

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL
CUSCO**

**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y
MECÁNICA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

**APLICACIÓN DE DEEP LEARNING Y
PROCESAMIENTO DE IMÁGENES SATELITALES PARA
EL ANÁLISIS TEMPORAL DEL RETROCESO
SUPERFICIAL DEL GLACIAR QUELCCAYA EN LA
CORDILLERA VILCANOTA, PERÚ**

Presentado por:

Br. Andres Benjamin Quispe Quispe

Para optar al título profesional de:

INGENIERO ELECTRÓNICO

Asesor:

Dr. Ing. Facundo Palomino Quispe

Cusco - Perú

2024

Dedicatoria

Dedico esta tesis a mi familia, pilar fundamental de cada uno de mis logros. A mi abuelita Rosa, quien me brindó su amor incondicional y me crió como a un hijo propio. A mi padre, Walter, por su apoyo constante y sus enseñanzas, que siempre me impulsan a seguir adelante. A mis queridas hermanas, Raquel y Yanina, quienes han sido como una madre para mí; sus palabras de aliento y ánimo han fortalecido mis metas en cada paso. A mi hermano Michael y a mis sobrinos, que me inspiran a continuar en esta carrera y han sido siempre mi refugio. Y, sobre todo, agradezco profundamente a Dios, quien fortalece y alimenta mi espíritu. Él ha sido mi confianza y mi fortaleza a lo largo de este proceso.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de esta investigación. Extiendo un agradecimiento especial a Dios quien ha sido mi guía a lo largo de este camino, a mi familia, quienes han sido mi sostén y mi fuente de motivación a lo largo de este proceso. Su apoyo incondicional ha sido fundamental para alcanzar este logro.

Agradezco, al Dr. Ing. Facundo Palomino, mi asesor, por su orientación, paciencia y constante preocupación. Sus consejos y su apoyo han sido una fuente de inspiración invaluable en cada etapa de este trabajo.

Agradezco también a la Dra. Ing. Ana Beatriz Álvarez, por su generosidad al compartir sus conocimientos y por su apoyo al laboratorio LIECAR, que fueron necesarios para llevar a cabo este estudio.

Finalmente, agradezco a todos mis amigos del laboratorio institucional LIECAR que, de una u otra forma, contribuyeron con su apoyo y compañía en este proceso, y a cada persona que, con sus palabras y acciones, me alentó a seguir adelante.

Resumen

En las últimas décadas, los glaciares tropicales han retrocedido significativamente debido al calentamiento global, afectando la hidrología y el paisaje montañoso, y aumentando los riesgos asociados. Su monitoreo es esencial para cuantificar estas variaciones, dado su rol como reservas de agua dulce e indicadores del cambio climático.

Las técnicas de teledetección, ampliamente utilizadas desde la década de 1970, permiten detectar vegetación, agua, glaciares, etc. La comunidad científica emplea estos datos para el monitoreo de la cobertura del suelo, incluida la detección de cuerpos glaciares. Estudios previos han usado índices espectrales para detectar y segmentar glaciares; sin embargo, estos métodos presentan limitaciones y pueden ser complejos cuando se trata de inventariar glaciares extensos con precisión.

Por otro lado, los métodos de aprendizaje profundo han demostrado ser muy eficientes para el procesamiento y segmentación de imágenes en diversas disciplinas. Este estudio analiza temporalmente la variación del glaciar Quelccaya, en la cordillera de Vilcanota entre Cusco y Puno, en el periodo 1991-2024. Se emplean métodos de aprendizaje profundo junto con datos ópticos de teledetección de las misiones Landsat, generando un conjunto de datos de 2400 muestras para tareas de segmentación semántica y el entrenamiento de un modelo de deep learning que extrae automáticamente cuerpos glaciares a partir de imágenes multiespectrales.

Se proponen tres modelos de deep learning: U-Net, DeepResUnet y DeepLabV3Plus, todos para segmentación semántica. Estos modelos se entrenaron con el 70 % de las muestras, usando el 15 % para validación y el 15 % para evaluación. Para evaluar la eficiencia de los modelos se utilizaron métricas como el Dice Coefficient (Dice), la Intersección sobre Unión

media (mIoU) y la Precisión por Pixel (PA). De estos, U-Net mostró los mejores resultados con una mejor precisión en segmentación y extracción glaciar por encima de los otros modelos. U-Net se empleó para mapear el glaciar Quelccaya desde 1991 hasta 2024 en el área de estudio y evaluar el impacto del cambio climático. Los resultados indican que la superficie del glaciar Quelccaya disminuyó notablemente en un 30.15 % durante este periodo, pasando de 50.81 km² en 1991 a 35.49 km² en 2024, con una tasa de disminución de 0.38 km²/año y un intervalo de confianza del 93 %. Los resultados de este estudio fueron comparados con los de investigaciones previas sobre el glaciar Quelccaya, mostrando una alta coincidencia con estos estudios anteriores.

Palabras clave: Teledetección; Landsat; procesamiento de imágenes; aprendizaje profundo; segmentación semántica.

Abstract

In recent decades, tropical glaciers have experienced a global retreat, primarily due to global warming. The loss of glacier surface has become increasingly severe, driving changes in hydrology and mountain landscape development, which poses multiple risks. Quantifying glacier variations is of great interest, both due to the importance of glaciers as freshwater resources and their role as indicators of climate change. For this reason, it is essential to monitor glacier dynamics in order to quantify these variations.

Remote sensing techniques, widely used since the 1970s, allow for the detection of vegetation, water, glaciers, and more. The scientific community employs this data for land cover monitoring, including the detection of glacier bodies. Previous studies have used spectral indices to detect and segment glaciers; however, these methods have limitations and can be complex when it comes to accurately inventorying extensive glaciers.

On the other hand, deep learning methods have proven to be highly efficient for image processing and segmentation across various disciplines. This study temporally analyzes the variation of the Quelccaya Glacier in the Vilcanota mountain range, between Cusco and Puno, over the period from 1991 to 2024. Deep learning methods are employed alongside optical remote sensing data from the Landsat missions, generating a dataset of 2400 samples for semantic segmentation tasks and training a deep learning model that automatically extracts glacier bodies from multispectral images.

Three deep learning models are proposed: U-Net, DeepResUnet, and DeepLabV3Plus, all for semantic segmentation. These models were trained with 70 % of the samples, using 15 % for validation and 15 % for evaluation. To assess the efficiency of the models, metrics such as the

Dice Coefficient (Dice), Mean Intersection over Union (mIoU), and Pixel Accuracy (PA) were used. Of these, U-Net showed the best results with superior segmentation accuracy and glacier extraction compared to the other models. U-Net was used to map the Quelccaya Glacier from 1991 to 2024 in the study area and evaluate the impact of climate change. The results indicate that the surface area of the Quelccaya Glacier decreased significantly by 30.15 % during this period, from 50.81 km² in 1991 to 35.49 km² in 2024, with a rate of decrease of 0.38 km²/year and a 93 % confidence interval. The results of this study were compared with previous research on the Quelccaya Glacier, showing a high degree of agreement with these earlier studies.

Índice general

Índice general	vii
Índice de tablas	x
Índice de figuras	xi
Introducción	1
1. Aspectos generales	3
1.1. Ámbito geográfico	3
1.2. Descripción de la realidad problemática	3
1.3. Formulación del problema	9
1.3.1. Problema general	9
1.3.2. Problemas específicos	10
1.4. Justificación	10
1.5. Objetivos	12
1.5.1. Objetivo general	12
1.5.2. Objetivos específicos	13
1.6. Hipótesis	13
1.6.1. Hipótesis general	13
1.6.2. Hipótesis específicos	13
1.7. Variables e indicadores	14
1.7.1. Identificación de variables	14

1.8.	Alcance de la investigación	15
1.9.	Viabilidad y factibilidad	15
1.10.	Limitaciones de la investigación	16
1.11.	Delimitación del estudio	17
2.	Marco teórico	18
2.1.	Antecedentes de estudio	18
2.2.	Bases teóricas	23
2.2.1.	Glaciares	23
2.2.2.	Retroceso glaciar	26
2.2.3.	Principales causas del retroceso glaciar	26
2.2.4.	Teledetección	28
2.2.5.	Radiación electromagnética	29
2.2.6.	Interacción de la radiación con la atmósfera	30
2.2.7.	Resolución de imágenes	32
2.2.8.	Programa Satélite Landsat 8	33
2.2.9.	Programa Satélite Landsat 5	34
2.2.10.	Landsat 8 OLI/TIRS Collection 2 Level 1	35
2.2.11.	Landsat Thematic Mapper (TM) Collection 2 Level 1	36
2.2.12.	Indice espectral	36
2.2.13.	Procesamiento de imágenes digitales	38
2.2.14.	Deep Learning	42
2.2.15.	Estructura de entradas y salidas	42
2.2.16.	Aprendizaje Supervizado	44
2.2.17.	Función de perdida	45
2.2.18.	Entrenamiento	46
2.2.19.	Testeo	47
2.2.20.	Redes Neuronales profundas	48

2.2.21. Aprendizaje no Supervizado	49
2.2.22. Redes neuronales artificiales	49
2.2.23. Redes Neuronales Convolucionales	50
3. Diseño del sistema de segmentación de cuerpos glaciares	61
3.1. Base de datos	62
3.1.1. Adquisición de imágenes	63
3.1.2. Acondicionamiento de las imágenes satelitales	63
3.1.3. Generación de máscaras	65
3.1.4. Tratamiento de imágenes	66
3.2. Arquitecturas de aprendizaje profundo (Deep learning)	68
3.2.1. Implementación de la arquitectura de deep learning	68
3.2.2. Resultados de entrenamiento	75
4. Experimentos y resultados	77
4.1. Experimentos	77
4.1.1. Evolución del glaciar Quelccaya	79
4.2. Resultados	83
5. Comparación con trabajos previos	85
5.1. Análisis del modelo	88
5.2. Análisis del rendimiento de segmentación de glaciares	89
5.3. Segmentación de glaciares y variación de glaciares	90
Bibliografía	101
Anexos	101

Índice de tablas

2.1. Referencia de bandas	35
2.2. Especificaciones de las bandas OLI	36
2.3. Especificaciones de las bandas TIRS	36
2.4. Referencia de bandas	37
2.5. Especificaciones de las bandas TM	37
3.1. Párametros de entrenamiento	74
4.1. Párametros de entrenamiento	77
4.2. Resultado cuantitativo de la estimación de superficie glaciar a travez del tiempo	83

Índice de figuras

1.1.	Ubicación del manto glaciar Quelccaya	4
1.2.	<i>Incidencia de anemia por grupo etario y según residencia.</i>	6
1.3.	<i>Incidencia de anemia por grupo etario y según residencia.</i>	6
1.4.	<i>Incidencia de anemia por grupo etario y según residencia.</i>	7
1.5.	Retroceso superficial del Glaciar Quelccaya vista a través de una imagen satelital.	7
2.1.	Componentes circundantes de un glaciar.	24
2.2.	Localización de las cordilleras glaciares en el Perú.	25
2.3.	Carbono negro en la superficie del glaciar Yanapaccha, cordillera Blanca. . . .	28
2.4.	Espectro electromagnético basada en el rango de longitud de onda y frecuencia.	29
2.5.	Transmisión Atmosférica.	31
2.6.	Ventanas Atmosféricas en Imágenes Ópticas Multiespectrales.	31
2.7.	Subsistemas de la plataforma L8.	33
2.8.	Subsistemas de la plataforma L5.	35
2.9.	Modélo basico de un sensor para adquirir imágenes.	40
2.10.	Ejemplo de adquisición y formación de una imagen digital	42
2.11.	Subconjuntos de la inteligencia artificial.	43
2.12.	Tareas de aprendizaje supervisado con salidas estructuradas.	44
2.13.	Modelo representativo de datos de entrenamiento, entrada/salida.	46
2.14.	Ajuste de la función de perdida.	47
2.15.	Visualización de la función de perdida.	48

2.16. Ajuste de la función de perdida en el entrenamiento.	48
2.17. Estructura jerárquica de una ANN	49
2.18. Proceso de convolución.	53
2.19. Convolución 2D aplicado a una imagen RGB	53
2.20. Representación de una CNN	54
2.21. Segmentación semántica.	54
2.22. Arquitectura U-Net	55
2.23. Arquitectura profunda ResUnet	58
2.24. Arquitectura profunda DeepLabV3+	59
 3.1. Flujo de trabajo en este estudio.	61
3.2. Procesamiento de datos para el análisis temporal.	62
3.3. Escenas de Landsat seleccionadas para el conjunto de datos.	64
3.4. Acondicionamiento de imágenes Landsat 8.	65
3.5. Corrección de agua y sombras.	67
3.6. Proceso de creación de máscaras	67
3.7. Arquitectura del modelo U-Net	69
3.8. Arquitectura del modelo DeepResUnet.	70
3.9. Bloques fundamentales de las redes neuronales.	71
3.10. Arquitectura del modelo DeepLabV3Plus.	71
3.11. Representación de muestras de conjuntos de datos (RGB y máscara) utilizadas para entrenar el modelo.	73
3.12. Gráficos de pérdida de entrenamiento y métricas de precisión para U-NET.	75
3.13. Gráficos de pérdida de entrenamiento y métricas de precisión para DeepResUnet.	76
3.14. Gráficos de pérdida de entrenamiento y métricas de precisión para DeepLabV3Plus.	76
 4.1. Resultado del indicador MIou evaluado para cada modelo.	78
4.2. Resultado del indicador PA evaluado para cada modelo.	78
4.3. Resultado del indicador Dice evaluado para cada modelo.	78

4.4. Resultado de convergencia de la función de pérdida (Loss) evaluado para cada modelo.	79
4.5. Análisis cualitativo y cuantitativo de los modelos evaluados.	80
4.6. Proceso de predicción y reconstrucción para calcular el área segmentada.	81
4.7. Retroceso superficial del glaciar Quelccaya.	81
4.8. Cuerpos glaciares extintos.	81
4.9. Contornos glaciares más antiguos y más recientes del glaciar Quelccaya.	82
4.10. Análisis temporal del glaciar Quelccaya.	84
5.1. Serie temporal de superficie glaciar Quelccaya.	87

Introducción

Esta investigación analiza el retroceso del glaciar Quelccaya mediante un estudio temporal desde 1991 hasta 2024. Para estimar la superficie glaciar a lo largo del tiempo, se emplearon imágenes multiespectrales de la misión Landsat y técnicas de deep learning. El objetivo principal de este proyecto es aplicar técnicas de deep learning para clasificar y segmentar cuerpos glaciares a partir de imágenes satelitales, permitiendo así realizar un análisis temporal y estimar el retroceso de la superficie del glaciar Quelccaya, ubicado en la Cordillera de Vilcanota, Perú. El estudio está estructurado en seis capítulos que abordan diversos aspectos del proyecto.

En el Capítulo 1, se presentan los aspectos generales de la investigación, incluyendo el área de estudio y la importancia de monitorear este glaciar, que tiene relevancia como fuente de agua dulce, además de su valor económico y cultural. También se definen los objetivos, hipótesis, alcance del estudio.

El Capítulo 2 proporciona el marco teórico necesario para comprender los conceptos básicos que sustentan los objetivos de este estudio, explorando temas relacionados con teledetección, deep learning y métodos de procesamiento de imágenes.

El Capítulo 3 se centra en el diseño de un sistema de segmentación de cuerpos glaciares. Aquí se detalla la metodología, desde la creación de la base de datos, que incluye adquisición y acondicionamiento de datos, generación de máscaras y tratamiento de imágenes, hasta la implementación de arquitecturas de deep learning. Se presentan los resultados de cada arquitectura evaluada para determinar la más adecuada para esta tarea.

En el Capítulo 4, se describen los experimentos y resultados obtenidos. Se detallan los

experimentos realizados con las distintas arquitecturas propuestas, identificando la más óptima para la segmentación de cuerpos glaciares. Se analiza el retroceso de la superficie glaciar entre 1991 y 2024 utilizando las imágenes segmentadas por la arquitectura seleccionada, y se realiza un análisis temporal de la superficie glaciar en los años estudiados.

En Capítulo 5 se comparan los resultados obtenidos con aquellos de estudios previos que también investigaron el retroceso del glaciar Quelccaya. Además se compara la metodología de esta investigación con la metodología de los estudios previos.

Finalmente, en el Capítulo 6 se realizó la discusión, donde se interpretan, analizan y comparan los resultados obtenidos.

Capítulo 1

Aspectos generales

1.1. Ámbito geográfico

El desarrollo de la tesis tendrá como área de estudio el glaciar Quelccaya, el cual se encuentra sobre una amplia meseta al sureste del Perú, en la Cordillera Vilcanota, entre los departamentos de Cusco y Puno. Considerada la segunda capa de hielo tropical más grande del mundo (Malone et al., 2022), (Taylor et al., 2022). La elevación media del margen de hielo es de 5300 m.s.n.m. y la elevación aproximada de la cumbre es de 5680 m.s.n.m. (Yarleque et al., 2018)

1.2. Descripción de la realidad problemática

El retroceso glaciar en los Andes del Perú es un fenómeno preocupante que ha estado ocurriendo en las últimas décadas y que tiene fuertes vínculos con la contaminación global y el cambio climático. Los glaciares en la región andina han experimentado una disminución significativa en su tamaño y volumen, lo que se conoce como retroceso glaciar. El aumento de la temperatura a nivel global llega a tener impactos directos en los glaciares, pues contribuye al derretimiento acelerado y al retroceso de las masas de hielo en las altas montañas (Drenkhan et al., 2018).

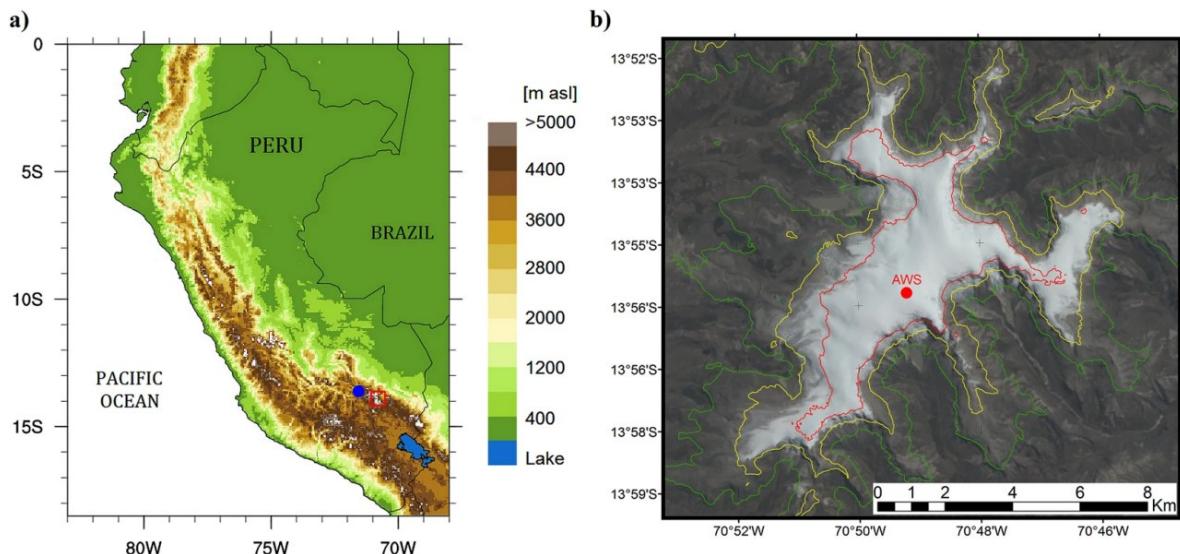


Figura 1.1: Ubicación del manto glaciar Quelccaya, (a) Topografía de los Andes, ubicación del glaciar Quelccaya (cuadrado rojo). (b) Imagen RGB LANDSAT 8 del glaciar Quelccaya obtenida el 2 de agosto de 2017. La ubicación AWS se muestra con un punto rojo. Los contornos de color representan las isolíneas 5100 (verde), 5300 (amarillo) y 5500 m s.n.m. (rojas)

Fuente: (Yarleque et al., 2018)

El Perú, hogar de una porción significativa de la biodiversidad global, concentra el 68 % de los glaciares tropicales del mundo según (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023). Sin embargo, debido al acelerado retroceso de la superficie de los glaciares, se estima que para el 2030 disminuya en un 6 % la disponibilidad hídrica en la vertiente del pacífico, esto llega a generar una gran problemática para el consumo humano, sistemas de riego y generación de energía eléctrica (Ruiz, 2023). Pues estos glaciares en el Perú llegan a ser grandes reservas de agua dulce, sin embargo, recientes estudios de (Aliaga & Cosme, 2018) indican que en los últimos 35 años se ha perdido cerca del 22 % de la superficie glaciar, esto fue evidenciado en una comparativa que se realizó entre los años 1970 teniendo una superficie glaciar equivalente a 1958 km², 2006 teniendo una superficie glaciar equivalente a 1370 km² y registros actualizados al 2010 equivalente a 1230 km² de superficie .

Además, La Cordillera del Vilcanota provee de agua al valle del río Vilcanota, en este valle se concentra el 75 % de la población del Departamento del Cusco, 82 % de terrenos de cultivos lo cual llega a constituir parte importante de la economía de los pobladores. En este valle se encuentran ciudades como Tinta, Sicuani, Combapata, Urcos, Pisaq, Calca, Urubamba, Ollantaytambo, Machupicchu, Quillabamba, etc. (Porcel, 2015). La Central Hidroeléctrica de

Machupicchu, fuente de energía que abastece a todo el departamento de Cusco, está ubicado en el curso medio del río Vilcanota, el río proporciona $131 \text{ m}^3/\text{s}$ de agua promedio anual para la Central Hidroeléctrica (Cabrera Paredes & Castillo Melendez, 2019)

Cabe recalcar que la Cordillera Vilcanota en el sur de Perú, es considerada como la segunda cadena glaciar con mayor presencia de nieve en Perú y sus nevados Cayangate, Tacusiri, Surimani, Qolpacunca, Chimboya, Chumpi, Ambroca, y el glaciar de Quelccaya, son de gran importancia a nivel hídrico en la región, siendo actualmente el Quelcaya el segundo glaciar más extenso del mundo y plano en la zona tropical y es de gran interés científico, pues permite estimar el ritmo de desglaciación de estas montañas y analizar los cambios climáticos ocurridos en el trópico desde la última era glaciar (Taylor et al., 2022). Además, alberga uno de los lagos alpinos más grandes, Sibinacocha. Es en esta zona donde el *Telmatobius marmoratus* (rana de agua veteada), *Rhinella spinulosa* (sapo andino) y *Pleurodema marmoratum* (rana de cuatro ojos veteada) han ampliado su área de distribución para habitar en estanques recién formados que fueron creados a causa de la deglaciación en curso, según (Seimon et al., 2017) revela cambios dinámicos en el hábitat que es afectada por la deglaciación en curso, pues se genera la formación de nuevos estanques, cambios en la vegetación en el hábitat de los anfibios. Por lo tanto, mantener la humedad en la Cordillera Vilcanota es esencial para garantizar la supervivencia de estas especies, adaptándose al cambio climático continuo.

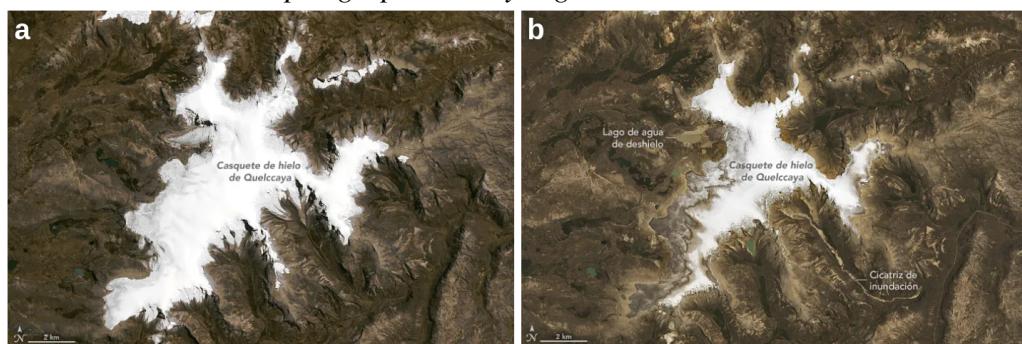
El casquete de hielo de Quelccaya, ubicado en una meseta de gran altitud en los Andes peruanos, está sufriendo los efectos del cambio climático a pesar de las temperaturas frías en estas alturas. Entre 1988 y 2023, la extensión del glaciar se redujo de 58 km^2 a 40 km^2 , según estimaciones de Christopher Shuman, glaciólogo de la Universidad de Maryland y colaborador de la NASA. Este retroceso ha sido documentado mediante imágenes satelitales Landsat, que también han revelado la aparición y desaparición de lagos glaciares durante este periodo. En particular, en noviembre de 2022, un desbordamiento violento de un lago glaciar causó una inundación, dejando una cicatriz visible en las imágenes de 2023 (NASA, 2024).

Desde 1974, Lonnie Thompson y su equipo de la Universidad de Ohio han liderado investigaciones sobre el casquete de Quelccaya, documentando su retroceso con imágenes terrestres

y satelitales. Además, el equipo ha perforado núcleos de hielo del glaciar, proporcionando un registro histórico de la temperatura y la composición atmosférica que se remonta a 1.800 años. Además estos glaciares tropicales ofrecen una valiosa oportunidad para comprender los cambios en la temperatura media global y el clima en áreas que representan más del 50 % de la superficie del planeta, donde vive más del 50 % de la población mundial. Sin embargo, se estima que el casquete de hielo de Quelccaya podría desaparecer antes de que termine el siglo XXI, dejando solo imágenes terrestres y satelitales como testimonio de su existencia (NASA, 2024).

Figura 1.2:

Incidencia de anemia por grupo etario y según residencia.

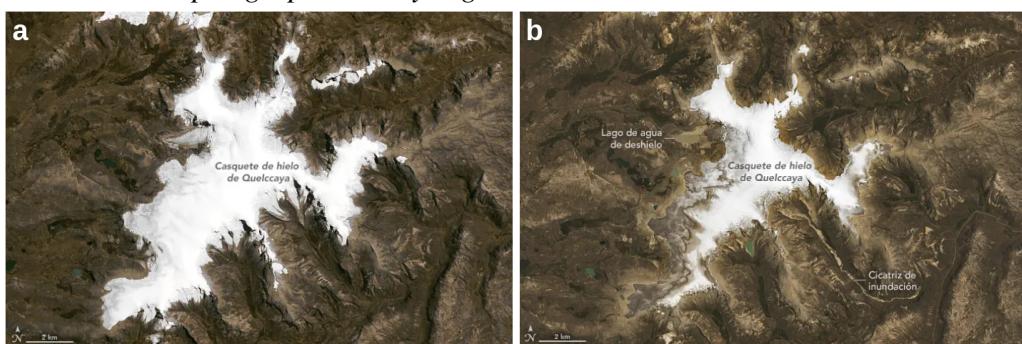


Nota: Retroceso superficial del Glaciar Quelccaya vista a través de una imagen satelital. a) Imagen de 3 de septiembre de 1988, b) Imagen del 22 de octubre de 2023.

Fuente: Elaboración propia.

Figura 1.3:

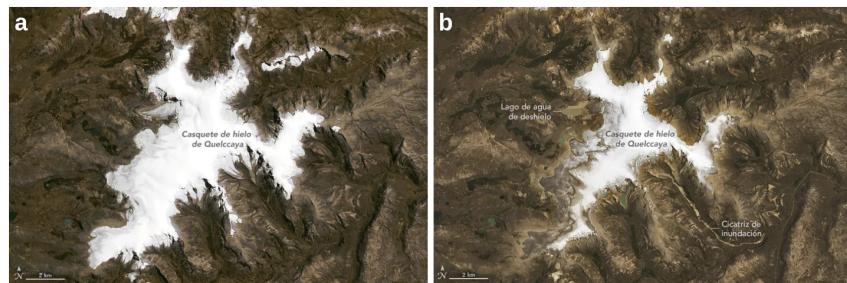
Incidencia de anemia por grupo etario y según residencia.



Nota: Retroceso superficial del Glaciar Quelccaya vista a través de una imagen satelital. a) Imagen de 3 de septiembre de 1988, b) Imagen del 22 de octubre de 2023.

Fuente: Elaboración propia.

Según la revista (Público, 2024), a casi 5,000 metros de altitud, la comunidad de Phinaya, en los Andes peruanos, trabaja para proteger uno de los glaciares tropicales más importantes del mundo: el Quelccaya. El calentamiento global está acelerando el deshielo de este vasto glaciar, pero las comunidades quechua han unido esfuerzos para mitigar los impactos de su



Nota: Retroceso superficial del Glaciar Quelccaya vista a través de una imagen satelital. a) Imagen de 3 de septiembre de 1988, b) Imagen del 22 de octubre de 2023.

Fuente: Elaboración propia.

Figura 1.4:

Incidencia de anemia por grupo etario y según residencia.

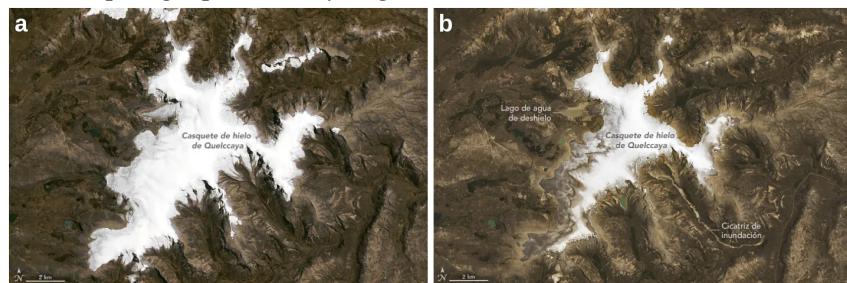


Figura 1.5: Retroceso superficial del Glaciar Quelccaya vista a través de una imagen satelital. a) Imagen de 3 de septiembre de 1988, b) Imagen del 22 de octubre de 2023.

Fuente: <https://ciencia.nasa.gov/ciencias-terrestres/el-casquete-de-hielo-de-quelccaya-antes-y-ahora/>.

retroceso. La revista indica que en 1982, el Quelccaya era el glaciar tropical más grande del mundo. Cuatro décadas después, su superficie ha disminuido un 46 %, dejándolo en segundo lugar tras el Nevado Coropuna. Esta pérdida ha dejado a las 200 familias de Phinaya en una zona amenazada por la desertificación, enfrentando serios problemas de acceso al agua.

Una entrevista por parte de la revista realizada a Yolanda Quispe, guardaparques y comunitaria de Phinaya, expresa con emoción que proteger el glaciar es un honor, dado que representa un pilar fundamental para la vida en la región. En 2019, las comunidades de Phinaya y la Asociación de Vivienda Salma Sallani se unieron al Área de Conservación Regional (ACR) Ausangate, protegiendo unas 66,500 hectáreas de ecosistemas frágiles en la cordillera de Vilcanota.

Además el ACR busca conservar las áreas montañosas, que son fuente de agua y hábitat para diversas especies. Sin embargo, el deshielo del Quelccaya no solo afecta a la biodiversidad, sino también a las familias que dependen del pastoreo de alpacas, la principal actividad económica de la comunidad. Con el cambio climático secando los pastizales, la crianza de estos

animales se ha vuelto insostenible, incrementando la migración hacia las ciudades cercanas (Público, 2024).

La migración es ahora impulsada por el deterioro de los recursos naturales, lo que ha provocado que muchos jóvenes abandonen la comunidad según estudios recientes de (Costa, 2022). Esta pérdida de población no solo amenaza la continuidad de las actividades tradicionales, sino que también abre la puerta a la explotación minera, lo que podría generar más conflictos y degradación ambiental. Frente a esta situación, el ACR Ausangate busca promover el ecoturismo como una estrategia para atraer a los jóvenes de regreso a la comunidad, protegiendo a la vez los ecosistemas y la cultura local (Público, 2024).

La función crucial de los glaciares como reservorios temporales de agua adquiere una relevancia especial en regiones tropicales, dada la presencia de dos estaciones bien definidas, una lluviosa y otra seca. A pesar de su papel fundamental, son escasas las investigaciones sobre el balance de masa en los glaciares de la Cordillera Vilcanota. (Porcel, 2015)

La conservación de las cuencas en la Cordillera del Vilcanota tiene gran importancia debido a que constituyen la fuente de suministro de agua para una extensa área en la región de Cusco. Además, son fundamentales para el desarrollo de la actividad turística, que atrae cada año a numerosos visitantes interesados en la práctica de deportes de aventura y en la apreciación de los hermosos paisajes que ofrece la zona.

Con el desarrollo de las tecnologías aeroespaciales, la detección de objetos a partir de imágenes satelitales se convirtió en un componente esencial para muchas aplicaciones desde 1970. Las aplicaciones mediante el sensoramiento remoto permiten el monitoreo ambiental y evaluación, agricultura de precisión, mapeo geográfico (Sánchez-Díaz, 2018). Así mismo, el cambio climático y el calentamiento global, conlleva a la importancia en el análisis de las conexiones entre las Series de Tiempo de Temperatura Superficial e Índices que son obtenidos mediante imágenes de satélite. (Zuluaga Gómez et al., 2021).

Muchos de los estudios realizados por la comunidad científica de glaciología utilizan métodos tradicionales para el evaluar cuerpos glaciares, los más populares son los métodos de

segmentación a partir del NDSI (Índice Diferencial Normalizado De Nieve) y el NDWI(Índice Diferencial Normalizado De Agua) (Drenkhan et al., 2018), (Durán-Alarcón et al., 2015), (Sood et al., 2020) y el método de umbralización OTSU (Garcia Bazan, 2022), (Gaddam et al., 2022). Los índices espectrales NDSI y NDWI ayudan a segmentar cuerpos glaciares y cuerpos de agua respectivamente, de esta manera es posible medir la superficie del cuerpo glaciar, en muchas investigaciones aún actuales para el análisis de cuerpos glaciares se suele eliminar los cuerpos de agua, puesto que la técnica de NDSI también suele confundir cuerpos de nieve con cuerpos de agua, y en algunos casos también con sombras, sin embargo, los cuerpos de agua tienden a tener menor intensidad de pixel que los cuerpos glaciares es por ello que en algunos casos suelen ser eliminados con facilidad aplicando un umbral, pero en otros casos la existencia de lagunas en estado glaciar no llegan a ser eliminados por la aproximación en la intensidad de píxeles e incluso se llegan a eliminar píxeles de cuerpos glaciares, otra problemática es que este método suele confundir sombras con cuerpos glaciares, esto impide tener mediciones correctas y para ello es necesario hacer una supervisión visual y eliminar las lagunas restantes de forma manual, dicho proceso tiende a ser laborioso y susceptible a errores.

Estas limitaciones han generado una creciente conciencia sobre la necesidad de desarrollar enfoques más sofisticados y precisos para el análisis de retroceso de superficie glaciar en áreas montañosas. Varias investigaciones emplean deep learning para abordar estas limitaciones y mejorar la precisión del análisis de retroceso glaciar en áreas montañosas.

1.3. Formulación del problema

1.3.1. Problema general

¿Cómo desarrollar un sistema basado en algoritmos inteligentes que clasifique y segmente cuerpos glaciares para realizar un análisis del retroceso de superficie glaciar?

1.3.2. Problemas específicos

- ¿Qué tipos de algoritmos inteligentes son las más adecuadas para la clasificación y segmentación de cuerpos glaciares?
- ¿Cómo impacta la falta de disponibilidad de datos confiables en el desarrollo de investigaciones de modelos de inteligencia artificial para la segmentación y clasificación de cuerpos glaciares?
- ¿Cómo llevar a cabo un análisis del retroceso de la superficie glaciar?
- ¿Cómo comparar resultados obtenidos de superficie glaciar a partir de modelos de inteligencia artificial?

