

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE CIENCIAS AGRARIAS
ESCUELA PROFESIONAL DE INGENIERIA TOPOGRAFICA Y
AGRIMENSURA



ESTIMACIÓN DE ÁREA GLACIAR UTILIZANDO REDES
NEURONALES CONVOLUCIONALES U-NET EN IMÁGENES
MULTIESPECTRALES SENTINEL 2 EN EL GLACIAR

AUSANGATE, 2019

TESIS

PRESENTADA POR:

PERCY ELBIS COLQUE CAILLAHUA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO TOPOGRAFO Y AGRIMENSOR

PUNO – PERÚ

2019

**UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE CIENCIAS AGRARIAS
ESCUELA PROFESIONAL DE INGENIERÍA TOPOGRÁFICA Y AGRIMENSURA
ESTIMACIÓN DE ÁREA GLACIAR UTILIZANDO REDES NEURONALES
CONVOLUCIONALES U-NET EN IMÁGENES MULTIESPECTRALES
SENTINEL 2 EN EL GLACIAR AUSANGATE, 2019**

TESIS PRESENTADA POR:

PERCY ELBIS COLQUE CAILLAHUA

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO TOPOGRAFO Y AGRIMENSOR



APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE

:

.....
Ing. VICTOR CIPRIANO FLORES FLORES

PRIMER MIEMBRO

:

.....
M.Sc. RAUL CORNEJO CALVO

SEGUNDO MIEMBRO

:

.....
Ing. HENRY PETERS CHINO CHOQUE

DIRECTOR / ASESOR

:

.....
M.Sc. VICTOR MANUEL ESPINOZA PINEDO

TEMA : Aplicación de los Métodos GeoEstadísticos al Tratamiento de Información Medioambiental y Topocartográficos

ÁREA : Ciencias Naturales

FECHA DE SUSTENTACIÓN 18 DE NOVIEMBRE DEL 2019

DEDICATORIA

A Dios con mucho amor, que siempre me levanta de mis tropiezos y siempre está a mi lado guiándome por el camino correcto.

Para mi Madre quien inculco en mí el hábito de la perseverancia, Para mi Padre quien me incentivo a ver la inmensidad del cielo, Incentivo mi curiosidad por las ciencias.

Para todas las personas del mundo que no les importa errar sueñan muy alto hasta llegar a las estrellas y quieren un mundo mejor.

Para ti música que siempre estás en mis bajos y los momentos altos de mi vida.

Percy Elbis

AGRADECIMIENTOS

A la plana Docente de la Universidad Nacional del Altiplano, Facultad de Ciencias Agrarias, Escuela Profesional de Ingeniería Topográfica y Agrimensura por ser parte de mi formación profesional a todos los docentes que me brindaron sus conocimientos y valores.

Agradezco a mi director y asesor M.Sc. Víctor Manuel Espinoza Pinedo por su orientación hacia un norte en esta Investigación.

A los miembros del Jurado, por sus instrucciones y orientaciones lo cual permitieron la culminación de este proyecto de investigación.

Agradezco a todos los desarrolladores y usuarios de todo el mundo de Google Earth Engine en especial Noel Gorelick.

Agradezco a todo el equipo de la Empresa de tecnología Google quien con sus servicios ha permitido concluir esta tesis.

ÍNDICE GENERAL

RESUMEN	12
ABSTRACT	13
I. INTRODUCCIÓN	14
1.1. PLANTEAMIENTO DEL PROBLEMA	15
1.2. FORMULACION DEL PROBLEMA	16
1.2.1. Problema General	16
1.2.2. Problemas Específicos	16
1.3. HIPOSTESIS DE LA INVESTIGACION	16
1.3.1. Hipótesis General	16
1.3.2. Hipótesis Específicos	16
1.4. JUSTIFICACION DE LA INVESTIGACION	16
1.5. OBJETIVOS DE LA INVESTIGACION	17
1.5.1. Objetivo General	17
1.5.2. Objetivos Específicos	17
II. REVISIÓN DE LITERATURA	18
2.1. ANTECEDENTES	18
2.2. MARCO TEORICO	20
2.1.1. Glaciar	20
2.1.2. Fundamentos de Percepción Remota Óptica	24
2.1.3. Índices Espectrales	29

2.1.4. Inteligencia Artificial	29
2.2.5. Redes Neuronales Convolucionales.....	31
2.2.6. El Programa Copérnico.....	41
2.2.7. Sistemas de Información Geográfica	44
2.2.8. Computación en la nube	44
III. MATERIALES Y MÉTODOS.....	47
3.1. UBICACIÓN DEL ÁREA DE ESTUDIO	47
3.1.1. Ubicación Política:.....	47
3.1.2. Ubicación Geográfica	47
3.2. MATERIALES E INSTRUMENTOS.....	48
3.2.1. Materiales.....	48
3.2.2. Hardware.....	48
3.2.3. Software	48
3.2.4. Librerías	48
3.2.5. Servicios.....	48
3.3. METODOLOGÍA PARA EL DESARROLLO DEL TRABAJO	49
3.3.1. Tipo de Investigación.....	49
3.3.2. Población y Muestra	49
3.3.3. Variables	50
3.3.4. Recolección de Datos.....	52
3.3.5. Procedimientos.....	57

IV. RESULTADOS Y DISCUSIÓN	75
4.1. Prueba T-Student para Muestras Relacionadas	75
4.2. Discusión	79
V. CONCLUSIONES	81
VI. RECOMENDACIONES	82
VII. REFERENCIAS	83
ANEXOS	91

ÍNDICE DE FIGURAS

Figura N° 1 Flujo glaciar desde la zona de acumulación a la de ablación.	21
Figura N° 2 Distribución de los glaciares en los andes tropicales.....	23
Figura N° 3 Espectro electromagnético basada en el rango de longitud de onda.....	25
Figura N° 4 Interacciones Energéticas en la atmosfera y en la superficie terrestre.....	26
Figura N° 5 Transmisión Atmosférica.....	27
Figura N° 6 Ventanas Atmosféricas en Imágenes Ópticas Multiespectrales.....	27
Figura N° 7 Curvas de reflectancia espectral de Diferentes Coberturas.....	28
Figura N° 8 Redes Neuronales Profundas totalmente conectadas.....	31
Figura N° 9 Funciones de Activación.....	33
Figura N° 10 Descenso del Gradiente.....	35
Figura N° 11 Flujo para clasificación de Imágenes con CNN.....	38
Figura N° 12 Proceso de la Convolución.....	38
Figura N° 13 Proceso de Agrupación máxima.	39
Figura N° 14 Segmentación Semántica	40
Figura N° 15 La Arquitectura U-Net	41
Figura N° 16 Nivel de Procesamiento en Sentinel 2.	42
Figura N° 17 Un diagrama Simplificado de la arquitectura del sistema GEE.....	46
Figura N° 18 Mapa de Ubicación del Área de Estudio.....	47
Figura N° 19 Vista del Glaciar Ausangante.....	49
Figura N° 20 Librería Espectral USGS y Firma Espectral de Nieve en Sentinel 2.....	51

Figura N° 21 Firma Espectral de Coberturas Terrestres en Sentinel 2.....	52
Figura N° 22 Puntos GPS para Validación.....	53
Figura N° 23 Recolección de puntos de entrenamiento y Validación.....	54
Figura N° 24 Firmas Espectrales Aleatorias del Dataset Entrenamiento.....	55
Figura N° 25 Imagen RGB y Su Target Correspondiente.....	56
Figura N° 26 Flujo de Trabajo.....	58
Figura N° 27 Almacenamiento de Dataset en Google Cloud Storage.....	63
Figura N° 28 Curva de Aprendizaje Durante el Entrenamiento.....	72
Figura N° 29 Estimación de área glaciar Ausangate.....	73
Figura N° 30 Área glaciar por el Algoritmos NDSI>0.4 en Diferentes Fechas.....	74
Figura N° 31 Diagrama de Cajas.....	75
Figura N° 32 Comparación U-Net y NDSI Fecha 2019/07/09.....	78
Figura N° 33 Serie de Tiempo en las 12 escenas en el Ausangate.....	78
Figura N° 34 Prueba en algunos glaciares importantes del Perú.....	80
Figura N° 35 Recolección de Puntos GPS para Validación.....	91

ÍNDICE DE TABLAS

Tabla N° 1 Cantidad y superficie de glaciares por cordilleras y altitudes	24
Tabla N° 2 Nivel de Procesamiento en Sentinel 2.....	43
Tabla N° 3 Bandas espectrales para los sensores SENTINEL-2 (S2A y S2B).	43
Tabla N° 4 Especificaciones Técnicas satélite Sentinel 2.	44
Tabla N° 5 Operacionalización de variables.....	50
Tabla N° 6 Reflectancia BOA en Coberturas Terrestres.	51
Tabla N° 7 Observaciones GPS en Campo.....	53
Tabla N° 8 Equivalencias de los atributos.	54
Tabla N° 9 Cantidad de Muestras.	54
Tabla N° 10 Imágenes Utilizadas para las pruebas estadísticas.	56
Tabla N° 11 Bandas que se utilizaron en el entrenamiento.	62
Tabla N° 12 Definición y Operación en cada Capa.....	68
Tabla N° 13 Tiempo de entrenamiento.....	70
Tabla N° 14 Resumen de los argumentos Utilizados para el Modelo U-net.	71
Tabla N° 15 Resultados de la Precisión del Modelo.	71
Tabla N° 16 Áreas Obtenidas por el Método NDSI>0.4 y U-Net.....	75

ÍNDICE DE ACRÓNIMOS

CNN: Convolutional Neural Network (Redes Neuronales Convolucionales).

ESA: Agencia Espacial Europea.

TOA: Top Of Atmosphere (Techo de la Atmosfera).

BOA: Bottom Of Atmosphere.

GEE: Google Earth Engine.

API: Interface de Programación de Aplicaciones.

IDE: Entorno de desarrollo Integrado.

GCP: Google Cloud Platform.

GCS: Google Cloud Storage.

NDSI: Índice de Nieve de Diferencia Normalizada.

AI: Inteligencia Artificial.

MSI: Instrumento Multiespectral.

NIR: Infrarrojo Cercano.

SWIR: Infrarrojo de Onda Corta.

ML: Machine Learning (Aprendizaje Automático).

DL: Deep Learning (Aprendizaje Profundo).

CONV: Capa Convolutiva.

POOL: Capa de Agrupamiento.

BN: Batch Normalization (Normalización por Lotes).

FC: Capas Totalmente Conectadas.

RESUMEN

La Investigación se realizó en el glaciar Ausangate ubicado entre las provincias de Canchis y Quispicanchis en la región del Cusco. Se probó el modelo U-Net en la estimación de área glaciar con imágenes multiespectrales Sentinel 2. El proceso de mapear glaciares por lo general se utiliza el Índice de Nieve de Diferencia Normalizada (NDSI). Con un umbral estándar de 0.4, la limitación que tiene es sobreestimar, es necesario eliminar polígonos que no pertenecen al área glaciar como lagunas glaciares, este proceso puede ser lento y estresante. Esta investigación propone utilizar las redes neuronales convolucionales (CNN) con una arquitectura U-Net como un método de mapeo en glaciares. Los algoritmos de Inteligencia Artificial como las CNN necesitan datos de entrenamiento y validación los cual es óptimo utilizar La infraestructura computacional de Google Earth Engine (GEE) para recolectar muestras de imágenes del satélite Sentinel 2A y 2B. En esta Investigación se recolecto 1883 muestras para el entrenamiento y 1470 para validación en el conjunto de glaciares tropicales del Perú. Para evaluar el rendimiento del modelo se eligieron 12 escenas de prueba de fechas que estén fuera del entrenamiento desde el 2016 al 2019 en el glaciar Ausangate. Los resultados muestran que el modelo U-Net logro precisiones de 97.89% en entrenamiento y 97.88 % en validación demostrando ser más robusto en mapear área glaciar y puede utilizar para cualquier glaciar, superando al $NDSI > 0.4$ donde se observó que sobrestima y confunde con lagunas glaciares y tiene que intervenir la eliminación manual de polígonos de lagunas glaciares.

Palabras Clave: U-Net, Glaciar, NDSI, Percepción Remota, Inteligencia Artificial.

ABSTRACT

The research was carried out in the Ausangate glacier located between the provinces of Canchis and Quispicanchis in the Cusco region. The U-Net model was tested in the estimation of glacier area with Sentinel 2 multispectral images. The process of mapping glaciers usually uses the Normalized Difference Snow Index (NDSI). With a standard threshold of 0.4, the limitation it has is overestimating, it is necessary to eliminate polygons that do not belong to the glacier area such as glacial lagoons, this process can be slow and stressful. This research proposes to use convolutional neural networks (CNN) with a U-Net architecture as a method of mapping in glaciers. Artificial Intelligence algorithms such as CNN need training and validation data which is optimal to use Google Earth Engine (GEE) computing infrastructure to collect samples of Sentinel 2A and 2B satellite imagery. In this Research, 1883 samples were collected for training and 1470 for validation in the set of tropical glaciers in Peru. To evaluate the performance of the model, 12 test scenes of dates were chosen that are out of training from 2016 to 2019 on the Ausangate Glacier. The results show that the U-Net model achieved accuracies of 97.89% in training and 97.88% in validation proving to be more robust in mapping glacier area and can be used for any glacier, exceeding the $NDSI > 0.4$ where it was observed that it overestimates and confuses with gaps glaciers and the manual removal of polygons from glacial lagoons has to intervene.

