizaje supervisado, se alimenta con las representaciones aprendidas en la parte (A) una cabeza de clasificación (B.1) y una de segmentación (B.2).

La decisión de emplear un modelo Swin Transformer preentrenado se basa en el éxito demostrado por este en investigaciones recientes [4]. En particular, en el artículo de referencia, este tipo de Transformer logró resultados destacados en una serie de métricas relevantes para problemas de segmentación de datos de teledetección. Este desempeño sobresaliente en tareas similares refuerza la confianza en su capacidad para abordar eficazmente la segmentación de imágenes en nuestro proyecto.

Es importante destacar que los Transformers están emergiendo como el estado del arte en el campo de la segmentación de imágenes. Estas arquitecturas han demostrado ser especialmente eficaces en la captura de relaciones de largo alcance y la comprensión de contextos complejos en las imágenes, lo que es fundamental en la teledetección, donde la información geoespacial puede ser intrincada y diversa.

En resumen, la elección de utilizar un Swin Transformer preentrenado se fundamenta en su destacado desempeño en investigaciones previas y su idoneidad para abordar desafíos de segmentación de imágenes en el contexto de la teledetección, donde la calidad y la precisión de la segmentación son de suma importancia.

6. Recursos utilizados

El presente capítulo se centra en describir y analizar las herramientas tecnológicas que se han empleado en el desarrollo de este proyecto de final de carrera. Estas herramientas se agrupan en dos categorías principales: software y hardware.

La elección de las herramientas tecnológicas adecuadas es un componente crítico en la consecución de los objetivos y metas del proyecto. Por lo tanto, comprender y documentar las soluciones tecnológicas utilizadas es esencial para proporcionar una visión clara de la infraestructura y el entorno de trabajo que respaldaron la investigación y el desarrollo de este proyecto.

En la primera sección de este capítulo, se abordarán las herramientas de software que desempeñaron un papel fundamental en la implementación y ejecución de las tareas requeridas.

La segunda sección se centrará en el hardware que respaldó la infraestructura tecnológica de este proyecto.

Este capítulo tiene como objetivo proporcionar una visión completa de las herramientas tecnológicas que formaron la base de este proyecto, destacando su importancia y relevancia en el contexto de la investigación y el desarrollo. Al comprender la infraestructura tecnológica utilizada, los lectores podrán evaluar mejor la validez y la aplicabilidad de los resultados obtenidos, así como obtener información valiosa para proyectos futuros.

6.1. Herramientas tecnológicas

6.1.1. Software

La implementación y ejecución de este proyecto se basaron en el lenguaje de programación Python, que ofrece una amplia gama de bibliotecas y herramientas para tareas de investigación y desarrollo. A lo largo de este proyecto, se aprovecharon diversas librerías y aplicaciones de software que desempeñaron un papel crucial en la consecución de los objetivos establecidos. A continuación, se detallan las principales herramientas de software utilizadas:

1. OpenCV, Rasterio y GDAL: Estas librerías se utilizaron para el procesamiento de imágenes, lo que permitió realizar operaciones de manipulación y análisis de datos geoespaciales de manera efectiva. OpenCV se centró en el procesamiento de imágenes, mientras que Rasterio y GDAL facilitaron la lectura y escritura de datos raster y vectoriales.

2. Matplotlib, Seaborn y Wandb (Weights & Biases): La visualización de resultados y datos desempeñó un papel crucial en la interpretación de los hallazgos. Matplotlib y Seaborn proporcionaron capacidades de visualización flexibles y personalizables, mientras que Weights & Biases (Wandb) ayudó en el seguimiento y la visualización interactiva de las métricas y resultados del modelado.

3. Scikit-learn: Esta biblioteca se utilizó para implementar las métricas de evaluación de los modelos.

4. Keras y TensorFlow: Para el modelado de datos y la partición de conjuntos de datos, se emplearon Keras y TensorFlow, dos frameworks ampliamente utilizados en el desarrollo de modelos de aprendizaje profundo. Estas herramientas proporcionaron una interfaz de alto nivel y una amplia gama de algoritmos para la construcción y entrenamiento de modelos de aprendizaje automático.

5. NumPy y Pandas: NumPy y Pandas se utilizaron para el tratamiento y la manipulación eficiente de datos. NumPy facilitó las operaciones numéricas y matriciales, mientras que Pandas simplificó la organización y el análisis de datos tabulares.

Además de estas librerías y herramientas específicas de Python, la gestión integral del proyecto se llevó a cabo utilizando dos herramientas de software esenciales:

1. Trello: Trello se utilizó para organizar y gestionar las tareas y actividades relacionadas con el proyecto. Esta plataforma de gestión de proyectos proporcionó una forma eficiente de planificar, seguir el progreso y coordinar las actividades de investigación y desarrollo, dividiendo las tareas en 3 bloques: pendientes, en curso y finalizadas y añadiendo una fecha de vencimiento a cada una de estas tareas.

2. GitHub: GitHub sirvió como plataforma central para el control de versiones y el alojamiento del código fuente del proyecto.

3. Teams y SharePoint para la organización de video llamadas con el equipo supervisor y la edición y supervisión de documentos.

La elección y utilización estratégica de estas herramientas de software desempeñó un papel fundamental en la consecución de los objetivos del proyecto, permitiendo la eficiencia en el desarrollo, la evaluación rigurosa de los resultados y la gestión efectiva de todas las etapas del trabajo realizado.

6.1.2. Hardware

El proyecto se benefició de un clúster de cómputo altamente eficiente compuesto por siete ordenadores interconectados. El nodo de acceso principal, vrhpcadm1.dsic.upv.es, sirvió como punto central para la preparación y gestión de trabajos. Los seis nodos de cómputo adicionales (vrhpc1.dsic.upv.es a vrhpc6.dsic.upv.es) desempeñaron un papel crucial en la ejecución de tareas de procesamiento intensivo. Cada uno de estos nodos de cómputo estaba equipado con 2 procesadores AMD EPYC 7453, cada uno con 28 núcleos, lo que proporcionó una capacidad de procesamiento significativa. Además, estos nodos contaban con 512 GB de memoria RAM y estaban equipados con 8 GPUs NVIDIA A40 de 48GB, lo que permitió el procesamiento paralelo y el entrenamiento eficiente de modelos de aprendizaje profundo. La gestión de recursos y la ejecución de trabajos en el clúster se realizaron mediante el sistema de planificación de trabajos Slurm, que aseguró una distribución eficiente de tareas en los nodos disponibles. Esta infraestructura de hardware proporcionó un entorno propicio para llevar a cabo investigaciones y análisis computacionales de alto rendimiento.

7. Experimentación y resultados

En esta sección, se presentan los resultados detallados de la investigación, que abarca la evaluación exhaustiva de tres modelos de aprendizaje profundo ampliamente reconocidos y aplicados en la tarea de segmentación semántica: el VGG16 U-Net, el Attention U-Net y el Swin Transformer.

Para lograr una comprensión profunda de los aspectos experimentales y evaluar la eficacia de los modelos, esta sección se estructura en torno a varios componentes críticos. Primero, se exploran las métricas empleadas para cuantificar el rendimiento de los modelos en términos de precisión, generalización y capacidad de discernimiento entre clases. Luego, se examina la selección de hiperparámetros que influyen en la arquitectura y la capacidad de adaptación de los modelos. La estrategia de entrenamiento y validación se desglosa para revelar cómo se abordó el proceso de optimización y adaptación de los modelos a los datos específicos. La función de pérdida, un componente esencial en el entrenamiento, se detalla en cuanto a su elección y su papel en la mejora del rendimiento del modelo.

Finalmente, los resultados obtenidos se presentan en términos de las métricas discutidas anteriormente. Se realiza una comparación crítica de los tres modelos, destacando sus puntos fuertes y debilidades en función de su capacidad para capturar con precisión las complejidades de los datos satelitales. Además, se expondrán los resultados de la segmentación de manera visual, añadiendo un componente gráfico para complementar la evaluación del rendimiento. Esta representación visual no solo enriquece nuestra comprensión de los resultados, sino que también brinda una perspectiva más intuitiva sobre cómo los modelos están logrando la tarea de segmentación.

7.1. Métricas utilizadas

Como se mencionó previamente en la sección 3.3, hay varias métricas (Figura 7.1, Figura 7.2 y Figura 7.3) disponibles para evaluar el rendimiento de un modelo de segmentación semántica. Para abordar este proyecto, se han elegido tres métricas de rendimiento distintas:

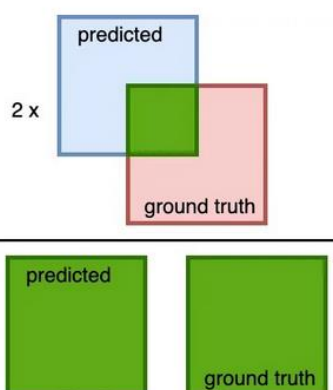
$$\text{Dice coefficient} = \frac{2 \times \text{area of overlapped (green)}}{\text{total area (green)}} = \frac{2 \times \text{area of overlapped (green)}}{\text{area of predicted (green)} + \text{area of ground truth (green)}}$$


Figure 7.1: Fórmula Dice Coefficient. Fuente: [136]

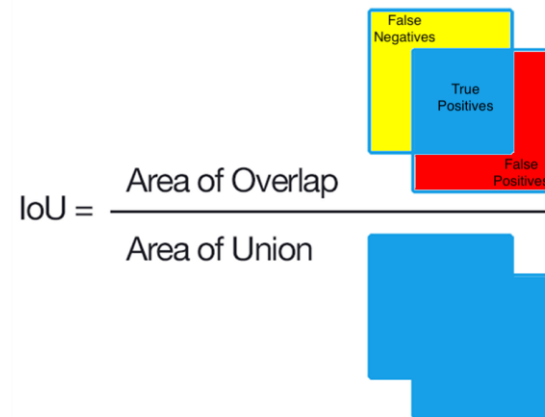


Figura 7.2: Fórmula IoU. Fuente: [137]

$$\text{Pixel Accuracy} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}$$

Figura 7.3: Fórmula de Pixel Accuracy. Fuente: [138]

La elección de estas métricas se ha fundamentado en las propiedades intrínsecas que cada una de ellas ofrece para evaluar con precisión la calidad de los modelos de segmentación semántica. Cada métrica aporta un enfoque único y complementario para capturar diferentes aspectos de la efectividad de la segmentación.

En primer lugar, el Pixel Accuracy se ha elegido debido a su simplicidad y facilidad de interpretación [139]. Proporciona una medida directa de la cantidad de píxeles correctamente clasificados en relación con el total de píxeles en la imagen. Esta métrica es valiosa para una evaluación rápida, brindando una vista panorámica del rendimiento general del modelo. También presenta una debilidad que es importante destacar y es que esta métrica es sensible al desbalanceo de clases [140]

Por otro lado, tanto el Dice Coefficient como el Mean IoU han sido seleccionados con una consideración especial hacia las condiciones específicas de nuestro proyecto. Uno de los desafíos comunes en la segmentación semántica es el desequilibrio entre clases, donde algunas categorías pueden ser significativamente más pequeñas que otras. Tanto el Dice Coefficient como el Mean IoU son menos susceptibles a este problema en comparación con métricas como la precisión. Estas métricas ponderan la superposición entre las predicciones y las etiquetas, lo que permite una evaluación más justa de la calidad de la segmentación, incluso en escenarios donde la distribución de clases es desigual [141].

Además, el Mean IoU y el Dice Coefficient son particularmente adecuados para medir el grado de solapamiento entre las máscaras predichas y las máscaras de referencia. En la segmentación semántica, es crucial evaluar cuán bien los objetos y regiones están siendo delineados correctamente por el modelo. Estas métricas capturan tanto la precisión (cuántos píxeles se superponen correctamente) como la exhaustividad (cuánto de la región real se cubre) de la segmentación, proporcionando así una medida más completa y rica de la calidad de la tarea [141].

En resumen, la elección de estas métricas no solo considera su idoneidad para evaluar diferentes aspectos de la segmentación semántica, sino que también refleja una toma de decisiones informada que tiene en cuenta las particularidades del proyecto, como el desequilibrio entre clases y la necesidad de medir el grado de superposición entre las predicciones y las etiquetas verdaderas.

7.2. Selección de hiperparámetros

En esta fase se aborda la estrategia seguida para la selección de hiperparámetros de las redes. En este TFG, la potencia del hardware en el que se realizaron los experimentos es un factor limitante a la hora de llevar a cabo esta fase, imponiendo restricciones tanto en la cantidad de hiperparámetros que se pueden optimizar como en el tipo de búsqueda que se puede llevar a cabo (como *Grid Search* [142], *Bayesian Search* [143] o *Random Search* [144]). Dado que la cantidad de hiperparámetros de cada red y la cantidad de imágenes son considerables, se optó por una estrategia de optimización selectiva y priorización de hiperparámetros clave, lo que permitió obtener resultados significativos dentro de las limitaciones de recursos disponibles.