1.4. Justificación

Los glaciares tropicales son catalogados como los principales indicadores del cambio climático debido a que son sensibles a las variaciones del clima. Los cambios en un glaciar se pueden medir en función de los cambios en su geometría como son el área, volumen o espesor el cual puede determinar el avance y retroceso que es muy importante en la evaluación de los recursos hídricos.

En el pasado reciente, las llamadas lenguas glaciares que se ubican en los alrededores del glaciar, son las más afectadas en comparación con los glaciares ubicados en zonas de pendiente, pues tienden a desaparecer más rápido, este derretimiento tiene como consecuencia la generación de nuevas lagunas y el incremento del área de estas, según (Jiménez Tejeda & Tecse López, 2021) este incremento se está dando de forma acelerada, en el periodo de análisis julio 2019 – agosto 2020, algunos de estos han duplicado su tamaño. como las que ocurrieron en la cordillera de Vilcanota.

Cabe mencionar que los glaciares andinos tienen una gran importancia económica ya que son fuentes de agua dulce que se encuentra en proceso de extinción, llegando a entregar

agua a las principales comunidades, poblaciones y regiones urbanas en épocas de escasez como la temporada seca, a su vez generan oportunidades para el aprovechamiento del potencial hidroeléctrico debido a sus favorables condiciones geográficas, sin embargo, al extinguirse los glaciares, traerá una disminución significativa, como ya se está observando en muchas zonas donde los glaciares ya se extinguieron o eso está sucediendo (Ramírez, 2008). El monitoreo del retroceso glaciar de la superficie de los glaciares ayudaría a evaluar y anticipar los impactos en la disponibilidad de agua dulce para comunidades y áreas urbanas, especialmente en épocas secas. Además, proporcionaría información crucial para planificar la gestión de recursos hídricos y evaluar la viabilidad de proyectos hidroeléctricos en función a la disminución de caudales.

El Quelccaya se extiende desde la cordillera oriental de los Andes y forma parte de la cordillera Vilcanota. Miguel Ángel Canal, subgerente regional de Recursos Naturales y Gestión del Medio Ambiente del Cusco, destaca la importancia de este macizo montañoso y de la laguna de Sibinacocha en el conjunto de nevados que conforman el Ausangate, el Apu tutelar del Cusco. Explica que este lugar “actúa como un indicador global donde se investiga la relación entre el calentamiento global y el deshielo de los glaciares”.

En el contexto actual del acelerado retroceso glaciar debido al cambio climático, es esencial implementar un monitoreo cuantitativo de los glaciares. Este seguimiento busca identificar amenazas futuras, especialmente relacionadas con la formación de lagunas causadas por el deshielo, que podrían representar un riesgo para las comunidades o poblaciones cercanas. Además, la necesidad de preservar el hábitat de especies anfibias en la laguna Sibinacocha y áreas adyacentes, donde nuevos estanques formados por la deglaciación han ampliado su especie habitable. El monitoreo proporcionaría datos esenciales para entender cómo el retroceso glaciar afecta la humedad del ecosistema, permitiendo implementar medidas para mantener las condiciones favorables para la vegetación y la supervivencia de estas especies ante los efectos del cambio climático.

Un proyecto de monitoreo del retroceso del Glaciar Quelccaya se justifica por su impacto directo en las comunidades cercanas, que dependen de estos recursos para actividades económicas, como la crianza de llamas y alpacas. La información del monitoreo permitiría anticipar

cambios en la disponibilidad de agua, ayudando a adaptar prácticas de crianza de animales y optimizar recursos hídricos, ayudaría a frenar la migración juvenil mediante nuevas estrategias de subsistencia para implementar proyectos de diversificación económica que ayuden a las comunidades a encontrar fuentes de sustento alternativo y reducir la necesidad de migración y serviría como herramienta para proteger el área contra la minería, promoviendo así un desarrollo sostenible en la región y mitigando riesgos ambientales y sociales.

Según a los antecedentes descritos anteriormente, es fundamental y necesario el monitoreo de la superficie glaciar, el cual tiene que ser cartografiado de manera correcta y precisa; sin embargo, los métodos tradicionales de segmentación y clasificación de cuerpos glaciares, a partir de los métodos tradicionales como el NDSI, llegan a ser métodos simples y en la mayoría de los casos requiere intervención manual para determinar el umbral (intensidad de pixel) más adecuado para segmentar cuerpos glaciares; sin embargo, no siempre funciona correctamente, puesto que casi siempre es necesario corregir y/o eliminar errores de segmentación y clasificación de otros cuerpos. La aplicación de métodos automáticos de segmentación basados en algoritmos de inteligencia artificial evitaría la intervención manual del hombre, con esto se lograría disminuir errores al momento de extraer el área glaciar.

La metodología aplicada en esta investigación puede ser replicada para analizar diferentes zonas glaciares, de igual manera los resultados y las conclusiones obtenidas pueden ser utilizadas como base para futuras investigaciones.

1.5. Objetivos

1.5.1. Objetivo general

Aplicar técnicas de deep learning para la clasificación y segmentación de cuerpos glaciares, para realizar un análisis temporal del retroceso de superficie del glaciar Quelccaya, ubicado en la Cordillera Vilcanota, Perú.

1.5.2. Objetivos específicos

- Utilizar algoritmos de segmentación semántica basados en deep learning para clasificar y segmentar cuerpos glaciares a partir de imágenes de satélite.
- Desarrollar una base de datos confiable y accesible de imágenes satelitales etiquetadas con cuerpos glaciares para entrenar y evaluar el modelo para tareas de clasificación y segmentación.
- Analizar los cambios en la superficie glaciar en intervalos de tiempo, utilizando las imágenes clasificadas y segmentadas por el modelo de deep learning, para cuantificar la superficie de retroceso glaciar.
- Comparar los resultados de superficie glaciar obtenidos con datos registrados por artículos científicos o instituciones gubernamentales del Perú.

1.6. Hipótesis

1.6.1. Hipótesis general

El uso de algoritmos inteligentes basados en deep learning para la segmentación de glaciares permitirá realizar un análisis temporal del retroceso glaciar y esto revelará la magnitud de pérdida de superficie glaciar.

1.6.2. Hipótesis específicos

- Se espera que los algoritmos de deep learning puedan alcanzar algún nivel de precisión en la clasificación y segmentación de cuerpos glaciares, aunque no se puede asegurar si esta precisión será alta.

- Con el desarrollo de una base de datos, se espera que el modelo implementado tenga precisión y efectividad para tareas de clasificación y segmentación de cuerpos glaciares, aunque la magnitud de esta precisión y efectividad es incierta.
- El modelo podría ser capaz de identificar y clasificar cuerpos glaciares a lo largo de diferentes estaciones del año, aunque es incierto si podrá hacerlo correctamente, dadas las posibles variaciones temporales en su apariencia
- Es incierto si los resultados de la investigación serán comparables y correlacionados con los datos registrados por artículos científicos o instituciones gubernamentales del Perú.

1.7. Variables e indicadores

1.7.1. Identificación de variables

Variable independiente

- **Algoritmos de deep learning:** Arquitectura del modelo (número de capas, tipo de activaciones, etc.)
- **Datos de entrenamiento:** Calidad de las imágenes satelitales, cantidad de datos disponibles y fuentes de datos (satélites específicos, temporadas del año)
- **Parámetros de entrenamiento:** Hiperparámetros del modelo.

Variable dependiente

- **Precisión de la clasificación y segmentación:** Eficacia del modelo para clasificar y segmentar correctamente los cuerpos glaciares.
- **Cobertura glaciar:** La extensión y la distribución espacial de la cobertura glaciar en cada imagen de satélite. Esta variable dependiente refleja el estado actual de los glaciares

en el momento de la captura de la imagen.

- **Retroceso de la superficie glaciar:** Área de superficie glaciar cuantificada a lo largo del tiempo.

1.8. Alcance de la investigación

Los alcances de este estudio pueden resumirse en:

- Desarrollar, entrenar y evaluar modelos de deep learning específicos para la clasificación y segmentación de cuerpos glaciares utilizando imágenes satelitales multiespectrales.
- Recopilar y etiquetar una base de datos de al menos mil imágenes obtenidas de imágenes satelitales que contengan cuerpos glaciares, asegurando su accesibilidad para el entrenamiento, validación y testeo de los modelos de deep learning.
- Realizar un análisis temporal del retroceso de la superficie glaciar utilizando los modelos de deep learning entrenados para identificar cambios en la superficie glaciar a lo largo del tiempo.
- Comparar los resultados obtenidos de la estimación de superficie glaciar con datos de referencia provenientes de artículos científicos, evaluando la precisión y fiabilidad del modelo.

1.9. Viabilidad y factibilidad

- **Disponibilidad de datos:** Se ofrecen una gran cantidad de datos de imágenes satelitales de forma gratuita a través de plataformas de libre acceso, como el Servicio Geológico de los Estados Unidos (USGS). Esto hace que sea relativamente fácil acceder a las imágenes necesarias para esta investigación.

- **Metodología establecida:** La segmentación semántica utilizando deep learning se ha utilizado con éxito en una variedad de campos, incluida la detección de cambios en la cobertura terrestre. Hay varias arquitecturas de redes neuronales pre-entrenadas disponibles que se puede adaptar para el análisis de imágenes multiespectrales de glaciares.
- **Avances en tecnología informática:** Con el aumento de la capacidad de procesamiento y el acceso a herramientas de deep learning, realizar análisis de imágenes satelitales a gran escala se ha vuelto más factible. Los marcos de trabajo de deep learning como TensorFlow o PyTorch proporcionan una infraestructura robusta para desarrollar y entrenar modelos de segmentación semántica.
- **Relevancia e importancia del tema:** El retroceso de los glaciares es un problema ambiental importante y de interés mundial debido a su relación con el cambio climático. Esta investigación tiene el potencial de contribuir al entendimiento de este fenómeno y sus implicaciones.
- **Costo relativamente bajo:** En comparación con estudios de campo o la adquisición de datos de imágenes satelitales de alta resolución, el costo de esta investigación puede ser relativamente bajo, especialmente si se aprovechan los recursos gratuitos disponibles.

1.10. Limitaciones de la investigación

Algunas limitaciones de este estudio pueden resumirse en:

- La capacidad del modelo para generalizar correctamente a nuevos datos y diferentes regiones geográficas puede ser limitada, afectando su aplicabilidad en diferentes contextos.
- La falta de datos satelitales históricos y etiquetados de alta calidad puede afectar la precisión, el entrenamiento del modelo de deep learning y los resultados obtenidos.
- La variabilidad estacional y los cambios climáticos pueden dificultar la clasificación y segmentación precisa de los cuerpos glaciares, ya que su apariencia puede variar significativamente a lo largo del año.

- Las discrepancias entre los resultados del modelo y los datos de referencia de artículos científicos o instituciones gubernamentales pueden surgir debido a diferencias en los métodos de recolección de datos y análisis.

1.11. Delimitación del estudio

La investigación se centrará en el uso de métodos de deep learning para clasificar y segmentar cuerpos glaciares, específicamente en el análisis temporal del retroceso de la superficie del glaciar Quelccaya, parte de la Cordillera Vilcanota, durante un período de al menos 10 años. Para el análisis remoto, se utilizarán imágenes satelitales con el objetivo de identificar y segmentar áreas de cobertura glaciar a lo largo del tiempo. Se emplearán técnicas de segmentación semántica para mejorar la precisión en la identificación de los cuerpos glaciares en las imágenes satelitales. La investigación se llevará a cabo sin visitas de campo ni mediciones directas en el terreno, centrándose exclusivamente en el análisis de datos de imágenes satelitales para comprender el retroceso de la superficie del glaciar Quelccaya.

Capítulo 2

Marco teórico

2.1. Antecedentes de estudio

La asociación Peruana de Ingeniería Hidráulica y Ambiental (APIHA), señaló que en los últimos 60 años ha desaparecido más del 50 % del área glaciar del Perú, advierten la existencia de un riesgo enorme de extinción en las siguientes décadas, esto debido al calentamiento global (Gerens,). Es por ello que especialistas en todo el mundo muestran su preocupación respecto al fenómeno del calentamiento global y sus futuros impactos, entre ellos destaca el retroceso de los glaciares. Para abordar esta problemática, se lleva a cabo numerosos estudios a nivel mundial, enfocándose en el análisis del retroceso glaciar en diversas cordilleras del planeta.

Entre las regiones objetivas de atención, las cordilleras del Perú emergen como un área de especial interés. En este contexto, se han llevado a cabo investigaciones específicas en las distintas cordilleras glaciares de nuestro país. Un ejemplo destacado es el estudio realizado por (Navarro et al., 2023), titulado **1975–2018: 43 Years of Glacial Retreat in the Incachiriasca Glacier (Nevado Salcantay, Vilcabamba Range, Peru)**. Esta investigación tiene como objetivo el estudio del retroceso del glaciar Incachiriasca, combinando resultados del monitoreo de 2007-2018, con el análisis de 28 imágenes de satélite desde el año 1975-2007. Donde describieron un análisis del retroceso del área glaciar Incachiriasca-II utilizando imágenes del satélite Landsat 2,4,5 y 7 para el análisis de retroceso glaciar, donde se demostró una pérdida del 51.4 % del área total del glaciar, de 0.53 km^2 (1975) a 0.26 km^2 (2018) el cual es equivalente a $0.0063 \text{ km}^2/\text{año}$

(1.2 % anual). Según la tendencia del análisis, se observó que la tasa anual del retroceso del glaciar incrementó significativamente, a partir de 2010, aumentando de 1 % (2001-2010) a 3 % (2010-2018).

Debido a que los glaciares llegan a ser ecosistemas altamente sensibles al cambio climático, en las últimas décadas los efectos son más evidentes sobre estas reservas de agua sólida, generando una significativa y notoria disminución de la masa glaciar trayendo como efecto la formación de nuevas lagunas, esto es evidenciado en un estudio relizado por (Drenkhan et al., 2018), titulado **Current and future glacier and lake assessment in the deglaciating Vilcanota-Urubamba basin, Peruvian Andes**. Este estudio tuvo como objetivo combinar un análisis del desarrollo actual (1988-2016) y futuro (2050-2100) de glaciares y lagos en la cuenca Vilcanota-Urubamba (Cusco, sur de Perú). Los resultados que se obtuvieron indican que el área total de los glaciares disminuyó un 37.35 % de 226.1 km² en 1988 a 141.7 km² en 2016, los lagos adyacentes aumentaron en área un 15.5 % de 23.3 km² (460 lagos) en 1968 a 26.9 km² (544 lagos) en 2016. El estudio supone que las futuras áreas de glaciares podrían disminuir entre un 40.7 % y un 44.9 % en las próximas décadas (2031-2060) y entre un 41.4 % y un 92.7 %, respectivamente, en este siglo (2071-2100). Por lo tanto, los paisajes andinos se transformarán en áreas en su mayoría libres de glaciares, con algunas cumbres cubiertas de hielo por encima de 6000 msnm.

En estudios recientes de (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023), ha destacado el valor de los productos satelitales como una fuente crucial de información para el monitoreo y estudio de los glaciares y lagunas. Uno de los proyectos más significativos desarrollados por el INAIGEM fue el **Inventario Nacional de Glaciares y Lagunas de Origen Glaciar (INGLOG 2)**. Este estudio tuvo como objetivo ofrecer datos clave sobre el estado de los glaciares y las lagunas de origen glaciar en Perú, con el fin de apoyar la toma de decisiones en su gestión. En este inventario, la obtención de información se basó en el uso de imágenes satelitales Sentinel-2, recolectadas durante los meses de menor precipitación y mínima presencia de nubes, optimizando así la calidad y precisión de los datos. Para el procesamiento de estas imágenes, se emplearon herramientas avanzadas de computación en la

nube, utilizando plataformas como Google Earth Engine (GEE) y lenguajes de programación como Python y JavaScript, que facilitaron el análisis masivo de datos y la automatización de procesos. En el proceso de segmentación de cuerpos glaciares, se aplicó el Índice Diferencial de Nieve (NDSI), un indicador que mide la diferencia de reflectancia entre el espectro visible verde y el infrarrojo de onda corta. Este índice resultó fundamental para identificar la presencia de nieve y hielo en las imágenes satelitales. Con la ayuda de algoritmos automatizados, se calculó el valor mínimo del NDSI para la colección de imágenes Sentinel-2. Se estableció un umbral mínimo de $NDSI \geq 0.1$, que fue utilizado como indicador de la presencia de cobertura glaciar. Sin embargo, con el objetivo de mejorar la precisión y reducir los márgenes de error en la identificación de glaciares, se implementó una revisión manual adicional. Esta revisión permitió identificar y corregir aquellos índices que, erróneamente, clasificaban sombras, nubes y humedales como parte de la cobertura glaciar. Además, se ajustaron los perímetros glaciales y se llenaron vacíos internos que no habían sido detectados por el NDSI, mejorando así la confiabilidad de los resultados. Este enfoque híbrido, que combinó algoritmos automatizados con correcciones manuales, permitió obtener una cartografía glaciar más precisa y confiable, sirviendo como base para futuros estudios y estrategias de conservación de los ecosistemas montañosos en Perú.

Recientemente los algoritmos basados en Machine Learning (ML) están tomando popularidad en aplicaciones de geocincencia. Un estudio regional en el uso de Machine Learning para la clasificación de la cobertura glaciar es la tesis de (Garcia Bazan, 2022), titulado **Estimación Del Área Superficial De Un Glaciar Utilizando Técnicas De Procesamiento Digital De Imágenes Satelitales Obtenidas Por El Perusat-1**. El objetivo de este estudio fue calcular la extensión superficial de un glaciar empleando métodos de procesamiento digital de imágenes satelitales captadas por el PeruSat-1 que se enfoca en la estimación del área superficial de glaciares, con un caso de estudio específico en el glaciar Incachiriasca, utilizando procesamiento digital de imágenes satelitales obtenidas por el satélite peruano PeruSAT-1. La metodología aplicada en este trabajo comenzó con un análisis tradicional de monitoreo de glaciares mediante técnicas de procesamiento digital de imágenes, para luego incorporar métodos de Machine Learning, tanto supervisados como no supervisados, que se aplicaron para mejorar la precisión

de la clasificación glaciar. Aunque estas técnicas son efectivas, el estudio resaltó que los métodos de Machine Learning son computacionalmente intensivos, lo que dificulta su aplicación a imágenes satelitales de gran tamaño y en escalas más amplias. Para validar los resultados obtenidos, se contrastaron con información proporcionada por el INAIGEM y el Servicio Nacional de Áreas Naturales Protegidas por el Estado (SERNANP). Finalmente, la tesis concluyó con una estimación del área superficial del glaciar Incachiriasca de 16,241.93 m² y una tasa de retroceso glaciar de 4.51 km²/año. Sin embargo, el trabajo también identificó que estos métodos de extracción de cuerpos glaciares enfrentan dificultades para detectar de manera precisa los límites de las superficies glaciares, especialmente en escenarios geográficos complejos. Esto limita su capacidad de ser aplicados de forma automática en cualquier entorno.

Un caso relevante en el uso de técnicas avanzadas de deep learning para la cartografía y monitoreo de masas de agua continentales es el estudio realizado por (Ghaznavi et al., 2024), titulado **Comparative performance analysis of simple U-Net, residual attention U-Net, and VGG16-U-Net for inventory inland water bodies**. El objetivo de esta investigación fue desarrollar un método basado en deep learning para inventariar y cartografiar dichas masas utilizando las bandas RGB-NIR (rojo, verde, azul, e infrarrojo cercano) de imágenes satelitales Sentinel-2. En este estudio, se emplearon tres algoritmos de deep learning diferentes, todos basados en la arquitectura U-Net, con el fin de segmentar las masas de agua continentales. Estos algoritmos incluyeron U-Net, Residual Attention U-Net y VGG16-U-Net. Los resultados del estudio indicaron que los algoritmos basados en U-Net son efectivos para inventariar cuerpos de agua de manera rápida, precisa y económica en términos de costo computacional. Esta capacidad de segmentación precisa y eficiente hace que estos métodos sean ideales para aplicaciones a gran escala, permitiendo un análisis eficiente de grandes volúmenes de datos satelitales. Además, los hallazgos de este estudio sientan las bases para la implementación de un mapeo preciso de la cobertura terrestre, utilizando imágenes satelitales. Al proporcionar un algoritmo rápido, objetivo y confiable, este trabajo abre la posibilidad de llevar a cabo inventarios de cobertura terrestre a nivel global con un alto grado de precisión y con menores costos en términos de procesamiento de datos.

Otro estudio realizado por (Rajat et al., 2022), titulado **Glacier retreat in Himachal from 1994 to 2021 using deep learning**. Cuyo objetivo fue, Aplicar la arquitectura U-Net para la identificación de glaciares, utilizando teledetección proveniente de satélites indios (IRS) y datos satelitales de Landsat, incorporando mejoras en los procesos de segmentación e identificación de glaciares. El cual se hizo en los glaciares de la provincia india de Himachal Pradesh, situada en Himalaya indio occidental, donde se realizó un mapeo glaciar de las últimas tres décadas, es decir, de 1994 a 2021, se evaluó el impacto del cambio climático utilizando técnicas de segmentación de deep learning en base a imágenes satelitales Landsat 4, 5 y 8, llegando a automatizar el proceso de mapeo de glaciar, el algoritmo utilizado fue U-NET el cual llegó a tener una precisión de identificación del 95 %, en dicho estudio se observó que el área glaciar de Himachal Pradesh disminuyó a un ritmo de 67,84 km² por año, de 4020,6 km² en 1994 a 2198,5 km² en 2021. Se observa que de 1994 a 2021, el área del glaciar disminuyó aproximadamente en un porcentaje de pérdida de 1,678 por año, con una tendencia decreciente del 2,31 % en 1994 a 2001 y 1.398 % en 2011 a 2021.

El estudio realizado por (Malone et al., 2022), titulado **The Evolution of the Two Largest Tropical Ice Masses since the 1980s**. El objetivo de esta investigación fue reevaluar el tamaño y las tasas de retroceso de las dos masas de hielo tropicales más grandes: el manto glaciar Quelccaya y el Nevado Coropuna, utilizando datos de la misión Landsat. El análisis se llevó a cabo durante la estación seca, empleando el Índice Diferencial de Nieve Normalizado (NDSI) para segmentar los cuerpos glaciares. Se realizaron correcciones en las imágenes donde el NDSI confundía sombras, nubes y cuerpos de agua con áreas glaciares. Además, se aplicó un umbral de 0.5, que permitió separar de manera más consistente los valores altos, asociados a la nieve, de los valores más bajos atribuidos a otros materiales de la superficie. Para mejorar la precisión, se llenaron pequeños vacíos en los polígonos (menores de 6000 m²) que habían sido afectados por el umbral, y se eliminaron polígonos pequeños (menores de 10,000 m²), asumiendo que estos representaban coberturas de nieve efímera en lugar de áreas glaciares permanentes. El período de análisis abarcó desde el año 1988 hasta 2020, concluyendo que el manto glaciar de Quelccaya experimentó un notable retroceso desde mediados de la década de 1980. Durante este periodo, el glaciar se redujo en aproximadamente un 26 %, pasando de 52.60 ± 2.58 km² en 1988

a 39.00 ± 1.92 km² en 2020. La mayor parte de la pérdida de hielo se produjo en elevaciones más bajas, lo que resultó en un aumento de la elevación media del glaciar de 5389 m.s.n.m. a 5449 m.s.n.m.. Esto indica que el manto glaciar de Quelccaya en 2020 es una versión más pequeña, elevada y restringida en comparación con su extensión en 1988. Durante el período de observación (1988-2020), la pérdida total de hielo en el glaciar Quelccaya fue de 13.60 ± 3.22 km², lo que se traduce en una tasa de pérdida media de 0.43 ± 0.10 km² por año. A pesar de esta reducción significativa, el estudio destacó que la tasa de retroceso ha sido relativamente constante a lo largo del tiempo. Finalmente, los hallazgos de este estudio fueron comparados con investigaciones previas sobre glaciares tropicales, concluyendo que ambas masas de hielo tropicales, aunque más pequeñas, se están retirando a un ritmo menor de lo que comúnmente se informa en la literatura científica.

2.2. Bases teóricas

2.2.1. Glaciares

Los glaciares son masas de hielo formadas por nieve compactada y recristalizada en tierra firme, compuestas por cristales de hielo, agua, aire, roca y nieve. Se originan cuando, a lo largo de los años, la acumulación de nieve supera su derretimiento, un proceso común en zonas de alta montaña. A medida que la nieve se comprime, se convierte en hielo, lo que forma el glaciar. Estos cuerpos de hielo se mueven lentamente cuesta abajo de forma continua durante largos períodos.

Los glaciares son dinámicos, ya que sus componentes y funcionamiento varían con el tiempo. Se distinguen tres áreas principales: la Zona de Acumulación, donde el glaciar gana masa; la Zona de Ablación, donde pierde masa a través del derretimiento, generando escorrentías que alimentan los ríos; y la Línea de Equilibrio Altitudinal, que marca el punto donde no hay ganancia ni pérdida de masa. La ubicación de estas zonas cambia en respuesta a las variaciones climáticas (Menzies, 1995), (Ceballos et al., 2012), (Instituto Nacional de Investigación en

Glaciares y Ecosistemas de Montaña, 2023).



Figura 2.1: Componentes circundantes de un glaciar.

Fuente: (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023)

Glaciares Tropicales

Los glaciares tropicales son aquellos ubicados en latitudes cercanas al ecuador, entre los trópicos de Cáncer y Capricornio. Aproximadamente el 99 % de estos glaciares a nivel mundial se concentra en la cordillera de los Andes, abarcando territorios de países como Venezuela, Colombia, Ecuador, Perú, Bolivia, Chile y Argentina.

Perú alberga el 68 % de los glaciares tropicales del mundo (Veettil & Kamp, 2017), distribuidos en 20 cordilleras a lo largo de las regiones norte, centro y sur del país. Estos glaciares son de gran relevancia debido a su sensibilidad al cambio climático, lo que los convierte en excelentes indicadores de las variaciones climáticas (Francou et al., 1999).

Clasificación de los glaciares tropicales

Cada glaciar posee características particulares, según INAIGEM clasificó cada glaciar por tipo de superficie: Glaciar libre de detritos, glaciar cubierto por detritos y glaciar rocoso.

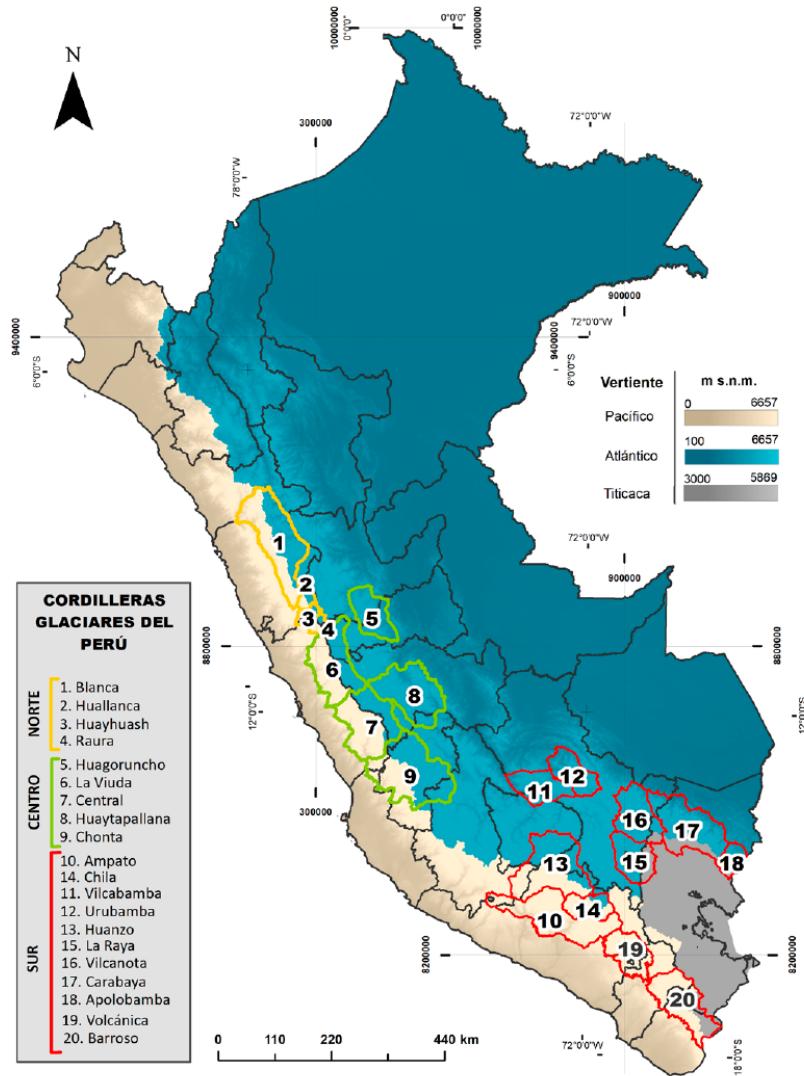


Figura 2.2: Localización de las cordilleras glaciares en el Perú.

Fuente: (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023)

- a) **Glaciar libre de detritos:** Son glaciares que no presentan un alto grado de impurezas o material particulado en su superficie; sin embargo, en algunos casos pueden contener un pequeño porcentaje de material detrítico, cenizas o impurezas (Lliboutry, 1956). Este material se acumula en la superficie del glaciar debido al transporte de partículas por los vientos, las cuales provienen del desprendimiento de laderas cercanas.
- b) **Glaciar cubierto por detritos:** Su característica principal es que están parcial o totalmente cubiertos por una capa de detritos supraglaciares. A diferencia de los glaciares libres de detritos, el material detrítico proviene principalmente de la fragmentación de rocas causada por procesos de meteorización (Anderson & Anderson, 2016).

c) **Glaciar rocoso:** Está formada principalmente por detritos, rocas y hielo que se desplazan cuesta abajo debido a la gravedad. Su característica más notable es la aparición de surcos y lóbulos en la superficie. En los glaciares rocosos, el hielo no suele aflorar en la superficie, ya que está presente en los espacios internos entre los escombros o confinado en un núcleo de hielo en el interior (Potter Jr, 1972).

2.2.2. Retroceso glaciar

Se define como retroceso glaciar al ascenso de la línea inferior de los glaciares de alta montaña cada vez hacia áreas de altitud más alta o latitudes más frías generando una disminución en su extensión, espesor y volumen de la masa glaciar a lo largo del tiempo. El retroceso glaciar ocurre por el desequilibrio entre la acumulación de hielo y nieve en la parte superior del glaciar (zona de acumulación) y la pérdida de hielo en la parte inferior (zona de ablación) (DEL PERU,).

2.2.3. Principales causas del retroceso glaciar

Cambio climático

El Panel Intergubernamental sobre Cambio Climático (IPCC, por sus siglas en inglés) define el cambio climático como una modificación en el estado del clima que se manifiesta en un cambio del valor promedio y/o en la variabilidad de sus propiedades, y que perdura durante un período prolongado (IPCC, 2019).

Asimismo, el IPCC advierte sobre un rápido y significativo cambio en los glaciares de todas las regiones, incluidos los Andes tropicales. En general, los estudios indican una tendencia de reducción en la cobertura glaciar en las últimas décadas (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023).

El grado de retroceso varía según las características de cada glaciar, siendo los más pe-

queños y a menor altitud los más vulnerables. Sin embargo, comprender la relación entre el retroceso glaciar y el cambio climático es complejo, ya que la dinámica glaciar está influenciada tanto por factores locales, como la pendiente y orientación del glaciar, como por características climáticas regionales. Además, se prevé que la cantidad de lagunas glaciares continuará aumentando. También se estima que, para finales del 2100, el caudal de las cuencas podría reducirse en un 10 % o más durante las temporadas secas en zonas montañosas (IPCC, 2019).

Contaminación por partículas absorbentes de luz

Además del cambio climático, los glaciares están siendo influenciados por las llamadas "partículas absorbentes de luz", como el carbono negro, carbono orgánico y polvo mineral. Estas partículas, transportadas por el viento desde sus fuentes, terminan depositándose sobre la superficie glaciar. Al estar en la superficie, reducen el albedo de la nieve, lo que provoca una mayor absorción de energía solar, incrementando el calor y acelerando el derretimiento de los glaciares (Bond et al., 2013), (Gilardoni et al., 2022).

El principal componente de las partículas absorbentes de luz es el carbono negro (figura 12), que tiene un impacto significativo en los glaciares, ya que incluso en pequeñas cantidades puede alterar el balance energético en la superficie de la nieve (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023). Este tipo de partículas se generan por la combustión incompleta de combustibles fósiles y biomasa, como ocurre con las emisiones de vehículos, industrias, incendios forestales y la quema de residuos (Bond et al., 2013).

En la cordillera Blanca, se ha observado que las altas concentraciones de partículas absorbentes de luz están relacionadas con la cercanía a áreas urbanas. Además, estudios en las cordilleras Blanca, Huaytapallana y Vilcanota han demostrado que la mayor deposición de estas partículas sobre los glaciares ocurre entre el invierno y la primavera, coincidiendo con la temporada seca (Torres et al., 2018), (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023).



Figura 2.3: Carbono negro en la superficie del glaciar Yanapaccha, cordillera Blanca.
Fuente: (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023)

2.2.4. Teledetección

Los fenómenos que ocurren sobre la superficie de la tierra, tienen un gran interés por parte de la comunidad científica, esto ha llevado a realizarse muchísimos estudios en las últimas décadas, se tiene un interés sobre lo que sucede en la atmósfera, océanos y la superficie de la tierra. Para poder evaluar, analizar y predecir estos fenómenos se tienen que monitorizar ciertos parámetro que llegan a relacionarse de manera directa o indirecta con procesos geofísicos y biofísicos. Es en este punto donde la técnica de teledetección toma un papel importante, puesto que nos permite medir dichos parámetros de forma remota sin tener la necesidad de entrar en contacto directo con la superficie terrestre. Según la Agencia Espacial Europea (Sobrino, 2001) define la teledetección como la técnica de recopilar, evaluar y analizar datos de un objeto para obtener información de este sin necesidad de estar en contacto directo con el objeto de estudio. A su vez se tienen que distinguir tres elementos esenciales para la teledetección, que llegan a ser:

1. Una plataforma que transporta los instrumentos.
2. Fuente de energía.
3. Medio de propagación.

4. Un objeto de observación.
5. Un instrumento o sensor que observe el objeto.
6. Sistema receptor.
7. Usuario.

La técnica de teledetección utiliza las propiedades que tienen los objetos para radiar energía, de esta manera obtener información del cuerpo observado.

2.2.5. Radiación electromagnética

La radiación electromagnética se puede caracterizar a través de su frecuencia y por su longitud de onda como: ondas de radio, microondas, infrarrojo, luz visible, ultravioleta, rayos X, rayos gamma, etc. Las longitudes de onda son continuas, pero pueden agruparse en bandas sin límites definidos, a esto se le conoce como espectro electromagnético. Tal como podemos observar en la Figura 2.4, comprendido desde ondas pequeñas como los rayos gamma hasta ondas de mayor longitud como las ondas de radio (Halliday et al., 2015).