Keywords: U-Net, Glacier, NDSI, Remote Sensing, Artificial Intelligence.

I. INTRODUCCIÓN

Una tarea común en la comunidad de glaciología es el análisis de imágenes satelitales como segmentar el área glaciar, Monitorear el retroceso glaciar, Series temporales y otros. El cartografiado en glaciares tropicales donde el cambio climático está afectando donde se están incrementado lagunas y bofedales por el derretimiento del hielo glaciar.(Turpo Cayo & Maria Chaves, 2017).Es importante estar constantemente monitoreando los glaciares para tomar decisiones Inteligentes para el presente y futuro de reservas de agua. Muchos investigadores concuerdan que los trabajos futuros se deben centrar en sistematizar los proceso de extracción de los límites de los glaciares en las imágenes satelitales y se pueda tener una base de datos de glaciares más consistente y de mayor calidad que se pueda usar para muchos propósitos científicos.(Racoviteanu, Paul, Raup, Khalsa, & Armstrong, 2009). El gran número de imágenes espectrales-temporales de alta calidad capturadas desde satélites de observación de la Tierra, Se están convirtiendo en una tecnología fundamental para la estimación de coberturas terrestres. (Ji, Zhang, Xu, Shi, & Duan, 2018). Los algoritmos de aprendizaje profundo (DL), se están introduciendo recientemente en la comunidad de ciencia de la tierra y percepción remota en el Análisis de Big Data.(Zhang, Zhang, & Du, 2016). Las redes neuronales convolucionales (CNN) han sido muy empleadas para resolver problemas de visión por computador y análisis de imágenes médicas.(Milletari, Navab, & Ahmadi, 2016), Actualmente las CNN están dando el cambio radical hacia una mayor automatización, viene con muchas promesas, incluidos resultados de mayor calidad y menos esfuerzo de ingeniería.(Stoian, Poulain, Inglada, Poughon, & Derksen, 2019). El potencial para clasificar imágenes multiespectrales no se han explotados a fondo, en particular en percepción remota óptica.(Mahdianpari, Salehi, Rezaee, Mohammadimanesh, & Zhang, 2018). La literatura muy reciente ha demostrado el rendimiento superior de las redes neuronales convolucionales profundas para muchas tareas de clasificación, incluido el análisis automático de imágenes geoespaciales de resolución espacial muy alta.(Postadjian, Le Bris, Sahbi, & Mallet, 2017).

El objetivo general de esta investigación es probar el modelo U-Net en la estimación de área glaciar con imágenes multiespectrales Sentinel 2. La nieve y el hielo tiene su propia reflectancia el cual es característico y en esta investigación se entrenara

su firma espectral en diferentes longitudes de onda que tiene el sensor Multispectral (MSI) del satélite Sentinel 2. Como objetivos específicos, Entrenar y validar la arquitectura U-Net. Toda la investigación está desarrollado con herramientas libres esto con el fin de evitar la dependencia del propietario, estimular la innovación tecnológica, facilitar la conectividad y rebajar costes. La ventaja más sobresaliente es que nos permite usar las aplicaciones de forma legal y gratuita. Ello elimina la situación de ilegalidad. (Rius & González, 2011).

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la comunidad científica de glaciología en el mundo utiliza el índice espectral NDSI como técnica para segmentar glaciares. Muchos aun lo siguen usando y otros desarrollaron modificaciones del NDSI según a su región. En el Perú la entidad responsable de monitorear los glaciares es el Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña (INAIGEM) hace público un manual metodológico de inventario nacional de glaciares en el cual utilizan el Índice espectral NDSI con un umbral estándar de 0.4. Es muy usado para temas de segmentar superficies de nieve, generalmente utiliza un umbral estándar 0.4 y otros umbrales 0.45, 0.49 en los glaciares andinos tropicales del Perú cuando se aplica el umbral 0.4 se confunden con lagunas alrededor del glaciar, esto tiene la necesidad de verificar visualmente con una composición RGB generalmente es de SWIR1/NIR/RED en donde la nieve se caracteriza muy bien y se tiene que hacer una supervisión visual y eliminación de las lagunas manualmente, todo este proceso toma tiempo, si se trata de inventariar todos los glaciares de los Andes Peruanos es muy laborioso, estresante y puede ser susceptible a errores.

Esta investigación propone probar las Redes neuronales Convolucionales (CNN), como un método de cartografiado de áreas glaciares, la cantidad de datos es enorme en la constelación Sentinel 2, utilizar como conjunto de entrenamiento y validación, proponemos utilizar servicios y software en internet de esta manera ahorrar costo computacional, las CNN han ganado popularidad en la clasificación de imágenes médicas, visión por computador, Conducción Autónoma, y muchos otros campos de la ciencia. En la ciencia de la Percepción Remota no ha sido muy utilizada las técnicas de aprendizaje profundo.

1.2. FORMULACION DEL PROBLEMA

1.2.1. Problema General

¿Será posible Estimar el área glaciar con redes neuronales convolucionales U-Net en Imágenes Multiespectrales Sentinel 2?

1.2.2. Problemas Específicos

¿La Recolección de muestras en la plataforma Google Earth Engine es rápido?

¿Cómo entrenar, validar y predecir con la red U-Net?

¿Las Redes Neuronales Convolucionales U-Net serán más eficientes en la estimación de Área Glaciar que el Índice de Nieve de Diferencia Normalizada (NDSI)?.

1.3. HIPOSTESIS DE LA INVESTIGACION

1.3.1. Hipótesis General

Las Redes Neuronales Convolucionales U-Net es un estimador muy eficiente de área glaciar.

1.3.2. Hipótesis Específicos

Google Earth Engine es potente para extraer conjunto de muestras a una velocidad muy rápida.

Separar el Dataset para entrenar, validar y predecir con la red U-Net.

Las Redes Neuronales Convolucionales U-Net son más eficientes que el NDSI en la estimación de área glaciar más eficiente.

1.4. JUSTIFICACION DE LA INVESTIGACION

El conocimiento de la extensión actual de la superficie glaciar es elemental para la gestión de recursos hídricos, monitoreo de superficie glaciar, series de tiempo, etc. y deben ser cartografiados de manera correcta. El problema de estimar el área de la superficie glaciar con el algoritmo NDSI desarrollado por (Dozier, 1989) es un método simple y establecido para identificar la capa de nieve en imágenes satelitales ópticas

Landsat. Las limitaciones de mapear con el algoritmo NDSI utilizan un umbral para determinar que un píxel está cubierto de nieve o sin nieve. Casi en todas las investigaciones lo toman como estándar 0.4 y en otros consideran un umbral dinámico de 0.35 a 0.49 y viendo cual da mejores resultados de acuerdo a su criterio y finalmente se tiene que digitalizar manualmente aquellos objetos que no son realmente glaciar como laguna glaciar, sombras lo cual lleva tiempo y muchas veces el ojo humano comete el error de confundirse.

La importancia de esta investigación es utilizar algoritmos de aprendizaje profundo como las redes neuronales convolucionales U-Net como una alternativa para el cartografiado en imágenes multiespectrales y automatización. Hoy en día las CNN han solucionado muchos problemas que antes eran imposible como la conducción autónoma, visión robótica, detección de enfermedades con imágenes médicas, etc. La cual muchos investigadores concuerdan que la Inteligencia Artificial será la cuarta revolución industrial. Además es bueno recordar que existen métodos más precisos como la adquisición de datos por métodos topográficos, geodésicos y fotogramétricos que son altamente costosos y demorosos. La percepción remota óptica es una herramienta eficientes en cuanto a costos y tiempo y cada vez hay más satélites multiespectrales con una alta resolución espacial actualmente tenemos Sentinel 2 de libre acceso con una resolución espacial de 10 metros y una resolución temporal de 5 días en los satélites 2A y 2B.

1.5. OBJETIVOS DE LA INVESTIGACION

1.5.1. Objetivo General

- Probar la arquitectura U-Net en la estimación de área glaciar con imágenes multiespectrales Sentinel 2.

1.5.2. Objetivos Específicos

- Recolectar Conjunto de muestras en Google Earth Engine.
- Entrenar, Validar y Predecir con el modelo U-Net.
- Analizar Diferencias de estimación de áreas entre U-Net y el algoritmo NDSI.

II. REVISIÓN DE LITERATURA

2.1. ANTECEDENTES

En el artículo denominado, “Evaluation of Remote Sensing Techniques for Ice-Area Classification Applied to the Tropical Quelccaya Ice Cap, Peru” Menciona que la digitalización a mano tiene un problema significativo, es la decisión subjetiva como de cuales pixeles incluir como hielo, esto es un problema menor en el interior de la capa de hielo pero se vuelve problemático a lo largo de los márgenes del hielo. Donde el hielo se encuentra con otros materiales circundantes ,los pixeles tiene probabilidad de ser mezclas de hielo con otros materiales.(Albert, 2002)

En la tesis denominada, “Análisis de la evolución de las lagunas de alta montaña en la cordillera del Vilcabamba (Cusco y Apurímac) entre los años 1991-2014 mediante métodos de sensoramiento remoto y SIG”. La investigación utilizó imágenes de satélite de la serie Landsat donde aplico el algoritmo Normalized Difference Snow Index (NDSI) para el cartografiado de los glaciares, Usando los siguientes umbrales para el NDSI de 0.35 y 0.4.(Guardamino Soto, 2014)

En la tesis denominado, “Determinación del Retroceso Glaciar de la Cordillera de la Raya Mediante Imágenes Satelitales Landsat,1975-2013” Donde la metodología que se utilizó para obtener área de los glaciares fue el método de digitalización visual de la superficie glaciar con una combinación de bandas R/G/B 7/4/2 en la imagen satelital Landsat 5 y 7.(Turpo cayo, 2014)

En el artículo denominado “Using atmospherically-corrected Landsat imagery to measure glacier area change in the Cordillera Blanca, Peru from 1987 to 2010”. Esta investigación se explora los efectos de la atmósfera y las correcciones topográficas al comparar las estimaciones de área glaciar sin escombros. Utilizando Imágenes Crudas e Imágenes Corregidas Atmosféricamente y topográficamente, Donde se aplicó el algoritmo NDSI para los tres tipos de contextos. Los resultados indican, que las correcciones pueden tener un impacto significativo en el área cuando se aplica el mismo umbral. Las estimaciones del área glaciar libre de escombros derivadas de escenas no corregidas son aproximadamente un 5% menos que las estimaciones del área glaciar libre de escombros derivadas de escenas corregidas atmosféricamente. Se determinó que las

estimaciones del área glaciario libre de escombros son más sensibles a la elección del umbral y los efectos topográficos. Respecto al Landsat 2 solo se digitalizó el contorno del glaciario visualmente ya que no contiene en el espectro la banda 5 o SWIR. (Burns & Nolin, 2014)

En el Artículo “A comparison of Normalised Difference Snow Index (NDSI) and Normalised Difference Principal Component Snow Index (NDPCSI) techniques in distinguishing snow from related land cover types” En este trabajo de investigación se propone una nueva técnica de cartografiado PCA con NDSI, denominado Índice de nieve del componente principal de diferencia normalizada (NDPCSI) para mejorar la precisión del mapeo de nieve. Los resultados muestran que tanto NDPCSI como NDSI con altas precisiones de clasificación de 84.9% y 76.8%, respectivamente, fueron efectivos en el mapeo de la nieve. Los resultados del estudio también indican que el NDSI fue sensible a los cuerpos de agua que se encuentran en los terrenos más bajos dentro del área de estudio, mientras que el PCA fue capaz de des correlacionar la nieve de los cuerpos de agua y sombras. Aunque el NDSI y el NDPCSI produjeron resultados comparables, el NDPCSI fue capaz de mapear la nieve de otras coberturas relacionadas con la tierra con mayor precisión. La superioridad del NDPCSI se puede atribuir particularmente a la capacidad del análisis de componentes principales para des correlacionar la nieve de cuerpos de agua y sombras. La precisión de ambas técnicas se evaluó utilizando una banda pancromática Landsat 8 de resolución espacial más alta y los datos del espectro radiómetro de imágenes de resolución moderada (MODIS) adquiridos el mismo día. Se sugieren que NDPCSI es una alternativa viable en el mapeo de la nieve, especialmente en paisajes heterogéneos que incluyen cuerpos de agua. (Sibandze, Mhangara, Odindi, & Kganyago, 2014)

En el artículo denominado “On the need for a time-and location-dependent estimation of the NDSI threshold value for reducing existing uncertainties in snow cover maps at different scales” El área cubierta de nieve a menudo se estima utilizando el índice de nieve de diferencia normalizada (NDSI). Sin embargo la representatividad espaciotemporal del umbral estándar de 0.4 es cuestionable a escala local. El análisis demostró que el umbral del NDSI en sitios diferentes no está correlacionado ($r = 0.17$) y demostró que existen grandes incertidumbres en la predicción del área cubierta de nieve de hasta el 24.1% en los mapas de cobertura de nieve satelital en los casos en que se

utiliza el umbral estándar de 0.4, pero un modelo polinomial cuadrático calibrado desarrollado explica la dinámica de los umbrales estacionales puede reducir este error. La investigación ha revelado que el uso del umbral estándar de 0.4 es adecuado para productos satelitales con un tamaño de pixel de 500 metros a más.(Härer, Bernhardt, Siebers, & Schulz, 2018)

En el artículo denominado “Detection of Glacier Calving Margins with Convolutional Neural Networks: A Case Study Yara” El mapeo continuo y preciso de los frentes de nacimiento de glaciares es esencial para monitorear y comprender los rápidos cambios de glaciares en la Antártida y Groenlandia, Esta Investigación utilizó técnicas de segmentación semántica utilizando redes neuronales convolucionales (CNN) con una arquitectura U-Net ha sido entrenado con un conjunto de datos de imágenes y sus correspondientes frentes de parto determinados manualmente con imágenes Landsat 5, 7 y 8. La red neuronal fue capaz de identificar el frente de parto en nuevas imágenes con una desviación media de 96.3 m de los frentes reales, equivalente a 1.97 píxeles en promedio, mientras que el error correspondiente para frentes determinados manualmente en las mismas imágenes de resolución es de 92.5 m (1.89 píxeles). Encontramos que la red neuronal entrenada supera significativamente las técnicas de detección de bordes comunes y se puede utilizar para mapear continuamente los frentes de hielo de parto con una variedad de productos de datos.(Mohajerani et al., 2019).