7.2.1. Vgg16

Dada esta restricción, la estrategia de selección de hiperparámetros que se ha seguido ha sido la siguiente:

1. Selección de los hiperparámetros a ajustar: se seleccionan 3 hiperparámetros para optimizar. Por un lado, el *learning rate*, evaluado con tasas de 0.00001, 0.0001 y 0.001. El *batch size*, evaluado para tamaños de 8, 4, 2 y 1 y por último el *solver*, en los que se ha probado el *adam* y el *stochastic gradient descent*.
2. La realización de un *Random Search* con el fin de visualizar ciertas tendencias en el valor de las métricas en función de los hiperparámetros. En la Figura 7.3 podemos ver los resultados de este *Random Search* donde observando cada línea podemos observar la tendencia en los coeficientes Dice, en el Mean IoU y en el Accuracy según la combinación de hiperparámetros:

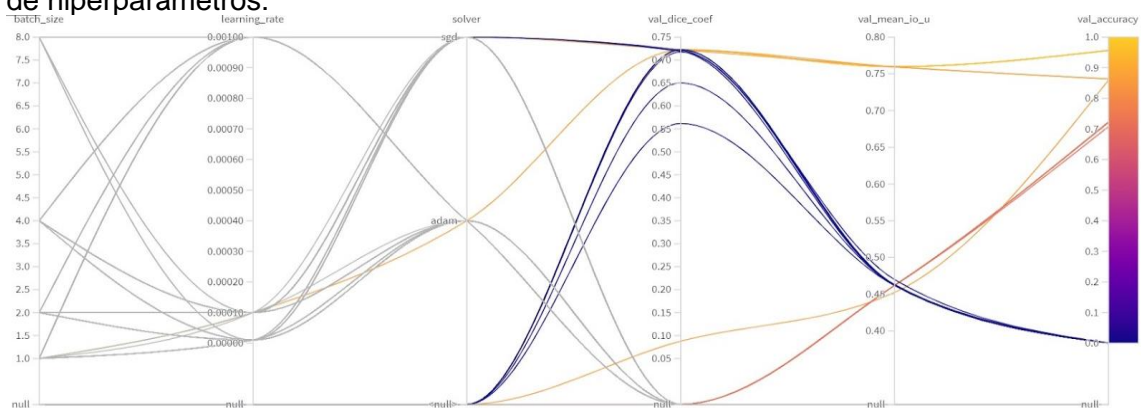


Figura 7.3: Random Search de los hiperparámetros. Fuente: Propia.

Puede observarse que el mejor en las 3 métricas de validación se da con un *batch size* de 1, un *learning rate* con valor de 0.0001 y el *solver* adam. Pueden observarse en la gráfica valores con *null*. Esto ocurre cuando el programa excede la capacidad del *hardware* y se da un error en los resultados. Puede observarse que para valores de *batch size* mayores que 1 prácticamente no se registra ninguna iteración de forma correcta, por lo que este *Random Search* nos dió pistas para realizar finalmente una prueba manual de hiperparámetros, acotando nuestro espacio de búsqueda.

Por otro lado, la función de pérdida seleccionada es la *Dice Coefficient Loss*. Esta función se ha elegido por encima de otras posibles en el problema de la segmentación por sus robustas propiedades frente a problemas desbalanceados [145]. También se probaron otras funciones de pérdida propuestas en [146] siendo el *Dice Coefficient Loss* la que mejor rendimiento proporcionaba.

Por tanto, los hiperparámetros seleccionados para el modelo son:

- *Learning rate*: 0.001
- *Solver*: Adam
- *Batch Size*: 1

7.2.2. Attention

Utilizando el conocimiento adquirido de la primera red entrenada y debido a las restricciones de potencia computacional, se adaptó la estrategia de ajuste de hiperparámetros para esta nueva red de manera sutil, siguiendo los siguientes pasos manualmente y prescindiendo del *Random Search*:

En primer lugar, se exploraron valores extremos de hiperparámetros que previamente habían causado problemas de exceso de memoria en la otra red. Esto se hizo para evaluar si la reducción en la cantidad de parámetros en esta nueva red permitiría manejar estos valores sin problemas de hardware. Se llevaron a cabo pruebas con tamaños de lote (*batch size*) superiores a 1, resultando nuevamente en errores de memoria excedida. Esto llevó a restringir el rango de valores de los hiperparámetros al rango previamente considerado.

El siguiente paso implicó un ajuste manual de los hiperparámetros que se habían probado en la red anterior con resultados exitosos. El objetivo era determinar si estos hiperparámetros podían conducir a la creación de un modelo de alta calidad en esta nueva configuración.

Paralelamente a lo anterior, se evaluaron diversas funciones de pérdida, y se encontró que la función de pérdida del Coeficiente de Dice (*Dice Coefficient Loss*) ofrecía la mejor calidad en términos de resultados, siguiendo una pauta similar a la que se había observado en el modelo anterior.

Finalmente, los hiperparámetros seleccionados para esta segunda red coinciden con los seleccionados para la primera:

- *Learning rate*: 0.001

- *Solver*: Adam
- *Batch Size*: 1

7.2.3. Swin Transformer

En esta situación, dado que estábamos empleando un modelo preentrenado, la cantidad de hiperparámetros para experimentar estaba determinada por un archivo de configuración que era imprescindible ajustar.

Entre todos los parámetros ajustables de la configuración, se realizaron pruebas únicamente con: el *learning rate*, la ponderación del algoritmo de optimización *Adam*, las tasas de regularización L2 [192] y el tamaño del *batch*.

El ajuste de estos parámetros se hizo por prueba y error, siendo la siguiente configuración la que mejor rendimiento obtuvo:

- *Batch size*: 1
- *Learning Rate*: 2e-6
- *L2*: 0.001
- *Adam betas*: (0.9,0.999)

7.3. Estrategia, entrenamiento y validación

Tanto para la *Attention Unet* como para la *VGG16-Unet*, el proceso inicial en el entrenamiento del modelo implicó dividir los datos en dos conjuntos: entrenamiento (train) y prueba (test). El conjunto de entrenamiento consta de 10189 imágenes, mientras que el conjunto de prueba consta de 3396 imágenes (representando el 75% y el 25% respectivamente del total).

Después de completar esta partición, se optó por entrenar y validar el modelo utilizando la estrategia de validación cruzada con K iteraciones (*K-fold cross validation*).

La validación cruzada de K iteraciones es una técnica común en la evaluación de modelos de aprendizaje automático. Consiste en dividir el conjunto de datos en K subconjuntos de aproximadamente igual tamaño. Luego, se realiza un proceso de entrenamiento y evaluación K veces, donde en cada iteración, uno de los pliegues se reserva como conjunto de prueba y los otros K-1 pliegues se utilizan como conjunto de entrenamiento.

Este enfoque permite utilizar todos los datos tanto para entrenamiento como para prueba a lo largo de las K iteraciones. Al finalizar las K iteraciones, se obtienen K puntuaciones de rendimiento. Estas puntuaciones se pueden promediar para obtener una medida más robusta del rendimiento del modelo en diferentes conjuntos de datos. Esto nos da una ligera idea de cómo se comportará el modelo en datos no vistos.

En nuestro contexto específico, aplicaremos una validación cruzada de 20 iteraciones (20-Fold Cross Validation) al conjunto de entrenamiento. Esto nos permitirá preservar el conjunto de prueba para futuras evaluaciones utilizando imágenes que el

modelo aún no haya encontrado previamente. A grandes rasgos, el funcionamiento de esta fase será el siguiente:

Iterar 20 veces:

Apartaremos 1/20 de las muestras, es decir, 509 imágenes.

Entrenamos el modelo con la muestra restante, es decir, con 9621 imágenes.

Medimos las métricas de interés sobre las 509 imágenes apartadas.

Esto implica que se llevarán a cabo 20 procesos de entrenamiento, lo que resultará en que las métricas finales del modelo sean el promedio de las métricas obtenidas en cada uno de estos 20 entrenamientos.

Tras completar este proceso, se avanza al siguiente paso, que implica entrenar un modelo definitivo utilizando la totalidad de las imágenes en el conjunto de entrenamiento.

Por último, este modelo definitivo se evalúa en el conjunto de prueba para comprender su desempeño con imágenes completamente nuevas para él.

En el caso del *Swin Transformer*, por temas computacionales, fue imposible realizar un *Cross Validation*, por lo que simplemente se realizó un *finetuning* de las últimas capas del modelo con el 80% de los datos. El 20% restante se utilizó para evaluar el rendimiento de este. Este modelo fue entrenado con 10 épocas al tener un elevado consumo de recursos computacionales.

7.4. Resultados

El presente apartado se centra en los resultados obtenidos durante el proceso de entrenamiento y validación de los modelos entrenados. Se expondrá una evaluación exhaustiva del rendimiento de los modelos a través de múltiples perspectivas, con el objetivo de comprender su capacidad para capturar las sutilezas de la composición y distribución de la tierra en áreas de interés.

En primer lugar, se presentarán las métricas derivadas del proceso de K-fold Cross Validation, ofreciendo una visión integral de la eficacia de los modelos en la segmentación semántica de datos LULC.

A continuación, se examinarán detalladamente las curvas de aprendizaje y convergencia obtenidas durante el entrenamiento de los modelos con el conjunto de entrenamiento completo. Estas gráficas no solo proporcionan una perspectiva sobre cómo los modelos han aprendido a lo largo del proceso de entrenamiento, sino que también pueden revelar pistas sobre posibles problemas de sobreajuste o subajuste.

Finalmente, para ilustrar concretamente la capacidad de los modelos para realizar segmentación semántica en datos LULC, se presentará un ejemplo visual de la segmentación realizada por dos de los modelos entrenados. Esta representación visual

no solo ofrece una comprensión intuitiva de las predicciones generadas por los modelos, sino que también resalta cómo capturan las características distintivas de diferentes clases de uso y cobertura de la tierra.

Resultados VGG16

En primer lugar se comprobarán los resultados del *Dice Coefficient* (Figura 7.4) en cada uno de los Folds tanto para el entrenamiento como para la validación. Esto nos dará una idea del rendimiento promedio de nuestro modelo, además de poder detectar otros problemas como *overfitting* o *underfitting*.

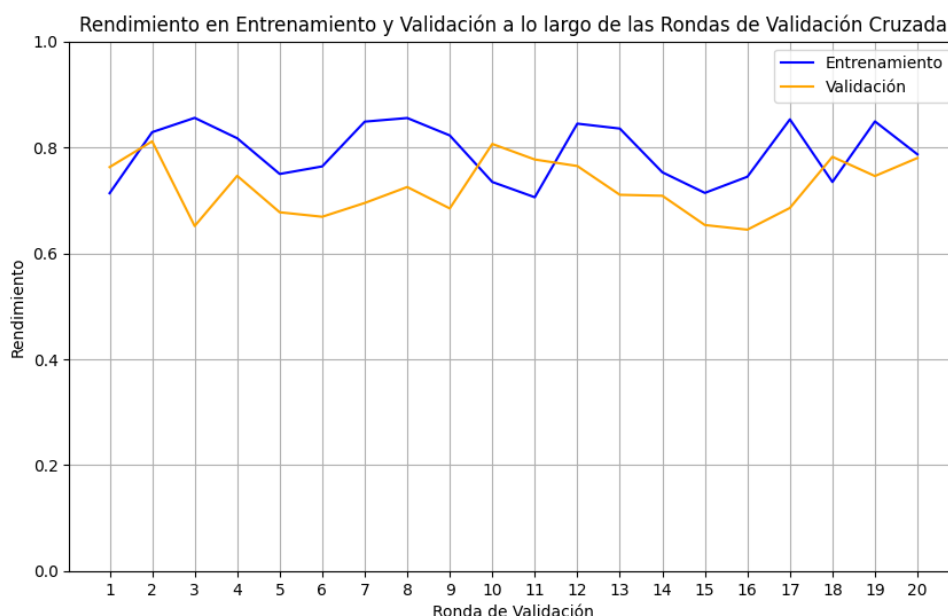


Figura 7.4: Resultados del CV del modelo. Fuente: Propia.

Se puede observar que el desempeño demostrado en cada ciclo de validación cruzada presenta una marcada similitud tanto en el conjunto de entrenamiento como en el de validación. Esta coherencia sugiere la existencia de un modelo altamente estable, el cual está respaldado por una capacidad razonable para identificar patrones en datos que no han sido previamente observados. Esta consistencia en el rendimiento del modelo en diferentes conjuntos de datos indica que no se está incurriendo ni en *overfitting* ni en *underfitting*.