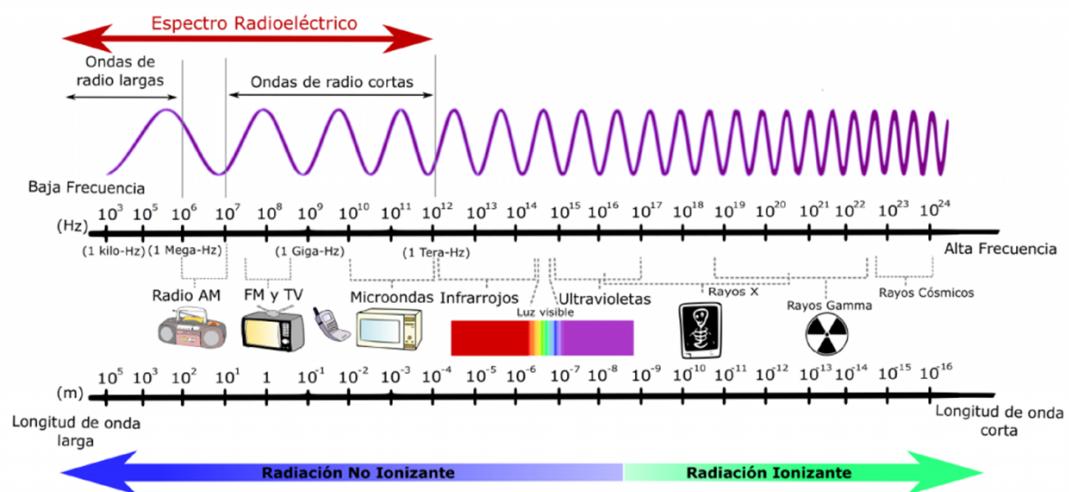


Figura 2.4: Espectro electromagnético basada en el rango de longitud de onda y frecuencia.
Fuente: (Rojas Monsalvo, 2013)

Normalmente en teledetección utiliza una serie de bandas que llegan a ser más útiles que otras, como:

Espectro visible (VIS): Componen de longitudes de onda que varían en el rango de (λ de 0,4 a 0,7 μm): compone tres bandas fundamentales, rojo (λ de 0,6 a 0,7 μm), verde (λ de 0,5 a 0,6 μm) y azul (λ de 0,4 a 0,5 μm), además esta porción del espectro es el rango completo de la energía electromagnética, al cual es sensible al ojo humano (Halliday et al., 2015).

Infrarrojo cercano (NIR): Componen de longitudes de onda que varían en el rango de (λ de 0,7 a 1,3 μm) esta porción del espectro permite discriminar masas vegetales así como las concentraciones de humedad (Halliday et al., 2015).

Infrarrojo de onda corta 1 (SWIR-1): Componen de longitudes de onda que varían en el rango de (λ de 1,57 a 1,65 μm), se emplea para discriminar el contenido de humedad en la vegetación o en los suelos (Halliday et al., 2015).

Infrarrojo de onda corta 2 (SWIR-2): Componen de longitudes de onda que varían en el rango de (λ de 2,08 a 2,35 μm), se emplea para discriminar propiedades del agua, minerales, etc (Halliday et al., 2015).

Infrarrojo termico (TIRS): Componen de longitudes de onda que varían en el rango de (λ de 8 a 14 μm), decisivo para localización de lugares con alta temperatura se emplea para localizar el calor proveniente de gran parte de las cubiertas terrestres (Halliday et al., 2015).

2.2.6. Interacción de la radiación con la atmósfera

Antes de que la radiación electromagnética llegue a la superficie terrestre, ocurren tres interacciones en la atmósfera, absorción, transmisión y dispersión.

Absorción atmosférica

La energía electromagnética que viaja a través del espacio en dirección a la atmósfera es parcialmente absorbida por varias moléculas, destacando en la absorción: Ozono (O₃), Vapor

de agua (H_2O), oxígeno molecular O_2 , dióxido de carbono (CO_2) y aerosoles atmosféricos (Geotig, Fecha de acceso), cada uno de ellos absorbe energía en distintos rangos del espectro, como se puede observar en la Figura 2.5. Además se observa que existe una gran cantidad de longitudes de onda que no son útiles, puesto que no pueden penetrar la atmósfera.

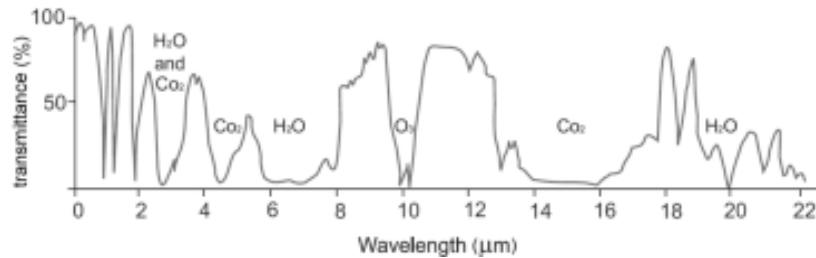


Figura 2.5: Transmisión Atmosférica.

Fuente: Camps-Valls et al., 2011

En la Figura 2.6 el color gris indica bandas de absorción y las áreas azules indican ventanas atmosféricas, es lo que los sensores de los satélites son capaces de ver en la superficie terrestre. Las ventanas de 0.4 a 2 μm la radiación en este rango es (Visible, NIR, SWIR) es principalmente energía reflejada. Los sensores que operan en este rango a menudo son llamados ópticos.(Tempfli et al., 2009)

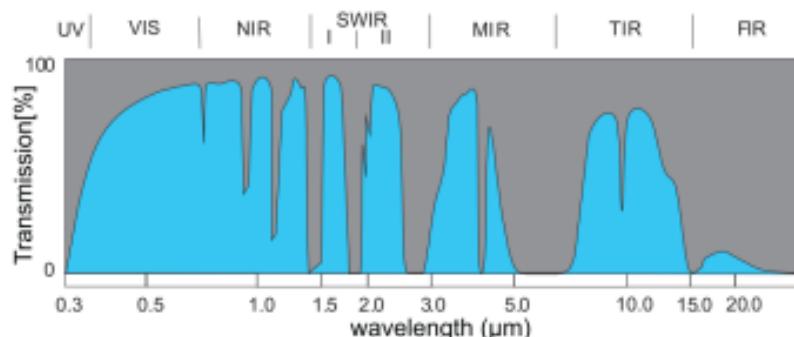


Figura 2.6: Ventanas Atmosféricas en Imágenes Ópticas Multiespectrales.

Fuente: (Camps-Valls et al., 2011)

Dispersión

La dispersión sucede cuando la radiación llega a interactuar con partículas en suspensión, tales como aerosoles y moléculas de aire. Estas partículas llegan a desviar la trayectoria de la radiación en diferentes direcciones, provocando que la luz sea dispersado en todo el espectro electromagnético (Geotig, Fecha de acceso).

2.2.7. Resolución de imágenes

Resolución espacial

Es el objeto más pequeño que se puede llegar a identificar en una imagen. Generalmente, se describe como el campo de visión instantáneo se define como el Ángulo de visión máxima en el que un sensor puede detectar eficazmente la energía electromagnética (Camps-Valls et al., 2011).

Resolución espectral

Es el número de bandas que tiene el sensor, la longitud onda de las bandas, las posiciones en que se ubican las bandas en el espectro electromagnético. Una alta resolución espectral proporciona una firma espectral más precisa (Camps-Valls et al., 2011).

Resolución radiométrica

Es la capacidad para detectar variaciones en la radiancia espectral que recibe. Está determinado por el número de niveles discretos en los que se puede dividir la radiación de la señal (Camps-Valls et al., 2011). Cuanto sea mayor la resolución radiométrica podrá interpretarse mejor la imagen (Chuvieco, 1996).

Resolución temporal

Es la frecuencia de cobertura que proporciona el sensor. Se refiere a la frecuencia con la que el sensor vuelve a visitar un área y toma imágenes periódicamente durante su vida útil (Meneses & Almeida, 2012).

2.2.8. Programa Satélite Landsat 8

El Programa Landsat, iniciado en 1972 por la NASA y el USGS, revolucionó el estudio de la Tierra con la serie más extensa de datos satelitales sobre cambios en la superficie terrestre. Actualmente, su octava misión, Landsat 8, incluye el satélite LDCM, equipado con los sensores OLI (Operational Land Imager) y TIRS (Thermal Infrared Sensor). Este satélite sigue trayectorias previas, asegurando la continuidad de los datos. Proporciona imágenes con resolución de hasta 15 metros y cobertura de 170 x 183 km, útiles para estudios de temperatura, áreas costeras y detección de nubes (Survey, 2013).

Plataforma del Satélite l8

El Landsat 8 cuenta con múltiples subsistemas (mecánico, control térmico, propulsión, RF, entre otros). Posee un panel solar de 9.75 x 2.6 metros que genera 3750 W y carga baterías de Ni-H2 con 125 Ah de capacidad. También dispone de un grabador de estado sólido de 3.14 Tb y una antena de banda X para transmitir datos captados por los sensores OLI y TIRS (Survey, 2013).

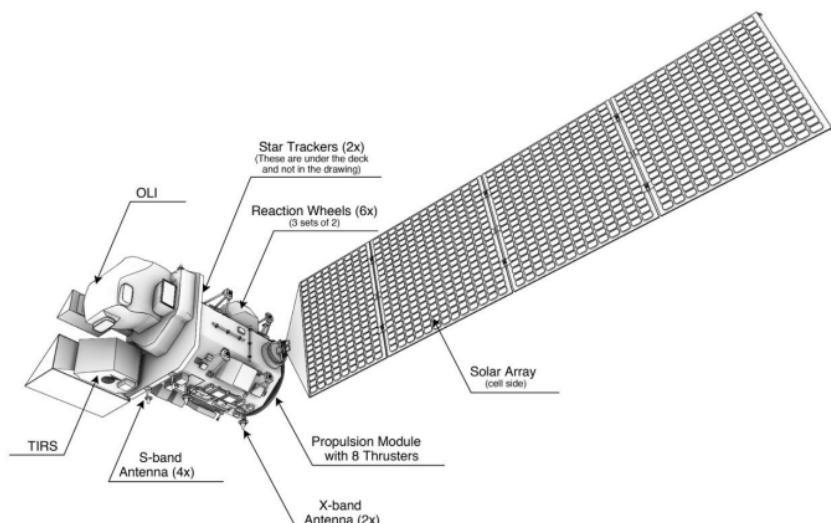


Figura 2.7: Subsistemas de la plataforma L8.

Fuente: (Survey, 2013)

A continuación se presenta una descripción técnica de los productos del L8:

Masa: 2,782 kg (6,113 lbs)

Órbita: Circular a 705 km con una inclinación de 98.2 grados

Estabilidad del objetivo: 6.02 microradianes

Almacenamiento de datos: Grabadora de estado sólido de 3.14 terabits

Transmisión de datos: Banda X, 384 Mbps (en más de dos canales)

Propulsión: 395 kg (870 lbs) de monopropelante de hidracina con 8 propulsores de 22

Newton

Vida útil: 5 años

2.2.9. Programa Satélite Landsat 5

Lanzado en marzo de 1982, el Landsat 5 tuvo como objetivo monitorear recursos terrestres mediante imágenes multiespectrales de alta resolución. Operó hasta junio de 2013, superando su vida útil inicial de 2 años. Utilizaba el sensor Thematic Mapper (TM), que capturaba imágenes con resolución de 30 metros (bandas visibles e infrarrojas) y 120 metros (banda térmica), cubriendo franjas de 185 km de ancho (eoPortal Directory, 2024).

Componentes de espacio y hardware

El Landsat 5 contaba con una matriz solar articulada que generaba 1430 W y dos baterías de NiCd con 100 Ah para la fase de eclipse. Incluía un sistema de propulsión de hidrazina para mantener la órbita y equipamiento avanzado como antena TDRS, GPS y un módulo de banda ancha. Su diseño destacó por la confiabilidad y longevidad, contribuyendo significativamente al monitoreo ambiental global (eoPortal Directory, 2024).

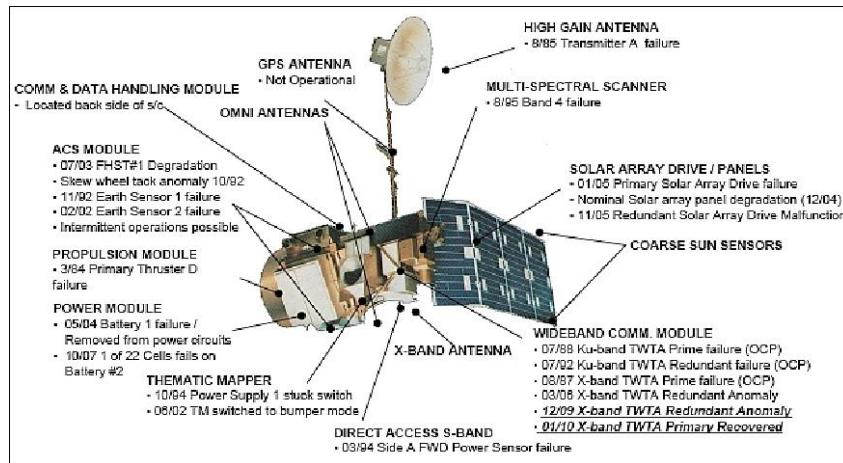


Figura 2.8: Subsistemas de la plataforma L5.

Fuente: (eoPortal Directory, 2024)

2.2.10. Landsat 8 OLI/TIRS Collection 2 Level 1

Los productos Level 1 Precision Terrain (L1TP) se entregan como números digitales (DN) en un formato entero de 16 bits sin signo, los cuales pueden convertirse en reflectancia o radiancia en la parte superior de la atmósfera (TOA) utilizando los factores de escala radiométrica proporcionados en los archivos de metadatos de cada escena. Además, están corregidos tanto radiométrica como geométricamente. Las Tablas 1 y 2 muestran las referencias y especificaciones de las bandas OLI, respectivamente (NASA, 2022).

Número de banda	Descripción de banda	Rango de banda (nm)
1	Coastal Aerosol (Operational Land Imager (OLI))	435-451
2	Blue (OLI)	452-512
3	Green (OLI)	533-590
4	Red (OLI)	636-673
5	Near-Infrared (NIR) (OLI)	851-879
6	Short Wavelength Infrared (SWIR) 1 (OLI)	1566-1651
7	SWIR 2 (OLI)	2107-2294
8	Panchromatic (OLI)	503-676
9	Cirrus (OLI)	1363-1384
10	Thermal Infrared Sensor (TIRS) 1	10600-11190
11	TIRS 2	11500-12510

Tabla 2.1: Referencia de bandas

Cada imagen de L1TP se encuentra en un archivo independiente. Cada banda es un archivo Cloud Optimized GeoTIFF (COG) en escala de grises, los archivos de imagen contienen las etiquetas y claves definidas por la especificación del formato de archivo de imagen con

Numero de banda	Identificador	Tipo de dato	Unidad	Rango
1	B1	UINT16	W/(m ² sr um)	[1 - 65535]
2	B2	UINT16	W/(m ² sr um)	[1 - 65535]
3	B3	UINT16	W/(m ² sr um)	[1 - 65535]
4	B4	UINT16	W/(m ² sr um)	[1 - 65535]
5	B5	UINT16	W/(m ² sr um)	[1 - 65535]
6	B6	UINT16	W/(m ² sr um)	[1 - 65535]
7	B7	UINT16	W/(m ² sr um)	[1 - 65535]
8	B8	UINT16	W/(m ² sr um)	[1 - 65535]
9	B9	UINT16	W/(m ² sr um)	[1 - 65535]

Tabla 2.2: Especificaciones de las bandas OLI

Numero de banda	Identificador	Tipo de dato	Unidad	Rango
10	B10	UINT16	W/(m ² sr um)	[1 - 65535]
11	B11	UINT16	W/(m ² sr um)	[1 - 65535]

Tabla 2.3: Especificaciones de las bandas TIRS

etiquetas geográficas (GeoTIFF), donde GeoTIFF define un conjunto de etiquetas Tagged Image File Format (TIFF), que describen información cartográfica y geodética asociada con imágenes geográficas TIFF.

2.2.11. Landsat Thematic Mapper (TM) Collection 2 Level 1

Los productos estándar L1TP, que se presentan como números digitales (DN) en un formato entero de 8 bits sin signo, pueden convertirse en reflectancia TOA mediante los factores de escala indicados en los metadatos del producto.

El producto L1TP incluye correcciones radiométricas, geométricas y de precisión, y utiliza un DEM para corregir el error de paralaje causado por el relieve topográfico local. Las Tablas 3 y 4 presentan las referencias y especificaciones de las bandas TM, respectivamente (NASA, 2022).

2.2.12. Índice espectral

Los índices espectrales llegan a ser combinaciones paramétricas de bandas o canales espectrales, estos índices nos ayudan a analizar aspectos territoriales, para el análisis de vege-

Numero de banda	Descripción de banda	Rango de banda (nm)
1	Blue (TM)	450-520
2	Green (TM)	520-600
3	Red (TM)	630-669
4	Near-Infrared (NIR) (TM)	760-900
5	Short Wavelength Infrared (SWIR) 1 (TM)	1550-1750
6	Thermal Infrared 1 (TM)	10400-12500
7	SWIR 2 (TM)	2080-2350

Tabla 2.4: Referencia de bandas

Numero de banda	Identificador	Tipo de dato	Unidad	Rango
1	B1	UINT8	W/(m ² sr um)	[1 - 255]
2	B2	UINT8	W/(m ² sr um)	[1 - 255]
3	B3	UINT8	W/(m ² sr um)	[1 - 255]
4	B4	UINT8	W/(m ² sr um)	[1 - 255]
5	B5	UINT8	W/(m ² sr um)	[1 - 255]
6	B6	UINT8	W/(m ² sr um)	[1 - 255]
7	B7	UINT8	W/(m ² sr um)	[1 - 255]

Tabla 2.5: Especificaciones de las bandas TM

tación, coberturas de agua, coberturas de nieve, proliferación de algas acuáticas, cálculos de niveles de humedad en el terreno, etc (EOS, fecha de acceso).

Índice diferencial normalizado de nieve (NDSI)

Al desear identificar la presencia de nieve, los sensores de los satélites incluyen mediciones a 0.66 y 1.6 mm. Al haber transparencia de la atmósfera en estas longitudes de onda, al tiempo que la nieve no es reflectante a 1,6 mm y muy reflectante a 0,66 mm. La capa de nieve llega a ser tan brillante como las nubes, esto hace que sea mu complicado diferenciarlas una de otra. Sin embargo, a 1,6 mm, la capa de nieve absorbe la luz solar y es por esa razón aparece representada más oscura que las nubes. Esto permite una distinción efectiva y notoria entre las nubes y los cuerpos de nieve. La imagen, por lo tanto, demuestra la habilidad de separar las nubes de la nieve usando observaciones en estas longitudes de onda. El índice NDSI es una medida de la magnitud relativa de la diferencia de reflectancia entre el rango visible del espectro (verde) y el infrarrojo de onda corta (SWIR). Este índice controla la variación de dos bandas, una de ellas en el infrarrojo cercano o en el infrarrojo de onda corta y la otra en las partes visibles del espectro. Esto es adecuado para el mapeo y análisis de nieve. La nieve es reflectante en el espectro visible

y además muy absorbente en el infrarrojo cercano (NIR) o en la parte infrarroja de onda corta del espectro mientras que la mayor parte de la reflectancia de las nubes continúa siendo alta en las mismas partes del espectro, en efecto esto permite una buena separación de la mayoría de las nubes y la nieve (EOS, fecha de acceso).

Fórmula del NDSI

La relación entre las bandas captadas que llegan a componer una imagen de satélite para cuerpos de nieve, tiene la siguiente fórmula.

$$\text{NDSI} = \frac{\text{GREEN}-\text{SWIR}}{\text{GREEN}+\text{SWIR}}$$

2.2.13. Procesamiento de imágenes digitales

En la actualidad, las imágenes constituyen un lenguaje en sí mismas, puesto que las imágenes llegan a transmitir distintos tipos de información. Por esto, es muy necesario contar con un soporte para la representación digital de las imágenes que nos permita luego modificar el mismo, esto con el fin de modificar el contenido visual, simbólico y obtener la información necesaria (Jähne, 2005).

Según (Gonzalez & Woods, 2008), una imagen puede definirse como una función bidimensional $f(x,y)$, donde x e y son coordenadas espaciales, y la amplitud de f en cualquier punto (x,y) se denomina intensidad o nivel de gris de la imagen en ese punto. Cuando x , y y los valores de intensidad de f son cantidades discretas y finitas, llamamos a la imagen una imagen digital. En el campo de las imágenes digitales, esto se refiere al procesamiento de estas mediante una computadora digital. Una imagen digital se compone de un número finito de elementos, cada uno con una ubicación y un valor específico; estos elementos se denominan comúnmente píxeles.

Además la visión humana, uno de nuestros sentidos más avanzados, está limitada a la percepción de la banda visual del espectro electromagnético (EM). Sin embargo, las máquinas

de imágenes pueden abarcar casi todo el espectro EM, desde ondas gamma hasta ondas de radio, y generar imágenes a partir de fuentes no convencionales para el ser humano, como el ultrasonido, la microscopía electrónica e imágenes generadas por computadora.

El procesamiento digital de imágenes cubre un campo amplio y diverso de aplicaciones. Además, existen campos como la visión artificial, cuyo objetivo es utilizar computadoras para emular la visión humana, incluyendo el aprendizaje, la capacidad de hacer inferencias y tomar decisiones basadas en entradas visuales. Esta área es una rama de la inteligencia artificial, destinada a imitar la inteligencia humana.

Formación de una imagen digital

La mayoría de las imágenes que nos interesan se generan mediante la combinación de una fuente de iluminación y la reflexión o absorción de energía de esa fuente por los elementos de la escena captada. Por ejemplo, la iluminación puede provenir de una fuente electromagnética, como radar, infrarrojos o rayos X, pero también podría originarse de fuentes menos tradicionales, como el ultrasonido. De manera similar, los objetos podrían ser familiares o incluso moléculas, formaciones rocosas, etc. Dependiendo de la fuente, la energía de iluminación se refleja desde los objetos o se transmite a través de ellos (Gonzalez & Woods, 2008).

Como menciona (Gonzalez & Woods, 2008), en la Figura 2.9 se muestran las tres disposiciones principales de sensores utilizados para transformar la energía incidente en imágenes digitales. La idea es simple: la energía entrante se convierte en un voltaje mediante una combinación de la energía eléctrica de entrada y el material del sensor, que responde al tipo de energía que se detecta. La forma de onda del voltaje de salida es la respuesta del sensor y se obtiene una cantidad digital al digitalizar esa respuesta. Un sensor conocido de este tipo es el fotodiodo, que está construido con materiales de silicio y cuya salida es un voltaje proporcional a la luz. El uso de un filtro delante del sensor mejora su selectividad; por ejemplo, un filtro óptico de transmisión verde favorece la luz en la banda verde del espectro de color. Como consecuencia, la salida del sensor sería más fuerte para la luz verde que para otros componentes de la luz

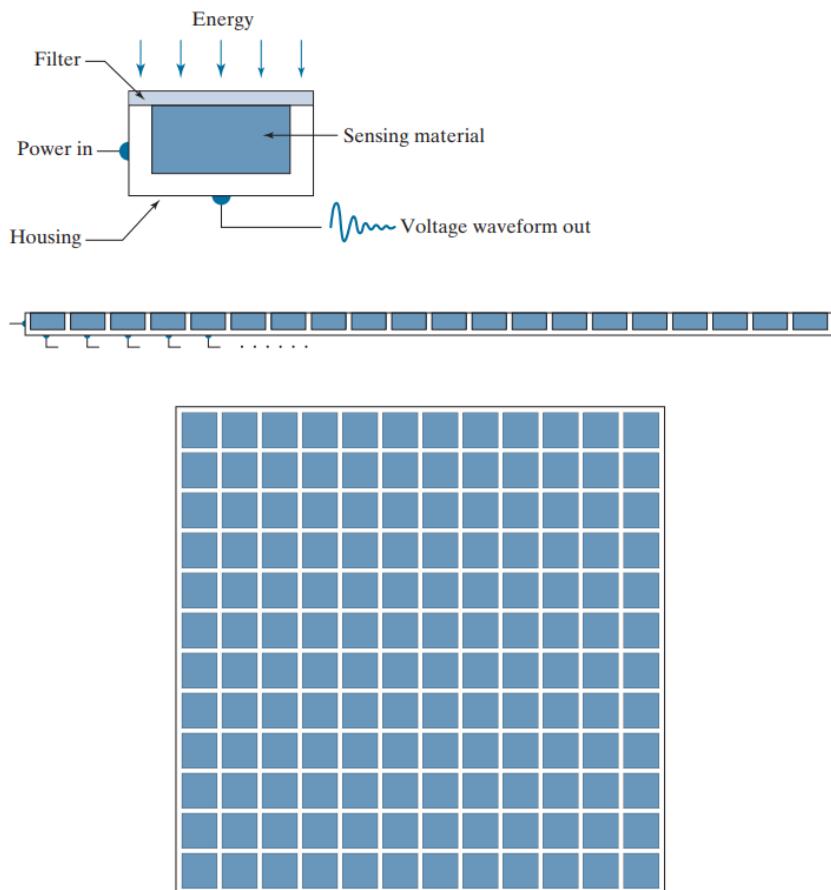


Figura 2.9: Modélo basico de un sensor para adquirir imágenes.
Fuente: (eoPortal Directory, 2024)

visible.

Para generar una imagen 2D utilizando un único elemento sensor, debe haber desplazamientos relativos en las direcciones x e y entre el sensor y el área que se va a fotografiar. La salida de los sensores se procesa mediante algoritmos de reconstrucción cuyo objetivo es transformar los datos detectados en imágenes transversales.

Como se mencionó anteriormente, una imagen se denota mediante funciones bidimensionales de la forma $f(x,y)$. El valor f en las coordenadas espaciales (x, y) es una cantidad escalar cuyo significado físico está determinado por la fuente de la imagen y cuyos valores son proporcionales a la energía irradiada por una fuente física (por ejemplo, ondas electromagnéticas). Como consecuencia, $f(x,y)$ debe ser no negativo y finito.

$$0 \leq f(x, y) < \infty \quad (2.1)$$

La función $f(x, y)$ se caracteriza por dos componentes: (1) la cantidad de iluminación de la fuente incidente sobre la escena que se está observando, y (2) la cantidad de luz reflejada por los objetos en la escena. Estos se denominan componentes de iluminación y reflectancia, y se denotan por $i(x, y)$ y $r(x, y)$ respectivamente. Las dos funciones se combinan como un producto para formar $f(x, y)$

$$f(x, y) = i(x, y)r(x, y) \quad (2.2)$$

Donde:

$$0 \leq i(x, y) < \infty \quad (2.3)$$

y

$$0 \leq r(x, y) < \infty \quad (2.4)$$

Por lo tanto, la reflectancia está limitada por 0 (absorción total) y 1 (reflectancia total). La naturaleza de $i(x, y)$ está determinada por la fuente de iluminación y $r(x, y)$ está determinada por las características de los objetos representados en la imagen (Gonzalez & Woods, 2008).

Un ejemplo de adquisición de imágenes digitales. (a) Fuente de iluminación (energía). (b) Una escena. (c) Sistema de imágenes. (d) Proyección de la escena sobre el plano de la imagen. (e) Imagen digitalizada.

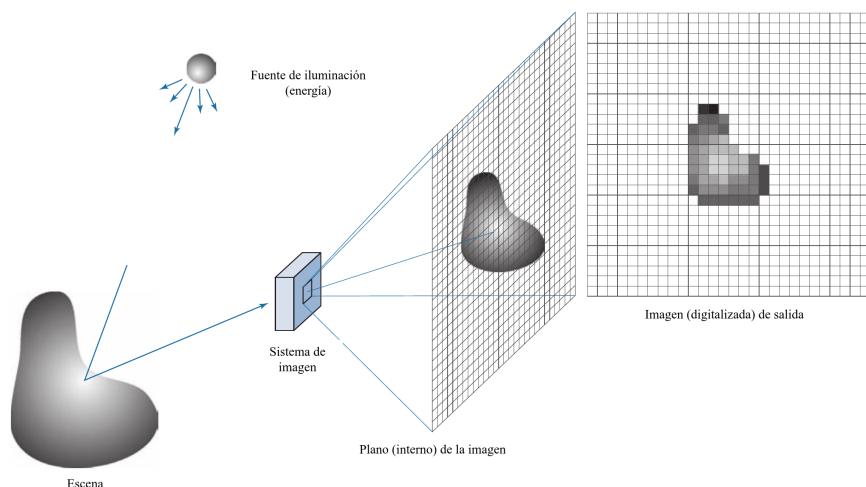


Figura 2.10: Ejemplo de adquisición y formación de una imagen digital: *Fuente de iluminación (energía).* (b) Una escena. (c) Sistema de imágenes. (d) Proyección de la escena sobre el plano de la imagen. (e) Imagen digitalizada.

Fuente: (Gonzalez & Woods, 2008)

2.2.14. Deep Learning

La inteligencia artificial (IA) se ocupa de construir sistemas que simulan comportamientos inteligentes. Dentro de la IA, el aprendizaje automático (Machine Learning) es un subconjunto que ajusta modelos matemáticos a datos observados para tomar decisiones. El aprendizaje profundo (Deep Learning) es un tipo de aprendizaje automático avanzado; sus redes profundas son hoy en día los modelos más potentes (Prince, 2023).

Es común usar algoritmos de procesamiento de lenguaje natural para traducir texto, visión artificial para analizar imágenes o asistentes digitales con reconocimiento de voz, esto promete transformar nuestro mundo, aunque no todos los efectos serán positivos.

2.2.15. Estructura de entradas y salidas

En la Figura 2.12 se puede observar que, en cada caso, hay una entrada significativa del mundo real, ya sea una oración, un archivo de sonido, una imagen, etc. Esta entrada se codifica como un vector de números, el cual forma la entrada del modelo. El modelo asigna esta entrada a un vector de salida, que luego se traduce nuevamente en una predicción significativa del mundo

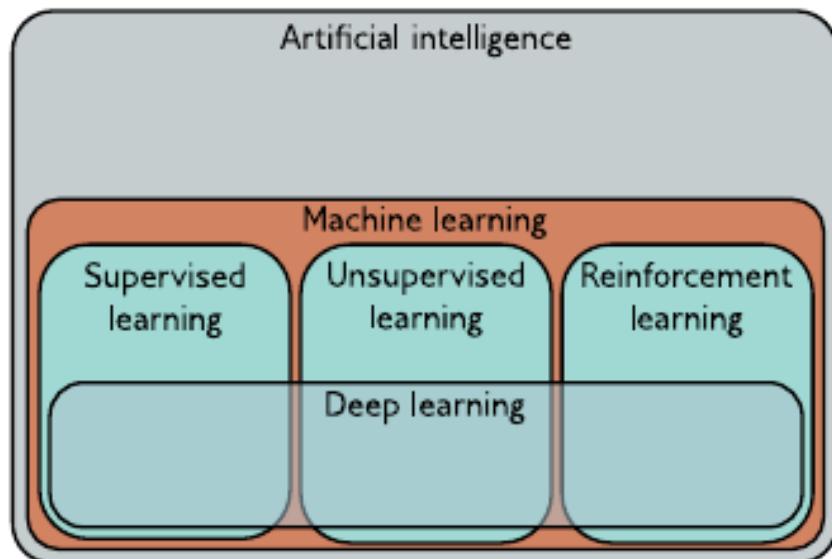


Figura 2.11: El Machine Learning viene ser un área de la IA, se puede dividir en aprendizaje supervisado, aprendizaje no supervizado y aprendizaje de refuerzo. El deep learning contribuye a cada una de estas tareas.

Fuente: (Prince, 2023)

real.

Por ahora, nos centramos en las entradas y salidas, tratando el modelo como una "caja negra" que recibe un vector de números y devuelve otro.

La entrada es un vector de longitud fija que contiene valores que caracterizan una propiedad. Por ejemplo, la Figura 2.12a representa un modelo de clasificación binaria multivariante para la segmentación semántica. Aquí, a cada píxel de una imagen de entrada se le asigna una etiqueta binaria que indica si pertenece a una vaca o al fondo. La Figura 2.12b muestra un modelo de regresión multivariante, donde la entrada es una imagen de una escena callejera y la salida es la profundidad en cada píxel.

En ambos casos, la salida es de alta dimensión y estructurada. Sin embargo, esta estructura está estrechamente ligada a la entrada, lo cual se puede explotar; si un píxel está etiquetado como "vaca", es probable que un píxel vecino con valores RGB similares tenga la misma etiqueta.

Las Figuras 2.12c–e representan tres modelos donde la salida tiene una estructura compleja que no está tan estrechamente ligada a la entrada. La Figura 2.12c muestra un modelo donde la entrada es un archivo de audio y la salida son las palabras transcritas de ese archivo. La Figura

2.12d representa un modelo de traducción, en el cual la entrada es un texto en inglés y la salida es su traducción al francés. La Figura 2.12e muestra una tarea muy desafiante: la entrada es un texto descriptivo, y el modelo debe generar una imagen que coincide con esa descripción (Prince, 2023).

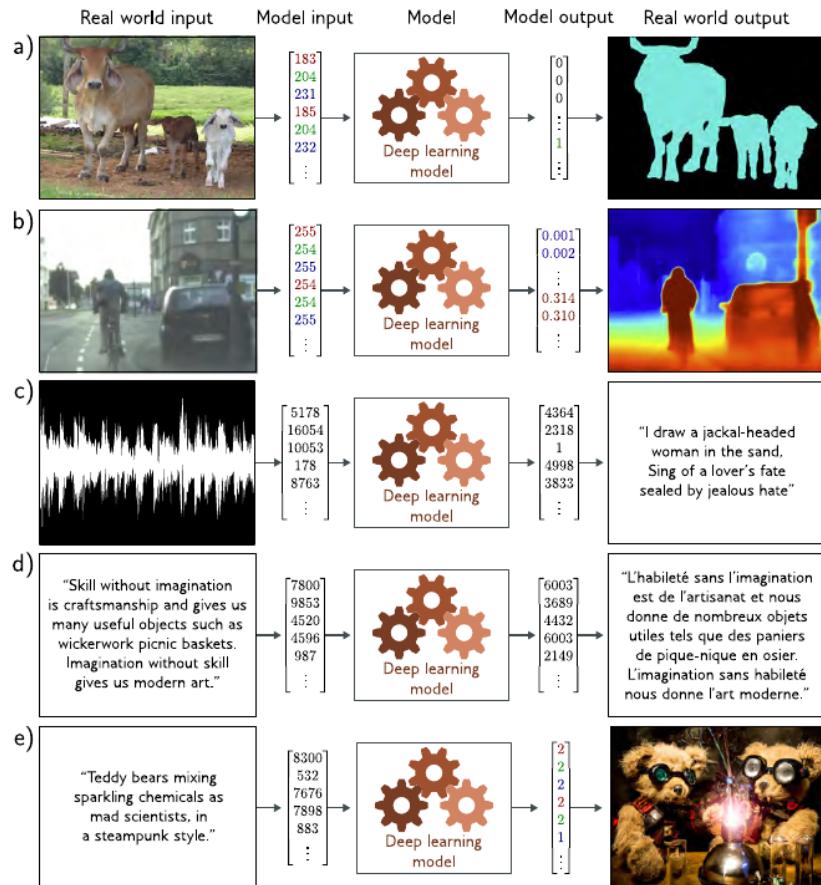


Figura 2.12: Tareas de aprendizaje supervisado con salidas estructuradas.
Fuente: (Prince, 2023)

2.2.16. Aprendizaje Supervizado

En la Figura 2.13, se observa un modelo de aprendizaje automático tratado como una "caja negra" que recibe un vector de entrada y produce un vector de salida. Ahora bien, ¿qué sucede dentro de esta caja? Imaginemos un modelo que predice la altura de un niño a partir de su edad. El modelo es una ecuación matemática que muestra cómo la altura promedio varía según la edad, representada por una curva en color cian. Así, si se ingresa una edad, el modelo devuelve una altura; por ejemplo, a los 10 años se predice una altura de 139 cms.