En el artículo denominado “Using the U-net convolutional network to map forest types and disturbance in the Atlantic rainforest with very high resolution images”. La red U-Net segmenta tipos de bosques a escala regional con una precisión general >95% utilizando bandas espectrales RGB con resolución espacial de 0.3 metros del satélite WorldView-3.(Wagner et al., 2019).

2.2. MARCO TEORICO

2.1.1. Glaciar

Los glaciares son masa de hielo ubicados en tierra firme, proveniente de la nieve compactada y recristalizada, poseen movimiento lento por su propio peso ayudado por la acción de la gravedad.(Rivera Mantilla, 2005). Los glaciares están formados por cristales de hielo, nieve, aire, agua y roca. Se formarán cada vez que una masa de nieve se acumule

sobre años sucesivos, compactos, y se convierte en hielo. Esto puede ocurrir en cualquier zona climática en la que la entrada de nieve supere la velocidad a la que se derrite, pero generalmente los glaciares se encuentran en altas montañas, latitudes altas y en lados occidentales de los continentes. (Huddart & Stott, 2010)

La dinámica de un glaciar está gobernada fundamentalmente por los siguientes procesos: mencionados por (European Space Agency, 2013).

- La acumulación de nieve y hielo en las partes superiores del glaciar.
- El descenso de esta masa de hielo hacia la parte inferior del glaciar.
- La ablación del hielo.
- La formación de torrentes de agua por el deshielo o el transporte del hielo/agua evaporada a través de la atmósfera.

La acumulación comprende todos los procesos que conducen a un aumento local de la masa del glaciar. La nieve acumulada se transforma en neviza (es decir, nieve antigua y compactada) y finalmente en hielo. La ablación incluye todos los fenómenos que provocan una pérdida local de la masa del glaciar. Dado que los glaciares tienden a equilibrar la acumulación y ablación de su masa en condiciones estables, ésta fluye lentamente desde la zona de acumulación a la de ablación. (European Space Agency, 2013).

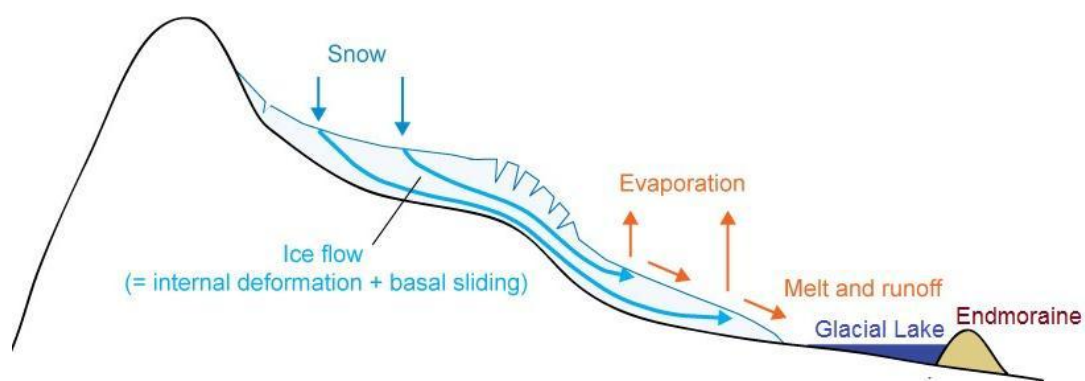


Figura N° 1 Flujo glaciar desde la zona de acumulación a la de ablación.

Fuente: (European Space Agency, 2013).

2.2.2. Laguna Glaciar

Un Lago glaciar se basa en dos perspectivas de glaciación y agua de deshielo glacial. Entonces un lago glaciar se podría definir como el agua natural suministrada principalmente por el agua de deshielo glaciar.(Yao, Liu, Han, Sun, & Zhao, 2018).Estos pueden formarse en una variedad de situaciones diferentes. Las lagunas glaciares pueden ser marginales, represadas por los glaciares en valles libre de hielo, también pueden ubicarse en frente de un margen glaciar se le denomina como laguna pro-glaciar, o pueden formarse en lugares donde el hielo enterrado se ha derretido para dar una depresión de deshielo.(Huddart & Stott, 2010). Los lagos glaciares no solo están estrechamente relacionados con el cambio climático y el movimiento de los glaciares, sino que también desempeñan un papel en la cadena de desastres de las montañas. Por lo tanto, muchos científicos y gobiernos han prestado cada vez más atención a los lagos glaciares.(Yao et al., 2018). A nivel internacional, hasta ahora no ha habido un estándar aceptado para el sistema de clasificación de los lagos glaciares. Algunas organizaciones y académicos propusieron los diferentes sistemas de clasificación de los lagos glaciares de acuerdo con sus propios fines de investigación.(Yao et al., 2018).

2.2.3. Glaciares Andinos Tropicales

La zona tropical de Sudamérica se puede dividir en dos sub-áreas basándose en las características de precipitación y la ubicación geográfica: Los trópicos interiores (Venezuela, Colombia y Ecuador) donde la ablación y acumulación de los glaciares ocurren simultáneamente todo el año, sin estacionalidad de las precipitaciones y los trópicos exteriores (Perú, Bolivia y el norte de Chile), donde la estación seca se produce de mayo a setiembre, con condiciones subtropicales y la estación lluviosa de octubre a marzo, con prevalencia de condiciones tropicales.(Veettil et al., 2016).

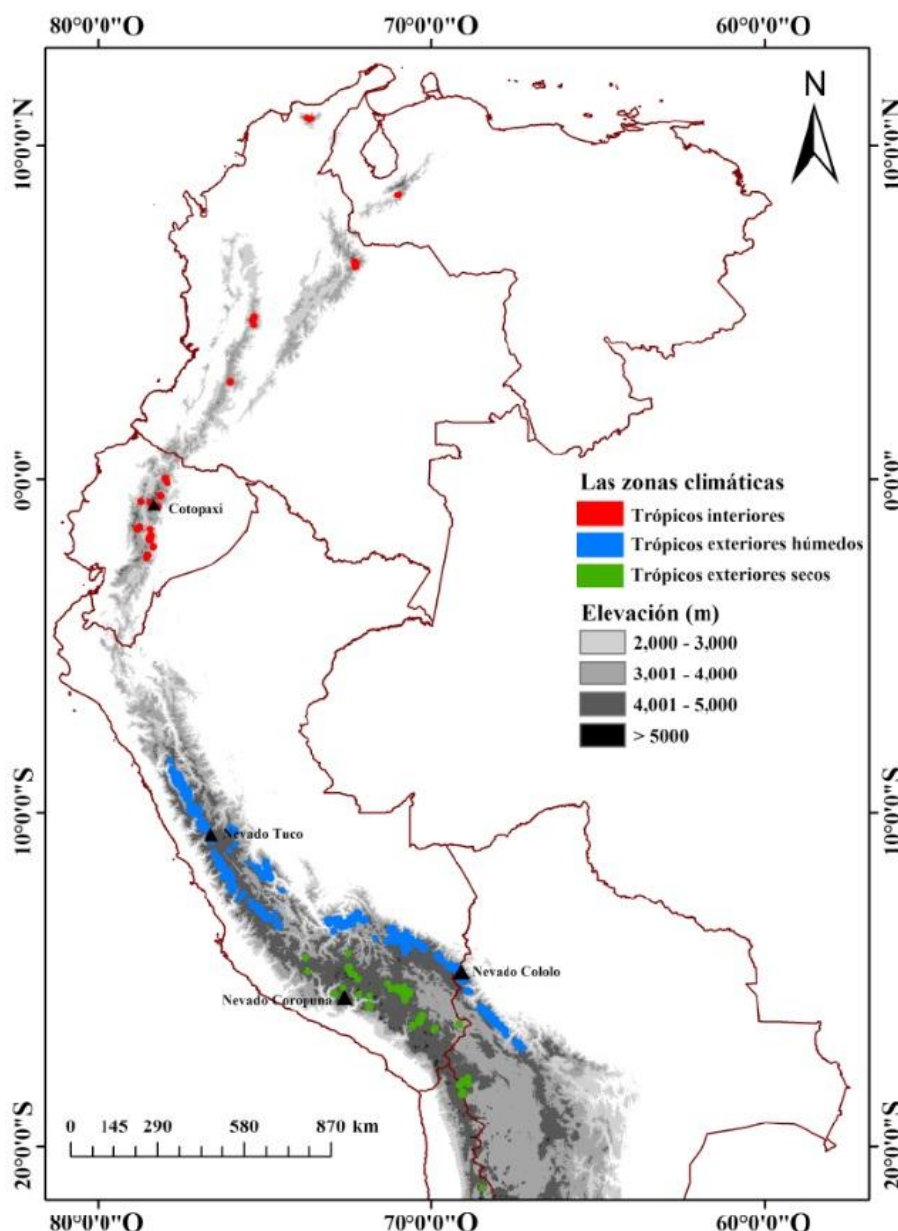


Figura N° 2 Distribución de los glaciares en los andes tropicales.

Fuente: (Veettil et al., 2016)

En el territorio Peruano se encuentra el 71% de los glaciares tropicales, Durante los últimos 54 años, como resultado del cambio climático, los glaciares tropicales andinos peruanos se han reducido en promedio en un 53.56%. (INAIGEM, 2017).

N°	Cordillera	Cantidad	Superficie	Año	Altitud máxima	Altitud mínima
		N°	Km ²		msnm	msnm
1	Blanca	755	527,62	2003	6701	4249
2	Huallanca	43	7,01	2007	5300	4750
3	Huayhuash	144	55,27	2007	6617	4303
4	Raura	102	28,34	2007	5701	4620
5	Huagoruncho	41	9,71	2009	5719	4624
6	La Viuda	65	6,03	2007	5406	4892
7	Central	174	51,91	2007	5888	4713
8	Huaytapallana	105	26,40	2009	5555	4532
9	Chonta	29	1,40	2009	5272	4923
10	Ampato	65	60,96	2010	6374	5044
11	Urubamba	117	26,39	2009	5800	4413
12	Vilcabamba	355	129,15	2009	6200	4187
13	Huanzo	31	4,51	2010	5445	5065
14	Chila	22	0,93	2010	5855	5224
15	La Raya	25	3,06	2010	5468	5107
16	Vilcanota	374	279,40	2009	6364	4592
17	Carabaya	148	34,53	2009	5804	4676
18	Apolobamba	69	45,25	2010	6000	4835
19	Volcánica	15	0,72	2009	6055	5548
Total		2 679	1 298,59			

Tabla N° 1 Cantidad y superficie de glaciares por cordilleras y altitudes

Fuente: (Autoridad Nacional del Agua, 2014)

2.1.2. Fundamentos de Percepción Remota Óptica

2.1.2.1. Percepción Remota

La percepción remota es una ciencia que tiene como objetivo obtener información de la superficie terrestre mediante la medición de las respuestas de las interacciones de la radiación electromagnética con los materiales terrestres.(Almeida & Meneses, 2012)

Una definición más clásica define a la percepción remota como una técnica que permite adquirir imágenes de la superficies terrestre desde sensores instalados en plataforma espaciales esto involucra interacción energética entre la tierra y el sensor.(Chuvienco, 1995)

2.1.2.2. Energía Electromagnética

La energía electromagnética (EM) viaja en ondas y abarca un amplio espectro desde ondas de radio muy largas hasta rayos gamma muy cortos. El ojo humano solo puede detectar una pequeña parte de este espectro llamado luz visible.(National Aeronautics and Space Administration, 2010).La radiación electromagnética puede ser onda o energía radiante que contiene partículas llamadas fotones.(Tempfli et al., 2009).