En la siguiente fase de entrenamiento, una vez acabado el K-Fold Cross Validation, se procede a entrenar el modelo definitivo con todas las imágenes disponibles en el conjunto *train*. A continuación, mostramos la función de pérdida del modelo, tanto en entrenamiento como en validación (Figura 7.5) y el rendimiento del *Dice Coefficient*, también para las dos fases (Figura 7.6).

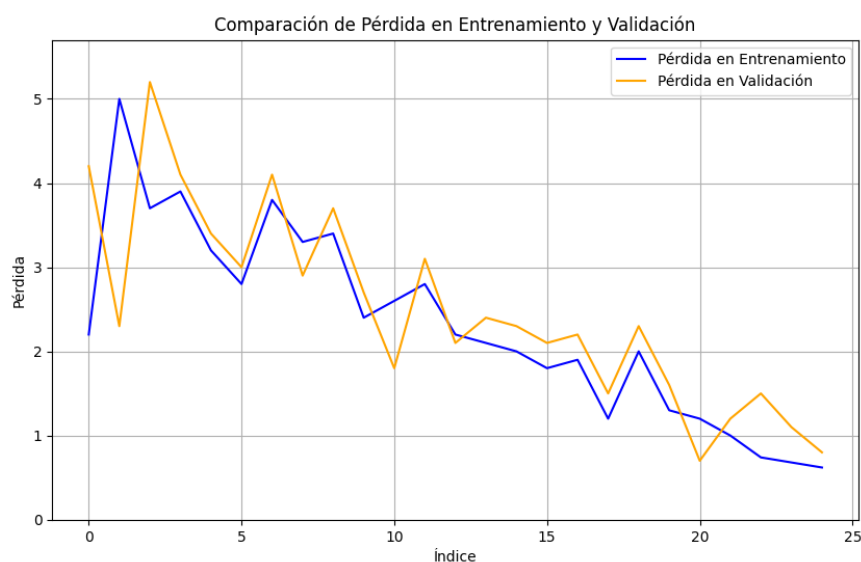


Figura 7.5: Función de pérdida del modelo. Fuente: Propia.

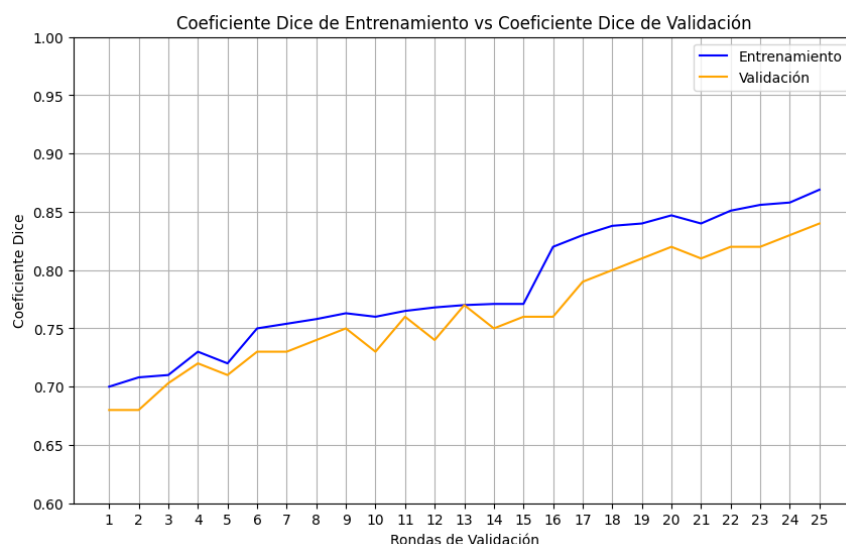


Figura 7.6: Métricas del modelo. Fuente: Propia.

Se puede notar que la función de pérdida (Figura 7.5) en el conjunto de validación muestra un leve aumento en comparación con la del conjunto de entrenamiento. Esta tendencia también se refleja en los valores del coeficiente Dice (Figura 7.6), donde el coeficiente es más alto en el conjunto de entrenamiento. Esta disparidad en el rendimiento entre los conjuntos podría sugerir la presencia de sobreajuste (*overfitting*). Sin embargo, es importante destacar que tanto la función de pérdida como la métrica exhiben patrones similares tanto en validación como en entrenamiento. Además, se puede observar que las discrepancias entre ambas funciones en cada época son mínimas. Estos indicios podrían señalar que el modelo no está sufriendo de sobreajuste significativo y que tiene la capacidad de generalizar de manera adecuada diversos patrones en los datos.

Para apoyar esta afirmación, vamos a mostrar las máscaras de segmentación predichas en algunas imágenes del conjunto *test* que el modelo no había visto previamente (Figura 7.7).

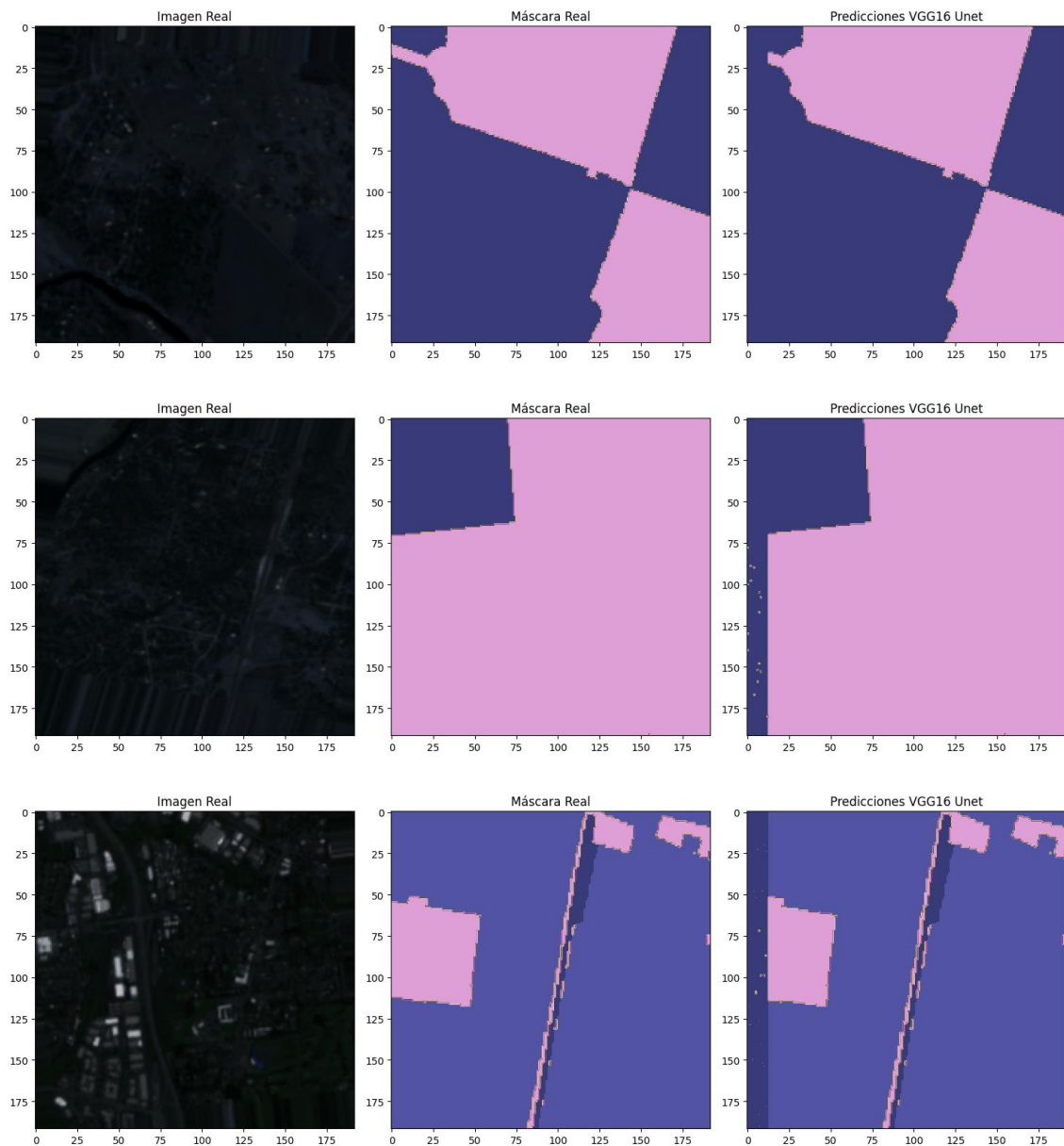


Figura 7.7: Imagen Original, Máscara Real y Máscara Predicha. Fuente: Propia.

La Figura 7.7 ofrece una visualización completa que incluye la imagen original, la correspondiente máscara de segmentación y la predicción generada por el modelo. Es evidente que las predicciones del modelo exhiben una notoria similitud con las máscaras de segmentación proporcionadas. A pesar de presentar algunas discrepancias, en general, las predicciones demuestran un alto grado de precisión. Este conjunto de resultados respalda firmemente la noción de que el modelo no está experimentando sobreajuste (*overfitting*).

Resultados Attention Unet

A continuación, se exponen los resultados logrados por el segundo modelo, conocido como 'Attention Unet'. Siguiendo la misma estructura, se presentan en primer lugar los resultados derivados del proceso de validación cruzada de 20 pliegues (Figura 7.8).

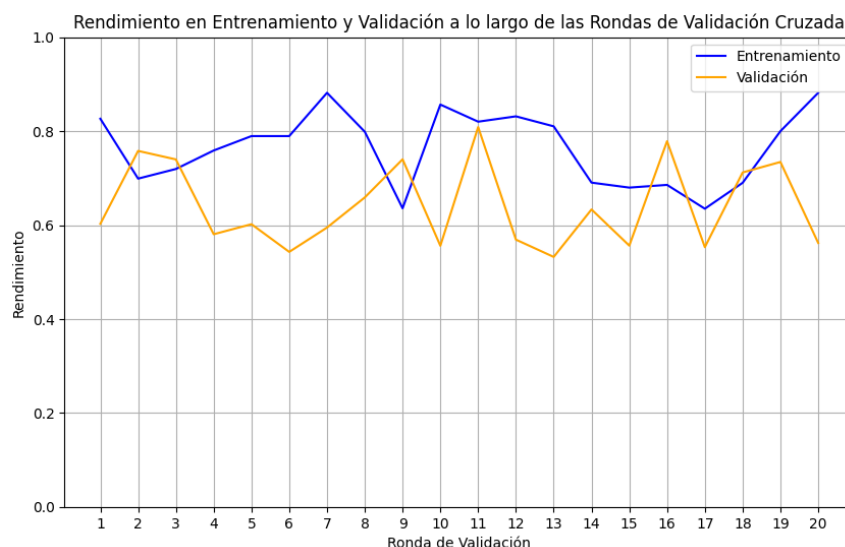


Figura 7.8: Resultados del CV del modelo. Fuente: Propia.

En esta ocasión, se observa un comportamiento mucho más irregular en la gráfica presentada (Figura 7.8). Además, se evidencian diferencias más marcadas entre los resultados obtenidos en el conjunto de entrenamiento y el conjunto de validación, siendo los resultados del conjunto de entrenamiento generalmente superiores. La variabilidad en estos resultados sugiere que el modelo podría estar enfrentando el problema de sobreajuste, lo que implica que no logra generalizar eficazmente hacia datos no previamente observados. No obstante, los ajustes en los hiperparámetros y las modificaciones realizadas en la arquitectura de la red son los que arrojan los mejores resultados, lo que lleva a la decisión de continuar con este modelo a pesar de los desafíos presentados.

Es importante señalar que en esta evaluación estamos focalizados exclusivamente en el coeficiente de Dice, lo que implica que contar con una gama más amplia de métricas y observaciones proporcionaría un conjunto más sólido de indicios para determinar si el modelo cumple con su objetivo o no.

Siguiendo la misma línea, procederemos a evaluar la pérdida (Figura 7.9) del modelo entrenado con el 75% total del conjunto de datos, así como el coeficiente de Dice (Figura 7.10) en cada época del entrenamiento y la calidad de las segmentaciones generadas (Figura 7.11) en comparación con las segmentaciones de referencia.

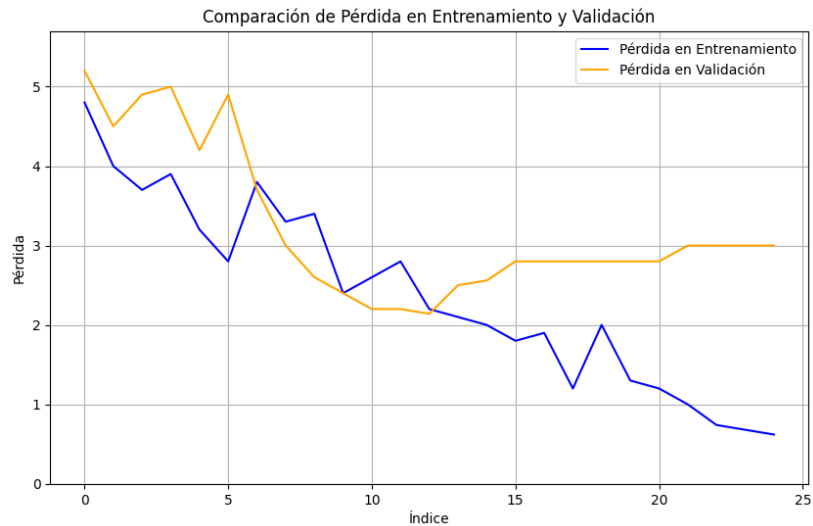


Figura 7.9: Función de pérdida del modelo. Fuente: Propia.