Más específicamente, el modelo pertenece a una familia de ecuaciones que mapea la entrada a la salida (una serie de curvas en cian). La ecuación particular se selecciona mediante datos de entrenamiento, que consisten en pares de datos de entrada y salida (marcados en naranja en la figura), y el modelo ajusta esta curva para que represente con precisión estos datos. Este ajuste se conoce como “entrenar el modelo” y se refiere a encontrar, en esta familia de ecuaciones, la que mejor describe los datos de entrenamiento.

En segmentación semántica, como el ejemplo de las vacas, se necesitan imágenes etiquetadas por un humano, lo cual se conoce como aprendizaje supervisado. Aquí, el modelo se comporta como una ecuación matemática donde al ingresar datos, se calculan resultados, proceso llamado “inferencia”. La ecuación tiene parámetros que definen la relación entre entrada y salida, los cuales se afinan durante el entrenamiento. Para hacer predicciones, el modelo utiliza una función $f[x]$ que toma la entrada x y devuelve la salida y , donde los parámetros ϕ determinan la relación precisa.

Durante el entrenamiento, el modelo ajusta los parámetros para que las predicciones se acerquen lo más posible a las salidas reales, minimizando una función de pérdida $L[\phi]$. La minimización de esta pérdida permite encontrar los mejores parámetros, haciendo que el modelo sea preciso en las predicciones. Después de entrenarlo, se evalúa su rendimiento utilizando un conjunto de prueba, midiendo así su capacidad para generalizarse a datos nuevos no vistos durante el entrenamiento (Prince, 2023).

2.2.17. Función de perdida

La función de pérdida o función de coste, mencionada anteriormente, mide el desajuste entre el modelo y el conjunto de datos de entrenamiento, el cual está compuesto por pares de datos de entrada/salida $[x_i, y_i]$ como se observa en la Figura 2.14a. En la Figura 2.14b-d presentan tres líneas que representan distintos conjuntos de parámetros. La línea verde ajusta los datos con mayor precisión, acercándose más a los puntos de datos. Para comparar los parámetros ϕ , cada selección recibe un valor numérico de pérdida, que indica cuánto se desvían

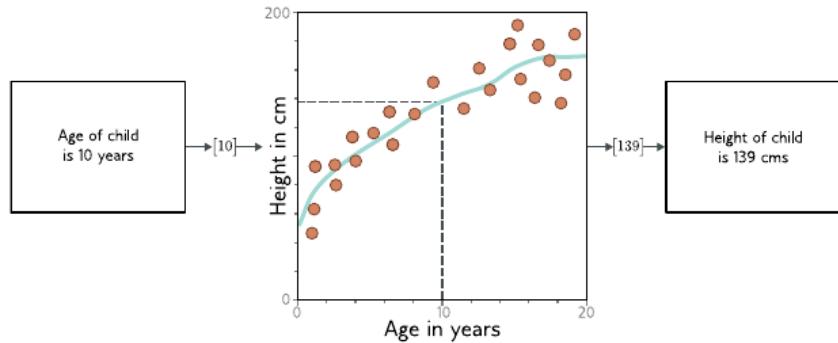


Figura 2.13: El modelo representa una familia de relaciones que relacionan la entrada (edad del niño) con la salida (altura del niño). La relación en particular se elige utilizando datos de entrenamiento, que consisten en pares de entrada/salida (puntos naranjas). Cuando entrenamos el modelo, buscamos entre las posibles relaciones una que describa bien los datos. Aquí, el modelo entrenado es la curva cian y se puede utilizar para calcular la altura para cualquier edad. Fuente: (Prince, 2023)

las predicciones del modelo respecto a los valores reales, el cual se representan mediante las líneas naranjas en las 2.14b-d. Esta desajuste total (pérdida total) se calcula sumando los cuadrados de las desviaciones para todos los pares de entrenamiento.

Dado que los mejores parámetros minimizan esta expresión, la llamamos pérdida por mínimos cuadrados. La operación de elevar al cuadrado significa que la dirección de desviación carece de importancia, es decir no importa si la línea está por encima o por debajo de los datos (Prince, 2023).

$$\mathcal{L}[\phi] = \sum_{i=1}^I (f[x_i, \phi] - y_i)^2 = \sum_{i=1}^I (\phi_0 + \phi_1 x_i - y_i)^2 \quad (2.5)$$

2.2.18. Entrenamiento

El proceso de encontrar los parámetros que minimicen la pérdida se llama ajuste del modelo o entrenamiento. Inicialmente, los parámetros se seleccionan al azar, y luego se ajustan utilizando la función de pérdida hasta llegar al mínimo, como se ilustra en la figura. Un enfoque común es calcular el gradiente de la superficie en la posición actual y dar un paso en la dirección de mayor descenso, lo que permite optimizar progresivamente los parámetros (Prince, 2023).

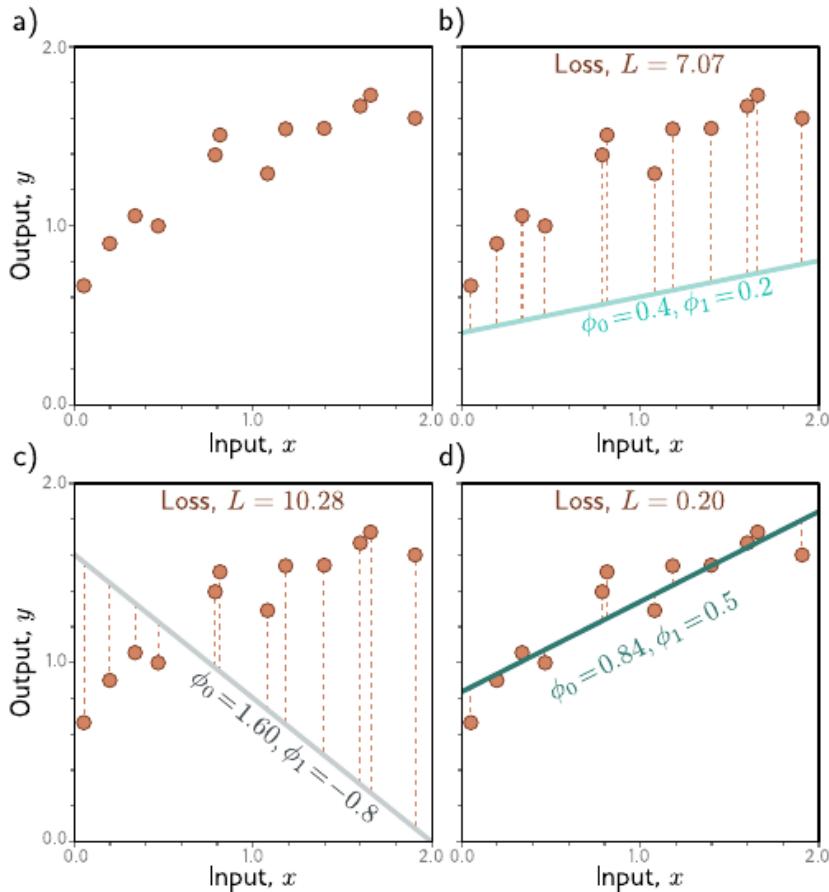


Figura 2.14: En (a), los datos de entrenamiento (puntos naranjas) constan de $I = 12$ pares de entrada/salida $[x_i, y_i]$. En (b)-(d), cada panel muestra el modelo de regresión lineal con distintos parámetros. Según la elección de los parámetros de intersección y pendiente $\phi = [\phi_0, \phi_1]$, los errores del modelo (líneas discontinuas) pueden ser mayores o menores. Los parámetros en las líneas de los paneles (b) y (c) tienen grandes pérdidas $L = 7,07$ y $L = 10,28$, respectivamente, debido a su ajuste pobre. La pérdida $L = 0,20$ en el panel (d) es menor, indicando un mejor ajuste; estos parámetros son los óptimos. Fuente: (Prince, 2023).

2.2.19. Testeo

Una vez que el modelo ha sido entrenado, es importante evaluar su rendimiento en el mundo real. Esto se hace calculando la pérdida sobre un conjunto de datos de prueba. La capacidad del modelo para generalizar sus predicciones depende de cuán representativos y completos sean los datos de entrenamiento. Sin embargo, también depende de la complejidad del modelo. Un modelo demasiado simple podría no capturar correctamente la relación entre entrada y salida (underfitting), mientras que uno demasiado complejo podría ajustarse a peculiaridades irrelevantes de los datos (overfitting) (Prince, 2023).

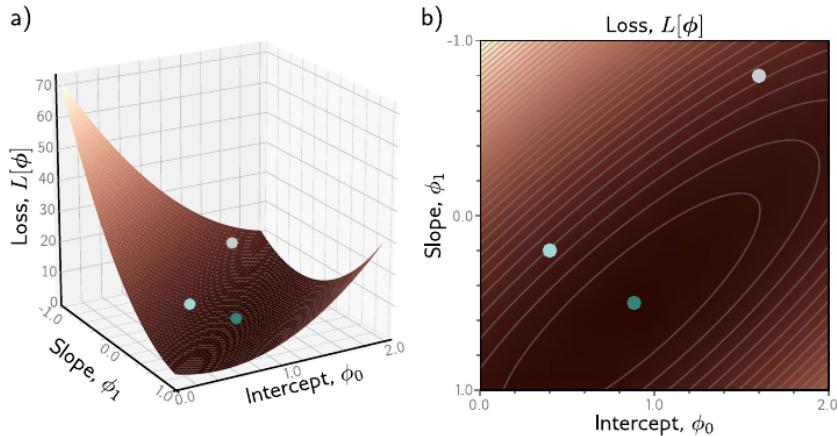


Figura 2.15: a) Cada combinación de parámetros $\phi = [\phi_0, \phi_1]$ tiene una pérdida asociada, la función de pérdida resultante $L[\phi]$ puede visualizarse como una superficie. En la figura, los tres círculos representan las tres líneas de la Figura 2.14b-d. b) La pérdida también puede visualizarse como un mapa de calor, donde las regiones más brillantes representan pérdidas mayores. El círculo verde tiene los parámetros con la menor pérdida. Fuente: (Prince, 2023).

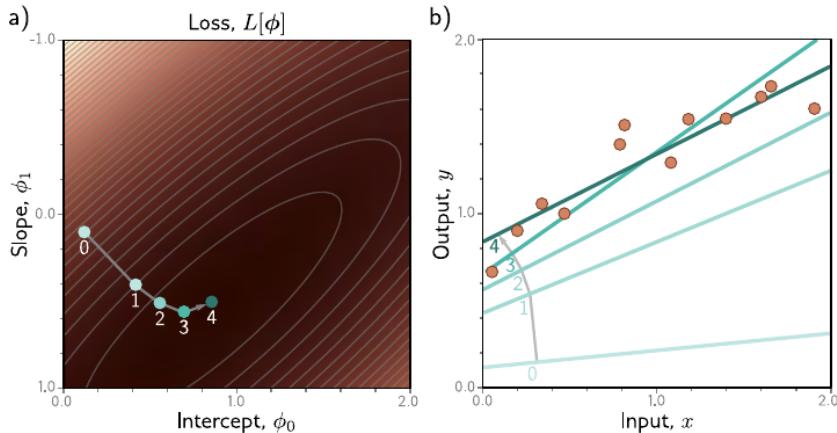


Figura 2.16: a) Algoritmos de entrenamiento iterativos inicializan los parámetros aleatoriamente y luego los mejoran «caminando cuesta abajo» hasta que ya no se puedan mejorar más. Aquí, empezamos en la posición 0 y nos movemos una cierta distancia cuesta abajo (perpendicular a las curvas de nivel) hasta la posición 1. A continuación, volvemos a calcular la dirección cuesta abajo y nos desplazamos a la posición 2. Finalmente alcanzamos el mínimo de la función (posición 4). b) Cada posición 0-4 del panel (a) corresponde a una intersección y una pendiente diferentes y, por lo tanto, representa una línea diferente. A medida que disminuye la pérdida, las líneas se ajustan más a los datos. Fuente: (Prince, 2023).

2.2.20. Redes Neuronales profundas

Son un tipo de modelo de aprendizaje automático, son ecuaciones que pueden representar una familia extremadamente amplia de relaciones entre la entrada y la salida. Estas redes neuronales profundas pueden procesar entradas muy grandes, de longitud variable y contener varios tipos de estructura interna. Pueden generar números reales únicos (regresión), números

múltiples (regresión multivariable) o probabilidades sobre una o más clases (clasificación binaria y multiclase) (Prince, 2023).

2.2.21. Aprendizaje no Supervizado

El aprendizaje no supervisado la construcción de un modelo a partir de datos de entrada sin las correspondientes etiquetas de salida se denomina aprendizaje no supervisado, la ausencia de etiquetas de salida significa que no puede haber "supervisión". En lugar de aprender una correspondencia entre la entrada y salida, el objetivo es comprender la estructura de los datos (Prince, 2023).

2.2.22. Redes neuronales artificiales

Parte de la inteligencia artificial, inspiradas en las redes neuronales biológicas, similares a las que componen el cerebro humano (Olabe, 1998), además estas redes neuronales artificiales tienen una composición jerárquica, pues dependen de una capa de entrada, capas ocultas, y capas de salida tal como se muestra en la Figura 2.17; sin embargo, para que estas redes neuronales artificiales funcionen adecuadamente, es necesario contar una gran cantidad de datos de entrenamiento para ajustar sus pesos y sesgos (IBM, 2021).

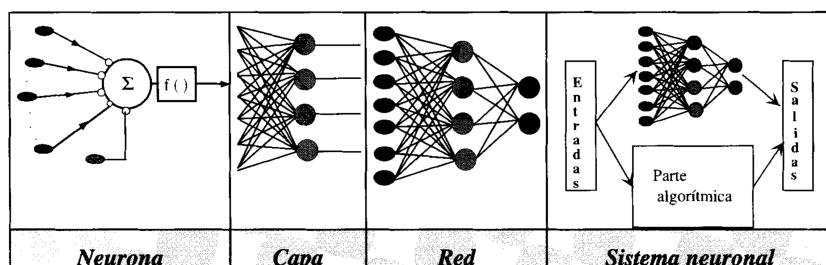


Figura 2.17: Estructura jerárquica de una ANN.
Fuente: (Brío & Molina, 2001)

2.2.23. Redes Neuronales Convolucionales

Es un enfoque de deep learning utilizado para resolver problemas complejos, adecuado para identificar patrones en una imagen (clasificación, segmentación, etc.), supera las limitaciones de enfoques tradicionales de machine learning, además resultan útiles para tratamiento y procesamiento de datos como audio y otros. (MathWorks,).

El término red neuronal convolucional "hace referencia a una red que emplea una operación matemática denominada convolución, que es un tipo especializado de operación lineal. En lugar de la multiplicación de matrices convencional usada en las redes neuronales tradicionales (tipo feedforward), las redes convolucionales utilizan convoluciones en una o más de sus capas (Goodfellow et al., 2016).

Generalmente, la operación utilizada en una red convolucional no se corresponde exactamente con la definición de convolución en campos como la ingeniería o las matemáticas. Normalmente, se denota con un asterisco:

$$s(t) = (x * w)(t) \quad (2.6)$$

En terminología de redes convolucionales, el primer argumento (función x) de la convolución a menudo se denomina entrada, y el segundo argumento (w) como kernel. La salida a veces se denomina mapas de características.

El índice de tiempo t puede entonces tomar solo valores enteros. Si ahora suponemos que x y w están definidos solo en el entero t, podemos definir la convolución discreta.

$$h(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (2.7)$$

En las aplicaciones de machine learning, la entrada suele ser una matriz multidimensional de datos y el kernel suele ser una matriz multidimensional de parámetros que se adaptan

mediante el algoritmo de aprendizaje. Nos referimos a estas matrices multidimensionales como tensores. Debido a que cada elemento de la entrada y el kernel debe almacenarse explicitamente por separado, generalmente se asumen que estas funciones son cero en todas partes, excepto en el conjunto finito de puntos para los que almacenamos en valores. Esto significa que en la práctica, podemos implementar la suma infinita como una suma sobre un número finito de elementos de la matriz (Goodfellow et al., 2016).

$$h(i, j) = (X * W)(i, j) = \sum_m \sum_n X(m, n)W(i - m, j - n) \quad (2.8)$$

La convolución es commutativa, lo que significa que podemos escribir de manera equivalente

$$h(i, j) = (W * X)(i, j) = \sum_m \sum_n X(i - m, j - n)W(m, n) \quad (2.9)$$

La propiedad commutativa de la convolución surge porque hemos invertido el kernel en relación con la entrada. La única razón para invertir el kernel es obtener la propiedad commutativa.

En cambio, muchas bibliotecas de redes neuronales implementan una función relacionada llamada correlación cruzada, que es lo mismo que la convolución pero sin invertir el kernel. Muchas bibliotecas de aprendizaje automático implementan la correlación cruzada pero la llaman convolución. En este texto seguimos esta convención de llamar a ambas operaciones convolución y especificamos si queremos invertir el kernel o no en contextos donde la inversión del kernel es relevante (Goodfellow et al., 2016).

$$h(i, j) = (W * X)(i, j) = \sum_m \sum_n X(i + m, j + n)W(m, n) \quad (2.10)$$

En el contexto del aprendizaje automático, el algoritmo de aprendizaje aprenderá los valores apropiados del kernel en el lugar apropiado, por lo que un algoritmo basado en convolución con inversión de kernel aprenderá un kernel que esté invertido en relación con el kernel

aprendido por un algoritmo sin la inversión (Goodfellow et al., 2016).

Por otro lado, las redes convolucionales, Se usa principalmente para procesar datos de imágenes, las capas convolucionales procesan cada región de imagen local de forma independiente, utilizando párametros compartidos en toda la imagen. El kernel convolucional ahora es un objeto 2D. Un kernel cuadrado aplicado a una entrada 2D, que comprende elementos X_{ij} .

Donde $W(m,n)$ son las entradas del kernel convolucional. Esto es simplemente una suma ponderada sobre una región de entrada cuadrada, el kernel se traslada tanto horizontal como verticalmente a través de la entrada 2D como se observa en la figura 2.18, para crear una salida en cada posición.

A menudo, la entrada es una imagen RGB, que se trata como una señal en 2D con tres canales tal como se muestra en la figura. Aquí, un kernel de 3x3 tendría pesos 3x3x3 y se aplicaría a los tres canales de entrada en cada una de las 3x3 posiciones para crear una salida 2D que tenga la misma altura y el ancho de la imagen de entrada (suponiendo un padding de cero). Para generar múltiples canales de salida, repetimos este proceso con diferentes pesos de kernel y agregamos los resultados para formar un tensor 3D. Si el kernel tiene un tamaño $K \times K$ y hay canales de entrada C_i , cada canal de salida es una suma ponderada de cantidades $C_i \times K \times K$ más un sesgo (bias). De ello se deduce que para calcular los canales de salida C_o , necesitaremos pesos $C_i \times C_o \times K \times K$ y sesgos C_o (Prince, 2023).

Segmentación semántica

La segmentación semántica de imágenes es una poderosa técnica de visión por computadora que implica la comprensión y el análisis de las imágenes a nivel de píxeles utilizando algoritmos de aprendizaje profundo. Tiene como objetivo asignar una etiqueta significativa a cada píxel de la imagen, dividiéndola efectivamente en diferentes objetos y/o regiones. A diferencia de la clasificación de imágenes, la cual asigna una sola etiqueta a la imagen completa, la segmentación semántica proporciona una comprensión más detallada y granular del contenido visual dentro de una imagen de acuerdo a diversas categorías. Es clave en aplicaciones como la

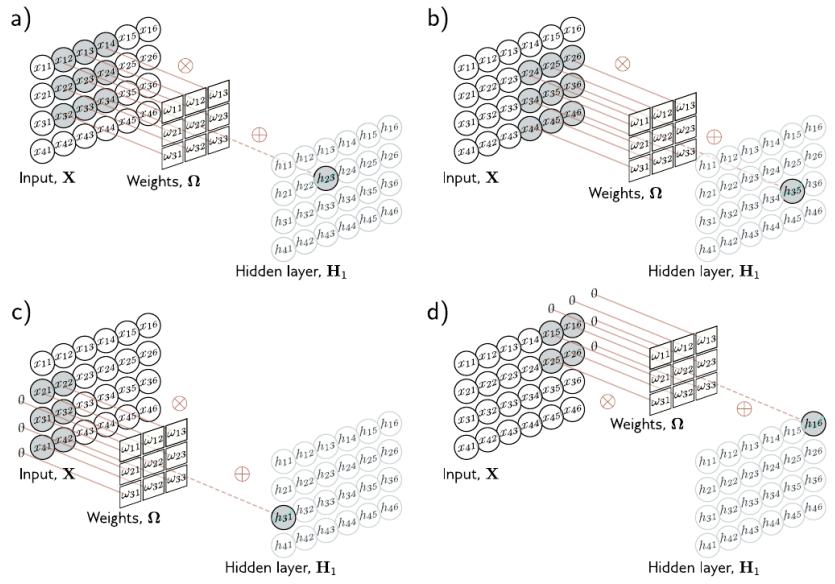


Figura 2.18: Cada salida H_{ij} calcula una suma ponderada de 3×3 entradas más cercanas, agrega un sesgo y pasa el resultado a través de una función de activación, a) Aquí, la salida h_{23} es una suma ponderada de las nueve posiciones desde x_{12} a x_{34} . b) Se calculan diferentes salidas trasladando el kernel a lo largo de la cuadrícula de la imagen en dos dimensiones. c-d) Con padding, las posiciones más allá del borde de la imagen se consideran cero.
Fuente: (Prince, 2023)

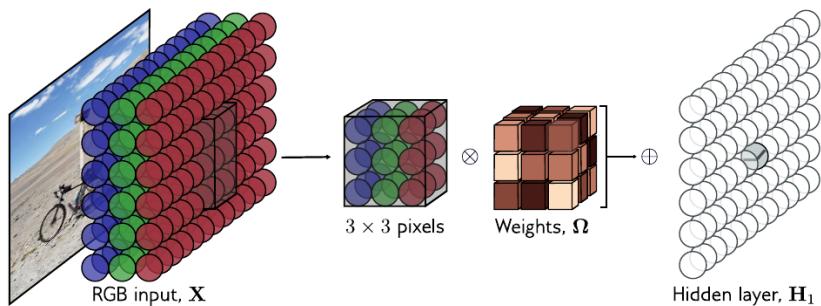


Figura 2.19: Convolución 2D aplicado a una imagen, donde la imagen es una entrada de tres canales RGB con un kernel de 3×3 , cada preactivación en la capa oculta se calcula multiplicando punto a punto los pesos del kernel $3 \times 3 \times 3$ por el parche de la imagen RGB 3×3 centrado en la misma posición, sumando y añadiendo el sesgo. Para calcular todas las reactivaciones en la capa oculta, "deslizamos" el kernel sobre la imagen tanto en la dirección horizontal como vertical. La salida es una capa 2D de unidades ocultas. Para crear múltiples canales de salida, se repite este proceso con múltiples kernels, lo que daría como resultado un tensor 3D de unidades ocultas en la capa oculta H_1 . Fuente: (Prince, 2023)

conducción autónoma y la medicina, donde es necesario identificar áreas relevantes dentro de una imagen. (Herdy, fecha de acceso).

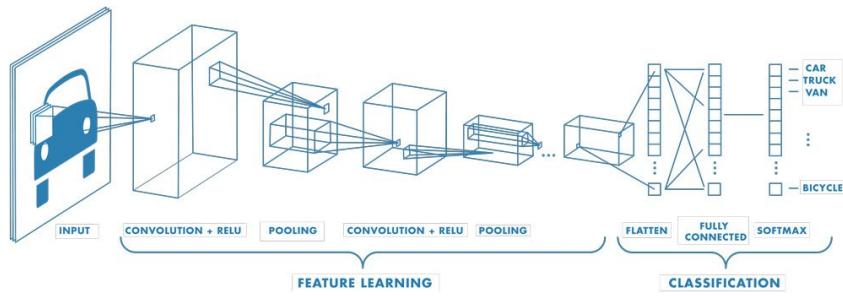


Figura 2.20: Representación de una CNN.

Fuente: (MathWorks,)



Figura 2.21: Resultado de la segmentación semántica, el resultado final se crea a partir de los mapas de probabilidad, utilizando un mapa heurístico para encontrar un mapa binario en función de las probabilidades y su proximidad espacial. Si hay evidencia suficiente, se agregan las clases subsiguientes y se combinan sus mapas de segmentación.

Fuente: (Herdy, fecha de acceso)

Arquitecturas de segmentación semántica

Existen muchas arquitecturas disponibles que han sido desarrolladas a lo largo de los años, sin embargo, las arquitecturas con un menor número de parámetros son los idóneos para esta investigación.

U-Net

U-Net es una arquitectura de red neuronal convolucional diseñada específicamente para segmentación de imágenes biomédicas a nivel de píxel. Desarrollada en 2015 por Olaf Ronneberger, Philipp Fischer y Thomas Brox, U-Net ha tenido un impacto considerable no solo en el ámbito biomédico, sino también en otros campos que requieren segmentación precisa de imágenes. Su estructura simétrica de codificación-decodificación permite la identificación y clasificación de características a diferentes escalas, destacándose por su eficiencia al aprender con conjuntos de datos limitados. Las aplicaciones de U-Net abarcan desde la detección de

patrones en teledetección y segmentación de imágenes satelitales hasta la creación de sistemas de apoyo al diagnóstico en medicina, lo que demuestra su versatilidad y capacidad de adaptación (Ronneberger et al., 2015).

La arquitectura U-Net se distingue por su estructura en forma de U, de donde proviene su nombre. Se compone de un camino de codificación y un camino de decodificación.

Camino de codificación: Esta parte de la red se encarga de capturar el contexto de la imagen de entrada mediante una serie de capas convolucionales y max-pooling, reduciendo la resolución de las dimensiones espaciales y "contrayendo" las imágenes originales.

Camino de decodificación: Esta ruta emplea capas convolucionales y de muestreo ascendente para generar un mapa de segmentación que mantiene las mismas dimensiones espaciales que la imagen de entrada, . "expandiendo" las imágenes previamente contraídas.

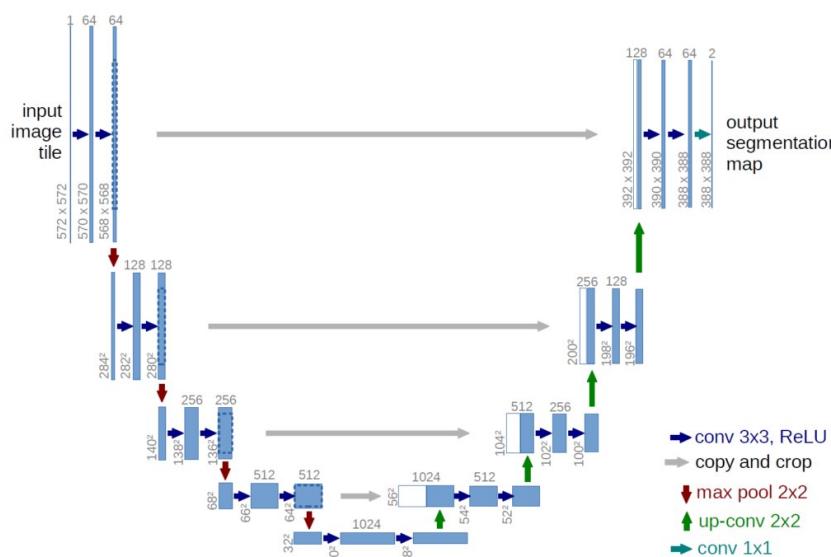


Figura 2.22: Arquitectura U-Net.
Fuente: (Ronneberger et al., 2015)

La fortaleza de U-Net en la segmentación semántica se debe al uso de conexiones de salto (representadas por las flechas grises en la figura 2.22), que vinculan las rutas de codificación y decodificación al fusionar características. Esto permite conservar los detalles espaciales que se pierden durante la reducción de resolución, manteniendo tanto el contexto local como global de la imagen. Al preservar esta información espacial, U-Net logra máscaras de segmentación más precisas. Estas conexiones de salto permiten que la red capte las relaciones entre las distintas

partes de la imagen, mejorando los resultados de segmentación.

La arquitectura U-Net incluye conexiones de salto que facilitan la fusión de características de bajo y alto nivel, lo que mejora la localización. Durante la etapa de reducción de resolución en la arquitectura, antes de realizar el MaxPooling, se guarda el tensor convolucional, que luego se concatena con un tensor muestreado ascendente que tiene la misma dimensión.

ResUNet

Inspirados por el aprendizaje profundo residual (He et al., 2016) y la arquitectura U-Net (Ronneberger et al., 2015), (Zhang et al., 2018) propuso el deep residual U-Net, una arquitectura que combina las fortalezas tanto del aprendizaje residual profundo como de U-Net. El deep residual U-Net (ResUnet) se basa en la estructura de U-Net, pero con dos diferencias clave. Primero, en lugar de unidades neuronales simples, utiliza unidades residuales como bloques básicos para construir el deepResUnet. Segundo, se elimina la operación de recorte de la red, lo que resulta en una arquitectura más elegante y con un mejor rendimiento.

Esta combinación trae dos beneficios: 1) la unidad residual facilitará el entrenamiento de la red; 2) las conexiones salteadas dentro de una unidad residual y entre niveles bajos y altos de la red facilitarán la propagación de información sin degradación, esto permite que el diseño tenga mucho menos parámetros y a su vez lograr rendimientos mucho mejor para la segmentación semántica.

Sin embargo, entrenar una red neuronal tan profunda es complicado, especialmente cuando los datos de entrenamiento son limitados. Una solución a este problema es utilizar una red preentrenada y luego ajustarla en el conjunto de datos específico. Otra opción es aplicar una amplia data augmentation. Además la estructura de U-Net también ayuda a mitigar las dificultades en el entrenamiento. La idea es que al copiar características de bajo nivel a los niveles superiores correspondientes, se crea un camino que facilita la propagación de información entre los diferentes niveles, lo que no solo mejora la retropropagación durante el entrenamiento, sino que también permite que los detalles finos de bajo nivel complementen las características se-

mánticas de alto nivel. Esta idea guarda cierta similitud con la de las redes neuronales residuales (He et al., 2016). Esto demostró que el rendimiento de U-Net mejoró aún más al reemplazar las unidades simples por unidades residuales.

ResidualUnit: Profundizar en la estructura de una red neuronal multicapa podría mejorar su rendimiento, pero también podría dificultar el entrenamiento, e incluso podría surgir un problema de degradación (He et al., 2016). Para superar estos problemas, (He et al., 2016) propuso la red neuronal residual, que facilita el entrenamiento y aborda el problema de degradación. La red neuronal residual está compuesta por una serie de unidades residuales apiladas. Cada unidad residual se puede representar en una forma general.

$$y_l = h(x_l) + F(x_l, W_l) \quad (2.11)$$

$$x_{l+1} = f(y_l) \quad (2.12)$$

Donde x_l y x_{l+1} son la entrada y salida de la unidad residual número l , $F(\cdot)$ es la función residual, $f(y_l)$ es la función de activación y $h(x_l)$ es una función de mapeo de identidad , donde una opción típica es $h(x_l) = x_l$. La 3.9 muestra la diferencia entre una unidad simple y una unidad residual. Dentro de una unidad residual, existen múltiples combinaciones de normalización por lotes (BN), activación ReLU y capas convolucionales. Además se utiliza unidades residuales con preactivación completa para construir el modelo deep residual U-Net.|

La arquitectura de DeepResUnet utiliza siete niveles para la extracción de características, como se muestra en la figura 2.23, además se compone de tres partes: codificación, puente y decodificación. La primera parte codifica la imagen de entrada en representaciones compactas. La última parte recupera las representaciones para una categorización a nivel de píxel, es decir, segmentación semántica. La parte intermedia actúa como un puente que conecta los caminos de codificación y decodificación. Todas las tres partes están construidas con unidades residuales que consisten en dos bloques de convolución de 3×3 y un mapeo de identidad.

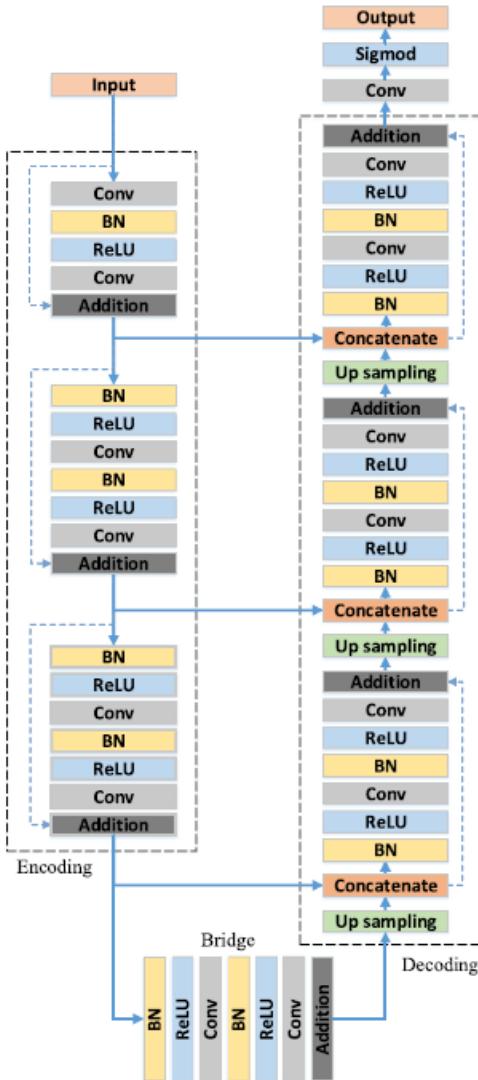


Figura 2.23: Arquitectura profunda ResUnet.

Fuente: (Ronneberger et al., 2015)

El camino de codificación y decodificación tiene tres unidades residuales. Antes de cada unidad, se realiza un muestreo ascendente de los mapas de características del nivel inferior y una concatenación con los mapas de características del camino de codificación correspondiente. Después del último nivel del camino de decodificación, se utiliza una convolución de 1×1 y una capa de activación sigmoidea para proyectar los mapas de características multicanal en la segmentación deseada. En total, tenemos 15 capas convolucionales en comparación con las 23 capas de U-Net.

DeepLabV3

DeepLabv3+ (Chen et al., 2018) es un modelo de redes neuronales completamente convolucionales dilatadas (Dilated FCN) que se utiliza comúnmente para la segmentación semántica, logrando un alto rendimiento en estas tareas. Esta arquitectura mejora el modelo original DeepLabv3 (Chen et al., 2017) al añadir un módulo de decodificación (Decoder). Este módulo Decoder permite integrar mejor las características de bajo y alto nivel, lo que incrementa la precisión en la definición de los bordes de segmentación. La estructura de DeepLabv3+ se representa en la 2.24 y sigue un enfoque global de codificador-decodificador, donde el componente de decodificación es una versión mejorada del DeepLabv3 original, mientras que la parte de codificación es una sección diseñada recientemente. Dentro de la arquitectura del codificador, la estructura principal se basa en Redes Neuronales Convolucionales Profundas (DCNN) que incorporan la convolución dilatada y el componente de Agrupamiento Piramidal Espacial Atrous (ASPP) como elementos clave (Chen et al., 2017). En esta estructura, la red troncal puede utilizar diferentes arquitecturas, como ResNet, MobileNet y Xception. ResNet, en particular, incluye un bloque de aprendizaje residual que ayuda a abordar el problema de la saturación de precisión y la degradación de rendimiento cuando la profundidad de la red aumenta. Además, investigaciones previas han demostrado que la red ResNet ofrece mejores resultados en segmentación semántica en comparación con otras arquitecturas (Lin et al., 2023).