La radiación electromagnética viaja a través del espacio en forma de perturbaciones periódicas de los campos eléctricos y magnéticos que oscilan simultáneamente en planos perpendiculares entre si y en la dirección de propagación a través del espacio a la velocidad de la luz ($c = 2.99792458 \times 10^8 \text{ m/s}$). (Malo, Tuia, Gómez-Chova, Camps-Valls, & Jiménez, 2011)

2.1.2.3. Espectro electromagnético

Conjunto de todas las ondas electromagnéticas dispuestas según la frecuencia o a la longitud de onda. (Malo et al., 2011)

$$c = \lambda * \nu$$

Donde:

c = Velocidad de la luz. (m/s)

λ = Longitud de onda (m)

ν = Frecuencia (Hz)

El espectro se divide en regiones basadas en la longitud de onda que varía desde las rayos gamma que tienen longitudes de onda de $10^{-6} \mu\text{m}$, hasta las ondas de radio las más largas de longitud de onda de muchos kilómetros. (Malo et al., 2011).

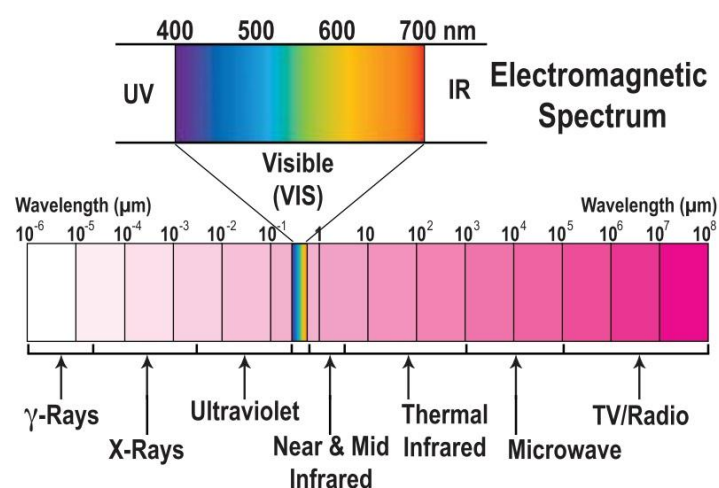


Figura N° 3 Espectro electromagnético basada en el rango de longitud de onda.

Fuente : (Malo et al., 2011)

La luz visible (VIS) se compone de longitudes de onda que varían de 400 a 700 nm, es decir, de azul a rojo. Esta porción estrecha del espectro es el rango completo de la energía electromagnética a la que es sensible el sistema visual humano. Un poco más allá del extremo rojo de la región visible (VIS) está la región de las ondas de energía infrarroja (IR): Infrarrojo cercano (NIR), infrarrojo de onda corta (SWIR), infrarrojo medio (MIR) y la energía térmica. Infrarrojo Térmico (TIR). (Malo et al., 2011)

Las regiones VIS hasta SWIR se usan comúnmente en la percepción remota óptica ya que dependen del sol como fuente de iluminación las longitudes ondas más usadas son de 400 nm a 3000 nm. (Malo et al., 2011).

Interacción energética en la Atmosfera

Antes de que la energía del sol llegue a la superficie de la tierra, ocurren tres interacciones relevantes en la atmosfera: Absorción, transmisión y dispersión. (Tempfli et al., 2009)

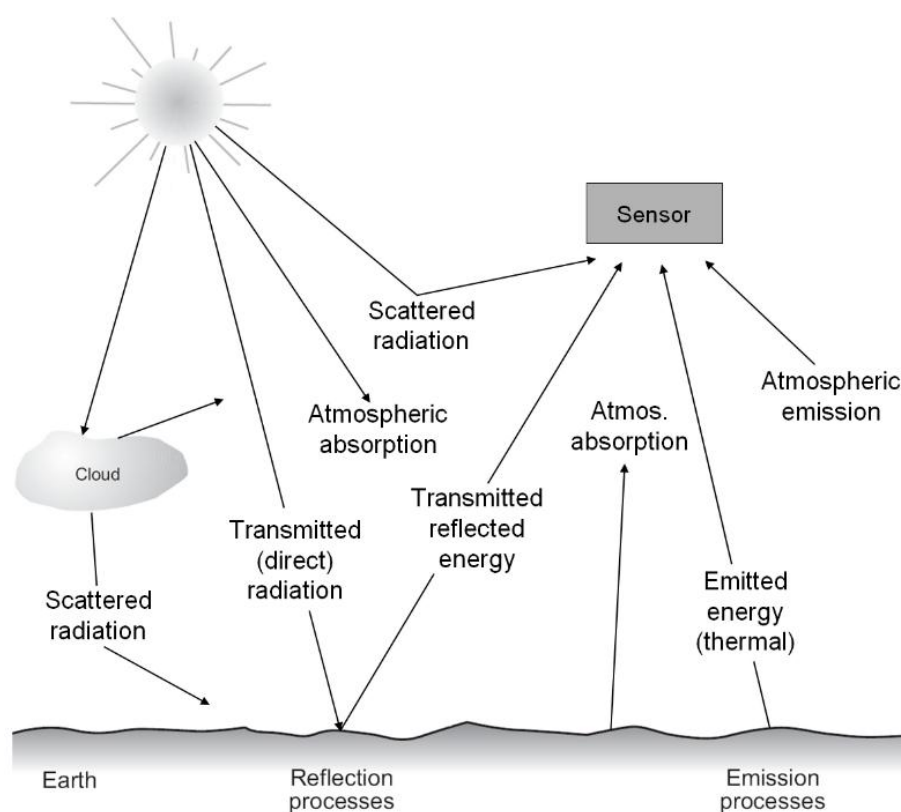


Figura N° 4 Interacciones Energéticas en la atmósfera y en la superficie terrestre.

Fuente : (Tempfli et al., 2009).

-Absorción y Transmisión

La EM que viaja a través de la atmosfera es parcialmente absorbida por varias moléculas. Los absorbentes más eficientes de la radiación solar en la atmosfera son el ozono (O₃), el vapor de agua (H₂O) y el dióxido de carbono (CO₂). (Tempfli et al., 2009)

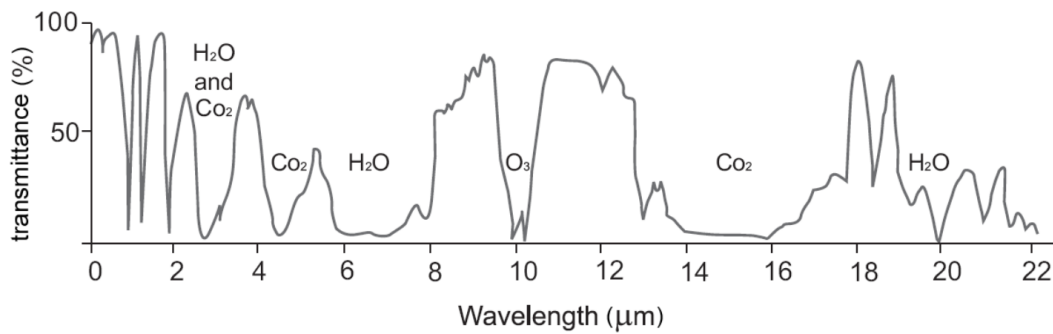


Figura N° 5 Transmisión Atmosférica.

Fuente : (Tempfli et al., 2009).

En la Figura N° 5 se puede observar que muchas de las longitudes de onda no son útiles, porque ninguna de las energías puede penetrar la atmosfera. La notación química (CO₂, H₂O, O₃) indica el gas responsable del bloqueo de la radiación electromagnética.

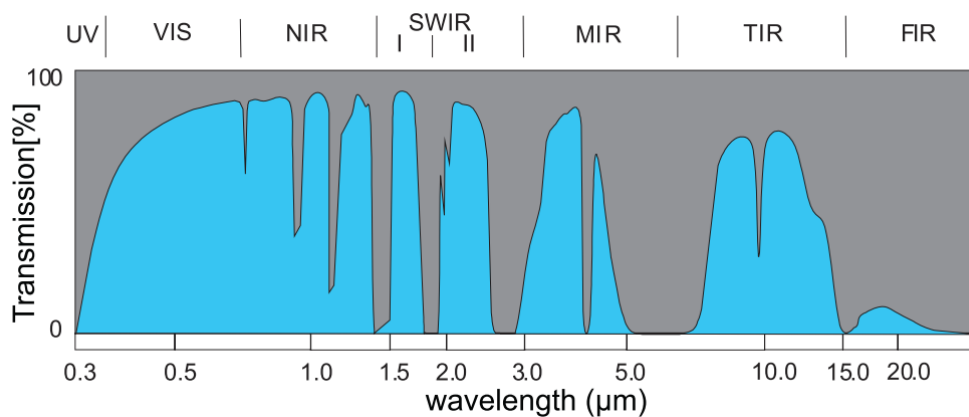


Figura N° 6 Ventanas Atmosféricas en Imágenes Ópticas Multiespectrales.

Fuente : (Malo et al., 2011).

En la figura N°6 Se Puede observar el color gris indica bandas de absorción y las áreas azules indican ventanas atmosféricas, es lo que los sensores son capaces de ver en la tierra.

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)

2.1.2.4. Curvas de Reflectancia Espectral

Cada tipo de material de interés tiene una curva de reflectancia. Dicha curva muestra la porción de la energía incidente que se refleja en función de su longitud de onda. (Tempfli et al., 2009)

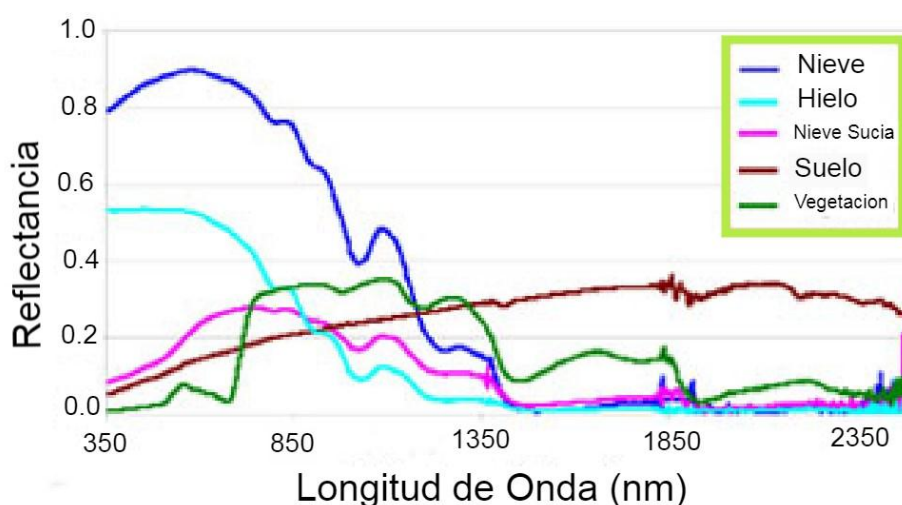


Figura N° 7 Curvas de reflectancia espectral de Diferentes Coberturas.

Fuente: (Kulkarni et al., 2007).

2.1.2.5. Resolución De Imágenes

Resolución espacial: Es el objeto más pequeño que se puede identificar en una imagen. (Almeida & Meneses, 2012). Generalmente se describe como el campo de visión instantáneo se define como el Angulo de visión máxima en el que un sensor puede detectar eficazmente la energía electromagnética. (Malo 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. (Almeida & Meneses, 2012). Una alta resolución espectral proporciona una firma espectral más precisa. (Malo et al., 2011)

Resolución Radiométrica: Es la capacidad para detectar variaciones en la radiancia espectral que recibe.(Chuvienco, 1995).Está determinado por el número de niveles discretos en los que se puede dividir la radiación de la señal.(Malo et al., 2011). Cuanto sea mayor la resolución radiométrica podrá interpretarse mejor la imagen.(Chuvienco, 1995)

Resolución Temporal: Es la frecuencia de cobertura que proporciona el sensor.(Chuvienco, 1995). Se refiere a la frecuencia con la que el sensor vuelve a visitar un área y toma imágenes periódicamente durante su vida útil.(Almeida & Meneses, 2012).

2.1.3. Índices Espectrales

Los índices espectrales son combinaciones paramétricas simples de varios canales espectrales, están diseñados para reforzar la sensibilidad a fenómenos biofísicos particulares, como el verdor de la vegetación, el contenido de agua. (Malo et al., 2011)

2.1.3.1. Normalized Difference Snow Index (NDSI)

El índice espectral (NDSI) fue desarrollado por (Dozier, 1989) y es un método simple y establecido para identificar glaciares en imágenes multiespectrales.

Es un índice para detección de glaciares está definido por la siguiente ecuación:

$$NDSI = \frac{Green_{0.53} - SWIR_{1.65}}{Green_{0.53} + SWIR_{1.65}}$$

Donde:

$Green_{0.53}$: Reflectancia en el Canal Verde.

$SWIR_{1.65}$: Reflectancia en el canal del Infrarrojo de onda corta.