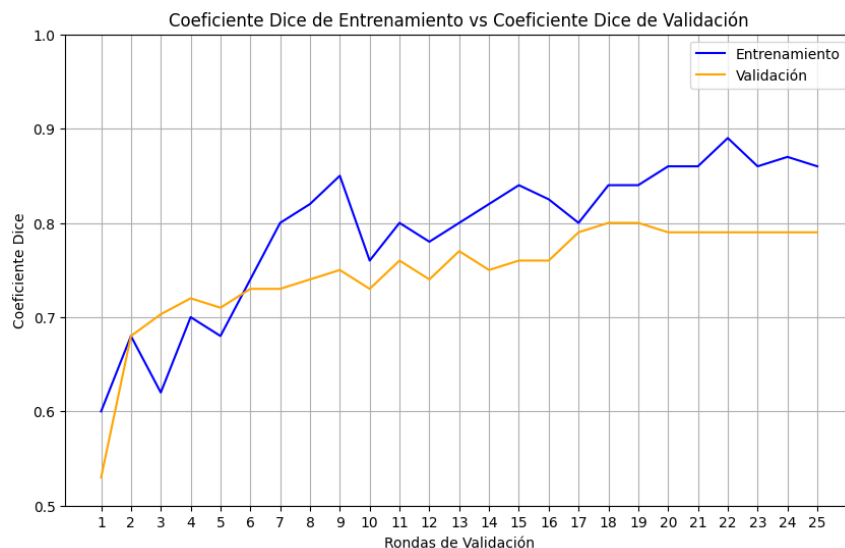


Figura 7.10: Métricas del modelo. Fuente: Propia.

En el primer gráfico (Figura 7.9), se puede apreciar una tendencia general en la que la función de pérdida en el conjunto de validación presenta valores superiores a los de la función de pérdida en el conjunto de entrenamiento. Asimismo, se observa que ambas funciones de pérdida experimentan fluctuaciones significativas hasta transcurrido un número considerable de épocas, momento en el cual parecen estabilizarse. Al analizar la gráfica (Figura 7.10) que representa la métrica del Coeficiente de Dice durante el proceso de entrenamiento, también se advierte que los valores en el conjunto de entrenamiento son más elevados que los del conjunto de validación.

Con base en toda esta información, es posible confirmar que el modelo está experimentando sobreajuste (overfitting). Sin embargo, a pesar de esta observación, al monitorear la métrica durante el proceso de entrenamiento, se puede notar que el conjunto de validación logra alcanzar un nivel aceptable en cuanto al Coeficiente de Dice. Por lo tanto, aunque se tenga conciencia de la presencia de este problema en el modelo, se opta por continuar con el proceso experimental y se procede a evaluar el modelo desde un enfoque gráfico (Figura 7.11).

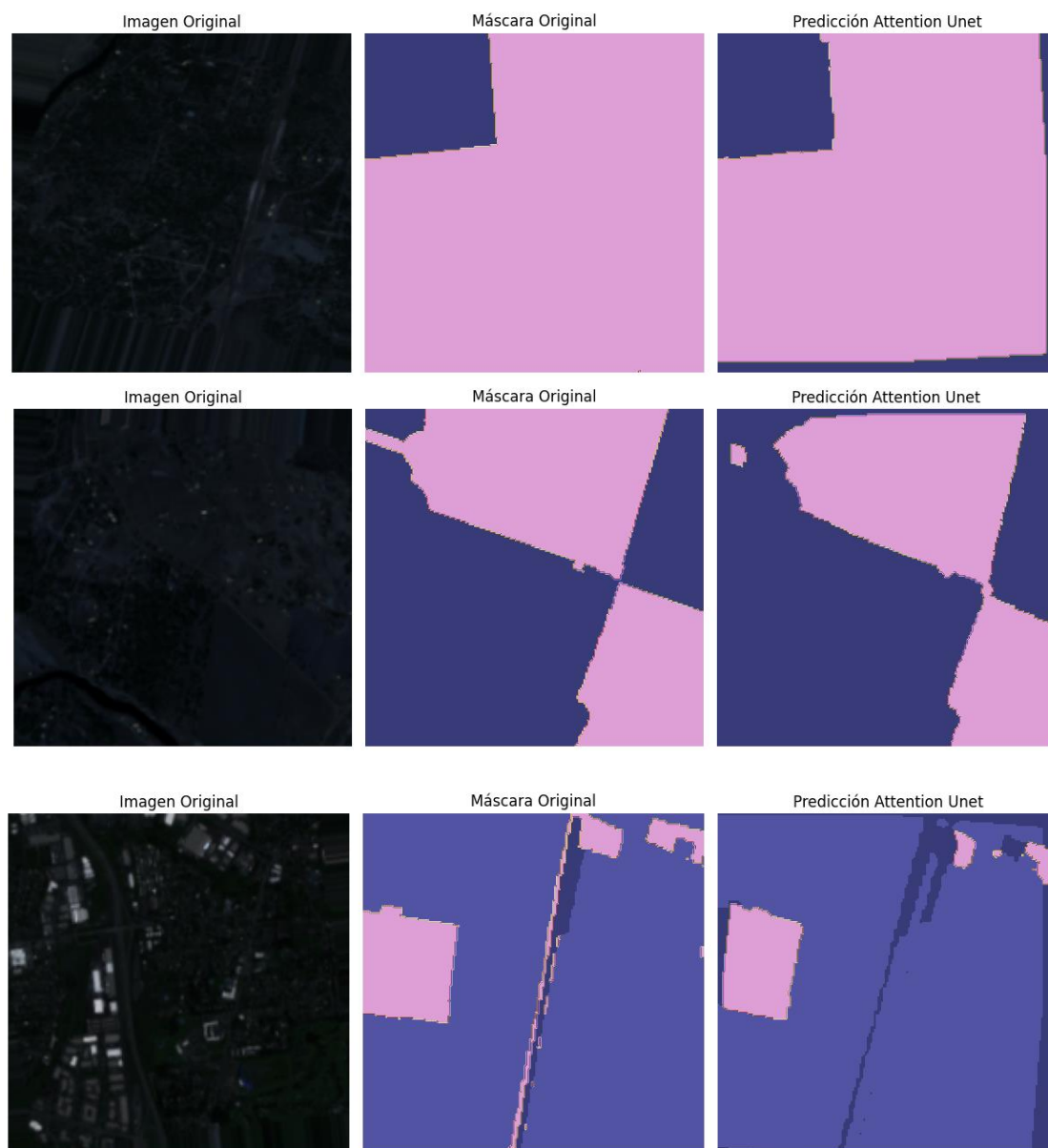


Figura 7.11: Imagen Original, Máscara Real y Máscara Predicha. Fuente: Propia.

Como se mencionó previamente, se presentan la imagen original junto con la máscara original y la máscara predicha por el modelo. Resulta evidente que el rendimiento del modelo es notablemente inferior en términos de la calidad de la segmentación, en comparación con el enfoque VGG16-Unet. Es evidente que este modelo enfrenta desafíos más significativos al segmentar detalles sutiles en comparación con el otro modelo que logra manejar esta tarea de manera más efectiva. En última instancia, se puede concluir que el fenómeno de sobreajuste impacta de manera considerable en el rendimiento del modelo, y por ende, el VGG16-Unet evaluado exhibe resultados de calidad superior.

Resultados Swin Transformer

Como hemos comentado anteriormente, el enfoque para el entrenamiento de este modelo es distinto respecto a los dos anteriores modelos. En este caso, no se ha

conseguido realizar un proceso de *Cross Validation* del modelo, por lo que simplemente se optó por un enfoque más simplista, entrenando el modelo en 10 épocas con una partición aleatoria del conjunto de datos en un 80% para el conjunto de entrenamiento y un 20% para el conjunto de test.

Los resultados de este modelo son los siguientes:

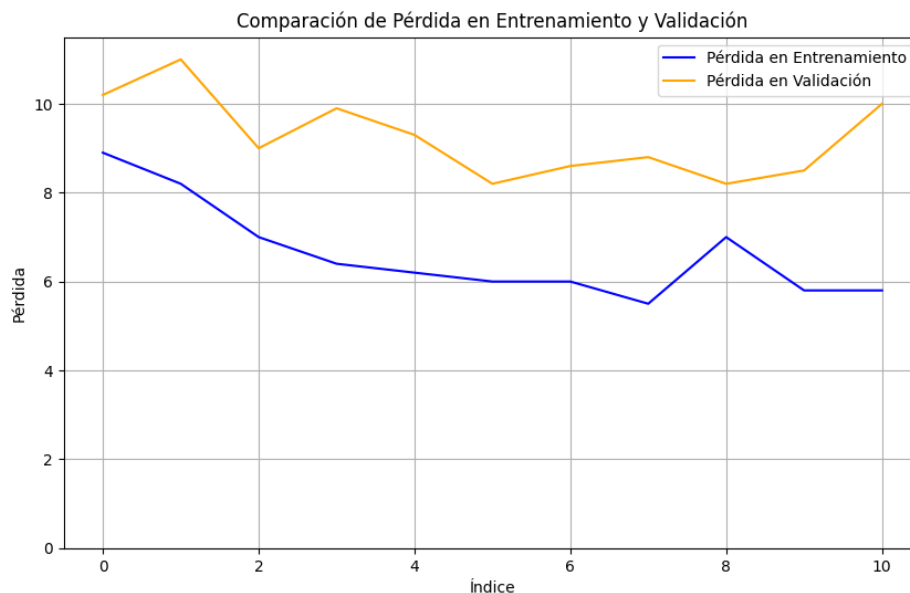


Figura 7.12: Función de Pérdida del modelo. Fuente: Propia.

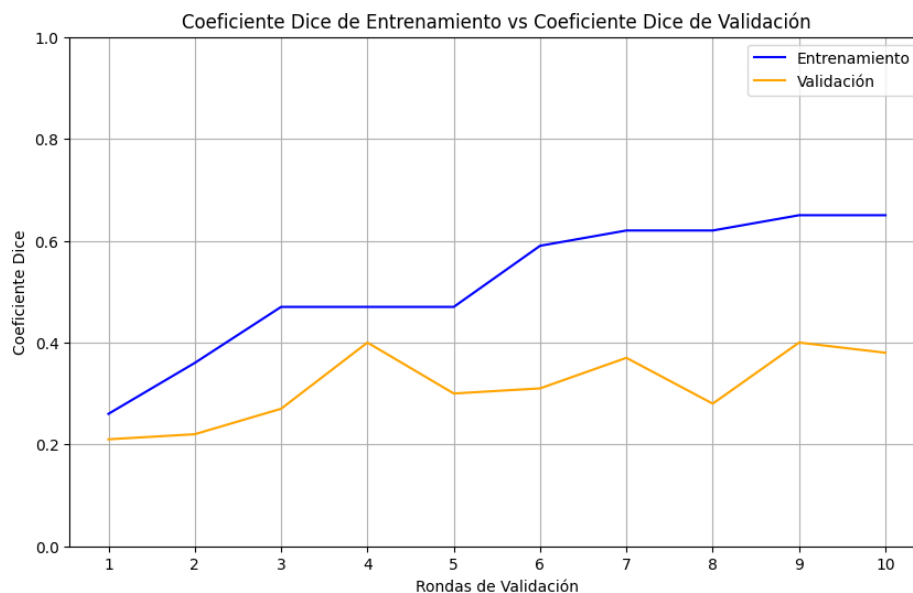


Figura 7.13: Métricas del modelo. Fuente: Propia.

Como puede observarse en ambas gráficas (Figura 7.12 y Figura 7.13), el modelo presenta un claro indicio de sobreajuste: la pérdida (Figura 7.12) en validación es significativamente mayor que la pérdida en el conjunto de entrenamiento. Además, esto puede observarse también en el monitoreo de las métricas (Figura 7.13): puede observarse como el rendimiento en el conjunto de entrenamiento es significativamente

mayor que en el conjunto de entrenamiento. Es también visible que el comportamiento de las funciones en ambas gráficas es ligeramente caótico y poco estable.