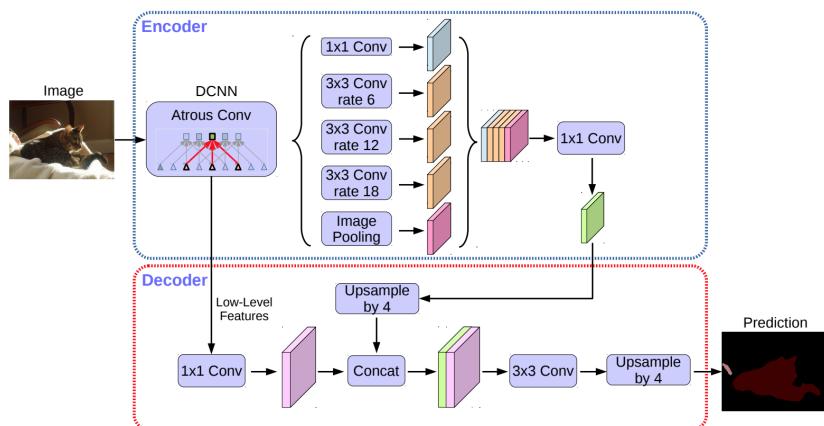


Figura 2.24: Arquitectura profunda DeepLabV3+.
Fuente: (Chen et al., 2018)

Atrous convolution: Es una herramienta poderosa que permite controlar la resolución de

las características calculadas por redes neuronales convolucionales profundas y ajustar el campo de visión del filtro para capturar información a múltiples escalas, ejecuta muchas convoluciones con varias tasas de dilatación siguiiendo el codificador, las salidas concatenadas y procesadas de varias convoluciones atrofias se utilizan para crear características ricas en contexto.

Al recopilar datos de diversas escalas y puntos de vista, el módulo ASPP mejora la comprensión de la red de los elementos en una escena. Es especialmente útil para superar los desafíos que presentan los elementos con diferentes tamaños y distribuciones espaciales.

Capítulo 3

Diseño del sistema de segmentación de cuerpos glaciares

Esta sección presenta el diseño y la implementación de un sistema para la clasificación y segmentación de cuerpos glaciares a partir de imágenes satelitales. El sistema permite analizar el retroceso de la superficie glaciar y, finalmente, comparar los resultados de la estimación del retroceso del glaciar Quelccaya con estudios previos.

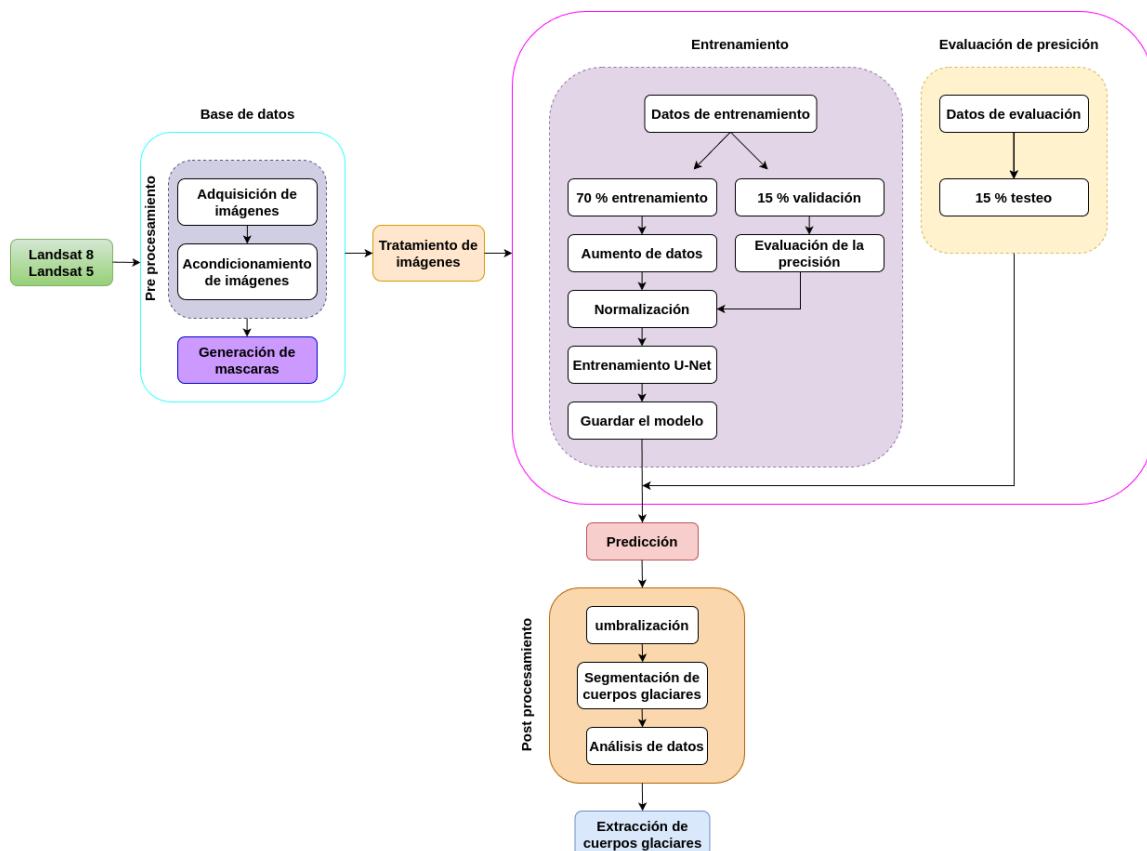


Figura 3.1: Flujo de trabajo en este estudio.

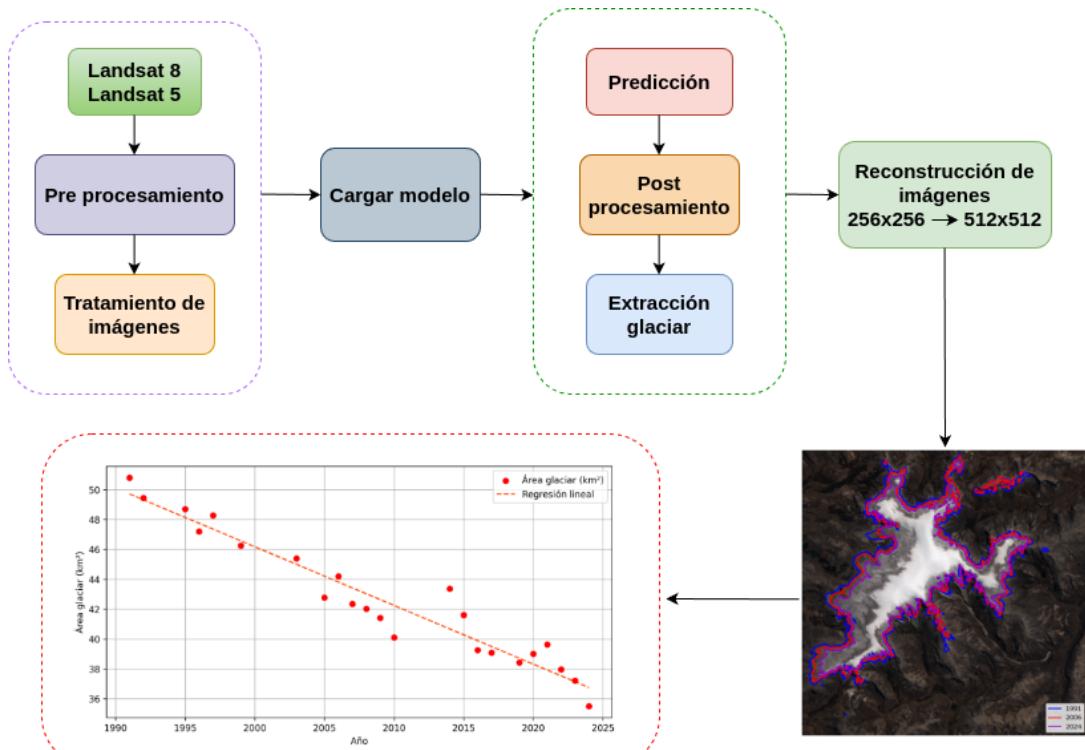


Figura 3.2: Procesamiento de datos para el análisis temporal.

3.1. Base de datos

Para implementar un sistema inteligente basado en deep learning que permita clasificar y segmentar cuerpos glaciares, es fundamental disponer de una base de datos adecuada. En este caso, se creó una base de datos utilizando imágenes del satélite Landsat 5 y Landsat 8 Collection 2 Level-1, obtenidas a través del Earth Explorer del U.S. Geological Survey (USGS) <https://earthexplorer.usgs.gov/>.

Según parte de la documentación oficial de USGS (Survey, 2024), se recomienda el uso de datos de Collection 2 Level-1 para análisis temporal, debido a que estos datos se someten a correcciones atmosféricas, lo cual permite mediciones más precisas de la reflectancia.

Los productos de Landsat Level-1 consisten en números digitales (DN) calibrados y escalados que representan los datos de imágenes multiespectrales. Los datos de Nivel 1 se clasifican en una estructura basada en niveles. Los datos de Nivel 1 de nivel más alto se calibran radiométricamente y se corrigen geométricamente utilizando puntos de control terrestre (GCP)

y datos de modelos de elevación digitales (DEM) para corregir el desplazamiento del relieve.

3.1.1. Adquisición de imágenes

Para construir una base de datos a partir de imágenes satelitales Landsat, este estudio utiliza datos ópticos multiespectrales obtenidos de satélites. Se accedió a estos datos a través de la plataforma de acceso abierto del Servicio Geológico de los Estados Unidos (USGS). La política de datos abiertos y gratuitos de USGS Landsat se ha mantenido constante desde su implementación. Los datos de la Colección 2 de Landsat, en formato Level 1, están disponibles para su descarga a través de EarthExplorer. Solo es necesario visitar la página web de acceso a datos de Landsat para aprender cómo buscar y descargar todos los productos disponibles desde los portales de datos de USGS.

Se seleccionaron escenas 001/071, 002/070, 003/070, 004/069, 007/067, 008/066 capturadas por los sensores TM de landsat 5 desde el año 1990 hasta el año 2011, y del sensor OLI de Landsat 8 desde el año 2013 hasta el 2024. Cada escena tiene una resolución de 7721x7591 para Landsat 5 y una resolución de 7800 × 7600 píxeles para Landsat 8. Se consideraron los siguientes criterios de selección: todas las escenas debían haber sido adquiridas durante la estación seca, entre junio y septiembre, para minimizar posibles errores de clasificación debido a la presencia de nieve temporal. Además, las escenas debían tener un porcentaje de nubosidad inferior al 10 %. Las escenas seleccionadas incluyen las Cordilleras Blanca, Huallanca, Huayhuash, Raura, Huagoruncho, La Viuda, Central, Huaytapallana, Chonta, Ampato, Urubamba, Vilcanota, Huanzo, Chila, La Raya, Vilcabamba, Carabaya, Apolobamba y la Cordillera Oriental en la Paz, Bolivia.

3.1.2. Acondicionamiento de las imágenes satelitales

Al realizar la descarga de la escena seleccionada, se logran descargar las bandas 1, 2, 3, 4, 5 y 7 de Landsat 5 y las bandas 2,3,4,5,6,7 para Landsat 8, para ello se usa la metodología de

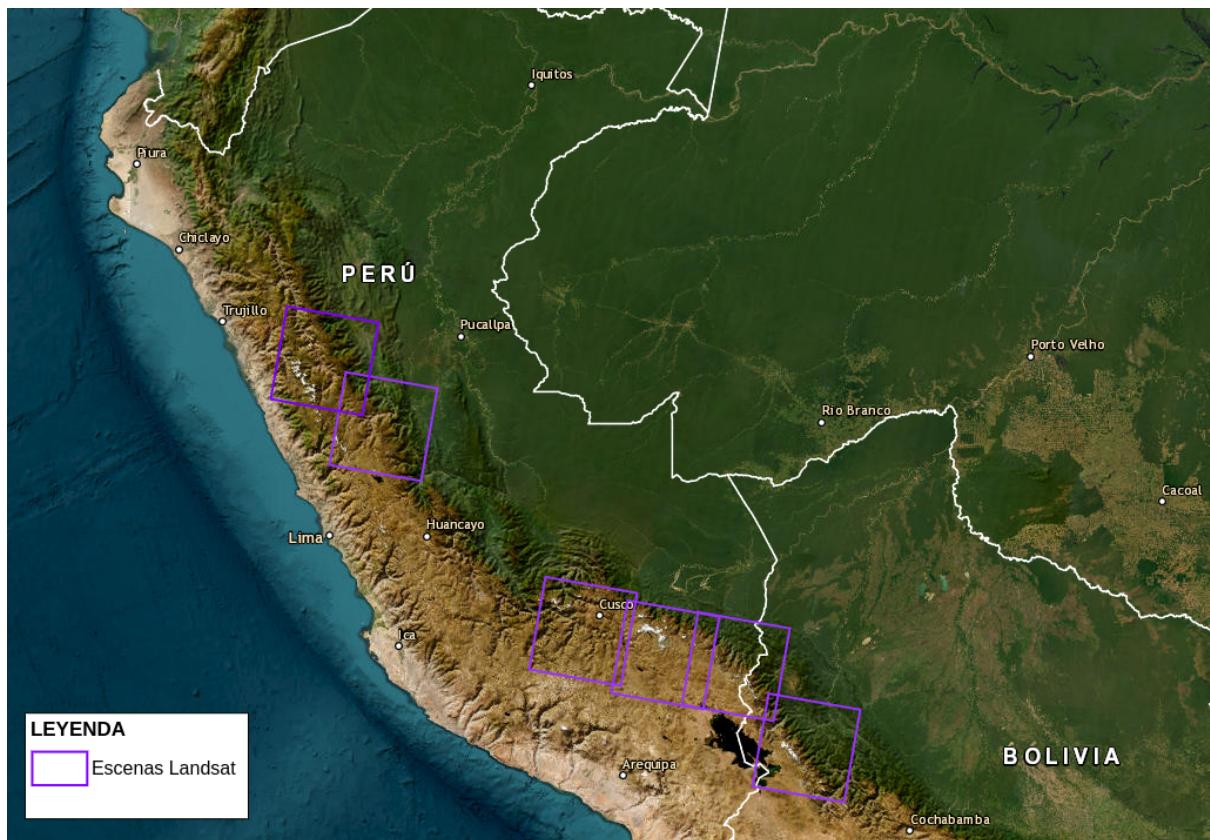


Figura 3.3: Escenas de Landsat seleccionadas para el conjunto de datos.

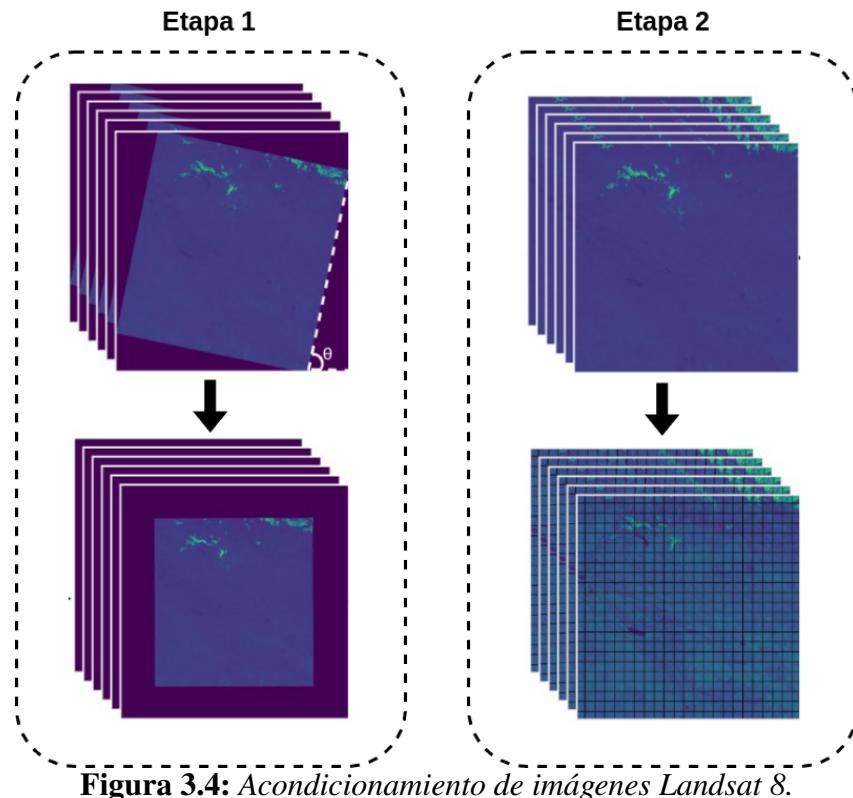
(Bezerra et al., 2023), (Perez-Torres et al., 2024).

Primeramente es necesario convertir cada banda de DN a Reflectancia TOA, posterior a ello fue necesario apilar estas 6 bandas para tener una sola imagen multiespectral de 6 canales.

Para corregir la orientación de las imágenes multiespectrales de Landsat 6 de seis canales, se determina el ángulo de inclinación de la imagen utilizando la transformada de Hough. Esta técnica es eficaz para identificar conjuntos de puntos colineales en una imagen, mapeando dichos puntos a un conjunto de parámetros (θ, ρ) que representan una línea en el espacio de parámetros. La búsqueda se restringe a un rango de $[0, 180^\circ]$ para θ y de $[-R, R]$ para ρ , lo cual permite definir un rectángulo que abarca la región de interés para cuantificar. Con el ángulo de inclinación θ establecido, se rota la imagen de satélite según este valor. Luego, el algoritmo de detección de bordes de Canny se aplica para distinguir los bordes relevantes de los irrelevantos. Con base en estos bordes detectados, se eliminan las áreas sin datos y se divide la imagen en parches de 256×256 píxeles, lo que reduce el costo computacional durante el entrenamiento. Este proceso se muestra en la Figura 3.4.

Para las imágenes de Landsat 5 no es posible corregir la orientación utilizando la metodología anterior, debido a que los bordes de estas imágenes no son uniformes. Por este motivo, las imágenes se dividieron en parches de 256x256 píxeles manteniendo la misma posición de la imagen.

Finalmente las imágenes multiespectrales tienen una forma de 6x256x256 (numero de canales=6, ancho = 256, alto = 256), un tamaño adecuado para incluirlo en la base de datos.



3.1.3. Generación de máscaras

Las máscaras de cada imagen multiespectral solamente tienen que incluir cuerpos glaciares, para ello se utilizó el Índice de Diferencia de Nieve Normalizado (NDSI, por sus siglas en inglés) con el objetivo de segmentar los cuerpos glaciares. Sin embargo, durante este proceso, algunos lagos fueron erróneamente clasificados como áreas de cobertura glaciar debido a sus características espectrales similares a la nieve y el hielo como se muestra en la Figura 3.5b. Para corregir este problema, se empleó el Índice de Diferencia de Agua Normalizado (NDWI, por sus siglas en inglés), el cual es más adecuado para la identificación de cuerpos de agua.

Posteriormente, se calculó la diferencia entre los valores del NDSI y el NDWI para mejorar la precisión de la segmentación. Este enfoque ayudó a reducir la confusión espectral entre los lagos y las zonas glaciares, atenuando la clasificación incorrecta de algunos píxeles que, debido a su similitud espectral, habían sido clasificados erróneamente como glaciares en el resultado original del NDSI. Para eliminar completamente estos píxeles mal clasificados, se estableció un umbral de 0.67, que permitió diferenciar los píxeles correspondientes a cuerpos de agua de aquellos clasificados como glaciares.

No obstante, este método también provocó la eliminación de algunos píxeles que, en realidad, pertenecían a áreas glaciares. Por lo tanto, fue necesario realizar una corrección manual en las imágenes segmentadas para garantizar la precisión de la clasificación final. Para ello se binarizó la imagen segmentada, posteriormente se corrigieron las imágenes binarias empleando Google Earth para verificar la segmentación y las correcciones aplicadas, asegurando que el resultado final fuera lo más fiel posible a la realidad. Este proceso se ilustra en la Figura 3.6, donde se puede observar la mejora en la identificación de las áreas glaciares y de los cuerpos de agua tras la aplicación de ambos índices y las correcciones manuales correspondientes.

Finalmente se obtuvieron máscaras que tienen una forma de 1x256x256 (numero de canales=1, ancho = 256, alto = 256), para Landsat 8 y Landsat 5, un tamaño adecuado para incluirlo en la base de datos.

3.1.4. Tratamiento de imágenes

Las imágenes multiespectrales y sus respectivas máscaras para el conjunto de datos solamente deben incluir áreas glaciares, cada pixel de la imagen multiespectral para Landsat 8 es de 16 bits sin signo (UINT16) y 8 bits sin signo (UINT8) sin signo para Landsat 5, lo que implica que los valores de los píxeles en estas bandas pueden variar desde 0 hasta 65535 y 0 hasta 255 respectivamente. Cuando se procesan imágenes, especialmente para análisis avanzados como algoritmos de deep learning o procesamiento digital de señales, los datos se convierten a menudo a float64 o float32 para facilitar las operaciones matemáticas. Convertir

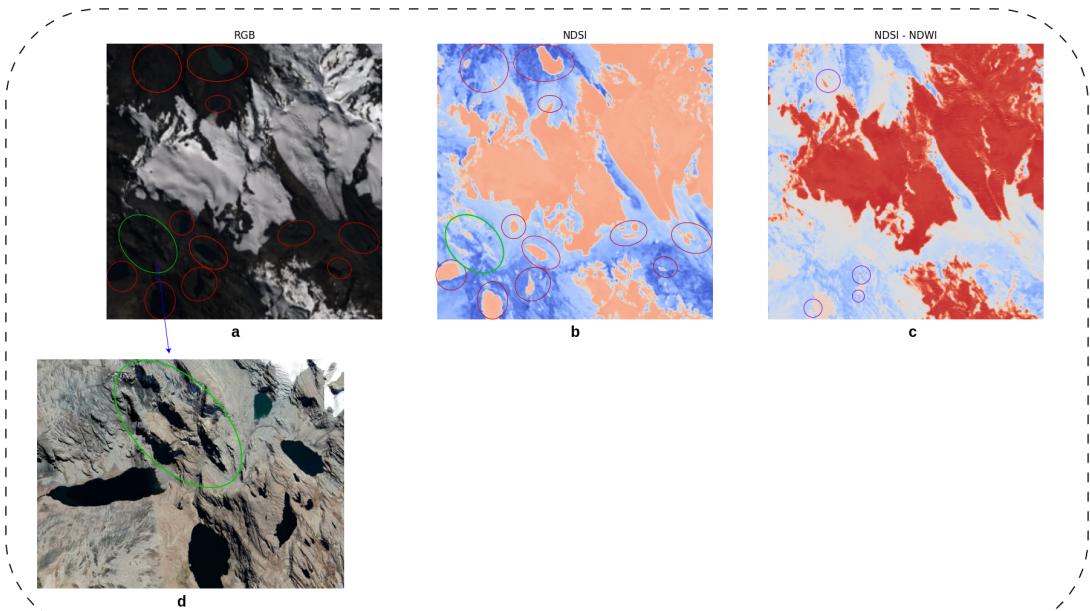


Figura 3.5: Corrección de agua y sombras. a) Imagen RGB. b) Índice NDSI, se observa que los cuerpos de agua también son segmentados. c) NDSI-NDWI. d) Visualización de sombras. El área circular roja encierra lagos, el verde encierra sombras y el morado encierra píxeles que aún tienen alto valor y necesitan ser eliminados.

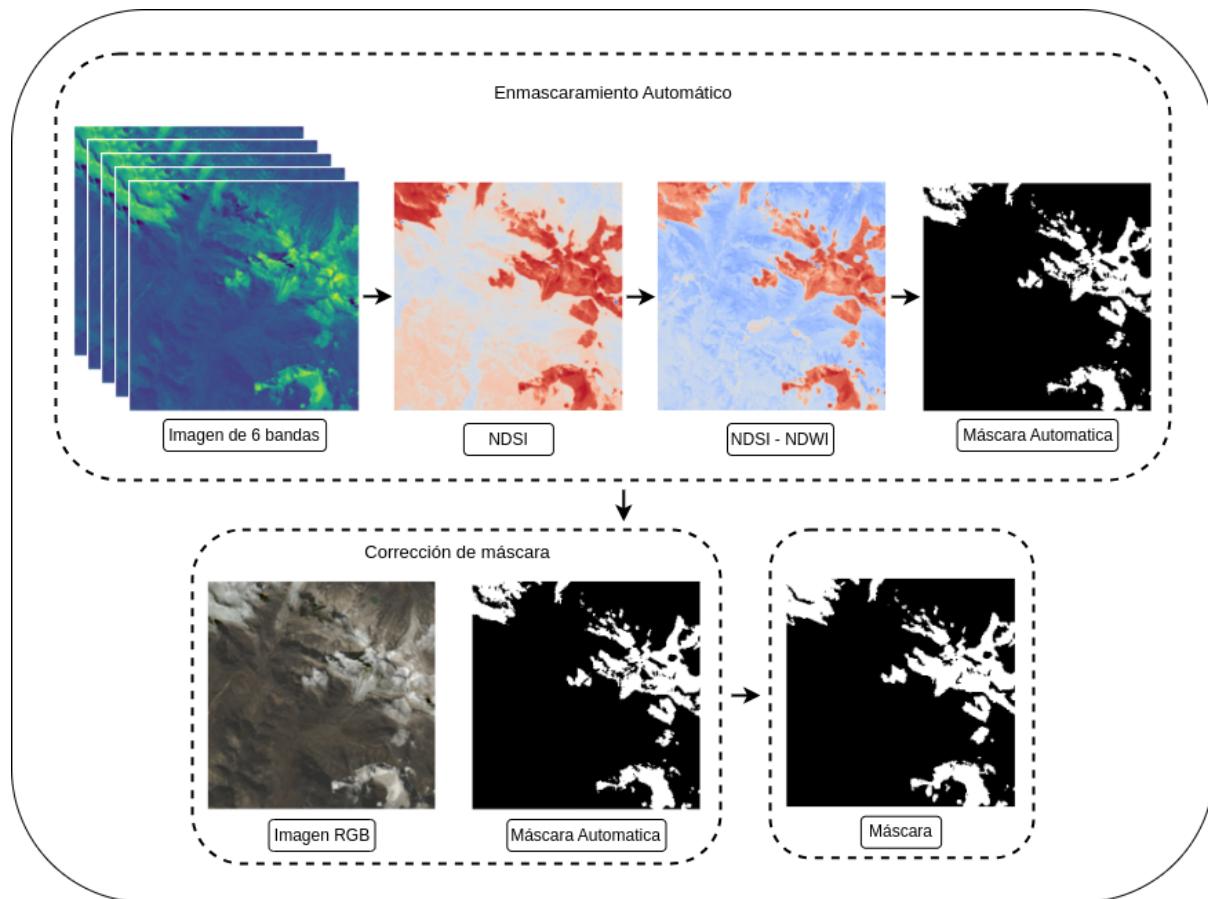


Figura 3.6: Proceso de creación de máscaras.

los datos a un formato de punto flotante permite realizar calculos mas precisos y evitar errores de redondeo que pueden ocurrir con los datos de menor presición.

Una práctica estándar en el procesamiento de imágenes y aprendisaje automático es normalizar los datos de imágenes a [0, 1] puesto que se mejora la eficiencia y efectividad del entrenamiento del modelo, garantiza la comparabilidad entre diferentes conjuntos de datos, evita problemas numéricos, y asegura que los modelos de machine learning y deep learning funcionen de manera óptima.

3.2. Arquitecturas de aprendizaje profundo (Deep learning)

3.2.1. Implementación de la arquitectura de deep learning

Se propuso 3 arquitecturas de segmentación semántica (U-Net, DeepResUNet y DeepLabV3Plus), para ello se implementó una red neuronal para cada modelo basandose en su arquitectura, dichas arquitecturas se muestran en las Figuras 3.7, 3.8 y 3.10. El tamaño de la imagen de entrada para cada arquitectura fue de 256x256 px.

El progreso de la segmentación semántica podría definirse como una tarea de clasificación a nivel de pixel para clasificar esos píxeles en cuerpos glaciares u otras clases. Las imágenes de los cuerpos glaciares segmentados con la máscara se compararon para minimizar la diferencia entre ellas durante el entrenamiento utilizando la pérdida de Dice. Se utilizaron Dice Coficient, mean Intersegtion Union (mIoU) y Pixel Accuracy (PA) como métricas.

Donde [0,1] representa el valor de la probabilidad predicha alcanzada por el modelo para la clase de verdad fundamental con etiqueta = 1.

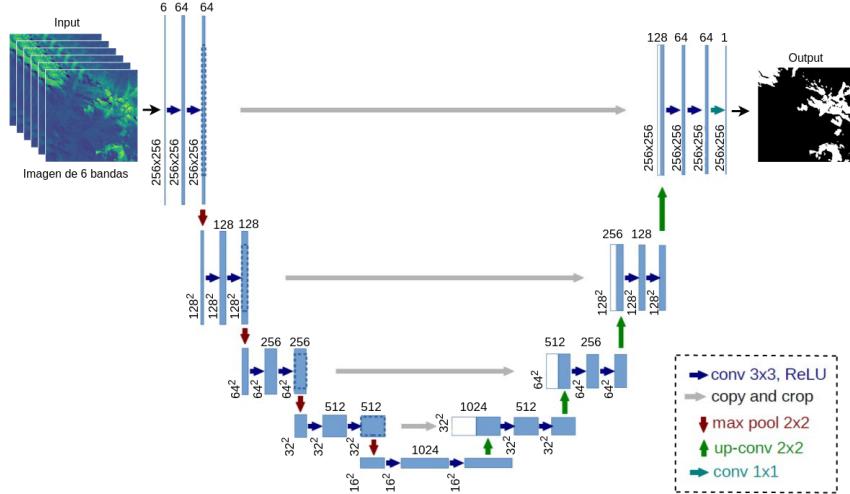


Figura 3.7: Arquitectura del modelo U-Net.

Función de pérdida

La función de pérdida utilizada para entrenar los modelos es una combinación ponderada de DICE Loss y BCE Loss (Binary Cross Entropy Loss), ya que ha demostrado obtener buenos resultados según (**montazerolghaem2023**). La razón detrás de combinar pérdidas como DICE Loss y BCE Loss es que cada una de ellas tiene sus propias ventajas y desventajas.

DICE Loss: Esta pérdida es especialmente útil para tareas de segmentación donde el tamaño de la región objetivo puede ser pequeño en comparación con el fondo. Maximiza el solapamiento entre la predicción y la verdad del terreno, lo que ayuda a lidiar con el desequilibrio de clases, es decir, cuando hay muchos más píxeles de fondo que de objeto en la imagen.

BCE Loss: Esta pérdida es una opción clásica para problemas de clasificación binaria, incluido cada píxel en una tarea de segmentación binaria. BCE mide la disimilitud entre las predicciones y las etiquetas verdaderas para cada píxel individualmente. Sin embargo, puede ser sensible al desequilibrio de clases.

Combinar ambas pérdidas permite aprovechar sus fortalezas individuales: BCE Loss puede proporcionar una señal de retroalimentación estable para píxeles individuales, mientras que DICE Loss puede ayudar a mejorar la predicción de los bordes del objeto y a manejar mejor el desequilibrio de clases.

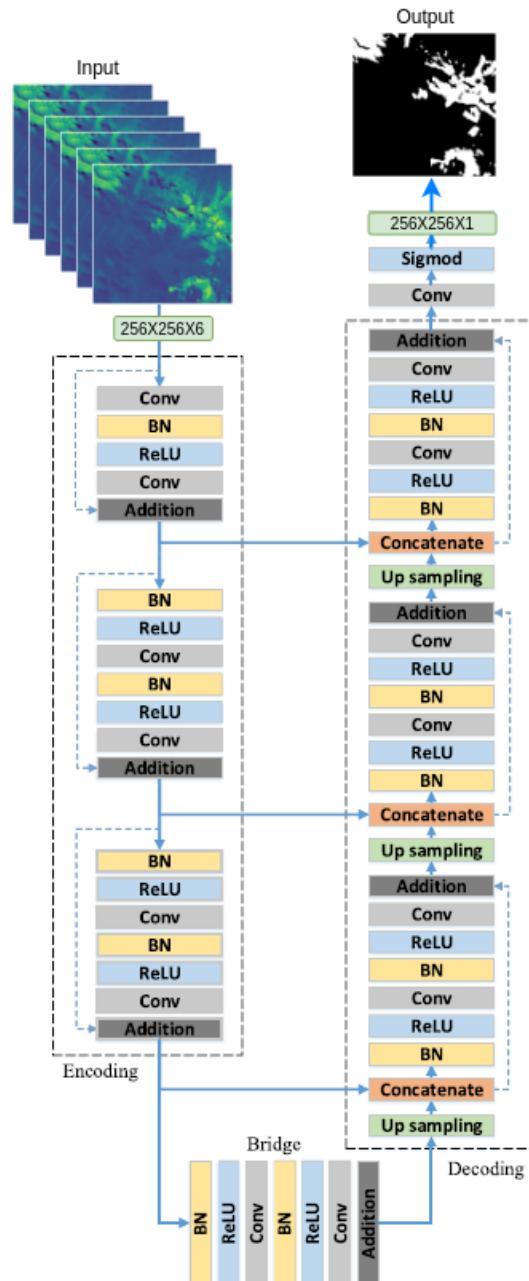


Figura 3.8: Arquitectura del modelo DeepResUnet.

Definición Matemática de la Combinación Ponderada

En la literatura, la combinación ponderada de funciones de pérdida se define típicamente como:

$$\text{Loss} = \alpha \cdot \text{DICE Loss} + \beta \cdot \text{BCE Loss} \quad (3.1)$$

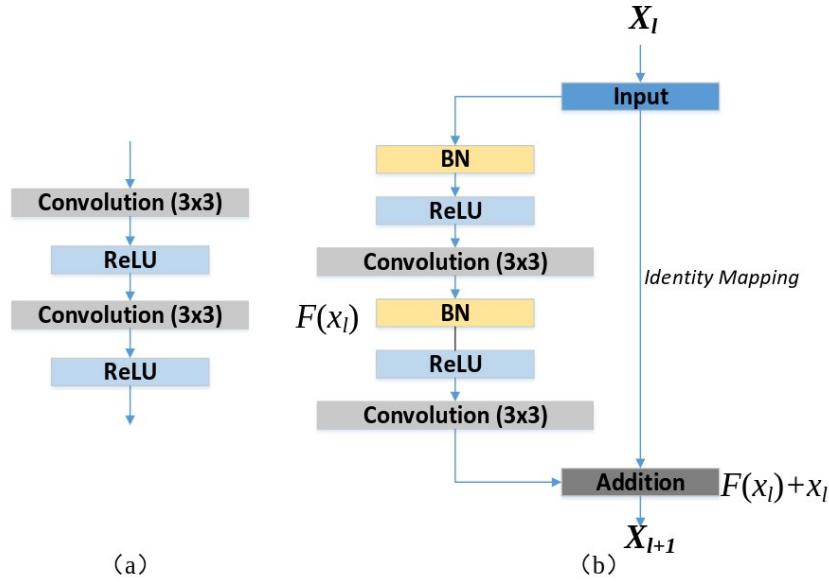


Figura 3.9: Bloques fundamentales de las redes neuronales. (a) Unidad neuronal simple utilizada en U-Net y (b) unidad residual con mapeo de identidad utilizada en el ResUnet propuesto.