2.1.4. Inteligencia Artificial

El término inteligencia artificial (AI), es muy difícil de definir porque se trata de la Inteligencia surgen preguntas como ¿Qué es la inteligencia? , ¿Cómo se puede medir la inteligencia?, ¿Cómo funciona el cerebro? Todas estas preguntas son significativas cuando se trata de entender la inteligencia artificial.(Wolfgang, 2017).

Las definiciones históricas de A, en 1955, John McCarthy, uno de los pioneros de la AI, fue el primero en definir el término inteligencia artificial, “El objetivo de la AI es desarrollar máquinas que se comporten como si fueran inteligentes”.

AI Es la capacidad de las máquinas de demostrar inteligencia o imitar capacidades cognitivas propias de seres inteligentes. Estas capacidades incluyen razonar, representar conocimiento, aprender en base a evidencias, planear en base a objetivos, tener curiosidad, y entender el lenguaje natural.(Florez, 2018).

Actualmente AI recibe influencia de diversos campos incluyendo lingüística, biología, psicología, economía, y muchos otros.(Florez, 2018). Una de las mentes científicas indica que la AI es la nueva electricidad.(Ng, 2018). La implementación de inteligencia artificial se basa en métodos de optimización matemática, inferencia estadística, y abstracción computacional.(Florez, 2018).

2.1.5. Aprendizaje Automático o Machine Learning

Machine Learning (ML) Es la ciencia y el arte de la programación de computadoras para que puedan aprender de los datos.(Géron, 2019).

Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E.—Tom Mitchell, 1997

Es un campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente. —Arthur Samuel, 1959.

2.1.3.1. Tipos de Aprendizaje Automático

Aprendizaje Supervisado: Los algoritmos de aprendizaje supervisado son, en términos generales, algoritmos de aprendizaje que aprenden a asociar algunas entradas con algunas salidas, dado un conjunto de ejemplos de entradas x y salidas y. En muchos casos, los resultados y pueden ser difíciles de recopilar automáticamente y deben ser proporcionados por un "supervisor" humano.(Goodfellow, Bengio, & Courville, 2016)

Aprendizaje No Supervisado: Los algoritmos de aprendizaje no supervisado son aquellos que experimentan solo características, pero no una señal de supervisión, se

refiere a la mayoría de intentos de extraer información de una distribución que no requiere trabajo humano.(Goodfellow et al., 2016).

2.2.6. Aprendizaje Profundo o Deep Learning

El aprendizaje profundo (DL) generalmente se implementa utilizando una arquitectura de red neuronal. El término "profundo" se refiere al número de capas en la red: cuantas más capas, más profunda es la red. Las redes neuronales tradicionales contienen solo 2 o 3 capas, mientras que las redes profundas pueden tener cientos.(Mathworks, 2018).

DL es un marco matemático para aprender representaciones a partir de datos.(Chollet, 2017)

2.2.5. Redes Neuronales Convolucionales

2.2.5.1. Redes Neuronales Artificiales

Una red neuronal es una referencia a la neurobiología, algunos de los conceptos centrales en el aprendizaje profundo se desarrollaron en parte al inspirarnos en nuestra comprensión del cerebro, los modelos de aprendizaje profundo no son modelos del cerebro. No hay evidencia de que el cerebro implemente algo como los mecanismos de aprendizaje utilizados en los modelos modernos de aprendizaje profundo.(Chollet, 2017)

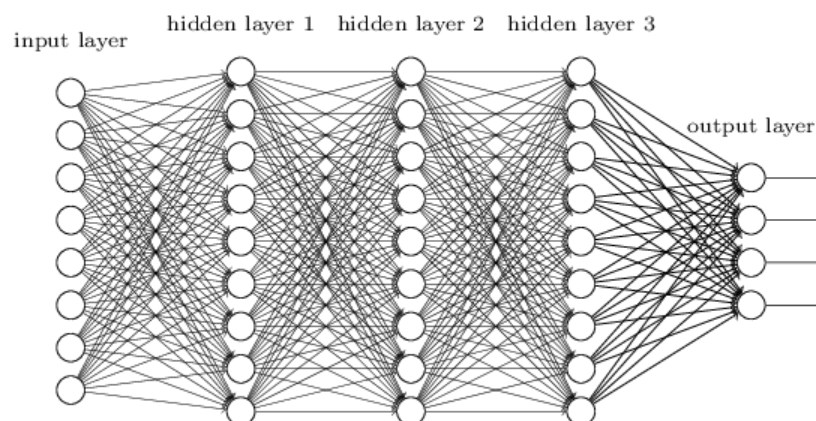


Figura N° 8 Redes Neuronales Profundas totalmente conectadas.

Fuente : (Nielsen, 2015).

- **Input Layer o Capa de entrada:** A menudo se define como sus datos de entrada sin procesar, como imágenes pueden ser valores de pixeles de diferentes canales de color. Además con diferentes dimensiones de datos de entrada, con diferentes estructuras como un vector unidimensional o una estructura similar a un tensor.(Bhardwaj, Di, & Wei, 2018)

- **Output Layer o Capa de Salida:** Es básicamente el valor de salida de la red y se configura según la forma del problema.(Bhardwaj et al., 2018)

- **Hidden Layers o Capas Ocultas:** Son capas entre las capas de entrada y salida. Las neuronas en las capas ocultas pueden adoptar diversas formas, como una capa de agrupación máxima, una capa convolucional, etc., Todas con diferentes funcionalidades matemáticas.(Bhardwaj et al., 2018)

2.2.5.2. Función de Activación

La función de activación en cada neurona artificial decide si las señales entrantes han alcanzado el umbral y deberían emitir señales para el siguiente nivel.(Bhardwaj et al., 2018)

- **Función Sigmoide o Logística:** Una función sigmoidea tiene una forma de S, su rango está entre 0 y 1. Su Expresión matemática es:

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

Su primera derivada, que se utiliza durante el backpropagation del paso de entrenamiento, tiene la siguiente expresión matemática:

$$\frac{d\sigma(x)}{d(x)} = \sigma(x) \cdot (1 - \sigma(x))$$

- **Función Tangente Hiperbólico (Tanh):** Su salida está centrada con un rango de -1 a 1.La fórmula matemática de la Tanh es la siguiente:

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

- **La Unidad Lineal Rectificada (ReLU):** Es la más usada y se ha vuelto bastante popular en los últimos años. Su fórmula matemática es la siguiente:

$$\sigma(x) = \begin{cases} \max(0, x) & , x \geq 0 \\ 0 & , x < 0 \end{cases}$$

En comparación con la Sigmoidea y Tanh, su cálculo es mucho más simple y más eficiente. La convergencia es seis veces mejor, posiblemente debido a que tiene una linealidad y forma no saturante. A diferencia de las funciones Tanh y Sigmoidea que implican operaciones exponencial costosa, Casi todos los modelos de aprendizaje profundo usan ReLu.(Bhardwaj et al., 2018).

Name	Equation	Derivative	1-D Graph	1-D Graph(derivative)
Binary Step	$\sigma(x) = \begin{cases} 1, & x > 0 \\ 0.5, & x = 0 \\ 0, & x < 0 \end{cases}$	$\sigma'(x) = \begin{cases} 0, & x \neq 0 \\ ?, & x = 0 \end{cases}$		
Identity	$\sigma(x) = x$	$\sigma'(x) = 1$		
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$	$\sigma'(x) = \sigma(x)(1 - \sigma(x))$		
Tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\sigma'(x) = 1 - \sigma(x)^2$		
Rectified Linear (ReLU)	$\sigma(x) = \max(0, x)$	$\sigma'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$		
Leaky ReLU	$\sigma(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases}$	$\sigma'(x) = \begin{cases} 1, & x \geq 0 \\ \alpha, & x < 0 \end{cases}$		

Figura N° 9 Funciones de Activación.

Fuente : (Bhardwaj et al., 2018).

2.2.5.3. Forward Propagation o Propagación hacia delante

Consiste básicamente en calcular los datos de entrada multiplicados por las redes. Pesos más el desplazamiento y luego pasarle la función de activación a la siguiente capa.

$$\begin{aligned}z_1 &= a_0 * w_1 + b_1 \\a_1 &= \sigma(z_1) \\z_2 &= a_1 * w_2 + b_2 \\a_2 &= \sigma(z_2)\end{aligned}$$

Donde:

z= Resultado de la suma ponderada.

a = datos.

w = Pesos.

b = sesgo o bias.

σ = Funcion de activación.

2.2.5.4. Backpropagation o Retro Propagación

Todas las redes aprenden del error y luego actualizan los pesos/parámetros de la red para reflejar los errores basadas en una función de costo dada. El gradiente es la pendiente que representa la relación entre los pesos de una red y su error.(Bhardwaj et al., 2018).El entrenamiento de una red neuronal se realiza con algoritmos como backpropagation. Este algoritmo de propagación de errores hacia atrás, utiliza el error observado en la salida de una red neuronal multicapa para ir ajustando paulatinamente los parámetros que definen la respuesta de la red ante una entrada dada, con la intención de reducir el error cometido por la red.(Berzal, 2018)

Una vez que se calcula la salida de la red para un conjunto de datos de entrada, evaluamos una función de error que nos permite comparar la salida obtenida con la salida deseada. Para minimizar esa función de error, ajustaremos. Iterativamente los pesos de la red de acuerdo al gradiente del error con respecto a los pesos.(Berzal, 2018)

2.2.5.5. El descenso del gradiente

Es un algoritmo de optimización muy genérico capaz de encontrar soluciones óptimas para una amplia gama de problemas. La idea general del descenso del gradiente

es ajustar los parámetros de forma iterativa para minimizar una función de costo. (Géron, 2019)

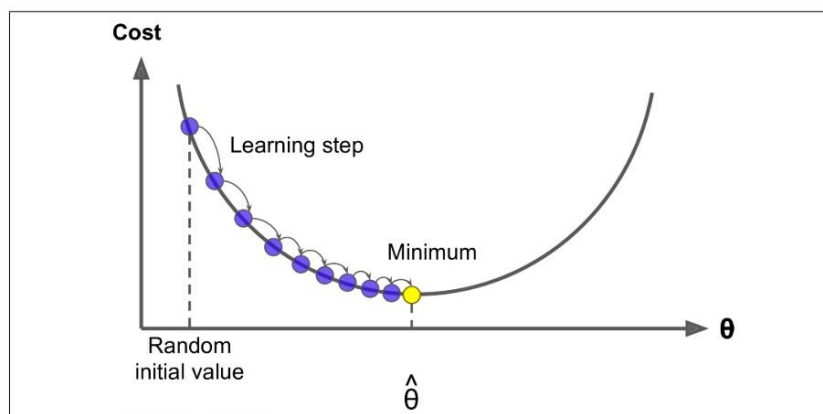


Figura N° 10 Descenso del Gradiente.

Fuente : (Géron, 2019).

Un parámetro importante en el descenso del gradiente es el tamaño de los pasos determinado por la tasa de aprendizaje. Si la tasa de aprendizaje es muy pequeña entonces el algoritmo tendrá que pasar muchas iteraciones para converger, lo cual lleva mucho tiempo, hablando computacionalmente.

Para implementar el descenso del gradiente, se debe realizar los siguientes pasos:

- Calcular la derivada parcial de la función de costo con respecto al parámetro θ_j

$$\frac{\partial}{\partial \theta_j} \text{MSE}(\boldsymbol{\theta}).$$

- Derivadas parciales de la función de costo en este caso el error medio cuadrático MSE para una regresión lineal.

$$\frac{\partial}{\partial \theta_j} \text{MSE}(\boldsymbol{\theta}) = \frac{2}{m} \sum_{i=1}^m (\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)}) x_j^{(i)}$$

- El vector gradiente de la función de costo

$$\nabla_{\theta} \text{MSE}(\theta) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} \text{MSE}(\theta) \\ \frac{\partial}{\partial \theta_1} \text{MSE}(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \text{MSE}(\theta) \end{pmatrix} = \frac{2}{m} \mathbf{X}^T (\mathbf{X}\theta - \mathbf{y})$$

Donde:

$\nabla_{\theta} \text{MSE}(\theta) = \text{Vector Gradiente}$

Pasos del descenso del gradiente.

$$\theta := \theta - \alpha \nabla_{\theta} \text{MSE}(\theta)$$

θ : = Próximo paso

α = la tasa de aprendizaje.

Finalmente lo que queremos lograr con el descenso del gradiente es cuales pesos y bias minimizan la función de coste.

2.2.5.6. La Función Softmax

Es una función de activación que a menudo se coloca en la capa de salida de una red neuronal. La función softmax normaliza cada valor entre 0.0 a 1.0 con la condición de que estos valores sea 1.0. El resultados es un vector que representa la probabilidad de que un dato de entrada pertenezca a cada clase disponible con cierto nivel de probabilidad.(Florez, 2018)

$$S(Y_j) = \frac{e^{Y_j}}{\sum_{k=1}^K (e^{Y_k})}$$

Donde:

Y = Salida de las capas ocultas.

K = Numero de clases en nuestro modelo.