También cabe destacar que el rendimiento del modelo, tanto en entrenamiento como en validación está muy por debajo de los mínimos conseguidos por los dos modelos anteriores. Pese a que claramente las métricas indican un rendimiento pobre y sustancialmente inferior al *VGG16 Unet* y al *Attention Unet*, vamos a realizar un análisis de las segmentaciones realizadas (Figura 7.14) en el conjunto de test por este modelo:

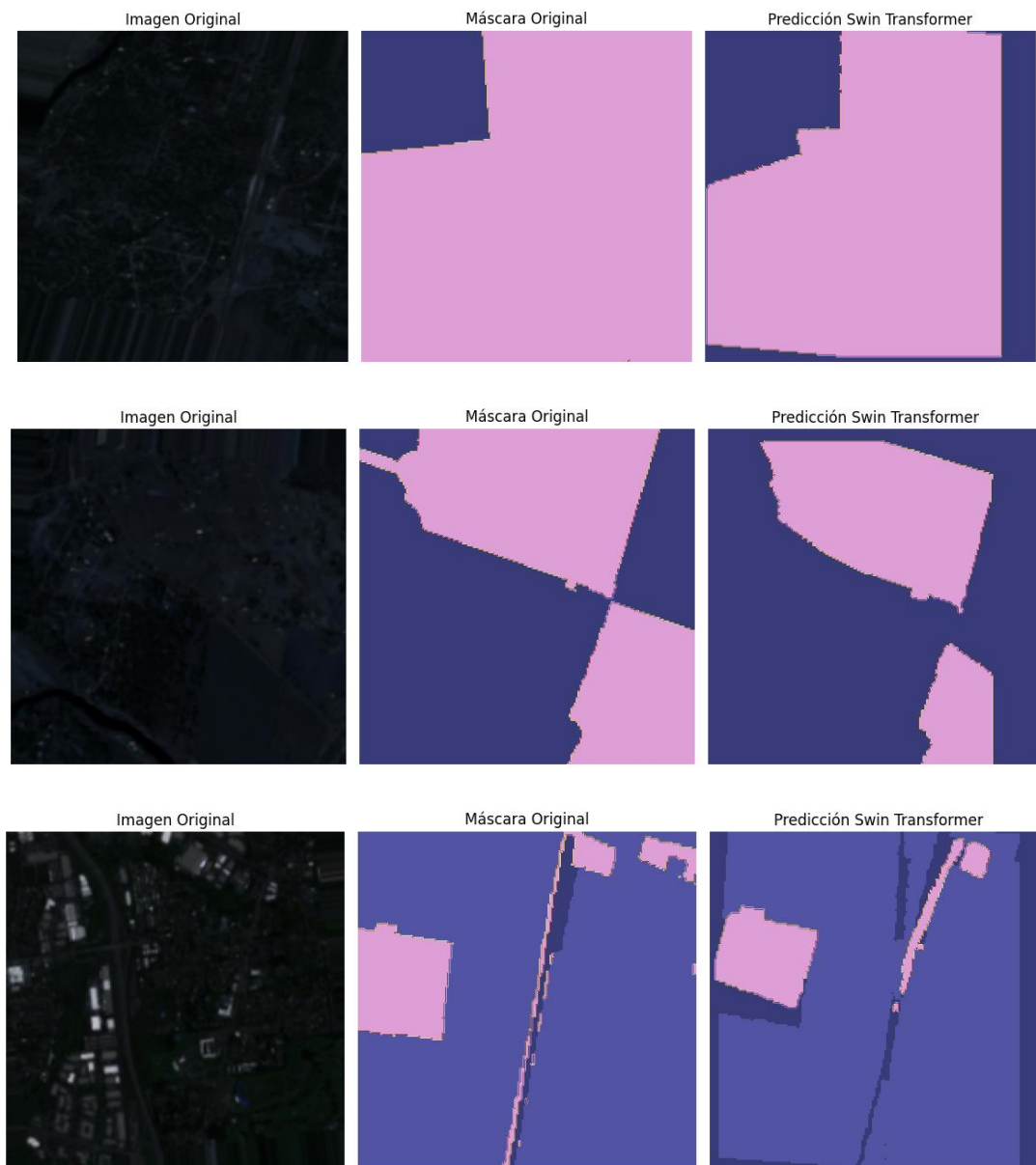


Figura 7.14: Imagen Original, Máscara Real y Máscara Predicha. Fuente: Propia.

Como habíamos anticipado, la calidad de la segmentación resultó ser notablemente deficiente, quedando muy por debajo de los estándares establecidos por modelos previos. En virtud de esta circunstancia, se ha optado por excluir este modelo del conjunto de resultados relevantes en el siguiente apartado. Esto se debe a que su calidad significativamente inferior lo sitúa en una posición desfavorable en comparación con los dos modelos anteriores, por lo que carece de sentido compararlo con estos.

Comparativa de ambos modelos.

A continuación, se ofrece una comparativa exhaustiva de los resultados obtenidos en las pruebas previas, abarcando todas las métricas y considerando los modelos VGG16-Unet y Attention Unet. Este análisis tiene como propósito fortalecer las conclusiones extraídas anteriormente, otorgándoles mayor sustento.

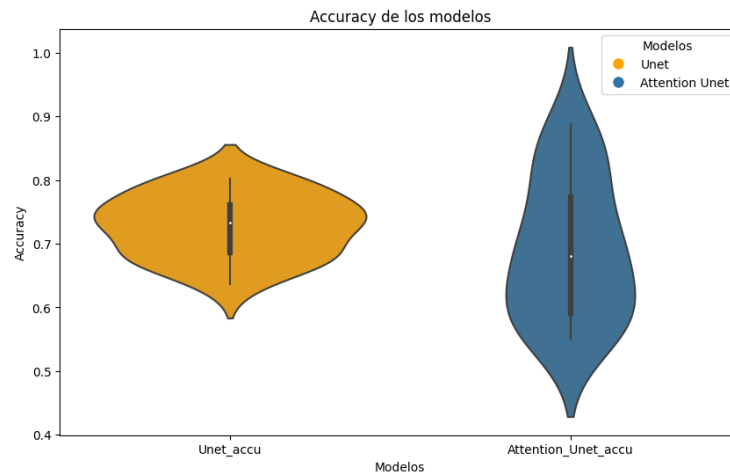


Figura 7.15: Accuracy de los modelos en el CV. Fuente: Propia.

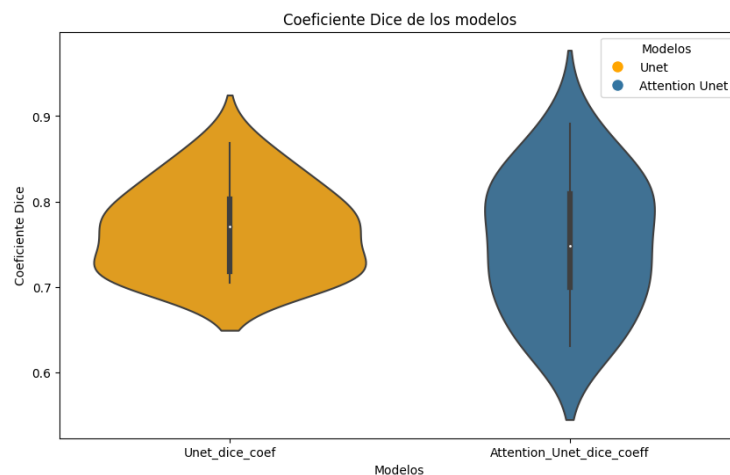


Figura 7.16: Coeficiente Dice de los modelos en el CV. Fuente: Propia.

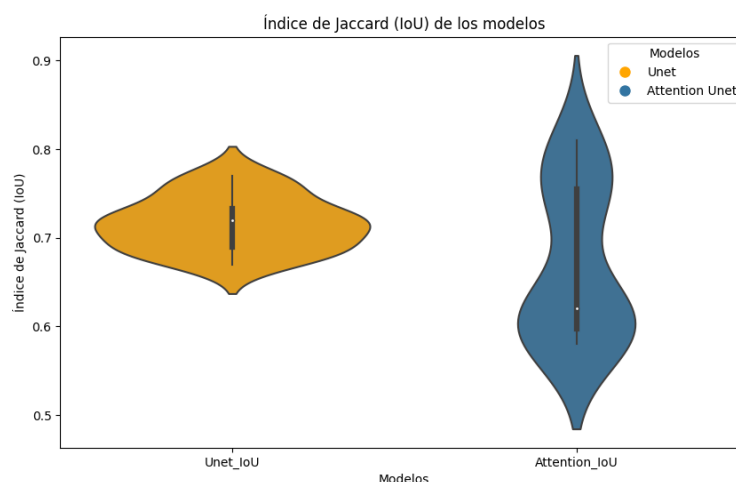


Figura 7.17: IoU de los modelos en el CV. Fuente: Propia.

Las Figuras 7.15, 7.16 y 7.17 muestran los gráficos de violín que ilustran la distribución de las métricas en el proceso de Validación Cruzada. Estos gráficos son útiles para obtener una visualización detallada de la distribución de los datos y su densidad de probabilidad. Esta representación combina elementos de los diagramas de caja y bigotes con los de densidad.

Es evidente que los resultados obtenidos mediante el uso de VGG16-Unet muestran una notable coherencia, exhibiendo una distribución de métricas considerablemente más simétrica. Esto se refleja en medianas notablemente centradas y en intervalos intercuartílicos más reducidos. En contraste, los resultados generados por el modelo Attention Unet revelan máximos ligeramente superiores en las métricas. No obstante, esta ventaja no logra contrarrestar la marcada variabilidad del modelo, la cual se manifiesta en distribuciones de resultados altamente asimétricas. Esta variabilidad excesiva contrarresta los beneficios de los valores máximos y sugiere dificultades en la capacidad del modelo para generalizar patrones aprendidos durante el entrenamiento.

Para finalizar con la sección de resultados, se adjunta una tabla resumen donde se recoge el resultado de las diferentes métricas en ambos modelos en el proceso de *Cross Validation*. Estos resultados se han agregado en diferentes estadísticos para aportar una visión más descriptiva del proceso.

	VGG16-Unet			Attention Unet		
	Pixel Accuracy	Dice Coefficient	Mean IoU	Pixel Accuracy	Dice Coefficient	Mean IoU
Media	0.728	0.767	0.715	0.690	0.751	0.669
Mediana	0.733	0.770	0.720	0.679	0.748	0.620
Varianza	0.002	0.002	0.000	0.012	0.006	0.007
Máximo	0.802	0.869	0.770	0.880	0.891	0.81
Mínimo	0.637	0.705	0.670	0.540	0.631	0.58

Tabla 7.1: Métrica en Validación Cruzada. Fuente: Propia.

Claramente se puede apreciar que los resultados presentados en la Tabla 7.1 respaldan todas las conclusiones que se derivaron del análisis exploratorio gráfico de manera consistente. En el caso de la variabilidad del modelo VGG16-Unet, se nota una constante reducción en comparación con los resultados del Attention Unet. Además, la mediana exhibe un incremento sistemático en el caso de VGG16-Unet. Aunque es evidente que los valores máximos logrados por el Attention Unet son consistentemente superiores, es importante destacar que los valores mínimos son consistentemente inferiores. Considerando el conjunto integral de estadísticas, se reafirma la superioridad en términos de rendimiento por parte de VGG16-Unet. Estos hallazgos respaldan y fortalecen las conclusiones previamente obtenidas.

8. Conclusiones

En este capítulo, se presentan las conclusiones clave del estudio realizado, que abordan tanto los resultados obtenidos como las implicaciones de la investigación. Además, se discutirá la reproducibilidad del proyecto, evaluando su capacidad de ser replicado o ampliado en el futuro. Asimismo, se analizará la relación de este proyecto con los estudios cursados a lo largo de la carrera, destacando cómo ha contribuido al desarrollo académico y profesional. Por último, se esbozarán las direcciones potenciales para investigaciones futuras, dando una visión de las oportunidades que se abren a partir de este trabajo.

Este capítulo culmina la investigación emprendida en este TFG y destaca la importancia de los hallazgos para la disciplina estudiada: como mejorar el trabajo que se realiza mediante IOTA con la creación de una herramienta más flexible. A través de las conclusiones, se busca consolidar los conocimientos adquiridos durante este proceso y brindar una perspectiva sólida sobre las posibilidades futuras que este trabajo pueda ofrecer.

8.1. Conclusiones

En este estudio, se han extraído conclusiones fundamentales que arrojan luz sobre las complejidades y desafíos inherentes a la tarea de crear un dataset a partir de múltiples fuentes de información. Se ha demostrado que esta tarea es complicada debido a la necesidad de amplio almacenamiento y recursos computacionales, la variabilidad en las resoluciones de los datos y la dificultad de alinear las zonas geográficas de origen, lo que, en última instancia, resalta la importancia de abordar con rigor la gestión de datos en proyectos similares.