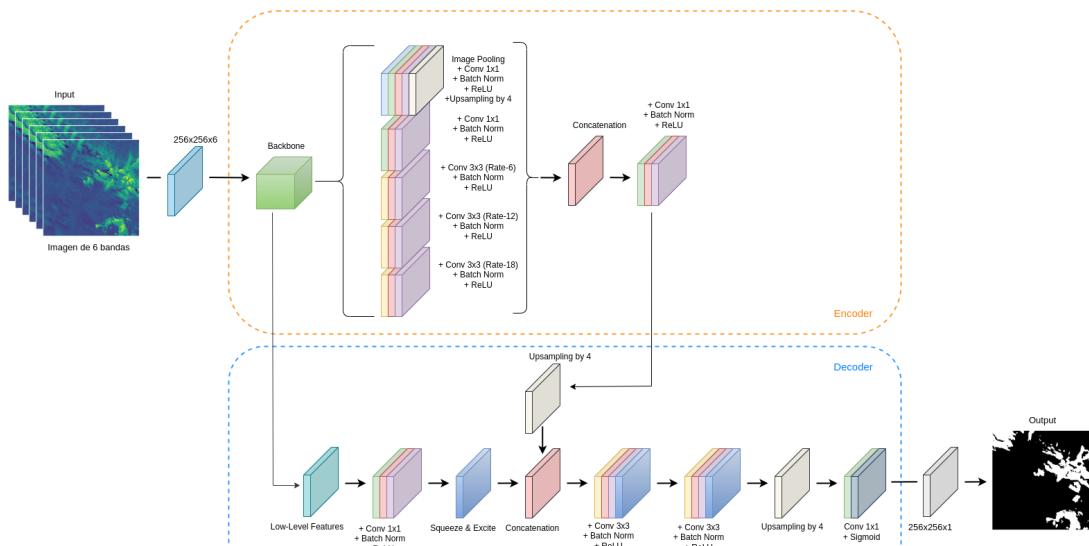


Figura 3.10: Arquitectura del modelo DeepLabV3Plus.

Donde:

α y β son hiperparámetros que determinan el peso de cada término de pérdida en la combinación total. Generalmente $\alpha + \beta = 1$, pero no necesariamente. Se pueden ajustar estos parámetros según cómo se requiera influir en cada término en el entrenamiento.

Métricas de evaluación

Para evaluar los modelos de segmentación semántica generados por la CNN, se utilizaron diversas métricas de evaluación: MIoU (Mean Intersection Over Union), PA (Pixel Accuracy) y Dice Coefficient (Dice).

El MIoU, definido en la ecuación 3.2, se utiliza como métrica estándar en aplicaciones de segmentación para medir la superposición promedio entre las áreas predichas y las áreas reales de todas las clases.

La PA, como se muestra en la ecuación 3.3, mide la proporción de píxeles correctamente clasificados en la imagen. Se calcula dividiendo el número de píxeles correctamente clasificados por el número total de píxeles en la imagen, proporcionando así una medida general de la precisión del modelo.

El Dice Coefficient, definido en la ecuación 3.4, evalúa la similitud entre dos conjuntos, permitiendo comparar la superposición entre una máscara de predicción y una máscara de referencia. Su valor varía entre 0 (sin coincidencias) y 1 (coincidencia perfecta), ofreciendo una medida de la precisión en la segmentación de áreas de interés.

En estas métricas, TP representa verdaderos positivos, TN indica verdaderos negativos, FP representa falsos positivos y FN señala falsos negativos.

$$\text{MIoU} = \frac{1}{k+1} \sum_{i=0}^N \frac{TP}{TP + FP + FN} \quad (3.2)$$

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

$$\text{Dice Coefficient} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3.4)$$

Proceso de entrenamiento

La plataforma computacional utilizada para implementar todos los modelos de segmentación semántica, incluyendo U-Net, ResUnet y DeepLabV3+, fue Python 3.11. Todos los marcos de aprendizaje profundo se desarrollaron utilizando el framework PyTorch 2.2.2, ejecutado sobre un sistema operativo Manjaro KDE/Linux. El hardware utilizado incluyó un procesador Intel Xeon y una GPU NVIDIA Quadro P2000 con 5 GB de memoria.

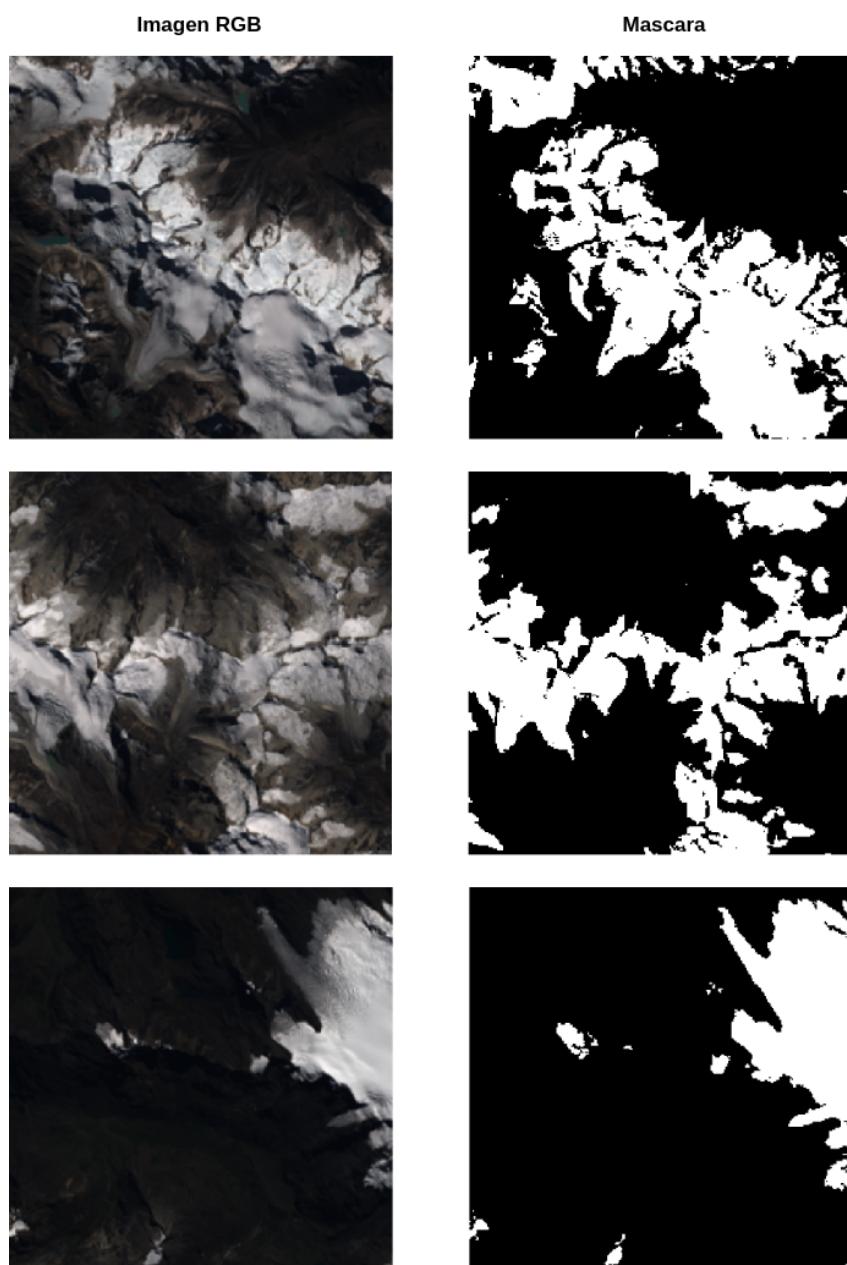


Figura 3.11: Representación de muestras de conjuntos de datos segmentados de diferentes ubicaciones (RGB y su respectiva máscara) utilizadas para entrenar los modelos propuestos.

Durante esta etapa, se llevó a cabo el entrenamiento de las tres arquitecturas de segmentación semántica propuestas. Se realizaron numerosos experimentos para identificar los parámetros óptimos de entrenamiento, tales como el optimizador, el número de épocas, la tasa de aprendizaje, el tamaño del batch, el número de trabajadores (num workers) y el factor de decaimiento de los pesos (weight decay). La selección cuidadosa de estos parámetros fue esencial para garantizar que los modelos convergieran de manera eficiente y alcanzaran los mejores pesos posibles. Esta optimización no solo maximiza el rendimiento de la red neuronal, sino que también mejora su capacidad para generalizar de manera efectiva a través de diferentes conjuntos de datos. Además, se evaluó el impacto de estos parámetros en la precisión y robustez de los modelos, asegurando que el proceso de entrenamiento resultara en una segmentación precisa y confiable.

Siguiendo las recomendaciones de la documentación de PyTorch (PyTorch, 2024), se utilizó el método ReduceLROnPlateau para ajustar dinámicamente la tasa de aprendizaje cuando una métrica de rendimiento dejaba de mejorar. Este enfoque es particularmente beneficioso, ya que la reducción de la tasa de aprendizaje en un factor de 2 a 10 puede ayudar a superar estancamientos durante el entrenamiento. En este contexto, el algoritmo monitoreó continuamente las métricas de evaluación y, si no se observaba ninguna mejora en un periodo de 5 épocas consecutivas, automáticamente reducía la tasa de aprendizaje. Esta estrategia permite que el modelo ajuste su proceso de aprendizaje de manera más refinada, optimizando así su capacidad para converger a una solución óptima.

Finalmente, para contar con un amplio conjunto de datos se realizó un data augmentation para evitar el sobreajuste del modelo, se aplicó rotaciones de 90° y reflejo de espejo horizontal y vertical a la imagen multiespectral de 6 canales y sus respectivas máscaras.

Modelo	Optimizer	Epochs	Learning rate	Batch size	num workers	weight decay
U-Net	Adam	25	1e-3	4	8	1e-2
ResUnet	Adam	25	1e-3	4	8	1e-2
DeepLabV3+	Adam	25	1e-4	4	8	1e-6

Tabla 3.1: Párametros de entrenamiento

3.2.2. Resultados de entrenamiento

Los modelos de redes neuronales convolucionales (CNN) propuestos fueron entrenados durante 25 épocas, guiados por los gráficos de precisión y pérdida tanto en el entrenamiento como en la validación, como se ilustra en las figuras 3.12, 3.13 y 3.14. Para asegurar un rendimiento óptimo y estabilidad en el proceso de entrenamiento, se asumió que todos los modelos se entrenaron correctamente siguiendo los valores de hiperparámetros óptimos, los cuales se encuentran detallados en la tabla 3.1. Estos valores de hiperparámetros fueron seleccionados después de entrenar múltiples veces los modelos con diferentes configuraciones, con el objetivo de encontrar la combinación que ofreciera el mejor desempeño y estabilidad.

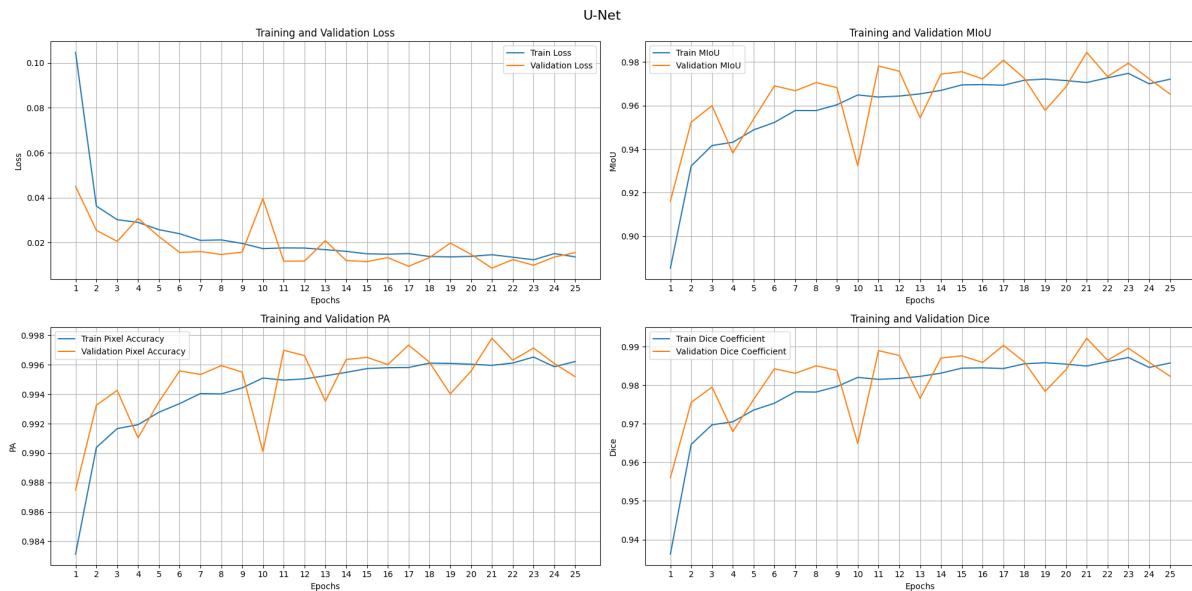


Figura 3.12: Gráficos de pérdida de entrenamiento y métricas de precisión para U-NET.

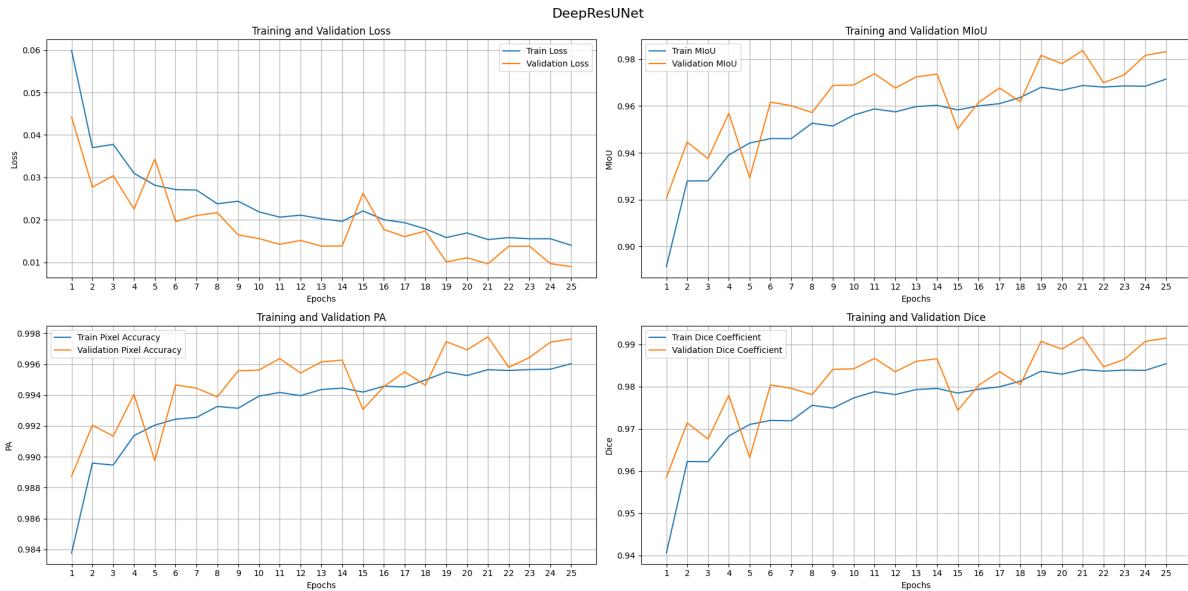


Figura 3.13: Gráficos de pérdida de entrenamiento y métricas de precisión para RESUNET.

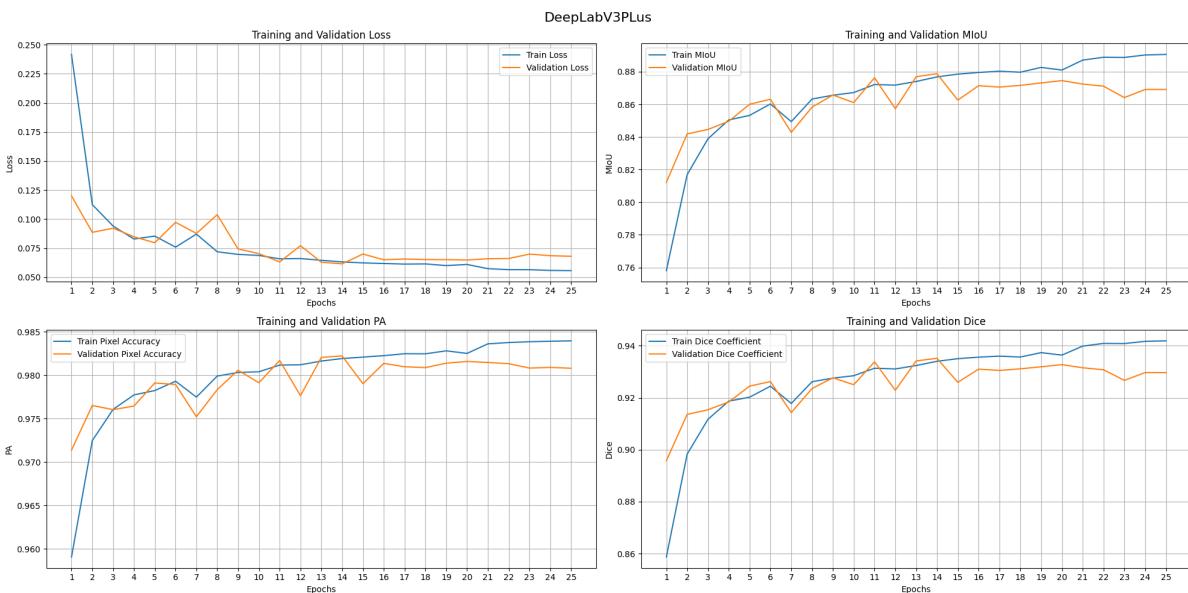


Figura 3.14: Gráficos de pérdida de entrenamiento y métricas de precisión para DeepLabV3Plus.

Capítulo 4

Experimentos y resultados

4.1. Experimentos

Una vez entrenados, los modelos fueron evaluados utilizando un conjunto de datos de prueba para medir su rendimiento en términos de las métricas descritas en las ecuaciones 3.2 - 3.4. Esta evaluación permitió comprobar la efectividad de los modelos propuestos, asegurando que los resultados obtenidos fueran representativos de su capacidad para generalizar y segmentar adecuadamente en diferentes escenarios.

En las Figuras 4.1, 4.2 y 4.3, se muestran los resultados de las métricas MIoU (Mean Intersection over Union), PA (Pixel Accuracy) y Dice Coefficient (Dice) evaluadas para cada modelo CNN utilizando los datos de evaluación.

Los modelos fueron evaluados utilizando un conjunto de datos de prueba compuesto por 360 imágenes, aplicando métricas como mIoU, Dice Coefficient, y PA para medir la precisión. Los resultados cuantitativos se presentan en la tabla 4.1.

Modelo	Loss	Miou	PA	Dice Coefficient
U-Net	0.0100	0.9810	0.9979	0.9904
ResUnet	0.0113	0.9782	0.9963	0.9882
DeepLabV3+	0.0704	0.8541	0.9824	0.9199

Tabla 4.1: Párametros de entrenamiento

Finalmente, U-Net demuestra un menor valor de pérdida y mayores puntajes en las

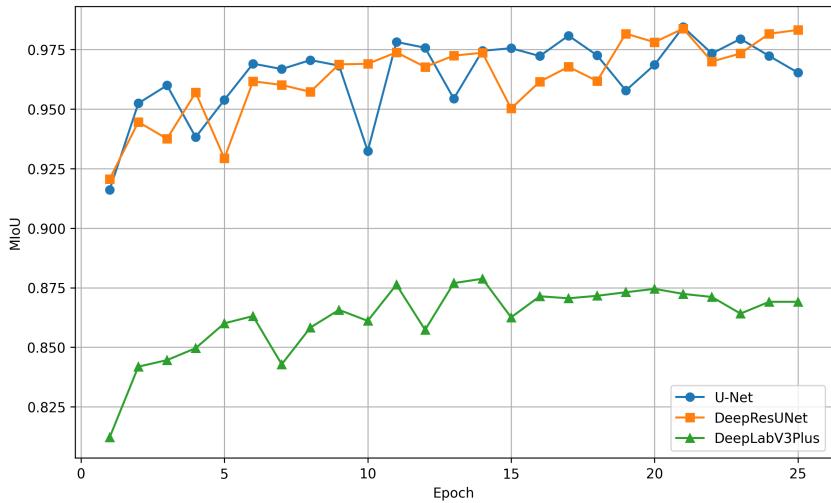


Figura 4.1: Resultado del indicador MIoU evaluado para cada modelo.

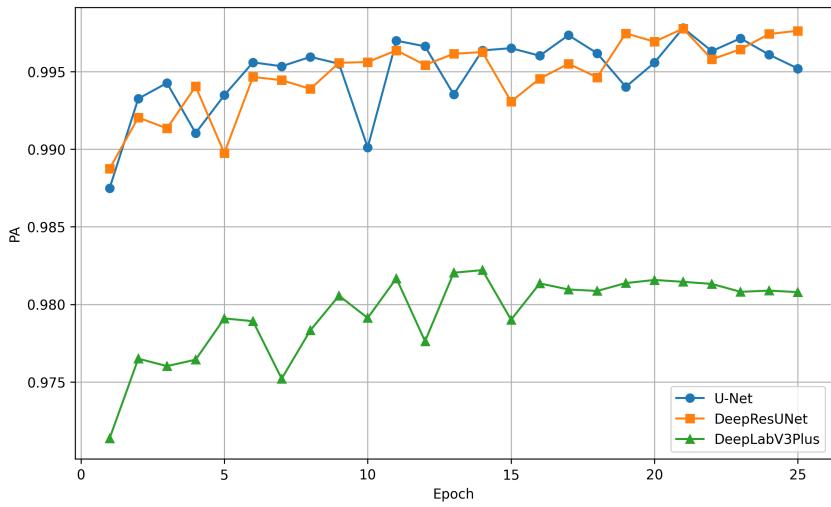


Figura 4.2: Resultado del indicador PA evaluado para cada modelo.

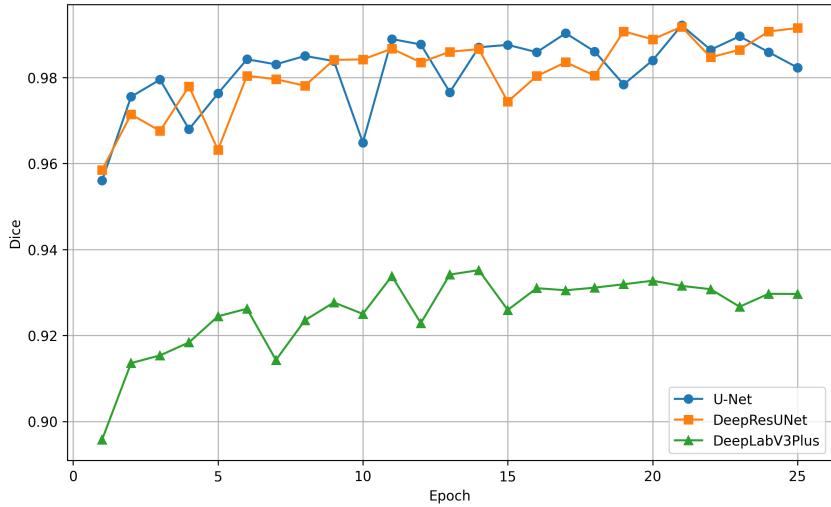


Figura 4.3: Resultado del indicador Dice evaluado para cada modelo.

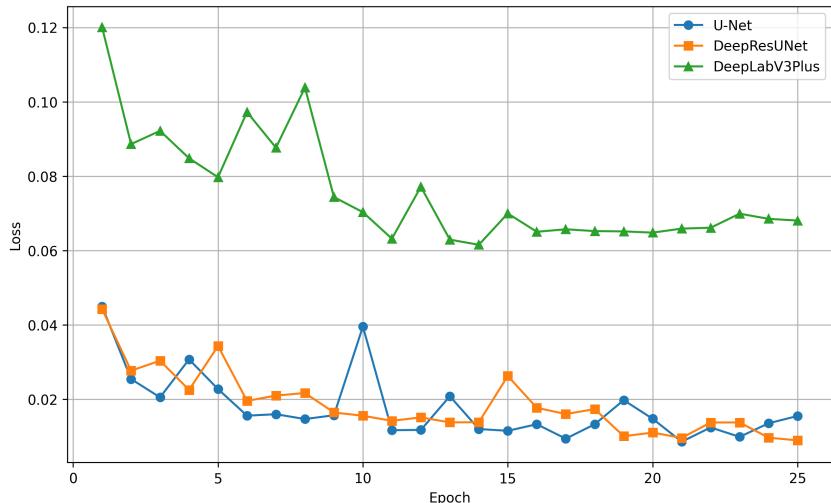


Figura 4.4: Resultado de convergencia de la función de pérdida evaluado para cada modelo.

métricas MIoU, PA y Dice Coefficient, consolidándose como el modelo más adecuado para tareas de segmentación semántica de cuerpos glaciares en el Perú.

La Figura 4.5 se muestra los resultados de segmentación obtenidos por los modelos propuestos.

El enfoque utilizado para el análisis temporal, con el fin de reducir la influencia de la nieve efímera, consistió en analizar múltiples escenas durante la estación seca y seleccionar aquellas que mejor representaran la extensión del glaciar. Para ello, se eligieron únicamente escenas sin nieve efímera visible en el paisaje circundante al glaciar, aplicando una técnica de preselección común.

4.1.1. Evolución del glaciar Quelccaya

Posteriormente, se creó un conjunto de datos que incluye únicamente las imágenes del glaciar Quelccaya, seleccionando las más representativas de los años de interés. Este dataset fue preprocesado y utilizado como entrada para la red neuronal U-Net, con el objetivo de generar imágenes predichas que luego se emplearían para calcular el área del glaciar en cada año.

Dado que la red neuronal acepta entradas de tamaño 256x256, fue necesario reconstruir las imágenes predichas a un tamaño de 512x512, adecuado para contener la totalidad del

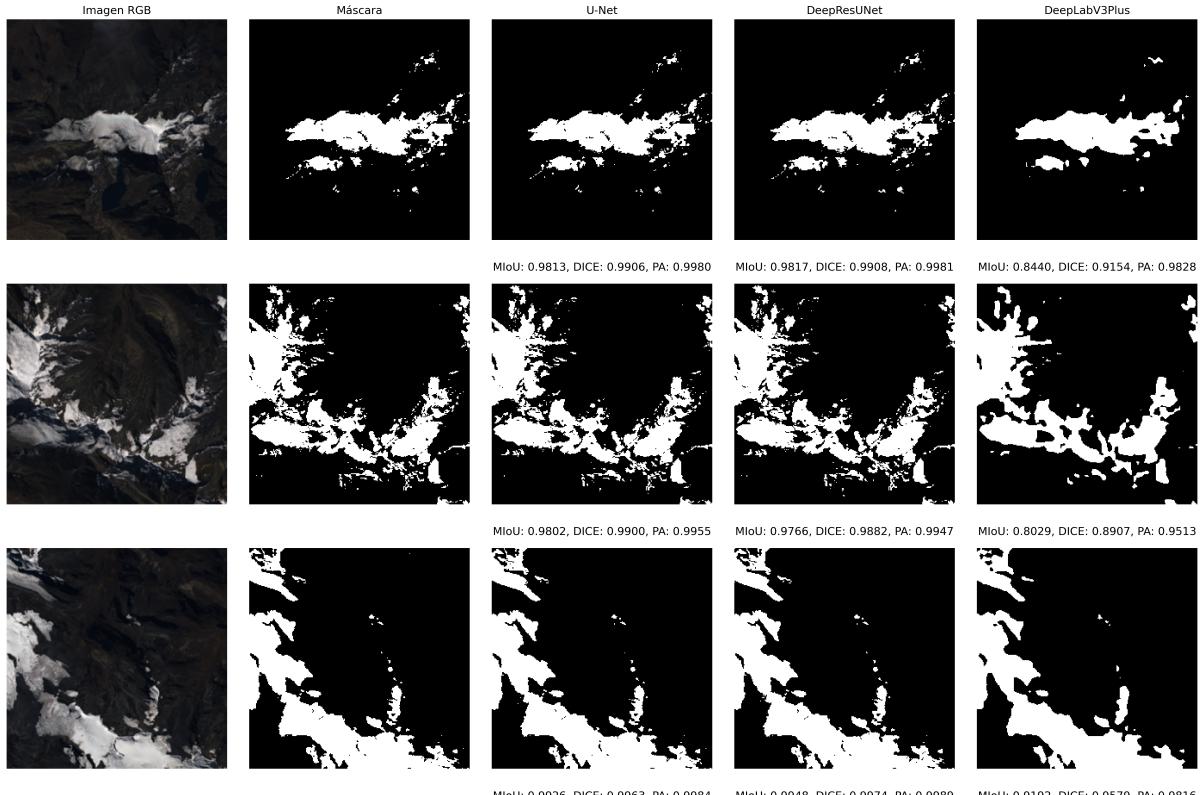


Figura 4.5: Cualitativamente, tanto U-Net como ResUnet muestran una gran similitud con las máscaras de referencia, logrando segmentar con precisión los bordes de los cuerpos glaciares. En contraste, aunque DeepLabV3Plus logra una segmentación aceptable, no alcanza el mismo nivel de precisión en la delimitación de los bordes. Cuantitativamente, U-Net muestra un rendimiento sobresaliente, mientras que DeepResUnet ofrece un rendimiento competitivo. En comparación, DeepLabV3Plus no alcanza un nivel competitivo frente a los otros modelos.

glaciar Quelccaya. Una vez reconstruida, la imagen resultante es completamente binaria, con un tamaño de 512x512, donde los píxeles blancos segmentados representan los cuerpos glaciares y los píxeles negros cualquier otra superficie.

Cada imagen binaria se procesa sumando los píxeles blancos, considerando que cada píxel tiene una resolución de 30 metros. De esta manera, se calcula el área del glaciar correspondiente a cada año. Este procedimiento se repite para los demás años de interés.

La Figura 4.7 muestra imágenes binarias que evidencian la reducción en el área del glaciar Quelccaya a lo largo del tiempo. En particular, se observa que algunos cuerpos glaciares disminuyeron drásticamente hasta desaparecer por completo, como se ilustra en la Figura 4.8 donde se observa que en la imagen de 1991, el recuadro rojo destaca la presencia de estos cuerpos glaciares; en 2010, aunque aún son visibles, su extensión es muy reducida. Para el año

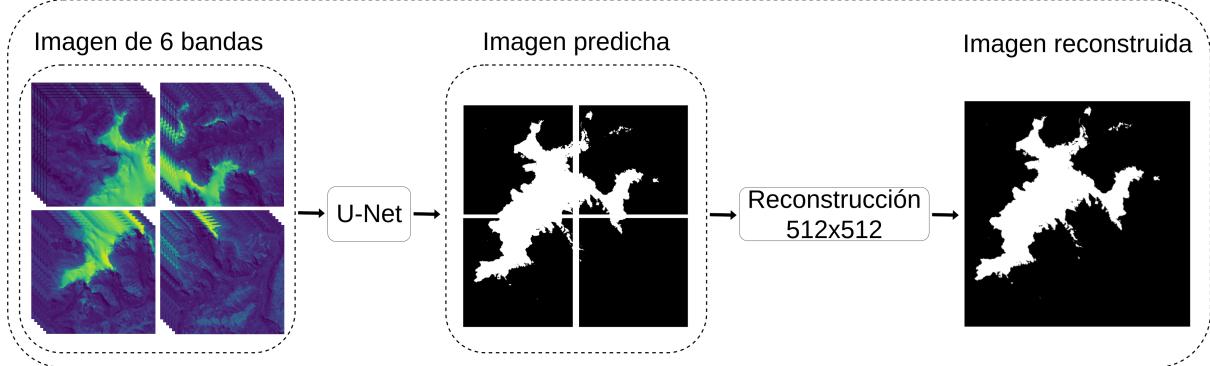


Figura 4.6: Proceso de predicción y reconstrucción para calcular el área segmentada.

2024, estos cuerpos glaciares han desaparecido completamente.

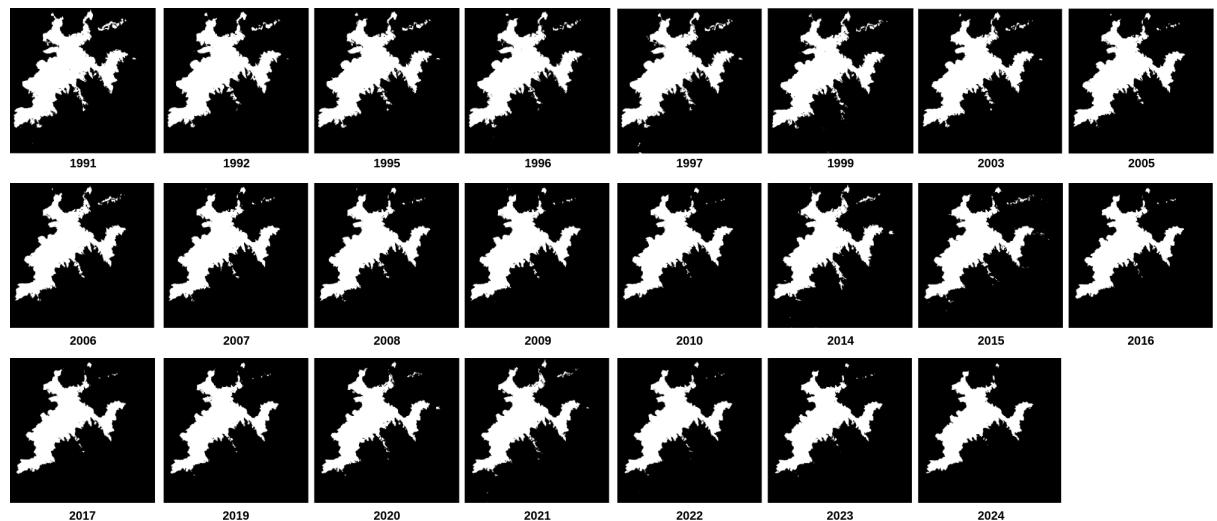


Figura 4.7: Retroceso superficial del glaciar Quelccaya visualizada mediante imágenes binarias segmentadas utilizando el modelo U-Net.

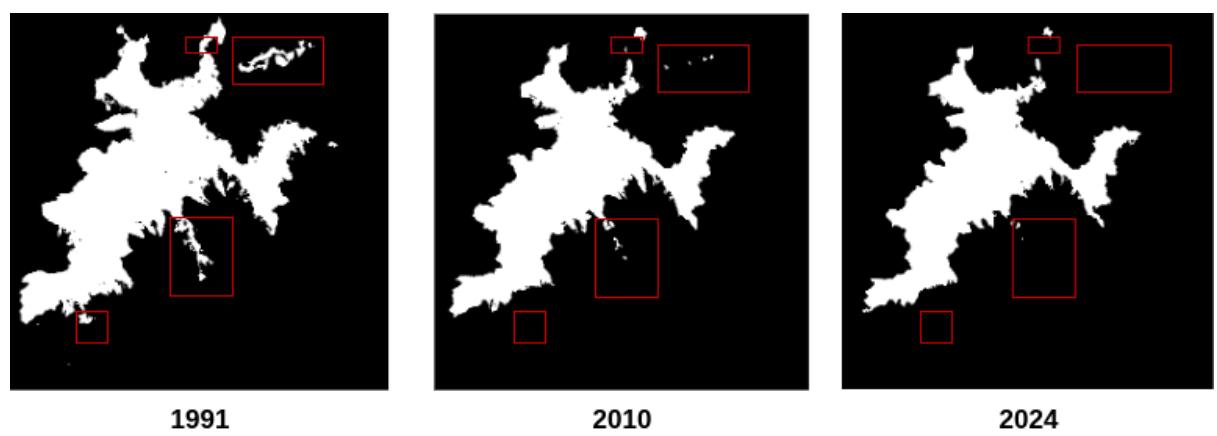


Figura 4.8: El recuadro rojo encierra cuerpos glaciares que se extinguieron gradualmente hasta llegar a desaparecer completamente.