2.2.5.7. Funciones de Perdidas

- **Entropía Cruzada:** Es la más utilizada para la clasificación CNN. Cuando se aplica tareas de segmentación, la entropía cruzada mide la divergencia de la probabilidad pronosticada de la etiqueta de la verdad del suelo para cada píxel por separado y luego promedia el valor sobre todos los píxeles del mini lote.(Kodym, Michalšpaněl, & Herout, 2018)

- **Dice Coefficient:** El coeficiente dice entre dos volúmenes binarios se expresa matemáticamente:

$$Dice = \frac{2 * |A \cap B|}{|A| + |B|}$$

Dónde: A= Verdadero y B= Predicción.

2.2.5.8. Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN) es un enfoque de aprendizaje profundo que se utiliza ampliamente para resolver problemas complejos. Supera las limitaciones de los enfoques tradicionales de aprendizaje automático.(Indolia, Goswami, Mishra, & Asopa, 2018).Las CNN son muy efectivas en áreas como el reconocimiento y las clasificación de imágenes.(Bhardwaj et al., 2018)

Las arquitecturas de CNN tiene varias variantes; por lo general, consisten en capas convolucionales y pooling (submuestreo), que se agrupan en módulos. Seguido de una o más capas completamente conectadas.(Rawat & Wang, 2017)

Algunos componentes fundamentales de construcción de CNN mencionados por (Bhardwaj et al., 2018)

- Capa Convolutiva (CONV).
- Capa de Activación.
- Capa de Agrupación o capa de submuestreo (POOL).
- Capa totalmente Conectada (FC).

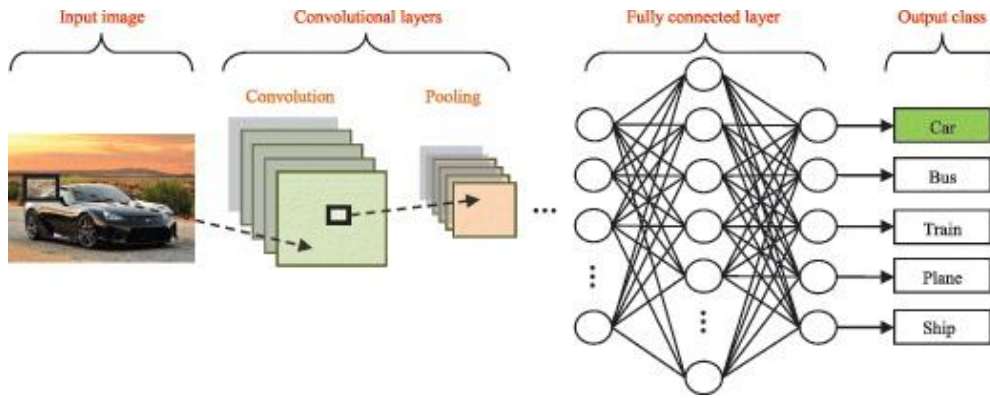


Figura N° 11 Flujo para clasificación de Imágenes con CNN.

Fuente : (Rawat & Wang, 2017)

- **Capa Convolutacional:** Las capas convolucionales sirven como extractores de características, y así aprenden las representaciones de sus imágenes de entrada.(Rawat & Wang, 2017)

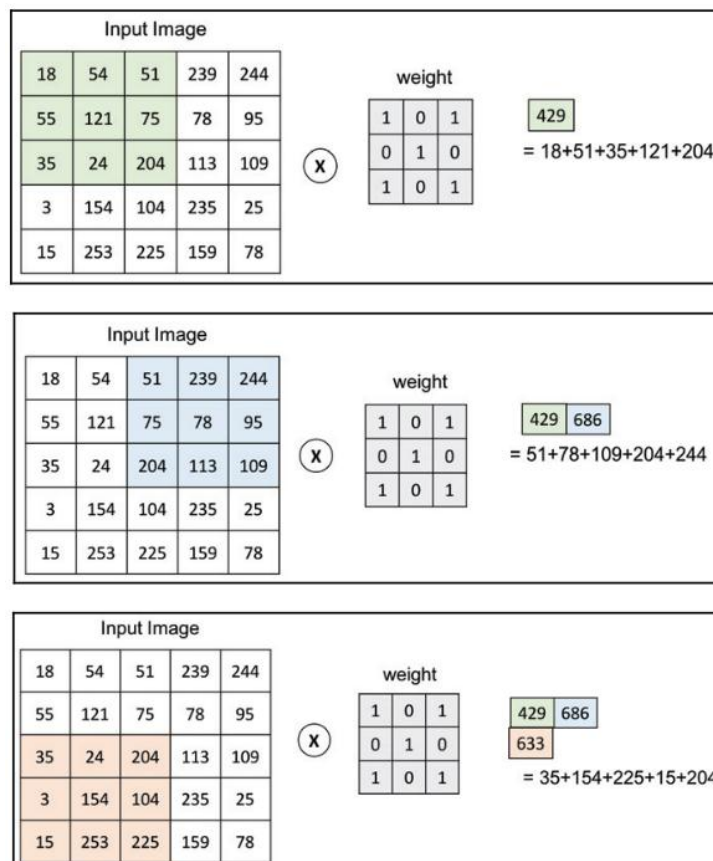


Figura N° 12 Proceso de la Convolución.

Fuente : (Bhardwaj et al., 2018).

- **Capa de Agrupación o Pooling:** Su objetivo es submuestreo es decir reducir la imagen de entrada para reducir la carga computacional, el uso de memoria y el número de parámetros.(Géron, 2019).La forma más común de capa de agrupación generalmente se aplica es la agrupación máxima.(Bhardwaj et al., 2018).En la figura N°13 se observa el proceso de agrupación máxima o más conocido como Max pooling. Una capa de agrupación funciona en cada banda de entrada de forma independiente, por lo que la profundidad de salida es la misma que la profundidad de entrada.

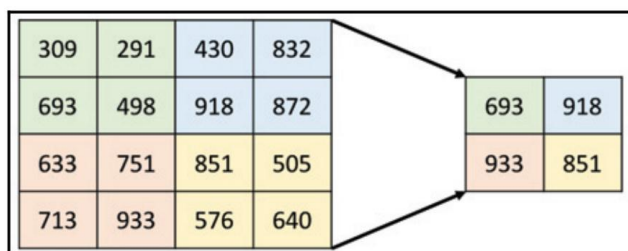


Figura N° 13 Proceso de Agrupación máxima.

Fuente : (Bhardwaj et al., 2018)

- **Capa Totalmente Conectada:** Las neuronas en una capa totalmente conectadas tienen conexiones completas a todas las activaciones de la capa anterior, lo que es diferente a las capa CONV. En las capas CONV, las neuronas están conectadas solo a una región local en la entrada, y muchas de las neuronas en un volumen CONV comparten parámetros. La capa totalmente conectada se utiliza a menudo en las dos últimas capas con una función softmax para reemplazar las otras funciones de activación para generar probabilidades.(Bhardwaj et al., 2018)

En resumen mencionado por (Bhardwaj et al., 2018), el bucle de entrenamiento incluye un paso hacia adelante y backpropagation.

- Para cada imagen de entrada, primero pasamos a través de la capa convolucional. Los resultados de estos se introducen a la función de activación en decir: CONV + ReLu.

- Después la función de máxima agrupación, la agrupación dará como resultado un tamaño más pequeño del parche ayudara a reducir el número de características.

- Las capas CONV + ReLu y POOL se repetirán varias veces antes de que se conecten a las capa completamente conectadas.
- La capa de salida a menudo está completamente conectada, pero con funciones softmax que ayudan a calcular la salida similar a la probabilidad.
- Después la salida se comparan con los datos verdaderos para generar valores de error y luego utilizan para calcular el costo. Normalmente la función de pérdida se define como el error cuadrático medio se utiliza en la fase de optimización y Finalmente los errores se propagan de forma posterior para actualizar los coeficientes y los valores de sesgo.

2.2.5.9. Segmentación Semántica

La segmentación semántica es la tarea de clasificación de objetos a nivel de pixel por pixel.(Avery Bick, Mullet, & Wang, 2018). Los marcos de segmentación semántica de última generación para imágenes de percepción remota se componen secuencialmente de subredes de codificador y decodificador.(Ma et al., 2019).Al mapear grandes territorios, la compensación entre el tiempo de procesamiento y la calidad de los resultados es una decisión central de diseño.(Stoian et al., 2019). En la figura N°14 se ve una segmentación semántica donde cada pixel se clasifica según la clase objeto a la que pertenece como por ejemplo; Carreteras, automóvil, peatón, edificio, etc.



Figura N° 14 Segmentación Semántica

Fuente : (Géron, 2019).

2.2.5.10. Arquitectura U-Net

Para el entrenamiento de una red U-Net se necesita las imágenes y sus correspondientes mapas de segmentación, se utilizan para entrenar la red con la

implementación del descenso del gradiente estocástico.(Ronneberger, Fischer, & Brox, 2015). La arquitectura U-Net consta de tres secciones: la contracción, el cuello de botella y la sección de expansión. En la figura N°15 se observa la arquitectura original U-Net para segmentar imágenes médicas.

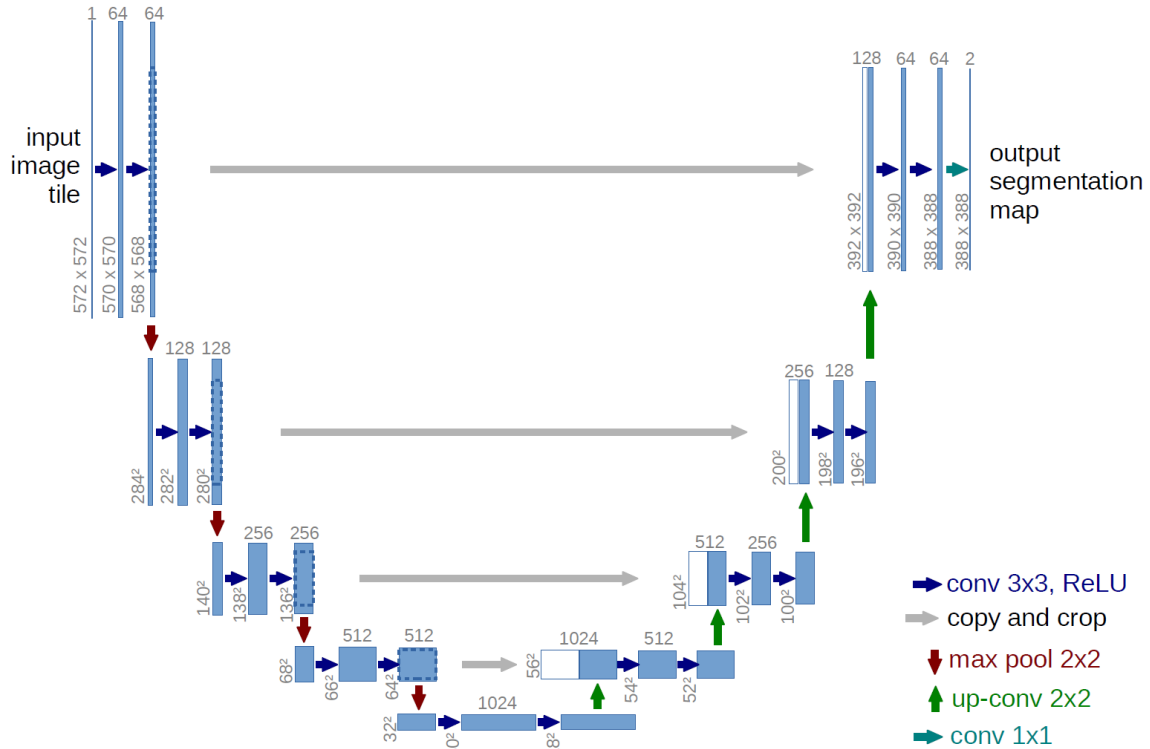


Figura N° 15 La Arquitectura U-Net

Fuente : (Ronneberger et al., 2015).

2.2.6. El Programa Copérnico

El programa Copérnico es el programa de observación de la tierra más ambicioso de la historia, diseñada para proporcionar información precisa, actualizada y de fácil acceso para mejorar la gestión del medio ambiente, comprender y mitigar los efectos del cambio climático y garantizar la seguridad ciudadana. Esta iniciativa está liderada por la comisión europea (CE), en colaboración con la Agencia Espacial Europea (ESA).

La ESA es la responsable de coordinar la distribución de datos recogidos por más de 30 satélites. (European Space Agency, 2014)

2.2.6.1. Sentinel 2

La misión Copernicus Sentinel-2 comprende una constelación de dos satélites en órbita polar colocados en la misma órbita síncrona solar, en fases de 180° entre sí. Su objetivo es monitorear la variabilidad en las condiciones de la superficie terrestre, y su gran ancho de franja (290 km) y su alto tiempo de revisión (10 días en el ecuador con un satélite, y 5 días con 2 satélites en condiciones sin nubes, lo que da como resultado 2-3 días en latitudes medias). Los límites de cobertura son entre las latitudes 56° sur y 84° norte. (European Space Agency, 2015).