Además, se ha confirmado que los modelos presentan dificultades para generalizar en áreas geográficas que nunca han sido previamente observadas. Específicamente, se ha constatado que los modelos preentrenados requieren considerables recursos computacionales y tiempo para realizar un *fine tuning* que les permita aprovechar el conocimiento previo y aplicarlo a nuevas zonas geográficas. Esta observación se aprecia claramente en los resultados obtenidos a través de la implementación del modelo *Swin Transformer*.

En lo que respecta a la selección de herramientas, se llega a la conclusión de que la implementación de un modelo *Unet*, en el que se emplea una arquitectura *VGG16* preentrenada con *Imagenet* como extractor de características (*encoder*), ha demostrado proporcionar resultados notables. Además, este enfoque se destaca por su eficiencia en términos de tiempo de *fine-tuning*, especialmente en comparación con otros modelos. Esto lo convierte en una opción atractiva y efectiva para la segmentación de zonas geográficas previamente no exploradas, lo que tiene aplicaciones prácticas significativas.

Por último, se debe señalar que los mecanismos de atención evaluados, tanto los intrínsecos al *Swint Transformer* como los integrados en la arquitectura *Unet*, no

proporcionaron resultados satisfactorios en este estudio. Esta conclusión sugiere la necesidad de explorar alternativas y perfeccionar la implementación de dichos mecanismos en investigaciones futuras, enfocadas en la segmentación geográfica.

En resumen, estas conclusiones consolidan los descubrimientos de esta investigación y ofrecen una visión crítica de los resultados en relación con los objetivos planteados, contribuyendo al conocimiento y la comprensión en el campo de estudio.

8.2. Reproducibilidad

La reproducibilidad en la investigación desempeña un papel fundamental para validar y consolidar los resultados obtenidos. En este sentido, este proyecto de TFG se destaca por su enfoque riguroso y la facilidad con la que puede ser replicado por otros investigadores y entusiastas interesados en el tema.

Uno de los pilares de la alta reproducibilidad de este proyecto radica en la disponibilidad de los datos utilizados. Los datos empleados en esta investigación se encuentran alojados en un repositorio específico de una investigación académica, como se detalla en el artículo [34]. Este enfoque garantiza la transparencia y la accesibilidad de los datos, permitiendo que cualquier persona interesada pueda acceder y utilizar la misma fuente de información que se utilizó en este estudio.

Además, el código fuente desarrollado para este proyecto se encuentra alojado en un repositorio público en GitHub, lo que facilita su acceso y reproducción. Esto implica que cualquier persona con acceso a Internet puede examinar el código, comprender la metodología empleada y replicar el proceso paso a paso. Esta disponibilidad fomenta la colaboración, la revisión y la mejora continua, contribuyendo así a la calidad y la confiabilidad de la investigación.

En resumen, la alta reproducibilidad de este proyecto se fundamenta en la disponibilidad de los datos en un repositorio académico de investigación y la publicación del código fuente en GitHub. Este enfoque refuerza la transparencia y la accesibilidad, lo que promueve la confiabilidad y la validez de los resultados obtenidos, así como la posibilidad de ampliar y construir sobre esta investigación en el futuro.

8.3. Relación con los estudios cursados

La realización de este proyecto de investigación ha servido como un puente sólido entre los conocimientos teóricos adquiridos a lo largo de mi carrera y su aplicación práctica en un contexto real. Entre los conocimientos teóricos destacados que se aplicaron en el proyecto se encuentran la visualización de datos, los algoritmos de Deep Learning, la programación, la limpieza y procesamiento de datos, en particular de imágenes, la evaluación de modelos, la toma de decisiones y el análisis exploratorio de datos. Estos pilares teóricos proporcionaron la base sólida para abordar con éxito la complejidad del proyecto.

Además, se aplicaron competencias transversales clave que enriquecieron el proceso. La planificación y gestión del tiempo (CT-12) fueron esenciales para coordinar

eficazmente las tareas del proyecto y cumplir con los plazos establecidos. La aplicación del pensamiento práctico (CT-02) se reflejó en la capacidad para aplicar conocimientos teóricos de manera efectiva y establecer un proceso sólido para lograr los objetivos establecidos. La resolución de problemas (CT-03) desempeñó un papel central en la identificación y definición de los desafíos a superar en el proyecto. El diseño y proyecto (CT-05) se aplicaron en la creación y dirección efectiva de un enfoque metodológico hasta su realización. Por último, el conocimiento de problemas contemporáneos (CT-10) permitió abordar cuestiones relacionadas con la sostenibilidad y la actualidad, enriqueciendo la relevancia y la aplicabilidad de la investigación.

En resumen, este proyecto no solo se benefició de la sólida base teórica adquirida durante la carrera, sino que también destacó la importancia de las competencias transversales en la planificación, ejecución y resolución de problemas en un contexto de investigación aplicada. Estas habilidades y conocimientos teóricos combinados fueron fundamentales para el éxito de este proyecto.

8.4. Trabajos futuros

En vista de los resultados y las oportunidades identificadas a lo largo de este proyecto, se vislumbran diversas direcciones para futuras investigaciones y mejoras. Una de las posibilidades más prometedoras es la creación de un ecosistema en la nube dedicado a la captura y almacenamiento de imágenes a gran escala. Esta infraestructura podría facilitar la gestión de un volumen aún mayor de imágenes geoespaciales, lo que ampliaría las posibilidades de investigación y permitiría un análisis más exhaustivo. Además, la expansión de este ecosistema podría allanar el camino para trabajar con un mayor número de bandas, lo que, en teoría, podría mejorar significativamente la precisión y la versatilidad de los modelos de segmentación geográfica. Por último, en busca de una mayor optimización y eficiencia, sería beneficioso explorar otras arquitecturas de Transformers que puedan ofrecer un rendimiento aún más avanzado en la tarea de segmentación. Estas direcciones futuras prometen enriquecer y expandir aún más la aplicación de modelos de aprendizaje automático en el campo de la geoespacialidad.

Bibliografía

- [1]. *SENTINEL 2*. (s. f.). https://www.esa.int/Space_in_Member_States/Spain/SENTINEL_2
- [2]. Pravitasari, A. A., Iriawan, N., Almuhayar, M., Azmi, T., Irhamah, I., Fithriasari, K., Purnami, S. W., & Ferriastuti, W. (2020). UNET-VGG16 with transfer learning for MRI-based brain tumor segmentation. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 18(3), 1310.
- [3]. Sun, Y., Bi, F., Gao, Y., Chen, L., & Feng, S. (2022). A Multi-Attention UNET for semantic segmentation in remote sensing images. *Symmetry*, 14(5), 906.
<https://doi.org/10.3390/sym14050906>
- [4]. Scheibenreif, L., Hanna, J., Mommert, M., & Borth, D. (2022b). Self-supervised vision transformers for land-cover segmentation and classification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
<https://doi.org/10.1109/cvprw56347.2022.00148>
- [5]. *Coastal erosion 2 - eo Science for Society*. (2022, 29 noviembre). eo science for society.
<https://eo4society.esa.int/projects/coastal-erosion-2/>
- [6]. *EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification*. (2019, 1 julio). IEEE Journals & Magazine | IEEE Xplore.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8736785>
- [7]. Munyati, C. (2004). Use of Principal Component analysis (PCA) of remote sensing images in Wetland change detection on the Kafue Flats, Zambia. *Geocarto International*, 19(3), 11-22. <https://doi.org/10.1080/10106040408542313>
- [8]. Zhu, Q., Zhong, Y., Zhao, B., Xia, G., & Zhang, L. (2016). Bag-of-Visual-Words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 13(6), 747-751.
<https://doi.org/10.1109/lgrs.2015.2513443>
- [9]. Emsley, S. (s. f.). *ARGANS / Coastal change from space*. ARGANS Limited - Copyright 2020. <https://coastalerosion.argans.co.uk/>

- [10]. *Significance of Land Use / Land Cover (LULC) maps*. (s. f.).
<https://www.satpalda.com/blogs/significance-of-land-use-land-cover-lulc-maps>
- [11]. Canada, N. R. (2015, 20 noviembre). *Land cover & Land use*. <https://natural-resources.canada.ca/maps-tools-and-publications/satellite-imagery-and-air-photos/tutorial-fundamentals-remote-sensing/educational-resources-applications/land-cover-biomass-mapping/land-cover-land-use/9373>
- [12]. Zhang, C., & Li, X. (2022a). Land use and land cover mapping in the era of big data. *Land*, 11(10), 1692. <https://doi.org/10.3390/land11101692>
- [13]. Mora, B., Tsendbazar, N., Herold, M., & Arino, O. (2014). Global Land cover Mapping: Current status and future trends. En *Remote sensing and digital image processing* (pp. 11-30). https://doi.org/10.1007/978-94-007-7969-3_2
- [14]. *IOTA2 Classification — IOTA2 Documentation*. (s. f.).
https://docs.iota2.net/master/i2_classification_tutorial.html
- [15]. Jonášová, E. P. (2023). A review of Deep-Learning Methods for Change Detection in Multispectral Remote Sensing Images. *Remote Sensing*, 15(8), 2092.
<https://doi.org/10.3390/rs15082092>
- [16]. *Desarrollo Urbano Sostenible / EDUSI*. (s. f.). <http://edusi.es/content/desarrollo-urbano-sostenible>
- [17]. *THE 17 GOALS / Sustainable Development*. (s. f.). <https://sdgs.un.org/goals>
- [18]. Gamez, M. J. (2022, 24 mayo). *Objetivos y metas de Desarrollo sostenible - Desarrollo sostenible*. Desarrollo Sostenible.
<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [19]. Moran, M. (2020, 17 junio). *Ciudades - Desarrollo sostenible*. Desarrollo Sostenible.
<https://www.un.org/sustainabledevelopment/es/cities/>
- [20]. Moran, M. (2020a, junio 17). *Bosques, desertificación y diversidad biológica - desarrollo sostenible*. Desarrollo Sostenible.
<https://www.un.org/sustainabledevelopment/es/biodiversity/>

- [21]. Moran, M. (2020b, junio 17). *Cambio climático - desarrollo sostenible*. Desarrollo Sostenible. <https://www.un.org/sustainabledevelopment/es/climate-change-2/>
- [22]. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., Ramírez-Quintana, M. J., & Flach, P. A. (2021). CRISP-DM Twenty years Later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3048-3061.
<https://doi.org/10.1109/tkde.2019.2962680>
- [23]. Alem, A., & Kumar, S. (2020b). Deep Learning Methods for Land Cover and Land Use Classification in Remote Sensing: A Review. *IEEE*.
<https://doi.org/10.1109/icrito48877.2020.9197824>
- [24]. Luna, Z. (2022, 24 agosto). Understanding CRISP-DM and its importance in data science projects. *Medium*. <https://medium.com/analytics-vidhya/understanding-crisp-dm-and-its-importance-in-data-science-projects-91c8742c9f9b>
- [25]. Strahler, A. H. (1980). The use of prior probabilities in maximum likelihood classification of remotely sensed data. *Remote Sensing of Environment*, 10(2), 135-163.
[https://doi.org/10.1016/0034-4257\(80\)90011-5](https://doi.org/10.1016/0034-4257(80)90011-5)
- [26]. Settle, J. J., & Briggs, S. (1987). Fast maximum likelihood classification of remotely-sensed imagery. *International Journal of Remote Sensing*, 8(5), 723-734.
<https://doi.org/10.1080/01431168708948683>
- [27]. Ham, J., Chen, Y., Crawford, M. M., & Ghosh, J. (2005). Investigation of the random forest Framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 492-501.
<https://doi.org/10.1109/tgrs.2004.842481>
- [28]. Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8), 1778-1790. <https://doi.org/10.1109/tgrs.2004.831865>
- [29]. Han, W., Zhang, X., Wang, Y., Wang, L., Huang, X., Zhang, L., Wang, S., Chen, W., Li, X., Feng, R., Fan, R., Zhang, X., & Wang, Y. (2023). A survey of Machine learning

and Deep learning in remote sensing of geological Environment: Challenges, advances, and opportunities. *Isprs Journal of Photogrammetry and Remote Sensing*, 202, 87-113.
<https://doi.org/10.1016/j.isprsjprs.2023.05.032>