En la figura 4.9, se visualizan los cambios en la superficie del glaciar Quelccaya para los años 1991, 2006 y 2024, con el propósito de proporcionar una mejor representación del

retroceso glaciar. En esta figura se muestran los bordes correspondientes a cada año: el borde azul indica la extensión del glaciar en 1991, el borde rojo en 2006 y el borde rosado en 2024.

Es importante mencionar que para delinear los bordes de los glaciares en distintos años se empleó el algoritmo de Canny, ampliamente utilizado para detectar todos los contornos presentes en una imagen. En este caso, el algoritmo identifica los bordes de los glaciares a partir de una imagen binaria, permitiendo observar los cambios en su extensión a lo largo del tiempo.

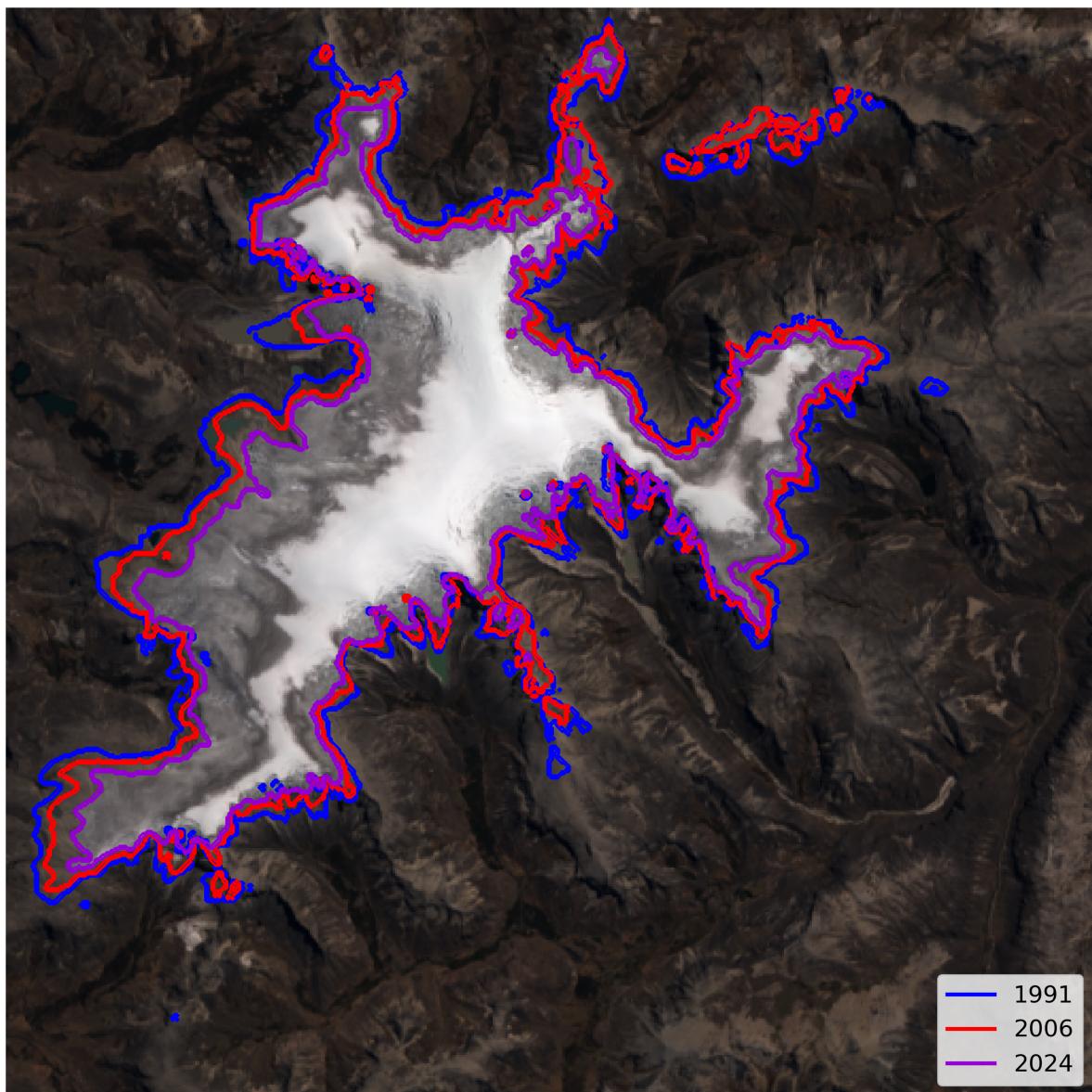


Figura 4.9: Contornos glaciares más antiguos y mas recientes del glaciar Quelccaya.

4.2. Resultados

Los resultados indican que la superficie del glaciar Quelccaya ha disminuido notablemente en un 30.15 % durante el periodo 1991-2024, pasando de 50.81 km² en 1991 a 35.49 km² en 2024. Esto representa una tasa de disminución de 0.38.99 km²/año, con todas las incertidumbres expresadas como intervalos de confianza del 93 %. En área superficial del año actual 2024 es una versión más pequeña y más restringida a su versión de 1991.

En el análisis se excluyeron algunos años, ya que se comprobó que no representaban con precisión la extensión de la capa de hielo. Estos años mostraban áreas anómalas, probablemente debido a la presencia de nieve efímera, una fuente potencialmente significativa de inexactitud al analizar escenas con esta cobertura temporal.

Año	Área glaciar (km ²)
1991	50.8131
1992	49.4514
1995	48.7134
1996	47.2077
1997	48.2877
1999	46.2609
2003	45.4023
2005	42.7689
2006	44.1945
2007	42.3459
2008	42.0192
2009	41.4171
2010	40.0995
2014	43.3746
2015	41.5998
2016	39.2517
2017	39.0825
2019	38.4210
2020	38.9988
2021	39.6297
2022	37.9539
2023	37.2051
2024	35.4951

Tabla 4.2: Resultado cuantitativo de la estimación de superficie glaciar a travez del tiempo

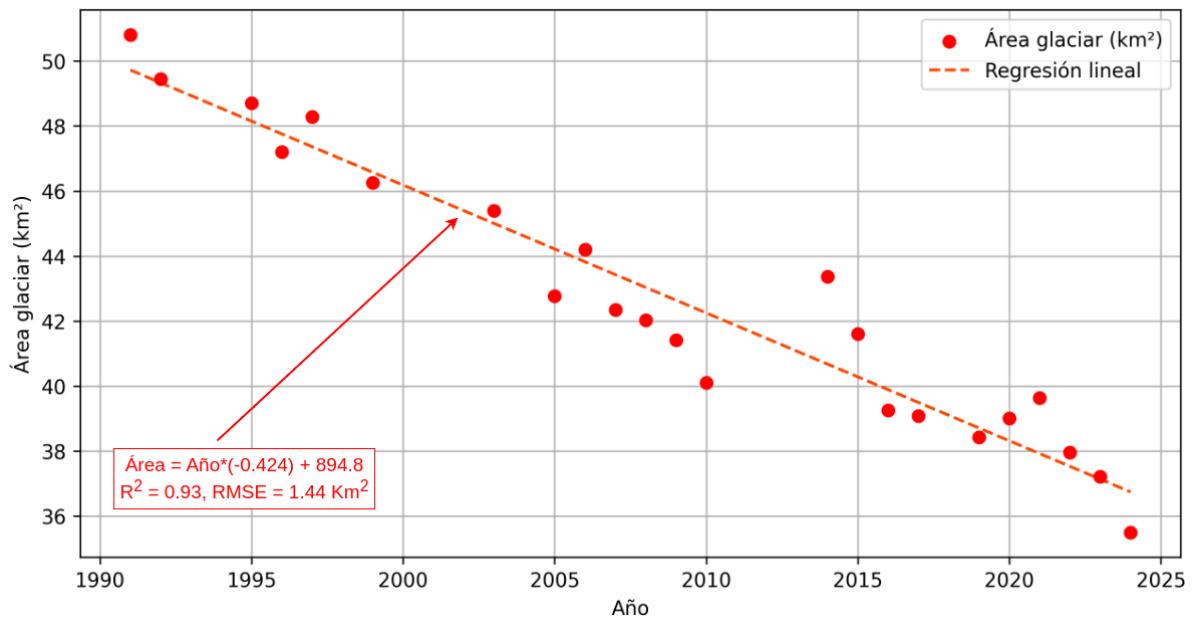


Figura 4.10: Análisis temporal del glaciar Quelccaya.

Capítulo 5

Comparación con trabajos previos

Los antecedentes sobre la estimación del área del glaciar Quelccaya son limitados, ya que la mayoría de los estudios se han enfocado en regiones glaciares más extensas. Además, los límites glaciares utilizados en estos estudios no suelen ser consistentes, ya que los investigadores frecuentemente seleccionan manualmente las imágenes satelitales que parecen no tener nieve efímera para evitar resultados inexactos. Para comparar los resultados de esta investigación, se consideraron los estudios de (Malone et al., 2022), (Taylor et al., 2022) y (Hanshaw & Bookhagen, 2014). Estos estudios emplearon el método NDSI (Índice Normalizado de Diferencia de Nieve) para segmentar el glaciar Quelccaya y posteriormente estimar el área glaciar mediante un análisis temporal, usando imágenes satelitales de la serie Landsat con una resolución de 30 metros. En particular, (Malone et al., 2022) y (Taylor et al., 2022) son investigaciones recientes que incluyen un análisis temporal detallado del glaciar Quelccaya.

Los resultados de la estimación de área glaciar obtenidos en este estudio presentan valores muy cercanos a los reportados en investigaciones previas. En años específicos, como 1991, 1992, 1995, 1999, 2006, 2009, 2015, 2019 y 2020, las estimaciones de área glaciar de este estudio son notablemente similares a las obtenidas en estudios anteriores. En la Figura 5.1 correspondiente, se observa que varios de estos valores se superponen o son muy cercanos a los resultados de otros estudios, aunque también existen pequeñas discrepancias. En general, los datos de este trabajo se asemejan más a los resultados de (Malone et al., 2022). Sin embargo, las diferencias menores en algunos años podrían atribuirse a variaciones en la delimitación de los bordes del

glaciar o a fenómenos climáticos interanuales, como El Niño o La Niña. Cabe destacar que los datos de esta investigación cuentan con una resolución espacial de 30 metros, equivalente a la de los estudios comparados.

En cuanto a los resultados específicos, para el año 1991 (inicio del análisis temporal en este estudio), la estimación del área glaciar en (Taylor et al., 2022) fue de 50.21 km², en (Malone et al., 2022) fue de 50.89 km², mientras que en este estudio fue de 50.81 km². Para el año 2020, último año de análisis en (Taylor et al., 2022) y (Malone et al., 2022), los valores de área glaciar fueron 41.58 km² en (Taylor et al., 2022), 39.03 km² en (Malone et al., 2022) y 38.99 km² en este estudio. Se observa que los resultados obtenidos en esta investigación son muy similares a los de (Malone et al., 2022). La tasa de retroceso glaciar estimada por (Malone et al., 2022) fue de 0.43 km²/año, mientras que en este estudio se obtuvo una tasa de 0.393 km²/año. Es importante señalar que (Taylor et al., 2022) no proporciona una tasa de retroceso específica para el glaciar Quelccaya.

Este estudio amplía el análisis temporal hasta 2024, año en el cual el área glaciar se ha reducido a 35.49 km². En la Figura 5.1 correspondiente se pueden observar algunos años excluidos del análisis; esto se debe a que las imágenes de esos años presentaban nieve temporal o nubes, lo cual afectaría la precisión de los resultados. Además, algunas imágenes no estaban disponibles para todos los años, lo que limitó la continuidad en ciertas observaciones.

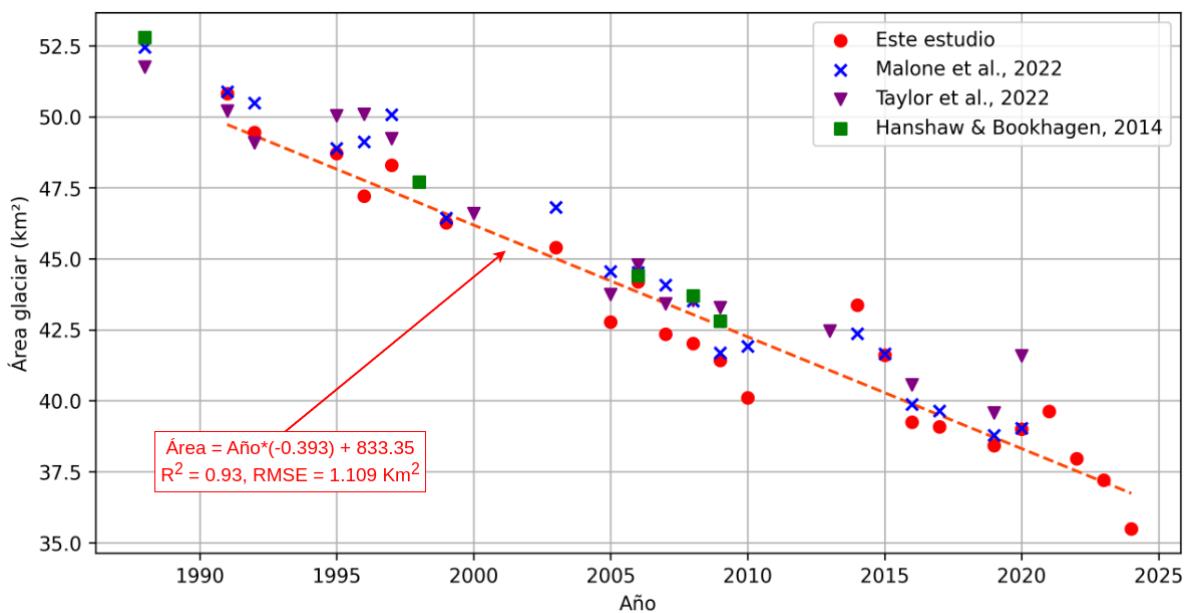


Figura 5.1: Serie temporal del área glaciar del Quelccaya. Los puntos rojos representan el área glaciar estimada en cada año según los resultados de este estudio, mientras que los demás símbolos corresponden a las estimaciones de estudios previos. La línea entrecortada muestra la regresión lineal obtenida para este estudio, con un coeficiente de determinación del 93 % y un error cuadrático medio (RMSE) de 1.109 km^2 . Además, se destaca la tasa de retroceso de la superficie glaciar, estimada en 0.393 $\text{km}^2/\text{año}$.

Discusión

Este estudio empleó un modelo de deep learning U-Net para segmentar los cuerpos glaciares del segundo glaciar más grande del mundo, el Quelccaya en Perú, utilizando imágenes de teledetección de Landsat correspondientes a varios años. El objetivo fue analizar los cambios ocurridos en el glaciar a lo largo de las últimas tres décadas.

5.1. Análisis del modelo

En la sección de resultados de entrenamiento, los modelos propuestos fueron evaluados durante 25 épocas, ya que se observó que no se obtenían mejoras significativas después de este número de iteraciones. Para lograr un rendimiento óptimo, fue necesario ajustar los valores de hiperparámetros, los cuales se presentan en la Tabla 3.1. Este ajuste se realizó mediante múltiples entrenamientos para identificar la configuración más adecuada. Las Figuras 4.1, 4.2 y 4.3 muestran los resultados de las métricas MIoU, PA y Dice, evaluadas para cada modelo. En estos gráficos, U-Net y DeepResUnet presentan un comportamiento consistente y similar, lo que sugiere que ambos modelos logran una segmentación semántica efectiva. En contraste, el modelo DeeplabV3Plus no alcanzó un desempeño competitivo en comparación con U-Net y DeepResUnet.

5.2. Análisis del rendimiento de segmentación de glaciares

La Figura 4.4 muestra la evolución de la función de pérdida en cada modelo durante las evaluaciones. Aquí, U-Net y DeepResUnet logran una convergencia más efectiva, mientras que DeepLabV3Plus mantiene una pérdida elevada en comparación, indicando un menor ajuste en la segmentación.

Los resultados cuantitativos están resumidos en la Tabla 4.1, donde se observa que DeepLabV3Plus tiene el menor rendimiento en segmentación según las métricas. DeepResUnet mejora considerablemente sobre DeepLabV3Plus al evaluar las mismas imágenes y métricas, mientras que U-Net destaca como el modelo de mayor precisión en la segmentación de cuerpos glaciares, alcanzando una precisión MIoU del 98.10 % y una pérdida de entrenamiento del 1 %.

Comparando con el estudio de (Rajat et al., 2022), que también mapea superficies glaciares usando imágenes satelitales, U-Net mostró inicialmente una precisión de píxeles baja en su caso, aumentando rápidamente después de la época 20. En su estudio, entrenaron U-Net durante 200 épocas, obteniendo una precisión del 95.89 % y una tasa de pérdida de entrenamiento de solo 0.07 %. Este menor valor de pérdida podría deberse a la mayor cantidad de épocas (200) en comparación con las 25 épocas utilizadas en este estudio, así como a posibles diferencias en la cantidad y calidad de datos de entrenamiento.

En cuanto a la precisión observada en este estudio, U-Net muestra un rendimiento superior en comparación con (Rajat et al., 2022). Esto puede estar relacionado con el uso de 2,400 imágenes del conjunto de datos, de las cuales el 70 % se destinó a entrenamiento, y además se implementó técnicas de aumento de datos en el algoritmo de entrenamiento.

Basado en la integración de las bandas 2, 3, 4, 5, 6 y 7 para Landsat 8 y las bandas 1, 2, 3, 4, 5 y 7 para Landsat 5 Collection 2 Level-1, U-Net demostró ser el modelo de mejor rendimiento, con una alta precisión en la clasificación y segmentación de diversos glaciares.

DeepResUnet también mostró un comportamiento similar a U-Net en cuanto a precisión;

sin embargo, tiene el mayor número de parámetros de entrenamiento en comparación con U-Net y DeepLabV3Plus, debido al uso del mecanismo residual que, aunque mejora el rendimiento, también incrementa el costo computacional.

5.3. Segmentación de glaciares y variación de glaciares

Como se observa en la Figura 4.5, los modelos propuestos presentan diferentes niveles de precisión en la segmentación de cuerpos glaciares. DeepLabV3Plus, aunque aceptable en términos generales, no logró segmentar los detalles finos de los cuerpos glaciares con la precisión esperada. En contraste, DeepResUnet mostró una segmentación más detallada de los límites glaciares, superando a DeepLabV3Plus en su capacidad para capturar los contornos de manera más completa. Sin embargo, el mejor rendimiento fue alcanzado por U-Net, que proporcionó la segmentación más precisa y detallada, demostrando su eficacia en la delineación exacta de los bordes glaciares.

En la Figura 4.7, se presentan las imágenes binarias que reconstruyen la segmentación del glaciar Quelccaya desde 1991 hasta 2024 utilizando el modelo U-Net. Claramente, la extensión glaciar disminuye a lo largo de los años, y las áreas señaladas en los recuadros rojos de la Figura 4.8 ilustran glaciares que desaparecen progresivamente. Sin embargo, la estimación precisa de la extensión glaciar a partir de imágenes satelitales requiere evitar escenas con nieve efímera, ya que esta puede introducir errores significativos en la estimación de la superficie glaciar.

En la Figura 4.9 Se observa claramente un retroceso significativo en la superficie glaciar a lo largo del tiempo, siendo más pronunciado en el lado occidental del glaciar en comparación con el lado oriental. Además, conforme el área glaciar ha disminuido, según los estudios de (Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña, 2023) se ha evidenciado la formación y expansión de varias lagunas en la zona.

A diferencia de (Malone et al., 2022), (Taylor et al., 2022) y (Hanshaw & Bookhagen, 2014), quienes utilizaron el metodo NDSI para segmentar y extraer cuerpos glaciares, en este

estudio se utilizó modelos de aprendizaje profundo para realizar la misma tarea, calcular los errores de las estimaciones no es una práctica común, a pesar de la importancia de estimar correctamente la superficie glaciar, según (Montoya-Jara et al., 2024) si existe una clasificación errónea de los píxeles se debe a la insertidumbre del umbral NDSI, para ello es común trabajar con umbrales que oscilen entre 0.55 y 0.66.

Es importante notar que el ritmo de disminución de la superficie glaciar varía anualmente. La tecnología satelital, combinada con herramientas avanzadas de segmentación automática, facilita el monitoreo de glaciares en retroceso, aunque sigue enfrentando desafíos, especialmente en la diferenciación entre agua, sombras y campos de nieve. Esto es particularmente relevante para glaciares cubiertos parcial o totalmente por escombros, donde una mayor resolución espacial ofrece ventajas significativas, no solo en términos de precisión de los límites glaciares, sino también en la visibilidad de características específicas, como grietas y texturas superficiales. Cabe mencionar que se están realizando muchos esfuerzos para estudiar la deglaciación en los Andes peruanos. Sin embargo estos estudios son complejos debido a los numerosos picos montañosos y a las características únicas en cuanto a ubicación e interacción con el medio ambiente.

En general, la estimación de la superficie glaciar de Quelccaya obtenida en este estudio es varia ligeramente en comparación con estudios previos. Esto se debe en parte a que el modelo U-Net mejora la propagación de características, reduce la pérdida de información, aumenta el peso de las secciones de glaciares y maneja la amplia gama de categorías de píxeles del glaciar.

Además fue optimizado para segmentar específicamente las superficies glaciares, lo que aumenta la confiabilidad de los resultados obtenidos. Este método automático permite segmentar el glaciar en cada imagen multiespectral de forma precisa, lo que respalda la viabilidad de esta técnica para estudios futuros de monitoreo glaciar. Estas imágenes multiespectrales de teledetección son muy buenas para segmentar cuerpos glaciares y su combinación con aprendizaje automático tienen sus propias ventajas sobre el método tradicional NDSI de extracción de glaciares.

Conclusiones

1. Se aplicaron con éxito técnicas de deep learning para clasificar y segmentar cuerpos glaciares a nivel de píxel en imágenes multiespectrales. Los modelos utilizados mostraron un rendimiento eficaz en la clasificación y segmentación de estos cuerpos, logrando resultados precisos. Sin embargo, fue necesario seleccionar el modelo de aprendizaje profundo con mejor desempeño, y se eligió U-Net por los resultados sobresalientes que presentó.

El análisis se realizó con imágenes del glaciar Quelccaya tomadas entre 1991 y noviembre de 2024. Para asegurar la precisión, se seleccionaron manualmente imágenes libres de nieve temporal y nubes, evitando así resultados falsos. Hubo algunos años en los que no se encontraron imágenes adecuadas, pero se incluyeron todas las disponibles para analizar y evaluar la estimación de la superficie glaciar de Quelccaya.

A partir de este conjunto de datos, se efectuó un análisis temporal que muestra una clara disminución en la extensión del glaciar a lo largo de las décadas. Sin embargo, se observaron algunos años en los que la superficie glaciar aumentó ligeramente, probablemente debido a la presencia mínima de nieve temporal o a variaciones meteorológicas. Este patrón de incremento también aparece en otros estudios con los que se compararon los resultados.

Finalmente, el método de segmentación mediante modelos de deep learning y el análisis temporal del retroceso de la superficie glaciar fueron llevados a cabo satisfactoriamente.

2. Se utilizaron modelos basados en redes neuronales convolucionales (CNN) para la segmentación semántica de cuerpos glaciares, empleando tres arquitecturas: U-Net, DeepResUNet y DeepLabV3Plus. Los tres modelos demostraron ser efectivos en la tarea de

segmentación, gracias a sus arquitecturas encoder-decoder, que permiten aprender características relevantes de los glaciares de manera automática. En comparación con los métodos tradicionales como el índice NDSI, que a menudo es lento y generalmente brinda resultados inapropiados al calcular incorrectamente el agua como nieve, los modelos de deep learning presentaron mejores resultados, con U-Net destacándose por su alta precisión. Las métricas obtenidas con U-Net fueron sobresalientes, con un MIoU de 0.9810, PA de 0.9979, Dice Coefficient de 0.9904 y un Loss de 0.01. Esto valida la hipótesis de que los algoritmos de deep learning pueden ofrecer una precisión significativa en la segmentación de cuerpos glaciares. Además, se sugiere que incluir más variables relevantes, como factores climáticos o topográficos, podría mejorar aún más el rendimiento del modelo.

3. Se logró desarrollar una base de datos confiable de 2400 imágenes satelitales, con una distribución de datos del 70 % para entrenamiento, 15 % para validación y 15 % para testeo. Para garantizar la calidad de los datos, se seleccionaron cuidadosamente imágenes multiespectrales que contenían exclusivamente cuerpos glaciares, y se etiquetaron solo los píxeles correspondientes a características glaciares, estas etiquetas se pueden observar en la Figura 3.11. Además, se utilizó Google Earth para verificar que los píxeles etiquetados realmente correspondieran a áreas glaciares. Con esta base de datos, los modelos de deep learning lograron una segmentación muy buena y efectiva de los cuerpos glaciares, tal como se puede observar en la Figura 4.5, donde los tres modelos segmentan cuerpos glaciares con una alta precisión para clasificar píxeles con características glaciares, validando la hipótesis de que la creación de una base de datos adecuada contribuye al rendimiento de los modelos, aunque la magnitud de la precisión puede variar.
4. Se analizaron los cambios en la superficie glaciar utilizando las imágenes segmentadas por el modelo U-Net y se cuantificó el área glaciar en diferentes años mediante un análisis temporal. Los resultados muestran que la extensión glaciar en 1991 era de 50.8131 km², mientras que en noviembre de 2024 se redujo a 35.4951 km², lo que representa una tasa de retroceso de aproximadamente 0.3936 km² por año. Sin embargo, en algunos años, las estimaciones mostraron un área mayor que la de años previos, lo que puede estar

relacionado con la presencia de nieve temporal en las imágenes. Como se puede evidenciar en el año 2010, la estimación de área glaciar para este estudio fue de 40.0995 km² y para el año 2014 la estimación glaciar fue de 43.3746 km²; sin embargo, tal fenómeno también ocurrió con el análisis de (Malone et al., 2022), quien en el año 2010 obtuvo 41.91 km² y para el año 2014 obtuvo una estimación glaciar de 42.37 km².

Para obtener estimaciones más precisas, es esencial evitar las imágenes que contengan nieve temporal, ya que esto puede introducir errores significativos en la medición. Se recomienda analizar múltiples imágenes durante la estación seca (junio-septiembre) y seleccionar aquellas con mínima presencia de nieve temporal. No obstante, incluso con estas precauciones, la estimación de la superficie glaciar puede seguir siendo incierta debido a las variaciones climáticas y las condiciones de cobertura en cada año. Además, algunos años no fueron considerados debido a la presencia excesiva de nieve temporal, nubosidad alta o la falta de imágenes disponibles en las fechas correspondientes.

5. Los antecedentes sobre la estimación de la extensión del glaciar Quelcaya son limitados, y no se encontraron datos oficiales de instituciones gubernamentales. Sin embargo, se realizó una comparación de los resultados de superficie glaciar obtenidos en este estudio con los de artículos científicos, especialmente con el trabajo de (Malone et al., 2022), (Taylor et al., 2022) y (Hanshaw & Bookhagen, 2014). Los resultados del análisis temporal muestran una alta concordancia con los estudios previos, con estimaciones de superficie glaciar muy cercanas, como se observa en los años 1991, 1995, 1999, 2009, 2015, 2019 y 2020. En este estudio, las estimaciones de superficie glaciar fueron de 50.81, 48.71, 46.26, 41.41, 41.59, 38.42 y 38.99 km², mientras que (Malone et al., 2022) reportó valores de 50.89, 48.89, 46.44, 41.69, 41.65, 38.79 y 39.03 km² en los mismos años. Para los años 1991, 1992, 2006, 2016, 2019 . En este estudio, las estimaciones de superficie glaciar fueron de 50.81, 49.45, 44.19, 39.25, 38.42 km², mientras que (Taylor et al., 2022) reportó valores de 50.21, 49.08, 44.79, 40.57, 39.58 km². Para el año 2006 en este estudio, la estimación glaciar fue de 44.19 km², mientras que en (Hanshaw & Bookhagen, 2014) reportó un valor de 44.4 km². Se puede concluir que muchos de los valores obtenidos en este estudio son muy próximos a los antecedentes mencionados, en general las estimaciones de superficie

glaciar obtenidas en este estudio muestran que, los valores obtenidos son muy cercanos a los valores reportados por (Malone et al., 2022), (Taylor et al., 2022) y (Hanshaw & Bookhagen, 2014), lo que sugiere que los resultados obtenidos mediante aprendizaje profundo son comparables a los obtenidos con métodos tradicionales como el NDSI.

Es importante destacar que el método tradicional NDSI utilizado en los estudios previos y el método de segmentación semántica basado en deep learning empleado en este estudio son técnicamente diferentes. Sin embargo, esta comparación demuestra que el aprendizaje profundo puede ser una alternativa competitiva y precisa para la estimación de la superficie glaciar, ofreciendo resultados similares a los de métodos más tradicionales.

Recomendaciones

1. Para mejorar la precisión y coherencia en los resultados, se recomienda utilizar datos ópticos con mejor resolución, como los del satélite Sentinel, que tiene una resolución de 10 metros por píxel en las bandas de color visible (rojo, verde, azul) y en el infrarrojo cercano, y 20 metros por píxel en las bandas de infrarrojo de onda corta y otras bandas del espectro visible. Alternativamente, se puede utilizar el satélite peruano PerúSAT-1, que ofrece una resolución de 0.7 metros por píxel en su modo pancromático (blanco y negro) y aproximadamente 2 metros por píxel en modo multiespectral (imágenes en color y en diferentes bandas del espectro). Gracias a esta resolución, se pueden capturar detalles de gran precisión en la superficie terrestre, lo que resulta útil para aplicaciones en cartografía, monitoreo de recursos naturales, agricultura, y estudios medioambientales. Sin embargo, estas imágenes no cuentan con datos históricos.
2. También se recomienda incluir datos de bandas térmicas junto con la temperatura promedio y precipitación de la zona o región, para enriquecer el conjunto de datos. El uso de más variables que afectan a los glaciares contribuirá a precisar los modelos de deep learning, ya que cuanto más datos se utilicen para entrenar el modelo, más coherentes y precisos serán los resultados.
3. Para futuras investigaciones, se recomienda una combinación de datos ópticos, radares y sensores LIDAR. Esta metodología permitirá construir series temporales mucho más precisas y coherentes para los cambios en los perímetros de los glaciares. Aunque esta información es aún limitada, ya existen antecedentes, como los estudios realizados por (Montoya-Jara et al., 2024) para los cuerpos glaciares de Suyuparina y Quisoquipina.

Bibliografía

- Aliaga, J. A., & Cosme, L. A. (2018). Retroceso glaciar del Nevado Huaytapallana como consecuencia del calentamiento global. *Naturaleza y Sociedad*, 1(1).
- Anderson, L. S., & Anderson, R. S. (2016). Modeling debris-covered glaciers: response to steady debris deposition. *The Cryosphere*, 10(3), 1105-1124.
- Bezerra, E., Mafalda, S., Alvarez, A. B., Uman-Flores, D. A., Perez-Torres, W. I., & Palomino-Quispe, F. (2023). A Cloud Coverage Image Reconstruction Approach for Remote Sensing of Temperature and Vegetation in Amazon Rainforest. *Applied Sciences*, 13(23), 12900.
- Bond, T. C., Doherty, S. J., Fahey, D. W., Forster, P. M., Berntsen, T., Deangelo, B. J., Flanner, M. G., Ghan, S., Kärcher, B., Koch, D., Kinne, S., Kondo, Y., Quinn, P. K., Sarofim, M. C., Schultz, M. G., Schulz, M., Venkataraman, C., Zhang, H., Zhang, S., . . . Zender, C. S. (2013). Bounding the role of black carbon in the climate system: A scientific assessment. *Journal of Geophysical Research: Atmospheres*, 118(11), 5380-5552. <https://doi.org/10.1002/jgrd.50171>
- Brío, B. M. d., & Molina, A. S. (2001). *Redes neuronales y sistemas borrosos*. Ra-Ma.
- Cabrera Paredes, J. F., & Castillo Melendez, G. (2019). Las curvas de capacidad de potencia y el análisis gráfico de los límites de operación del generador síncrono.
- Camps-Valls, G., Tuia, D., Gómez-Chova, L., Jiménez, S., & Malo, J. (2011). Remote Sensing Image Processing. Synthesis Lectures on Image, Video, and Multimedia Processing.
- Ceballos, J., Rodríguez Murcia, C., & Real-Núñez, E. (2012). Glaciares de Colombia, más que montañas con hielo. *Bogotá: IDEAM*, 344.
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision (ECCV)*, 801-818.
- Chuvieco, E. (1996). Fundamentos de la teledetección espacial, España: Rialp. SA,
- Costa, R. M. (2022). Migrações na comunidade campesina de Phinaya, Cordilheira Vilcanota, Peru: contexto das mudanças climáticas e socioambientais.
- DEL PERU, E. L. M. A. (s.f.). RETROCESO GLACIAR E IMPACTO AMBIENTAL.
- Drenkhan, F., Guardamino, L., Huggel, C., & Frey, H. (2018). Current and future glacier and lake assessment in the deglaciating Vilcanota-Urubamba basin, Peruvian Andes. *Global and Planetary Change*, 169, 105-118.
- Durán-Alarcón, C., Gevaert, C. M., Mattar, C., Jiménez-Muñoz, J. C., Pasapera-Gonzales, J. J., Sobrino, J. A., Silvia-Vidal, Y., Fashé-Raymundo, O., Chavez-Espiritu, T. W., &

- Santillan-Portilla, N. (2015). Recent trends on glacier area retreat over the group of Nevados Caullaraju-Pastoruri (Cordillera Blanca, Peru) using Landsat imagery. *Journal of South American Earth Sciences*, 59, 19-26.
- eoPortal Directory. (2024). Landsat-4, -5 [Last accessed: October 19, 2024]. <https://www.eoportal.org/satellite-missions/landsat-4-5#landsat-5-1>
- EOS. (fecha de acceso). *NDSI - Índice de Nieve y Hielo Normalizado*. <https://eos.com/es/make-an-analysis/ndsi/#:~:text=El%20%C3%ADndice%20NDSI%20es%20una,las%20partes%20visibles%20del%20espectro>
- Francou, B., Fabre, D., Pouyaud, B., Jomelli, V., & Arnaud, Y. (1999). Symptoms of degradation in a tropical rock glacier, Bolivian Andes. *Permafrost and Periglacial Processes*, 10(1), 91-100.
- Gaddam, V. K., Boddapati, R., Kumar, T., Kulkarni, A. V., & Bjornsson, H. (2022). Application of “OTSU”—An image segmentation method for differentiation of snow and ice regions of glaciers and assessment of mass budget in Chandra basin, Western Himalaya using Remote Sensing and GIS techniques. *Environmental Monitoring and Assessment*, 194(5), 337.
- Garcia Bazan, L. C. (2022). Estimación del área superficial de un glaciar utilizando técnicas de procesamiento digital de imágenes satelitales obtenidas por el PERUSAT-I.
- Geotig, E. (Fecha de acceso). *Absorción y Dispersión*. <https://blogs.upm.es/espacio-geotig/absolucion-y-dispersion/>
- Gerens. (s.f.). Más del 50 % de la superficie glaciar en el Perú ha desaparecido.
- Ghaznavi, A., Saberioon, M., Brom, J., & Itzerott, S. (2024). Comparative performance analysis of simple U-Net, residual attention U-Net, and VGG16-U-Net for inventory inland water bodies. *Applied Computing and Geosciences*, 21, 100150.
- Gilardoni, S., Di Mauro, B., & Bonasoni, P. (2022). Black carbon, organic carbon, and mineral dust in South American tropical glaciers: A review. *Global and Planetary Change*, 213, 103837. <https://doi.org/10.1016/j.gloplacha.2022.103837>
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* [<http://www.deeplearningbook.org>]. MIT Press.
- Halliday, D., Resnick, R., & Krane, K. S. (2015). *Physics*, vol. 2. Wiley Custom Learning Solutions.
- Hanshaw, M. N., & Bookhagen, B. (2014). Glacial areas, lake areas, and snow lines from 1975 to 2012: status of the Cordillera Vilcanota, including the Quelccaya Ice Cap, northern central Andes, Peru. *The Cryosphere*, 8(2), 359-376.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- Herdy, S. (fecha de acceso). *PyTorch Semantic Image Segmentation*. <https://medium.com/@stefan.herdy/pytorch-semantic-image-segmentation-b72658962e3>
- IBM. (2021). *¿Qué son las redes neuronales?* <https://www.ibm.com/mx-es/topics/neural-networks>
- Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña. (2023). *Inventario Nacional de Glaciares y Lagunas de Origen Glaciar 2023*. INAIGEM.
- IPCC. (2019). *El océano y la criósfera en un clima cambiante. Informe especial del Grupo Intergubernamental de Expertos sobre el Cambio Climático. Resumen para responsables de políticas*. Unidad de Apoyo Técnico del Grupo de Trabajo II, Panel Intergubernamental sobre el Cambio Climático.
- Jähne, B. (2005). *Digital image processing*. Springer Science & Business Media.