2.2.6.2. Niveles de Procesamiento

Los productos Level-1C y Level-2A están a disposición para los usuarios a través del Centro de acceso abierto Copernicus. El Productos para Level-1C están en reflectancia en el techo de la atmosfera (TOA) orto rectificada, con registro Multispectral subpixel, también con máscaras de nubes, tierra y agua. (European Space Agency, 2015b). El Producto Level-2A está en reflectancia orto rectificada en el fondo de la atmosfera (BOA), con registro Multispectral de subpixel. además se incluye un mapa de clasificación de la escena (Nube, sombras de nubes, vegetación, suelos, agua, nieve, etc.). (European Space Agency, 2015b). En Figura N°16 se observa el Nivel 1C (Izquierda) y Nivel 2A (Derecha).



Figura N° 16 Nivel de Procesamiento en Sentinel 2.

Fuente: (European Space Agency, 2015b)

Nombre	Descripción de alto nivel	Producción y Distribución	Volumen de datos
Nivel-1C	Reflectancia de la atmósfera en la geometría cartográfica.	Generación sistemática y distribución on-line.	600 MB (cada 100x100 km ²)
Nivel-2A	Reflectancia del fondo de la atmósfera en geometría cartográfica.	Generación sistemática y distribución y generación en línea en el lado del usuario (utilizando la caja de herramientas de Sentinel-2)	800 MB (cada 100x100 km ²)

Tabla N° 2 Nivel de Procesamiento en Sentinel 2.

Fuente: (European Space Agency, 2015a).

Número de banda	S2A		S2B		Resolución espacial (m)
	Longitud de onda central (nm)	Ancho de banda (nm)	Longitud de onda central (nm)	Ancho de banda (nm)	
1	442.7	21	442.2	21	60
2	492.4	66	492.1	66	10
3	559.8	36	559	36	10
4	664.6	31	664.9	31	10
5	704.1	15	703.8	16	20
6	740.5	15	739.1	15	20
7	782.8	20	779.7	20	20
8	832.8	106	832.9	106	10
8a	864.7	21	864	22	20
9	945.1	20	943.2	21	60
10	1373.5	31	1376.9	30	60
11	1613.7	91	1610.4	94	20
12	2202.4	175	2185.7	185	20

Tabla N° 3 Bandas espectrales para los sensores SENTINEL-2 (S2A y S2B).

Fuente: (European Space Agency, 2015a).

Especificaciones Técnicas	
Satelite	Sentinel 2
# Satelite	S2A & S2B
Altitud del Satelite (Km)	786
Orbita	Sincronica al Sol
Sensor	MSI
Tipo	Push - Broom
Rango Espectral (nm)	442.2 - 2202.4
Resolucion Espacial (mts)	10,20,60
Resolucion Espectral (# Bandas)	13
Resolucion Radiometrica (Bit)	12
Resolucion Temporal (Dia)	5
Ancho de Franja (Km)	290

Tabla N° 4 Especificaciones Técnicas satélite Sentinel 2.

Fuente: (European Space Agency, 2015a).

2.2.7. Sistemas de Información Geográfica

Un Sistema de Información Geográfica (SIG) es un sistema informático que ayuda a recopilar, procesar, editar, almacenar, administrar, compartir, analizar, modelar y visualizar grandes volúmenes de datos para comprender relaciones espaciales, patrones, tendencias, y tomar decisiones.(Tian, 2017).Un SIG ha permitido realizar gestión de datos espaciales, análisis, generación de resultados.(Olaya, 2011).

2.2.8. Computación en la nube

La computación de la nube o Cloud Computing se refiere a las aplicaciones entregadas como servicios a través de internet y el software de hardware y sistemas en los centros de datos que proporcionan esos servicios.(Armbrust et al., 2010).

El Cloud Computing se basa en el uso de la memoria, almacenamiento y cálculos computacionales a través de la compartición e interconexión de internet, accesibles a través de servicios ofrecidos en la web. La expresión de "Cloud Computing". Indica que la informática, orientada a servicios, proporcionado al usuario, está cambiando de curso debido a la posibilidad de usar computadoras menos potentes que pueden conectarse a Web y use todas las herramientas en línea. Con esto, la computadora se convierte simplemente en una plataforma de acceso a las aplicaciones, que estarían en una gran nube - la Internet.(Santaella, 2012). La nube es una colección de servicios que ayuda a

los desarrolladores a centrarse en su proyecto en lugar de en la infraestructura que lo potencia.(Geewax, 2018)

2.2.8.1. Google Earth Engine

Google Earth Engine (GEE) es una plataforma basada en la nube para el análisis geoespacial a escala planetaria que aporta las capacidades computacionales masivas de Google a una variedad de problemas sociales de alto impacto. Es único en el campo como una plataforma integrada diseñada para capacitar no solo a científicos tradicionales de la teledetección , sino también a una audiencia muchos más amplia que carece de la capacidad técnica necesario para utilizar supercomputadoras tradicionales o recurso de computación en nube de productos básicos a gran escala.(Gorelick et al., 2017)

GEE Se accede y se controla mediante una interfaz de programación de aplicaciones (API) accesible a través de Internet y un entorno de desarrollo interactivo (IDE) asociado que permite la creación rápida de prototipos y la visualización de resultados.(Gorelick et al., 2017)

Los Principales Componentes de Earth Engine

- **Dataset:** Un archivo a escala de petabytes de imágenes de sensores remotos disponibles públicamente y otros datos. (Google Developers, 2015).
- **Compute Power:** La infraestructura computacional de Google optimizada para el procesamiento paralelo de datos geoespaciales.(Gorelick et al., 2017)
- **APIs:** API para JavaScript y Python, para realizar solicitudes a los servidores de Earth Engine. (Google Developers, 2015).
- **Code Editor:** Entorno de desarrollo integrado (IDE) en línea para la creación rápida de prototipos y la visualización de análisis espaciales complejos utilizando la API de JavaScript. (Google Developers, 2015).

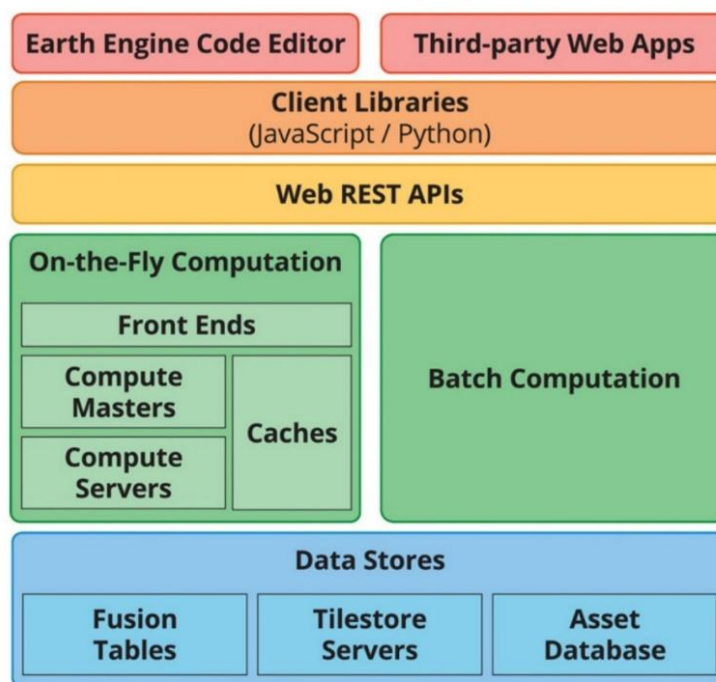


Figura N° 17 Un diagrama Simplificado de la arquitectura del sistema GEE.

Fuente : (Gorelick et al., 2017).

2.2.8.2. Google Cloud Platform

Google Cloud Platform (GCP) es una colección de productos que permite al mundo utilizar parte de la infraestructura interna de Google. Esta colección incluye muchas cosas que son comunes en todos los proveedores de nube, como las máquinas virtuales bajo demanda a través de Google Compute Engine o el almacenamiento de objetos para almacenar archivos a través de Google Cloud Storage (GCS). También incluye API para algunas de las tecnologías más avanzadas creadas por Google, como Bigtable, Cloud Datastore o Kubernetes.(Geewax, 2018). GCS es el sistema de almacenamiento de objetos predeterminado en Google Cloud Platform.(Geewax, 2018).

III. MATERIALES Y MÉTODOS

3.1. UBICACIÓN DEL ÁREA DE ESTUDIO

3.1.1. Ubicación Política:

El área de estudio está ubicado:

- Lugar** : Glaciar Ausangante.
- Distrito** : Pitumarca y Ocongate
- Provincia** : Canchis y Quispicanchi
- Departamento** : Cusco

3.1.2. Ubicación Geográfica

El área de estudio está ubicado en las siguientes coordenadas geográficas: Latitud: $13^{\circ}47'39.77''S$ y Longitud: $71^{\circ}13'46.32''O$.

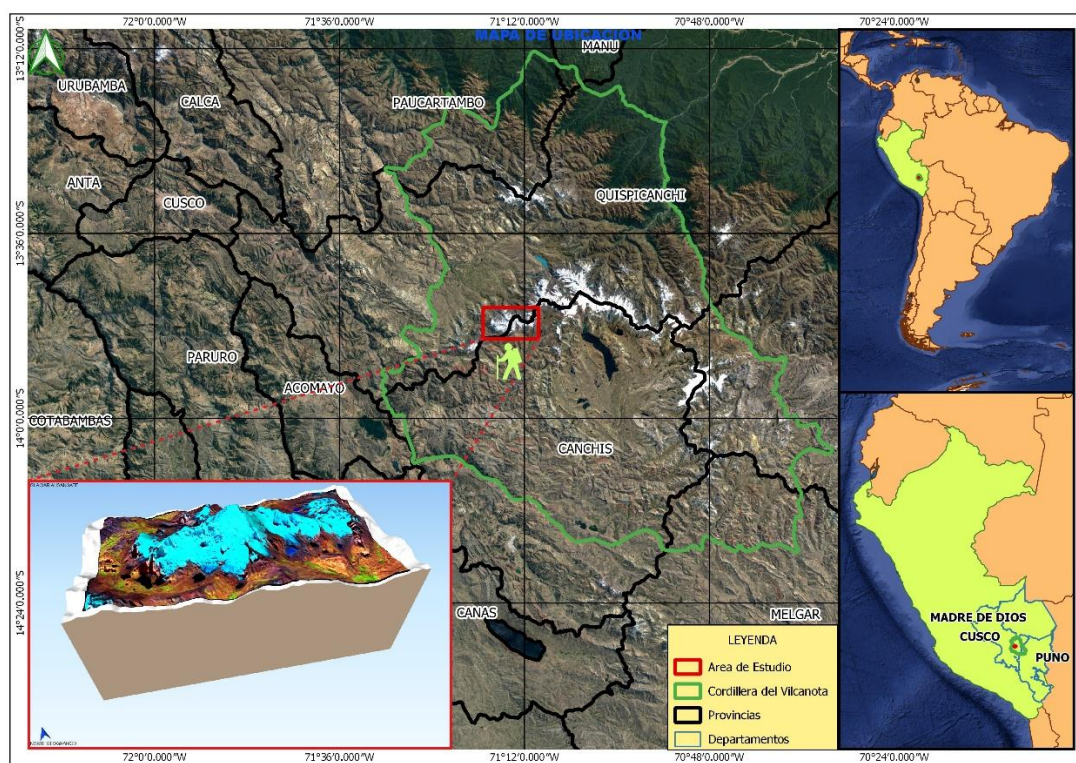


Figura N° 18 Mapa de Ubicación del Área de Estudio.

Fuente: (Propio, 2019)

3.2. MATERIALES E INSTRUMENTOS

3.2.1. Materiales

- Imágenes Satelitales Ópticas Sentinel 2 (2A y 2B).
- Carta Nacional 1:100000 (29-t).
- Cartografía de Glaciares ANA, (Vectorial).
- DEM ALOS PALSAR de Resolución 12.5 metros.
- Lentes Polarizados (Protección para el reflejo del Glaciar).
- Materiales de Escritorio.

3.2.2. Hardware

- Laptop Gamer I5.
- Cámara Fotográfica (Celular).
- GPS Garmin GPSMAP 64s.
- Impresora Multifuncional EPSON.

3.2.3. Software

- QGIS 3.8.
- Python 3.6.8.
- SNAP 6.0.9.
- Microsoft Word 2013.
- R Statics 3.6.1.
- Microsoft Excel 2013.

3.2.4. Librerías

- Tensorflow 2.0.0.
- Tensorboard 2.0.0.
- Keras 2.2.4.
- Folium 0.8.3
- EarthEngine 0.1.175.
- Sen2Cor-02.05.05.

3.2.5. Servicios

- Internet 12 Mbps.
- Google Drive.
- Google Earth Engine.
- Colaboratory.
- Google Cloud Storage.

3.3. METODOLOGÍA PARA EL DESARROLLO DEL TRABAJO

3.3.1. Tipo de Investigación

La Investigación es de tipo Explicativo, La investigación explicativa intenta aclarar por qué y cómo existe una relación entre dos fenómenos. Es decir intenta explicar.(Kumar, 2010). pretenden establecer las causas de los fenómenos que se estudian.(Hernandez Sampieri, Fernández Collado, & Baptista Lucio, 2014).