- [30]. Zhang, W., Tang, P., & Zhao, L. (2019). Remote sensing image scene classification using CNN-CapsNet. *Remote Sensing*, 11(5), 494. <https://doi.org/10.3390/rs11050494>
- [31]. Wang, Q., Zhang, X., Chen, G., Fan, D., Gong, Y., & Zhu, K. (2018). Change detection based on faster R-CNN for high-resolution remote sensing images. *Remote Sensing Letters*, 9(10), 923-932.
- [32]. Shirmard, H., Farahbakhsh, E., Heidari, E., Pour, A. B., Pradhan, B., Müller, R. D., & Chandra, R. (2022). A comparative study of convolutional neural networks and conventional machine learning models for lithological mapping using remote sensing data. *Remote Sensing*, 14(4), 819. <https://doi.org/10.3390/rs14040819>
- [33]. Hong, D., Yokoya, N., Xia, G., Chanussot, J., & Zhu, X. X. (2020). X-ModalNet: a semi-supervised deep cross-modal network for classification of remote sensing data. *Isprs Journal of Photogrammetry and Remote Sensing*, 167, 12-23.
<https://doi.org/10.1016/j.isprsjprs.2020.06.014>
- [34]. Johnson, N. R., Treible, W., & Crispell, D. (2022). OpenSentinelMap: a Large-Scale land use dataset using OpenStreetMap and Sentinel-2 imagery. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
<https://doi.org/10.1109/cvprw56347.2022.00139>
- [35]. Wójtowicz M., Wójtowicz A., Piekarczyk J. (2016). Application of remote sensing methods in agriculture. *Communications in Biometry and Crop Science* 11, 31–50.
- [36]. King, C., Baghdadi, N., Lecomte, V., & Cerdan, O. (2005). The application of remote-sensing data to monitoring and modelling of soil erosion. *Catena*, 62(2-3), 79-93.
<https://doi.org/10.1016/j.catena.2005.05.007>
- [37]. Rango, A. (1994). Application of remote sensing methods to hydrology and water resources. *Hydrological Sciences Journal-journal Des Sciences Hydrologiques*, 39(4), 309-320. <https://doi.org/10.1080/02626669409492752>

- [38]. Ghassemian, H. (2016). A review of remote sensing image fusion Methods. *Information Fusion*, 32, 75-89. <https://doi.org/10.1016/j.inffus.2016.03.003>
- [39]. Veneros, J., García, L., Morales, E., Gómez, V., Torres, M., & López-Morales, F. (2020). Aplicación de sensores remotos para el análisis de cobertura vegetal y cuerpos de agua. *Idesia*, 38(4), 99-107. <https://doi.org/10.4067/s0718-34292020000400099>
- [40]. Toth, C. K., & Józków, G. (2016). Remote Sensing Platforms and Sensors: A survey. *Isprs Journal of Photogrammetry and Remote Sensing*, 115, 22-36. <https://doi.org/10.1016/j.isprsjprs.2015.10.004>
- [41]. Blumenfeld, J. (2022). Passive sensors. *Earthdata*. <https://www.earthdata.nasa.gov/learn/backgrounders/passive-sensors>
- [42]. Kogut, P. (2023, 8 junio). Teledetección satelital: tipos, usos y aplicaciones. *EOS Data Analytics*. <https://eos.com/es/blog/teledeteccion/>
- [43]. *Glosario: Espectro electromagnético*. (s. f.). https://ec.europa.eu/health/scientific_committees/opinions_layman/es/lamparas-bajo-consumo/glosario/def/espectro-electromagnetico.htm
- [44]. Alonso, D. (2023). Combinación de bandas en imágenes de satélite Landsat y Sentinel. *MappingGIS*. <https://mappinggis.com/2019/05/combinaciones-de-bandas-en-imagenes-de-satelite-landsat-y-sentinel/>
- [45]. Lavender, S., & Lavender, A. (2015). Practical handbook of remote sensing. En *CRC Press eBooks*. <https://doi.org/10.1201/b19044>
- [46]. Pandey, P. C., Koutsias, N., Petropoulos, G. P., Srivastava, P. K., & Dor, E. B. (2019). Land use/Land cover in view of Earth observation: data sources, input dimensions, and classifiers—A review of the state of the art. *Geocarto*
- [47]. *Types of sensor resolutions applicable to remote sensing applications: radiometric, spatial, spectral and temporal- OneStopGIS (GATE-GeoInformatics 2024)*. (s. f.). <https://www.onestopgis.com/Aerial-Photography/Digital-Imaging/Digital-Image/2-Sensor-Resolutions-Radiometric-Spatial-Spectral-and-Temporal.html>

- [48]. Young, N. E., Anderson, R. S., Chignell, S. M., Vorster, A. G., Lawrence, R. L., & Evangelista, P. (2017). A survival guide to Landsat preprocessing. *Ecology*, 98(4), 920-932. <https://doi.org/10.1002/ecy.1730>
- [49]. CNICE. (s. f.-b). *Plataformas de teledetección*.
<http://concurso.cnice.mec.es/cnice2006/material121/unidad3/satelite1.htm>
- [50]. Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A. Y., & Jie, W. (2015). Remote sensing Big data Computing: challenges and opportunities. *Future Generation Computer Systems*, 51, 47-60. <https://doi.org/10.1016/j.future.2014.10.029>
- [51]. Yang, H., Kong, J., Hu, H., Du, Y., Gao, M., & Chen, F. (2022). A review of Remote sensing for water quality Retrieval: Progress and challenges. *Remote Sensing*, 14(8), 1770. <https://doi.org/10.3390/rs14081770>
- [52]. West, H., Quinn, N., & Horswell, M. (2019). Remote sensing for drought monitoring & Impact Assessment: progress, past challenges and future opportunities. *Remote Sensing of Environment*, 232, 111291. <https://doi.org/10.1016/j.rse.2019.111291>
- [53]. Hao, X., Zhang, G., & Ma, S. (2016). Deep learning. *International journal of semantic computing*, 10(03), 417-439. <https://doi.org/10.1142/s1793351x16500045>
- [54]. Zou, J., Yong, H., & So, S. (2008). Overview of artificial neural networks. En *Methods in molecular biology* (pp. 14-22). https://doi.org/10.1007/978-1-60327-101-1_2
- [55]. *A review of machine learning and deep learning applications*. (2018, 1 agosto). IEEE Conference Publication | IEEE Xplore.
https://ieeexplore.ieee.org/abstract/document/8697857?casa_token=SoevXdhZ_qAAAAA:7MsoO3HOy4Wmey7rRu6KpaMolvu6lPYjKVsdZjaEWwdv5TpHerCiQscjafg5bgN3KF4lpBDiWA
- [56]. Madani, A., Arnaout, R., Mofrad, M. R. K., & Arnaout, R. (2018). Fast and accurate view classification of echocardiograms using deep learning. *npj digital medicine*, 1(1).
<https://doi.org/10.1038/s41746-017-0013-1>
- [57]. Zohuri, B., & Moghaddam, M. A. (2020). Deep Learning Limitations and flaws. *Modern approaches on material science*, 2(3). <https://doi.org/10.32474/mams.2020.02.00013>
- [58]. Garrido-Muñoz, I., Montejo-Ráez, A., Martínez-Santiago, F., & Ureña-López, L. A. (2021). A survey on Bias in Deep NLP. *Applied sciences*, 11(7), 3184.
<https://doi.org/10.3390/app11073184>

- [59]. Krogh, A. (2008). What are artificial neural networks? *Nature Biotechnology*, 26(2), 195-197. <https://doi.org/10.1038/nbt1386>
- [60]. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of mathematical biophysics*, 5(4), 115-133.
<https://doi.org/10.1007/bf02478259>
- [61]. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
<https://doi.org/10.1037/h0042519>
- [62]. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. <https://doi.org/10.1038/323533a0>
- [63]. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
<https://doi.org/10.1109/5.726791>
- [64]. Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International journal of engineering applied science and technology*, 04(12), 310-316. <https://doi.org/10.33564/ijeast.2020.v04i12.054>
- [65]. Rasamoelina, A. D., Adjailia, F., & Sincak, P. (2020). A Review of Activation Function for Artificial Neural Network. *World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. <https://doi.org/10.1109/sami48414.2020.9108717>
- [66]. Jadon, S. (2022, 3 febrero). Introduction to different activation functions for deep learning. *Medium*. <https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092>
- [67]. *Gradient descent in Machine Learning - JavatPoint*. (s. f.). www.javatpoint.com.
<https://www.javatpoint.com/gradient-descent-in-machine-learning>
- [68]. Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3), 400-407. <https://doi.org/10.1214/aoms/1177729586>
- [69]. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1412.6980>

- [70]. Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. *ACM*. <https://doi.org/10.1145/2623330.2623612>
- [71]. Ruder, S. (s/f). *An overview of gradient descent optimization algorithms*. Arxiv.org. Recuperado el 29 de agosto de 2023, de <http://arxiv.org/abs/1609.04747>
- [72]. V. Bushave, "Understanding RMSprop — faster neural network learning", 2018.
- [73]. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two Time-Scale update rule converge to a local Nash equilibrium. *arXiv (Cornell University)*, 30, 6626-6637. <https://arxiv.org/pdf/1706.08500>
- [74]. Duchi, J., Hazan, E., & Singer, Y. (2011). *Adaptive subgradient methods for online learning and stochastic optimization*. Journal of machine learning research.
- [75]. Basodi, S., Ji, C., Zhang, H., & Pan, Y. (2020). Gradient amplification: an efficient way to train deep neural networks. *Big data mining and analytics*, 3(3), 196-207. <https://doi.org/10.26599/bdma.2020.9020004>
- [76]. Hu, Z., Zhang, J., & Ge, Y. (2021). Handling vanishing gradient problem using artificial derivative. *IEEE Access*, 9, 22371-22377. <https://doi.org/10.1109/access.2021.3054915>
- [77]. DeepAI. (2020). Exploding gradient problem. *DeepAI*. <https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem#:~:text=Exploding%20gradients%20are%20a%20problem,updates%20are%20small%20and%20controlled>.
- [78]. Fukushima, K. (1980). NeoCognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193-202. <https://doi.org/10.1007/bf00344251>
- [79]. Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. *International Joint Conference on Artificial Intelligence*, 1237-1242. <https://doi.org/10.5591/978-1-57735-516-8/ijcai11-210>
- [80]. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Q., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of Deep

- Learning: concepts, CNN architectures, challenges, applications, future directions.
Journal of Big Data, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- [81]. colaboradores de Wikipedia. (2023). Convolución. *Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/wiki/Convoluci%C3%B3n>
- [82]. Wolfram Research, Inc. (s. f.). *Convolution -- from Wolfram MathWorld*.
<https://mathworld.wolfram.com/Convolution.html>
- [83]. Yamashita, R., Nishio, M., Gian, R. K., & Togashi, K. (2018). Convolutional Neural Networks: an Overview and application in Radiology. *Insights Into Imaging*, 9(4), 611-629. <https://doi.org/10.1007/s13244-018-0639-9>
- [84]. *SuperDataScience*. (s. f.). <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation>
- [85]. Ajit, A., Acharya, K., & Samanta, A. K. (2020). A review of convolutional neural networks. *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. <https://doi.org/10.1109/ic-etite47903.2020.049>
- [86]. GeeksforGeeks. (2023). CNN Introduction to pooling layer. *GeeksforGeeks*.
<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>
- [87]. *Max-pooling / Pooling - Computer Science Wiki*. (s. f.).
https://computersciencewiki.org/index.php/Max-pooling/_/_Pooling
- [88]. Pokhrel, S. (2021, 11 diciembre). Beginners guide to Convolutional Neural Networks - towards Data science. *Medium*. <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d#:~:text=An%20activation%20function%20is%20the,function%20in%20a%20convolution%20layer.>
- [89]. Yeh, W., Lin, Y., Liang, Y., & Lai, C. (2021). Convolution neural network hyperparameter optimization using simplified swarm optimization. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2103.03995>
- [90]. *How to choose the optimal kernel size? / Devron*. (s. f.).
<https://www.devron.ai/kbase/how-to-choose-the-optimal-kernel-size>

- [91]. NeuralNet, M. (s. f.). *Calculating the output size of convolutions and transpose convolutions*. <http://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html>
- [92]. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.1409.0473>
- [93]. Varea, I. G. (2003). *Traducción automática estadística: modelos de traducción basados en máxima entropía y algoritmos de búsqueda*.
<https://dialnet.unirioja.es/servlet/tesis?codigo=90220>
- [94]. Oliver, A., Boleda, G., Melero, M., & Badia, T. (2005). Traducción automática estadística basada en n-gramas. *Procesamiento Del Lenguaje Natural*, 35(35), 77-84.
<http://dblp.uni-trier.de/db/journals/pdln/pdln35.html#GonzalezBMB05>
- [95]. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural Image caption Generation with visual attention. *International Conference on Machine Learning*, 3, 2048-2057.
<http://proceedings.mlr.press/v37/xuc15.pdf>
- [96]. Li, Y., Yang, L., Xu, B., Wang, J., & Lin, H. (2019). Improving user attribute classification with text and social network attention. *Cognitive Computation*, 11(4), 459-468. <https://doi.org/10.1007/s12559-019-9624-y>
- [97]. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., & Liu, W. (2018). Attention-Based transactional context embedding for Next-Item recommendation. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, 32(1).
<https://doi.org/10.1609/aaai.v32i1.11851>
- [98]. Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, 452, 48-62. <https://doi.org/10.1016/j.neucom.2021.03.091>
- [99]. Weng, L. (2018, 24 junio). Attention? attention! *Lil'Log*.
<https://lilianweng.github.io/posts/2018-06-24-attention/>