- Jiménez Tejeda, L. A., & Tecse López, C. A. (2021). Revisión sistemática de desglaciación, extensión superficial y formación de lagos en la cordillera de los Andes-Perú mediante métodos de teledetección, 2020.
- Lin, R., Mei, G., & Xu, N. (2023). Accurate and automatic mapping of complex debris-covered glacier from remote sensing imagery using deep convolutional networks. *Geological Journal*, 58(6), 2254-2267.
- Lliboutry, L. (1956). *Nieves y glaciares de Chile: fundamentos de glaciología*. Ediciones de la Universidad de Chile. <https://libros.uchile.cl/1339>
- Malone, A. G., Broglie, E. T., & Wrightsman, M. (2022). The evolution of the two largest tropical ice masses since the 1980s. *Geosciences*, 12(10), 365.
- MathWorks. (s.f.). *¿Qué es una red neuronal convolucional? | 3 cosas que debe saber* [Recuperado 25 de agosto de 2023]. <https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html>
- Meneses, P. R., & Almeida, T. d. (2012). Introdução ao processamento de imagens de sensoriamento remoto. *Universidade de Brasília, Brasília*.
- Menzies, J. (1995). Modern glacial environments: processes, dynamics, and sediments.
- Montoya-Jara, N., Loayza, H., Gutiérrez-Rosales, R. O., Bueno, M., & Quiroz, R. (2024). Estimation of Glacier Outline and Volume Changes in the Vilcanota Range Snow-Capped Mountains, Peru, Using Temporal Series of Landsat and a Combination of Satellite Radar and Aerial LIDAR Images. *Remote Sensing*, 16(20), 3901.
- NASA. (2022). Landsat 8 and 9 OLI/TIRS C2 L1 Data Format Control Book [Accessed: 2024-11-01]. https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/LSDS-1822_Landsat8-9-OLI-TIRS-C2-L1-DFCB-v6.pdf
- NASA. (2024). *El casquete de hielo de Quelccaya: Antes y ahora* [Accedido: 2024-11-12]. <https://ciencia.nasa.gov/ciencias-terrestres/el-casquete-de-hielo-de-quelccaya-antes-y-ahora/>
- Navarro, Á., Úbeda, J., Gómez, J., & Pellitero, R. (2023). 1975–2018: 43 Years of Glacial Retreat in the Incachiriasca Glacier (Nevado Salcantay, Vilcabamba Range, Peru). En *Sustainable Development Goals in Europe: A Geographical Approach* (pp. 263-278). Springer.
- Olabe, X. B. (1998). *Redes neuronales artificiales y sus aplicaciones*. Publicaciones de la Escuela de Ingenieros.
- Perez-Torres, W. I., Uman-Flores, D. A., Quispe-Quispe, A. B., Palomino-Quispe, F., Bezerra, E., Leher, Q., Paixão, T., & Alvarez, A. B. (2024). Exploratory Analysis Using Deep Learning for Water-Body Segmentation of Peru's High-Mountain Remote Sensing Images. *Sensors*, 24(16), 5177.
- Porcel, E. M. (2015). Iniciación de un monitoreo del balance de masa en el glaciar Suyuparina, Cordillera Vilcanota, Perú. *Cambio Climático en los Andes Tropicales*, 2(2), 1-14.
- Potter Jr, N. (1972). Ice-cored rock glacier, Galena Creek, northern Absaroka Mountains, Wyoming. *Geological Society of America Bulletin*, 83(10), 3025-3058.
- Prince, S. J. (2023). *Understanding deep learning*. MIT press.
- Público, O. (2024). *El pacto milenario: Los guardianes del hielo en el Cusco* [Accedido: 2024-11-12]. <https://ojopublico.com/4494/el-pacto-milenario-los-guardianes-del-hielo-el-cusco>
- PyTorch. (2024). *torch.optim.lr_scheduler.ReduceLROnPlateau* [Accessed: 2024-08-26]. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

- Rajat, S., Singh, B. R., Prakash, C., & Anita, S. (2022). Glacier retreat in Himachal from 1994 to 2021 using deep learning. *Remote Sensing Applications: Society and Environment*, 28, 100870.
- Ramírez, E. (2008). Impactos del cambio climático y gestión del agua sobre la disponibilidad de recursos hídricos para las ciudades de La Paz y El Alto. *Revista Virtual LEDESMA*, 2(3), 49-61.
- Rojas Monsalvo, K. (2013). Radiación electromagnética.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III* 18, 234-241.
- Ruiz, N. d. l. A. T. (2023). Deglaciación del nevado Palcaraju y probable desbordamiento de la laguna Palcacocha. Consecuencias futuras y posibles acciones. *Justicia Ambiental. Revista del Poder Judicial del Perú especializada en la Protección del Ambiente*, 3(3), 93-114.
- Sánchez-Díaz, B. (2018). La teledetección en investigaciones ecológicas como apoyo a la conservación de la biodiversidad: una revisión. *Revista científica*, (33), 243-253.
- Seimon, T. A., Seimon, A., Yager, K., Reider, K., Delgado, A., Sowell, P., Tupayachi, A., Konecky, B., McAloose, D., & Halloy, S. (2017). Long-term monitoring of tropical alpine habitat change, Andean anurans, and chytrid fungus in the Cordillera Vilcanota, Peru: Results from a decade of study. *Ecology and Evolution*, 7(5), 1527-1540.
- Sobrino, J. A. (2001). *Teledetección*. Universitat de Valencia.
- Sood, V., Singh, S., Taloor, A. K., Prashar, S., & Kaur, R. (2020). Monitoring and mapping of snow cover variability using topographically derived NDSI model over north Indian Himalayas during the period 2008–19. *Applied Computing and Geosciences*, 8, 100040.
- Survey, U. G. (2013). *Landsat Data Continuity Mission (LDCM) / Landsat 8: Science Data Users Handbook*. United States Geological Survey (USGS). <https://www.un-spider.org/sites/default/files/LDCM-L8.R1.pdf>
- Survey, U. G. (2024). Landsat Collection 2 Level 1 Data [Accessed: 2024-10-14]. <https://www.usgs.gov/landsat-missions/landsat-collection-2-level-1-data>
- Taylor, L. S., Quincey, D. J., Smith, M. W., Potter, E. R., Castro, J., & Fyffe, C. L. (2022). Multi-decadal glacier area and mass balance change in the Southern Peruvian Andes. *Frontiers in Earth Science*, 10.
- Torres, C., Suárez, L., Schmitt, C., Estevan, R., & Helmig, D. (2018). Measurement of light absorbing particles in the snow of the Huaytapallana glacier in the central Andes of Peru and their effect on albedo and radiative forcing. *Óptica Pura y Aplicada*, 51(4), 1-14. <https://doi.org/10.7149/OPA.51.4.51004>
- Veettil, B. K., & Kamp, U. (2017). Remote sensing of glaciers in the tropical Andes: a review. *International journal of remote sensing*, 38(23), 7101-7137.
- Yarleque, C., Vuille, M., Hardy, D. R., Timm, O. E., De la Cruz, J., Ramos, H., & Rabatel, A. (2018). Projections of the future disappearance of the Quelccaya Ice Cap in the Central Andes. *Scientific reports*, 8(1), 15564.
- Zhang, Z., Liu, Q., & Wang, Y. (2018). Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5), 749-753.
- Zuluaga Gómez, O. A., Patiño Quinchía, J. E., & Valencia Hernández, G. M. (2021). Modelos implementados en el análisis de series de tiempo de temperatura superficial e índices de vegetación: una propuesta taxonómica en el contexto de cambio climático global. *Revista de Geografía Norte Grande*, (78), 323-344.

Anexos

Anexo A: Código para cargar el dataset

```
1 import torch
2 from torch.utils.data import DataLoader, Dataset, random_split
3 from torchvision import transforms as T
4
5 import os
6 import rasterio
7 import numpy as np
8
9 # Clase para cargar datos
10 class My_dataloader(Dataset):
11     def __init__(self, data_dir, transform=None):
12         self.data_dir = data_dir
13         self.transform = transform
14         self.image_dir = sorted(os.listdir(os.path.join(data_dir, "images")))
15         self.mask_dir = sorted(os.listdir(os.path.join(data_dir, "masks")))
16     def __len__(self):
17         return len(self.image_dir)
18
19     def __getitem__(self, idx):
20         img_path=os.path.join(self.data_dir, 'images', self.image_dir[idx])
21         mask_path=os.path.join(self.data_dir, 'masks', self.mask_dir[idx])
22         #image_name = img_path[idx].split('/')[-1] # Obtiene el nombre de la imagen
23         with rasterio.open(img_path) as img_dataset:
24             # Leemos la imagen multiespectral
25             img = img_dataset.read()
26             # Normalizamos la imagen para que se encuentre en el rango de [0,1]
27             img = img.astype(np.float32) / np.max(img)
28             # Conversión a tensor de pytorch
29             img = torch.tensor(img)
30
31         with rasterio.open(mask_path) as mask_dataset:
32             # Leemos la mascara
33             mask = mask_dataset.read()
34             # Normalizamos la mascara para que se encuentre en el rango de [0,1]
35             #mask = mask.astype(np.float32) / 255.0
36             mask = mask.astype(np.float32) / np.max(mask)
37             # Conversión a tensor de pytorch
38             mask = torch.tensor(mask)
39
40         return img, mask
41
42
```

Anexo B: Código de entrenamiento

```
trainDA.py > main
1 import os
2 import csv
3 import sys
4 import click
5 import traceback
6 import albumentations as A
7 import torch
8 import torch.optim as optim
9 from albumentations.pytorch import ToTensorV2
10 from torchvision import transforms as T
11 from tqdm import tqdm
12 from torch.utils.data import DataLoader
13 from my_dataset import My_Dataloader
14 #from dataset import CustomDataset
15 from modules_DeepLabV3 import DeepLabV3Plus
16 from modules_Unet import Unet
17 from modules_ResNet import DeepResUNet
18
19 from metrics import calculate_metrics, calculate_metrics2, DICE_BCE_Loss, DICE_BCE_Loss2
20
21 import matplotlib.pyplot as plt
22 import numpy as np
23
24 # image shape
25 INPUT = (256, 256)
26 CLASSES = 1
27 @click.command()
28 @click.option("-T", "--data-dir-train", type = str, required = True, help = "Path for data train directory")
29 @click.option("-V", "--data-dir-val", type = str, required = True, help = "Path for data val directory")
30 @click.option(
31     "-E",
32     "--num-epochs",
33     type=int,
34     default=25,
35     help="Number of epochs to train the model for. Default - 25",
36 )
37 @click.option(
38     "-L",
39     "--learning-rate",
40     type=float,
41     default=1e-3,
42     help="Learning Rate for model. Default - 1e-4",
43 )
```

```

43     )
44     @click.option(
45         "-B",
46         "--batch-size",
47         type=int,
48         default=4,
49         help="Batch size of data for training. Default - 4",
50     )
51     @click.option(
52         "-A",
53         "--augment",
54         type=bool,
55         default=True,
56         help="Opt-in to apply augmentations to training set. Default - True",
57     )
58     @click.option(
59         "-S",
60         "--early-stop",
61         type=bool,
62         default=True,
63         help="Stop training if val_loss hasn't improved for a certain no. of epochs. Default - True",
64     )
65     def main(data_dir_train,
66             data_dir_val,
67             num_epochs,
68             learning_rate,
69             batch_size,
70             augment: bool,
71             early_stop:bool,
72             ):
73         click.secho(message="⚡ Training...", fg="blue", nl=True)
74         os.makedirs("output", exist_ok=True)
75         #train_transform = T.Compose([T.Resize([INPUT[0], INPUT[1]]])
76         #val_transform = T.Compose([T.Resize([INPUT[0], INPUT[1]])])
77         """ Etapa de data augmentation """
78         if augment:
79             train_transform = T.Compose([
80                 T.Resize([INPUT[0], INPUT[1]]),
81                 T.RandomHorizontalFlip(),
82                 T.RandomVerticalFlip(),
83                 T.RandomRotation(degrees=90),
84                 T.ToTensor()
85             ])

```

```

86
87     else:
88         pass
89
90     val_transform = T.Compose([T.Resize([INPUT[0], INPUT[1]])])
91
92     Training_dataset=My_dataloader(data_dir_train, transform=train_transform)
93     Validating_dataset=My_dataloader(data_dir_val, transform=val_transform)
94
95
96     train_dataloader = DataLoader(Training_dataset, batch_size=batch_size,num_workers=8, shuffle=True)
97     val_dataloader=DataLoader(Validating_dataset, batch_size=batch_size, num_workers=8, shuffle=True)
98
99     # Definimos el modelo de entrenamiento
100    model = DeepLabV3plus(num_classes=CLASSES)
101    #model = Unet(6,1)
102    #model = DeepResUNet(6,1)
103    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
104    model.to(device)
105    # Definimos la función de perdida
106    criterion = DICE_BCE_Loss2()
107    optimizer = optim.Adam(model.parameters(), lr=learning_rate, weight_decay=1e-6, betas=(0.9, 0.999 ))
108    scheduler = optim.lr_scheduler.ReduceLROnPlateau(
109        optimizer, mode="min", patience=5, factor=0.1
110    )
111    # Detener el entrenamiento
112    patience_epochs = 24
113    no_improvement_epochs = 0
114    csv_file = os.path.abspath("output/training_logs_DeepLabV3Plus_L5_L8_C2L1_P1.csv")
115    csv_header = [
116        "Epoch",
117        "Avg Train Loss",
118        "Avg Val Loss",
119        "Avg MIoU Train",
120        "Avg MIoU Val",
121        "Avg Pix Acc Train",
122        "Avg Pix Acc Val",
123        "Avg Dice Coeff Train",
124        "Avg Dice Coeff Val",
125        "Learning Rate",
126    ]
127
```

```

127
128     # Guardar el mejor modelo
129     best_val_loss = float("inf")
130     click.echo(
131         f"\n{click.style(text='Train Size: ', fg='blue')}{Training_dataset.__len__()} {click.style(text='Test Size: ', fg='blue')}{Validating_dataset.__len__()}\n"
132     )
133     # Bucle principal
134     with open(csv_file, "w", newline="") as f:
135         csv_writer = csv.writer(f)
136         csv_writer.writerow(csv_header)
137
138     #Parametros para actualizar lr
139     # Crear listas para almacenar las perdidas d entrenamiento y validacion en cada epoca
140     training_losses, validating_losses = [], []
141     training_dices, validating_dices = [], []
142     training_ious, validating_ious = [], []
143     training_pixels, validating_pixels = [], []
144
145     for epoch in range(num_epochs):
146         # Entrenamiento
147         model.train()
148         # Inicializar metricas
149         train_loss = 0.0
150         total_iou_train = 0.0
151         total_pixel_accuracy_train = 0.0
152         total_dice_coefficient_train = 0.0
153
154         train_dataloader = tqdm(
155             train_dataloader, desc=f"Epoch {epoch + 1}/{num_epochs}", unit="batch"
156         )
157
158         current_lr = optimizer.param_groups[0]["lr"]
159
160         for images, masks in train_dataloader:
161             images, masks = images.to(device), masks.to(device)
162
163             optimizer.zero_grad()
164             # Realizar la retropropagación hacia adelante y el calculo de loss
165             outputs = model(images)
166             t_loss = criterion(outputs, masks)
167             # Realizar la retropropagación y actualización de los parametros
168             t_loss.backward()
169             optimizer.step()

```

```

167     # Realizar la retropropagación y actualización de los parametros
168     t_loss.backward()
169     optimizer.step()
170
171     train_loss += t_loss.item()
172     # Calcular las metricas para entrenamiento
173     with torch.no_grad():
174         pred_masks = outputs > 0.5
175         iou_train, dice_coefficient_train, pixel_accuracy_train = calculate_metrics(
176             pred_masks, masks
177         )
178         total_iou_train += iou_train
179         total_dice_coefficient_train += dice_coefficient_train
180         total_pixel_accuracy_train += pixel_accuracy_train
181
182     # Visualizar el progreso de las metricas
183     train_dataloader.set_postfix(
184         loss=t_loss.item(),
185         train_iou=iou_train,
186         train_pix_acc=pixel_accuracy_train,
187         train_dice_coef=dice_coefficient_train,
188         lr=current_lr,
189     )
190
191     train_loss /= len(train_dataloader)
192     avg_iou_train = total_iou_train / len(train_dataloader)
193     avg_pixel_accuracy_train = total_pixel_accuracy_train / len(train_dataloader)
194     avg_dice_coefficient_train = total_dice_coefficient_train / len(train_dataloader)
195     # Guardar las métricas de entrenamiento en las listas
196     training_losses.append(train_loss)
197     training_dices.append(avg_dice_coefficient_train)
198     training_ious.append(avg_iou_train)
199     training_pixels.append(avg_pixel_accuracy_train)
200     print("Training Losses: ", training_losses)
201
202     # Validación
203     model.eval()
204     val_loss = 0.0
205     total_iou_val = 0.0
206     total_pixel_accuracy_val = 0.0
207     total_dice_coefficient_val = 0.0
208
209     val_dataloader = tqdm(val_dataloader, desc=f"Validation", unit="batch")
210

```

```

209     val_dataloader = tqdm(val_dataloader, desc="Validation", unit="batch")
210
211     with torch.no_grad():
212         for images, masks in val_dataloader:
213             images, masks = images.to(device), masks.to(device)
214             outputs = model(images)
215
216             v_loss = criterion(outputs, masks)
217             val_loss += v_loss.item()
218
219             # Calcular metricas para Validación
220             pred_masks = outputs > 0.5
221             iou_val, dice_coefficient_val, pixel_accuracy_val = calculate_metrics(
222                 pred_masks, masks
223             )
224
225             total_iou_val += iou_val
226             total_pixel_accuracy_val += pixel_accuracy_val
227             total_dice_coefficient_val += dice_coefficient_val
228
229             val_dataloader.set_postfix(
230                 val_loss=val_loss.item(),
231                 val_iou=iou_val,
232                 val_pix_acc=pixel_accuracy_val,
233                 val_dice_coef=dice_coefficient_val,
234                 lr=current_lr,
235             )
236
237             val_loss /= len(val_dataloader)
238             avg_iou_val = total_iou_val / len(val_dataloader)
239             avg_pixel_accuracy_val = total_pixel_accuracy_val / len(val_dataloader)
240             avg_dice_coefficient_val = total_dice_coefficient_val / len(val_dataloader)
241
242             validating_losses.append(val_loss)
243             validating_ious.append(avg_iou_val)
244             validating_pixels.append(avg_pixel_accuracy_val)
245             validating_dices.append(avg_dice_coefficient_val)
246             print("Validation Losses: ", validating_losses)
247
248             scheduler.step(val_loss)
249
250             print(
251                 f"\nEpoch {epoch + 1}/{num_epochs}\n"
252                 f"Avg Train Loss: {train_loss:.4f}\n"

```

```

253                 f"Avg Validation Loss: {val_loss:.4f}\n"
254                 f"Avg MIoU Train: {avg_iou_train:.4f}\n"
255                 f"Avg MIoU Val: {avg_iou_val:.4f}\n"
256                 f"Avg Pix Acc Train: {avg_dice_coefficient_train:.4f}\n"
257                 f"Avg Pix Acc Val: {avg_pixel_accuracy_val:.4f}\n"
258                 f"Avg Dice Coeff Train: {avg_dice_coefficient_train:.4f}\n"
259                 f"Avg Dice Coeff Val: {avg_dice_coefficient_val:.4f}\n"
260                 f"Current LR: {current_lr}\n"
261                 f"\u2022\u2022\u2022\n"
262
263
264             # Guardar el mejor modelo
265             if val_loss < best_val_loss:
266                 no_improvement_epochs = 0
267                 click.echo(
268                     message=f"\n*\u2022\u2022\u2022 val_loss improved from {best_val_loss:.4f} to {val_loss:.4f}\n",
269                     fg="green",
270                 )
271                 best_val_loss = val_loss
272                 torch.save(model.state_dict(), "./output/best_model_DeepLabV3Plus_L5_L8_C2L1_P1.pth")
273                 click.echo(message="Saved Best Model! \ud83d\udcbb", fg="green")
274                 print(f"\u2022\u2022\u2022\n")
275
276             else:
277                 no_improvement_epochs += 1
278                 click.echo(
279                     message=f"\u2022\u2022\u2022 val_loss did not improve from {best_val_loss:.4f}\n", fg="yellow"
280                 )
281                 print(f"\u2022\u2022\u2022\n")
282
283             # Apilar los valores de entrenamiento y validación en un CSV
284             csv_writer.writerow([
285                 epoch + 1,
286                 train_loss,
287                 val_loss,
288                 avg_iou_train,
289                 avg_iou_val,
290                 avg_pixel_accuracy_train,
291                 avg_pixel_accuracy_val,
292                 avg_dice_coefficient_train,
293                 avg_dice_coefficient_val,
294                 current_lr,

```

```

293     |         avg_dice_coefficient_val,
294     |         current_lr,
295     |
296     |     )
297
298     # Detención temprana
299     if early_stop:
300         if no_improvement_epochs >= patience_epochs:
301             click.echo(
302                 message=f"\nEarly Stopping: val_loss did not improve for {patience_epochs} epochs.\n",
303                 fg="red",
304             )
305             break
306
307     click.echo(message="Training Done!", fg="blue", nl=True)
308
309 ##### Guardar Graficos de metricas #####
310
311 epochs = list(range(1, len(training_losses) + 1))
312
313 plt.figure(figsize=(20, 10))
314 plt.subplot(2, 2, 1)
315 plt.plot(epochs, training_losses, label='Train Loss')
316 plt.plot(epochs, validating_losses, label='Validation Loss')
317 plt.xticks(ticks=list(range(1, len(training_losses) + 1)))
318 plt.title('Training and Validation Loss')
319 plt.xlabel('Epochs')
320 plt.ylabel('Loss')
321 plt.grid()
322 plt.tight_layout()
323 plt.legend()
324
325 plt.subplot(2, 2, 2)
326 plt.plot(epochs, training_ious, label='Train MIoU')
327 plt.plot(epochs, validating_ious, label='Validation MIoU')
328 plt.xticks(ticks=list(range(1, len(training_losses) + 1)))
329 plt.title('Training and Validation MIoU')
330 plt.xlabel('Epochs')
331 plt.ylabel('MIoU')
332 plt.grid()
333 plt.tight_layout()
334 plt.legend()
335
336 plt.subplot(2, 2, 3)

```

```

327 plt.plot(epochs, validating_ious, label='Validation MIoU')
328 plt.xticks(ticks=list(range(1, len(training_losses) + 1)))
329 plt.title('Training and Validation MIoU')
330 plt.xlabel('Epochs')
331 plt.ylabel('MIoU')
332 plt.grid()
333 plt.tight_layout()
334 plt.legend()
335
336 plt.subplot(2, 2, 3)
337 plt.plot(epochs, training_pixels, label='Train Pixel Accuracy')
338 plt.plot(epochs, validating_pixels, label='Validation Pixel Accuracy')
339 plt.xticks(ticks=list(range(1, len(training_losses) + 1)))
340 plt.title('Training and Validation PA')
341 plt.xlabel('Epochs')
342 plt.ylabel('PA')
343 plt.grid()
344 plt.tight_layout()
345 plt.legend()
346
347 plt.subplot(2, 2, 4)
348 plt.plot(epochs, training_dices, label='Train Dice Coefficient')
349 plt.plot(epochs, validating_dices, label='Validation Dice Coefficient')
350 plt.xticks(ticks=list(range(1, len(training_losses) + 1)))
351 plt.title('Training and Validation Dice')
352 plt.xlabel('Epochs')
353 plt.ylabel('Dice')
354 plt.grid()
355 plt.legend()
356 #plt.text(1, min(training_dices), 'd', fontsize=12, verticalalignment='bottom', horizontalalignment='right')
357 #plt.text(len(epochs) / 2, min(training_dices) - 0.01, 'd', fontsize=12, ha='center') # Colocar la letra 'd' debajo de la gráfica
358 # Añadir título general
359 plt.suptitle('DeepLabV3Plus', fontsize=16)
360
361 plt.tight_layout(rect=[0, 0, 1, 0.95]) # Ajustar layout para no solapar con el título
362
363 plt.tight_layout()
364 plt.savefig('./output/training_metric_Deeplabv3plus_L5_L8_C2L1_P1.png')
365
366 return
367 if __name__ == "__main__":
368     main()

```

Anexo C: Código de evaluación

```
1 import torch
2 from torchvision import transforms as T
3 from torch.utils.data import DataLoader, Dataset, random_split
4 from metrics import calculate_metrics, DICE_BCE_Loss, DICE_BCE_Loss2
5 from my_dataset import My_dataloader
6 from modules_DeepLabV3 import DeepLabV3Plus
7 from modules_ResNet import DeepResUNet
8 from modules_Unet import Unet
9 from typing import Any
10 import click
11 from tqdm import tqdm
12 import matplotlib.pyplot as plt
13
14 CLASSES = 1
15 INPUT = (256, 256)
16
17 @click.command()
18 @click.option("--D", "--data-dir", type=str, required=True, help="Path for Data Directory")
19
20 def main(data_dir):
21     click.echo(message="🏃 Evaluation...", fg="blue")
22     device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
23     # Función de perdida
24     criterion = DiceLoss()
25     criterion = DICE_BCE_Loss2()
26     # Modelo
27     #model = DeepLabV3Plus(num_classes=CLASSES).to(device)
28     #model = Unet(6,1).to(device)
29     model = DeepResUNet(6,1).to(device)
30
31     model.load_state_dict(torch.load("./output/best_model_DeepResUNet_L5_L8_C2L1_P1.pth"))
32     #model.to(device)
33     model.eval()
34     # Inicializar variables
35     total_loss_eval = 0.0
36     total_iou_eval = 0.0
37     total_pixel_accuracy_eval = 0.0
38     total_dice_coefficient_eval = 0.0
39     # Pre-procesamiento de datos
40     eval_transform=T.Compose([T.Resize([INPUT[0], INPUT[1]])])
41     eval_dataset=My_dataloader(data_dir, transform=eval_transform)
42     eval_dataloader=DataLoader(eval_dataset, batch_size=1, num_workers=8, shuffle=False)
43     click.echo(message=f"\n{click.style('Evaluation Size: ', fg='blue')}{eval_dataset.__len__()}")
44     eval_dataloader = tqdm(eval_dataloader, desc="Evaluation", unit="image")
45     # Calculo de gradiente
46     with torch.no_grad():
47         for images, masks in eval_dataloader:
48             images, masks = images.to(device), masks.to(device)
49             output = model(images)
50
51             eval_loss = criterion(output, masks)
52             #print()
53             total_loss_eval += eval_loss.item()
54
55             pred_masks = output > 0.5
56             #pred_masks = output
57             iou_eval, dice_coefficient_eval, pixel_accuracy_eval = calculate_metrics(
58                 pred_masks, masks
59             )
60
61             pred_masks, masks
62
63             total_iou_eval += iou_eval
64             total_dice_coefficient_eval += dice_coefficient_eval
65             total_pixel_accuracy_eval += pixel_accuracy_eval
66
67             # Displaying metrics in the progress bar description
68             eval_dataloader.set_postfix(
69                 total_loss_eval=eval_loss.item(),
70                 eval_iou=iou_eval,
71                 eval_pix_acc=pixel_accuracy_eval,
72                 eval_dice_coef=dice_coefficient_eval,
73             )
74             #total_loss_eval /= len(eval_dataloader)
75             avg_total_loss = total_loss_eval / len(eval_dataloader)
76             avg_iou_eval = total_iou_eval / len(eval_dataloader)
77             avg_pixel_accuracy_eval = total_pixel_accuracy_eval / len(eval_dataloader)
78             avg_dice_coefficient_eval = total_dice_coefficient_eval / len(eval_dataloader)
79
80             print(
81                 f"Avg Eval Loss: {avg_total_loss:.4f}\n"
82                 f"Avg mIOU EVal: {avg_iou_eval:.4f}\n"
83                 f"Avg Pix Acc EVal: {avg_pixel_accuracy_eval:.4f}\n"
84                 f"Avg Dice Coeff EVal: {avg_dice_coefficient_eval:.4f}\n"
85                 f"\n{'-'*50}"
86             )
87             click.echo(message="🏁 Evaluation Done!", fg="blue")
88             return
89
90 if __name__ == "__main__":
91     main()
```

```
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
```

Anexo D: Código Predicción

```
1 import torch
2 from torchvision import transforms as T
3 from torch.utils.data import DataLoader, Dataset, random_split
4 from metrics import calculate_metrics, calculate_metrics2, DICE_BCE_Loss, DICE_BCE_Loss2
5 from my_dataset import My_dataloader
6 from modules_DeepLabV3 import DeepLabv3Plus
7 from modules_ResNet import DeepResNet
8 from modules_Unet import Unet
9 from typing import Any
10 import click
11 from tqdm import tqdm
12 import matplotlib.pyplot as plt
13 import torch
14 import matplotlib.pyplot as plt
15 import rasterio
16 import os
17 # Función para mostrar la imagen, máscara y predicción
18 def display_single_image(image, mask, pred):
19     fig, ax = plt.subplots(1, 3, figsize=(20, 6))
20     #fig.tight_layout()
21     # Ajustar el espacio entre los subplots
22     fig.subplots_adjust(wspace=0.001) # Disminuye este valor para acercar las imágenes
23
24     # Si la imagen tiene más de 3 canales, mostrar los primeros 3 como RGB
25     #rgb_image = image[...,:3]
26     rgb_image = image[...,[2, 1, 0]]
27
28     ax[0].imshow(rgb_image)
29     ax[0].set_title('Imagen Original')
30     ax[0].axis('off')
31
32     ax[1].imshow(mask, cmap='gray')
33     ax[1].set_title('Ground Truth')
34     ax[1].axis('off')
35
36     ax[2].imshow(pred, cmap='gray')
37     ax[2].set_title('Unet')
38     ax[2].axis('off')
39
40     # Añadir las métricas en la parte inferior de la figura
41     #fig.suptitle(f'IoU: {iou:.4f}, DICE: {dice_coefficient:.4f}, Pixel Accuracy: {pixel_accuracy:.4f}', fontsize=16, y=0.05)
42     # Añadir las métricas en la parte inferior de la imagen de predicción
43     """
44     ax[2].text(0.5, -0.1, f'IoU: {iou:.4f} DICE: {dice_coefficient:.4f} Pixel Accuracy: {pixel_accuracy:.4f}', transform=ax[2].transAxes, fontsize=12)
45     """
46     plt.show()
47
48 def save_pred_img(array, output_path):
49     with rasterio.open(
50         output_path,
51         'w',
52         driver='GTiff',
53         height=array.shape[0],
54         width=array.shape[1],
55         count=1,
56         dtype='float32' # Asegurarse de que se guarda como float32
57     ) as dst:
58         dst.write(array, 1)
```

```
1 import torch
2 from torchvision import transforms as T
3 from torch.utils.data import DataLoader, Dataset, random_split
4 from metrics import calculate_metrics, calculate_metrics2, DICE_BCE_Loss, DICE_BCE_Loss2
5 from my_dataset import My_dataloader
6 from modules_DeepLabV3 import DeepLabv3Plus
7 from modules_ResNet import DeepResNet
8 from modules_Unet import Unet
9 from typing import Any
10 import click
11 from tqdm import tqdm
12 import matplotlib.pyplot as plt
13 import torch
14 import matplotlib.pyplot as plt
15 import rasterio
16 import os
17 # Función para mostrar la imagen, máscara y predicción
18 def display_single_image(image, mask, pred):
19     fig, ax = plt.subplots(1, 3, figsize=(20, 6))
20     #fig.tight_layout()
21     # Ajustar el espacio entre los subplots
22     fig.subplots_adjust(wspace=0.001) # Disminuye este valor para acercar las imágenes
23
24     # Si la imagen tiene más de 3 canales, mostrar los primeros 3 como RGB
25     #rgb_image = image[...,:3]
26     rgb_image = image[...,[2, 1, 0]]
27
28     ax[0].imshow(rgb_image)
29     ax[0].set_title('Imagen Original')
30     ax[0].axis('off')
31
32     ax[1].imshow(mask, cmap='gray')
33     ax[1].set_title('Ground Truth')
34     ax[1].axis('off')
35
36     ax[2].imshow(pred, cmap='gray')
37     ax[2].set_title('Unet')
38     ax[2].axis('off')
39
40     # Añadir las métricas en la parte inferior de la figura
41     #fig.suptitle(f'IoU: {iou:.4f}, DICE: {dice_coefficient:.4f}, Pixel Accuracy: {pixel_accuracy:.4f}', fontsize=16, y=0.05)
42     # Añadir las métricas en la parte inferior de la imagen de predicción
43     """
44     ax[2].text(0.5, -0.1, f'IoU: {iou:.4f} DICE: {dice_coefficient:.4f} Pixel Accuracy: {pixel_accuracy:.4f}', transform=ax[2].transAxes, fontsize=12)
45     """
46     plt.show()
47
48 def save_pred_img(array, output_path):
49     with rasterio.open(
50         output_path,
51         'w',
52         driver='GTiff',
53         height=array.shape[0],
54         width=array.shape[1],
55         count=1,
56         dtype='float32' # Asegurarse de que se guarda como float32
57     ) as dst:
58         dst.write(array, 1)
```