- [100]. Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning where to look for the pancreas. *arXiv (Cornell University)*. <http://export.arxiv.org/pdf/1804.03999>
- [101]. Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. En *Lecture Notes in Computer Science* (pp. 3-19). https://doi.org/10.1007/978-3-030-01234-2_1
- [102]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv (Cornell University)*, 30, 5998-6008. <https://arxiv.org/pdf/1706.03762v5>
- [103]. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv (Cornell University)*. <https://arxiv.org/pdf/1810.04805v2>
- [104]. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv (Cornell University)*. <https://arxiv.org/pdf/2010.11929>
- [105]. *Papers with code - Vision Transformer explained.* (s. f.). <https://paperswithcode.com/method/vision-transformer>
- [106]. Boesch, G. (2023). Vision Transformers (ViT) in Image Recognition – 2023 Guide. *viso.ai*. <https://viso.ai/deep-learning/vision-transformer-vit/>
- [107]. Educative. (s. f.). *What is the intuition behind the Dot product attention?* Educative: Interactive Courses for Software Developers. <https://www.educative.io/answers/what-is-the-intuition-behind-the-dot-product-attention>
- [108]. *Why does this multiplication of QK^T and KV^T have a variance of $\frac{1}{d_k}$, in scaled dot product attention?* (s. f.). Artificial Intelligence Stack Exchange. <https://ai.stackexchange.com/questions/21237/why-does-this-multiplication-of-q-and-k-have-a-variance-of-d-k-in-scaled>

- [109]. Storrs, E. (2021). Explained: Multi-head attention (Part 1). *Erik Storrs*.
<https://storrs.io/attention/>
- [110]. Bouchard, Louis (2020-11-25). "What is Self-Supervised Learning? | Will machines ever be able to learn like humans?". Medium. Retrieved 2021-06-09.
- [111]. Yarowsky, David (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods". Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, MA: Association for Computational Linguistics: 189–196. doi:10.3115/981658.981684. Retrieved 1 November 2022.
- [112]. Doersch, Carl; Zisserman, Andrew (October 2017). "Multi-task Self-Supervised Visual Learning". 2017 IEEE International Conference on Computer Vision (ICCV). IEEE. pp. 2070–2079. arXiv:1708.07860. doi:10.1109/iccv.2017.226. ISBN 978-1-5386-1032-9. S2CID 473729.
- [113]. Bai, W., Chen, C., Tarroni, G., Duan, J., Guitton, F., Petersen, S. E., Guo, Y., Matthews, P. M., & Rueckert, D. (2019). Self-Supervised Learning for cardiac MR image segmentation by Anatomical Position Prediction. En *Springer eBooks* (pp. 541-549).
https://doi.org/10.1007/978-3-030-32245-8_60
- [114]. Gopani, A. (2022). Contrastive vs non-contrastive self-supervised learning techniques. *Analytics India Magazine*. <https://analyticsindiamag.com/contrastive-vs-non-contrastive-self-supervised-learning-techniques/>
- [115]. Van Den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv (Cornell University)*.
<http://export.arxiv.org/pdf/1807.03748>
- [116]. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020). Big Self-Supervised models are strong Semi-Supervised learners. *arXiv (Cornell University)*.
<https://arxiv.org/pdf/2006.10029>
- [117]. "Demystifying a key self-supervised learning technique: Non-contrastive learning". ai.facebook.com. Retrieved 2021-10-05.

- [118]. Zhang, C., Zhang, K., Zhang, C., Pham, T. X., Yoo, C. D., & Kweon, I. S. (2022). How does SiMSiAM avoid collapse without negative samples? A unified understanding with self-supervised contrastive learning. *arXiv (Cornell University)*.
<https://doi.org/10.48550/arxiv.2203.16262>
- [119]. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using Deep Learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1.
<https://doi.org/10.1109/tpami.2021.3059968>
- [120]. Abril, R. R. (2023). Segmentación panóptica. *La Máquina Oráculo*.
<https://lamaquinaoraculo.com/deep-learning/segmentacion-panoptica/>
- [121]. Al-Amri, S. S., Kalyankar, N. V., & Khamitkar, S. (2010). Image segmentation by using threshold techniques. *arXiv (Cornell University)*. <https://arxiv.org/pdf/1005.4020>
- [122]. Krstinić, D., Skelin, A. K., & Slapničar, I. (2011). Fast two-step histogram-based image segmentation. *Iet Image Processing*, 5(1), 63. <https://doi.org/10.1049/iet-ipr.2009.0107>
- [123]. Preetha, M. M. S. J., Suresh, L. P., & Bosco, M. (2012). Image segmentation using seeded region growing. *IEEE*. <https://doi.org/10.1109/icceet.2012.6203897>
- [124]. Dhanachandra, N., Mangle, K., & Chanu, Y. J. (2015). Image segmentation using K - Means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54, 764-771. <https://doi.org/10.1016/j.procs.2015.06.090>
- [125]. Plath, N., Toussaint, M., & Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. *ACM*.
<https://doi.org/10.1145/1553374.1553479>
- [126]. Kato, Z., & Zerubia, J. (2011). Markov Random fields in image segmentation. *Foundations and Trends in Signal Processing*, 5(1-2), 1-155.
<https://doi.org/10.1561/20000000035>
- [127]. Yu, Y., Huang, J., Zhang, S., Restif, C., Huang, X., & Metaxas, D. N. (2011). Group sparsity based classification for cervigram segmentation. *IEEE*.
<https://doi.org/10.1109/isbi.2011.5872667>

- [128]. Milletari, F., Navab, N., & Ahmadi, S. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *IEE*.
<https://doi.org/10.1109/3dv.2016.79>
- [129]. Kaul, C., Manandhar, S., & Pears, N. (2019). Focusnet: An Attention-Based Fully Convolutional Network for Medical Image Segmentation. *arXiv*.
<https://doi.org/10.1109/isbi.2019.8759477>
- [130]. Rehman, M. U., Cho, S., Kim, J., & Chong, K. T. (2021). BrainSeg-Net: Brain tumor MR image segmentation via enhanced Encoder–Decoder Network. *Diagnostics*, 11(2), 169. <https://doi.org/10.3390/diagnostics11020169>
- [131]. *CORINE Land Cover — Copernicus Land Monitoring Service*. (s. f.).
<https://land.copernicus.eu/pan-european/corine-land-cover>
- [132]. *Open Access Hub*. (s. f.). <https://scihub.copernicus.eu/>
- [133]. *Bienvenido al proyecto QGIS!* (s. f.). <https://www.qgis.org/es/site/>
- [134]. *Sentinel-2 - Missions - Sentinel Online - Sentinel Online*. (s. f.). Sentinel Online.
<https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2>
- [135]. *OpenStreetMap*. (s. f.). OpenStreetMap.
<https://www.openstreetmap.org/#map=6/40.007/-2.488>
- [136]. Jing, K. H. (2019, 7 noviembre). Biomedical Image Segmentation - U-NeT. *Khok Hong Jing (Jingles)*. <https://jinglescode.github.io/2019/11/07/biomedical-image-segmentation-u-net/>
- [137]. *Guide to Image Segmentation in Computer Vision: Best Practices*. (s. f.). Encord.
<https://encord.com/blog/image-segmentation-for-computer-vision-best-practice-guide/>
- [138]. Parsad, N. M. (2018, 15 junio). Deep Learning in Medical Imaging V - DataDrivenInvestor. *Medium*. <https://medium.datadriveninvestor.com/deep-learning-in-medical-imaging-3c1008431aaf>
- [139]. Tiu, E. (2023, 1 agosto). Metrics to evaluate your semantic segmentation model. *Medium*. <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>

- [140]. Csurka, G., Larlus, D., & Perronnin, F. (2013). What is a good evaluation measure for semantic segmentation? *BMVC*. <https://doi.org/10.5244/c.27.32>
- [141]. Müller, D., Soto-Rey, I., & Kramer, F. (2022). Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15(1).
<https://doi.org/10.1186/s13104-022-06096-y>
- [142]. Liashchynskiy, P. (2019, 12 diciembre). *Grid search, random search, genetic algorithm: A big comparison for NAS*. arXiv.org. <https://arxiv.org/abs/1912.06059>
- [143]. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Neural Information Processing Systems*, 25, 2951-2959.
http://books.nips.cc/papers/files/nips25/NIPS2012_1338.pdf
- [144]. Blumenfeld, J. (2022). Passive sensors. *Earthdata*.
<https://www.earthdata.nasa.gov/learn/backgrounders/passive-sensors>
- [145]. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Cardoso, M. J. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. En *Lecture Notes in Computer Science* (pp. 240-248). https://doi.org/10.1007/978-3-319-67558-9_28
- [146]. Jadon, S. (2020). A survey of loss functions for semantic segmentation. *arXiv*.
<https://doi.org/10.1109/cibcb48159.2020.9277638>
- [147]. (S/f). Nasa.gov. Recuperado el 6 de septiembre de 2023, de
https://appliedsciences.nasa.gov/sites/default/files/2023-03/Fundamentals_of_RS_Span.pdf

OBJETIVOS DE DESARROLLO SOSTENIBLE

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.	X			X
ODS 3. Salud y bienestar.				
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.	X			
ODS 7. Energía asequible y no contaminante.	X			
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.		X		
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.	X			
ODS 14. Vida submarina.	X			
ODS 15. Vida de ecosistemas terrestres.	X			
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

El presente trabajo representa una contribución significativa a múltiples Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A través de la aplicación de modelos de Deep Learning para la generación de mapas de Clasificación de Uso del Suelo y Cobertura Terrestre, nuestro proyecto no solo busca avanzar en la comprensión y gestión de los recursos naturales, sino también en la promoción de un desarrollo sostenible y la mejora de la calidad de vida de las comunidades locales. A continuación, se justifica la contribución de este trabajo a los ODS específicos.

Contribución a los ODS:

ODS 3. Salud y bienestar:

La precisión en la clasificación de LULC permite un mejor monitoreo de la distribución de recursos naturales, lo que a su vez tiene un impacto directo en la salud y el bienestar de las poblaciones locales. La gestión adecuada de tierras y recursos hídricos contribuye a la prevención de desastres naturales y a la planificación de infraestructuras de salud y saneamiento.

ODS 6. Agua limpia y saneamiento:

La generación de mapas LULC es esencial para identificar áreas críticas de conservación y protección de recursos hídricos. Esto facilita la planificación de proyectos relacionados con el suministro de agua potable y el saneamiento básico, ayudando así a alcanzar el acceso universal a servicios de agua limpia y saneamiento.

ODS 7. Energía asequible y no contaminante:

La identificación precisa de áreas adecuadas para proyectos de energía renovable, como la ubicación de parques eólicos y solares, se beneficia enormemente de los mapas LULC. Esto promueve la expansión de fuentes de energía limpias y sostenibles, contribuyendo a la meta de energía asequible y no contaminante.

ODS 9. Industria, innovación e infraestructuras:

La aplicación de modelos de Deep Learning para la generación de mapas LULC impulsa la innovación en tecnologías de teledetección y análisis geoespacial. Esto puede ser fundamental para el desarrollo de infraestructuras más eficientes y sostenibles, así como para la toma de decisiones informadas en la industria y la planificación urbana.

ODS 11. Ciudades y comunidades sostenibles:

La disponibilidad de mapas LULC precisos es esencial para la planificación urbana sostenible. Ayuda a las autoridades locales a tomar decisiones informadas sobre el uso del suelo, la gestión de espacios verdes y la infraestructura, contribuyendo a la creación de ciudades más habitables y sostenibles.

ODS 12. Producción y consumo responsables:

El conocimiento detallado de la cobertura terrestre puede promover prácticas agrícolas y de uso de la tierra más responsables y sostenibles. Esto fomenta la producción y el consumo responsables al reducir la presión sobre los recursos naturales y minimizar el impacto ambiental.

ODS 13. Acción por el clima:

El monitoreo preciso de los cambios en la cobertura terrestre es esencial para evaluar y abordar el cambio climático. La información proporcionada por los mapas LULC puede contribuir a la planificación de estrategias de mitigación y adaptación al cambio climático.

ODS 14. Vida submarina y ODS 15. Vida de ecosistemas terrestres:

La conservación de la biodiversidad marina y terrestre está estrechamente relacionada con la gestión adecuada de la tierra y los recursos hídricos. La generación de mapas LULC ayuda a identificar áreas críticas para la conservación, contribuyendo a la protección de la vida submarina y de los ecosistemas terrestres.

En resumen, este trabajo desempeña un papel fundamental en la promoción de una serie de Objetivos de Desarrollo Sostenible al proporcionar información esencial para la toma de decisiones informadas y la gestión sostenible de los recursos naturales.