3.3.2. Población y Muestra

Población: Glaciares de la Cordillera del Vilcanota.

Muestras: Glaciar Ausangante.

El Muestreo que se aplico es el muestreo no probabilístico para población finita conocida.

n = 1 (Glaciar Ausangante).

Considerando los siguientes criterios:

- Por ser factible llegar al lugar, turístico en la región del Cusco, un glaciar complicado de cartografiar con métodos topográficos, el glaciar tiene mucha importancia hidrológica en la región, cosmovisión andina y el quinto glaciar más alto del Perú.

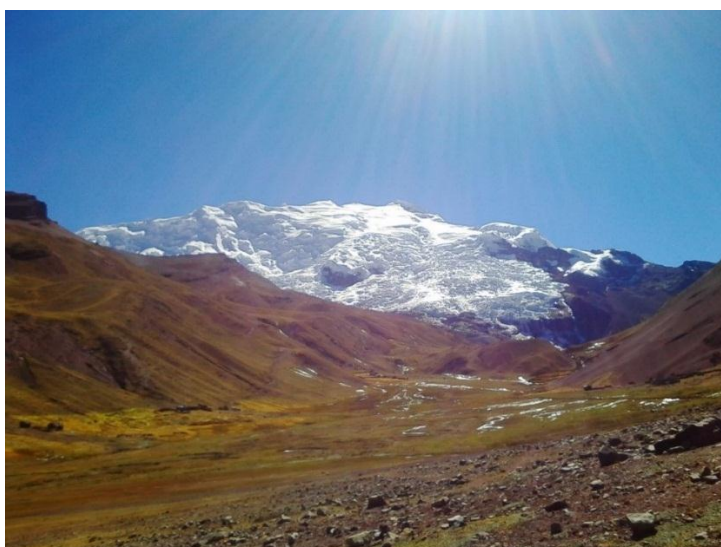


Figura N° 19 Vista del Glaciar Ausangante.

Fuente: (Propio, 2019).

3.3.3. Variables

Por ser una investigación experimental se tienen dos variables.

Variable Dependiente: Estimación de área glaciar.

Variable Independiente: Redes Neuronales Convolucionales U-Net.

Expresando matemáticamente es de la siguiente forma: $y = f(x)$

Estimación de área glaciar = $f(\text{Redes Neuronales Convolucionales U - Net})$

VARIABLE	DIMENSION	INDICADOR	INDICE
Variable Independiente	Factores Corrección del Imagen Óptica satelital	Corrección Atmosférica	Reflectancias en Superficie Normalizado = [0 - 1]
	Factores que Influyen en el entrenamiento de la CNN	Red Neuronal Convolutiva Arquitectura U-Net	Probabilidades = [0 - 1]
Variable Dependiente		NO ES GLACIAR	NO = 0
		SI ES GLACIAR	SI = 1

Tabla N° 5 Operacionalización de variables.

Fuente: (Propio, 2019).

3.3.3.1. Firma Espectral de la Nieve

El servicio Geológico de los Estados Unidos (USGS) Ha reunido una biblioteca espectral medida con espectrómetros de laboratorio, de campo y aerotransportada. Los instrumentos utilizados cubren longitudes de onda desde 0.2 a 200 micrómetros. Se midieron espectros de minerales, líquidos, plantas, compuestos químicos y materiales artificiales. (Kokaly et al., 2017)

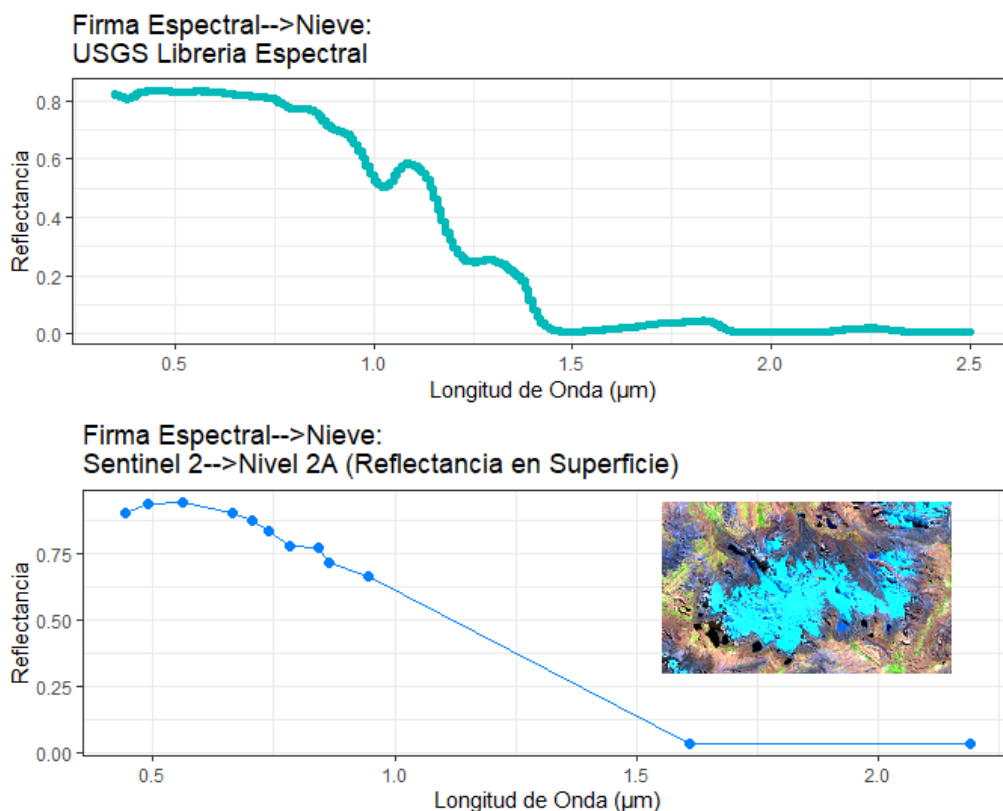


Figura N° 20 Librería Espectral USGS y Firma Espectral de Nieve en Sentinel 2.

Fuente: (Propio, 2019).

ID_Producto: S2B_MSIL2A_20190709T145739_N0213_R039_T19LBE_20190709T183447						
Longitud de Onda (nm)	Longitud	-71.2198	-71.2358	-71.2414	-71.2391	-71.2107
	Latitud	-13.7827	-13.8220	-13.8292	-13.8189	-13.8068
	Cobertura	Nieve	Agua	Bofedal	Superficie Rocosa	Agua Sucia
443	B1	0.9059	0.0001	0.0067	0.0280	0.1180
490	B2	0.9368	0.0001	0.0307	0.0562	0.1418
560	B3	0.9464	0.0001	0.0539	0.0824	0.1946
665	B4	0.9064	0.0001	0.0698	0.1014	0.2810
705	B5	0.8776	0.0002	0.1019	0.1151	0.2726
740	B6	0.8361	0.0001	0.1523	0.1278	0.1930
783	B7	0.7800	0.0002	0.1738	0.1390	0.1603
842	B8	0.7717	0.0001	0.1902	0.1360	0.1328
865	B8A	0.7187	0.0003	0.2035	0.1421	0.0986
945	B9	0.6656	0.0016	0.2385	0.1381	0.0303
1610	B11	0.0332	0.0049	0.1450	0.1777	0.0036
2190	B12	0.0333	0.0049	0.0749	0.1592	0.0040

Tabla N° 6 Reflectancia BOA en Coberturas Terrestres.

Fuente: (Propio, 2019)

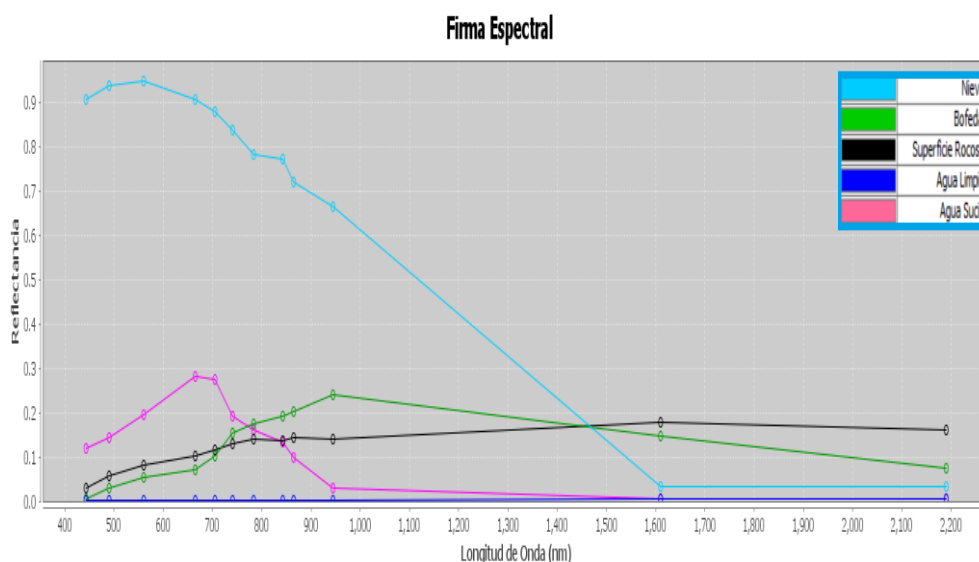


Figura N° 21 Firma Espectral de Coberturas Terrestres en Sentinel 2.

Fuente: (Propio, 2019)

En la figura N°21 Se observa que la reflectancia de la nieve es alta a diferencia de otras coberturas alrededor del glaciar.

3.3.4. Recolección de Datos

3.3.4.1. Metodología para el Objetivo Especifico 1

Actividad de Campo

- Se inspecciono las lagunas glaciares que se muestran la imagen satelital.
- Recorrido alrededor contorno del glaciar tomando puntos GPS para la validación.
- Registro de fotografías de las zonas de interés.

Puntos GPS para Validacion---Observaciones en Campo						
Equipo: GPS Garmin GPSMAP 64s.			Lugar:		Glaciar Ausangate.	
Fecha: 21/06/2019						
ID	Longitud	Latitud	Altura Elipsoidal-h	Altura Geoidal (EGM2008)	Cobertura	Etiqueta
1	-71.244111	-13.834408	4608.259	4559.725	Suelo Desn.	No Glaciar
2	-71.240776	-13.818401	4883.504	4834.97	Roca	No Glaciar
3	-71.239936	-13.817764	4895.581	4847.047	Roca	No Glaciar
4	-71.238996	-13.819499	4818.601	4770.067	Roca	No Glaciar
5	-71.242422	-13.827406	4684.281	4635.747	Pajonal	No Glaciar
6	-71.24145	-13.826268	4701.542	4653.008	Pajonal	No Glaciar
7	-71.24112	-13.82478	4721.111	4672.577	Pajonal	No Glaciar
8	-71.241911	-13.822992	4782.598	4734.064	Pajonal	No Glaciar
9	-71.238553	-13.825341	4659.608	4611.074	Pajonal	No Glaciar
10	-71.239042	-13.821932	4720.393	4671.859	Morrenas	No Glaciar
11	-71.238268	-13.822479	4692.008	4643.474	Morrenas	No Glaciar
12	-71.240764	-13.818874	4879.242	4830.708	Laguna	No Glaciar
13	-71.240689	-13.818147	4882.556	4834.022	Laguna	No Glaciar
14	-71.240689	-13.818147	4884.078	4835.544	Laguna	No Glaciar
15	-71.237836	-13.823834	4662.711	4614.177	Laguna	No Glaciar
16	-71.242513	-13.830007	4642.625	4594.091	Bofedal	No Glaciar
17	-71.239231	-13.825518	4660.436	4611.902	Bofedal	No Glaciar
18	-71.24273	-13.831059	4626.895	4578.361	Bofedal	No Glaciar

Tabla N° 7 Observaciones GPS en Campo.

Fuente: (Propio, 2019).



Figura N° 22 Puntos GPS para Validación.

Fuente: (Propio, 2019).

Actividad en Gabinete

Se recolectó muestras para el entrenamiento de 1883 muestras y validación 1470 muestras en total, Esto realizando una supervisión visual eligiendo zonas cubiertas de glaciario y no Glaciario con etiquetas de 1 y 0 respectivamente {0,1}, en las eco regiones Puna y Selva Alta es donde se encuentran los glaciares esto con el fin de optimizar el proceso. En la plataforma Google Earth Engine (GEE) se realizó una composición de los mejores píxeles con los canales R/G/B-11/8/4 es donde resaltan los glaciares. GEE tiene la colección de imágenes de Sentinel 2 en nivel 2A se realizó una intersección con las muestras de entrenamiento y validación se tiene escenas desde la fecha del 2018/12/13 hasta 2019/06/30, haciendo un total 3143 Imágenes satelitales de Nivel 2A.

VALOR	DESCRIPCION
0	No Glaciario
1	Glaciario

Tabla N° 8 Equivalencias de los atributos.

Fuente: (Propio, 2019).

Entrenamiento	1883 muestras
Validacion	1470 muestras

Tabla N° 9 Cantidad de Muestras.

Fuente: (Propio, 2019).

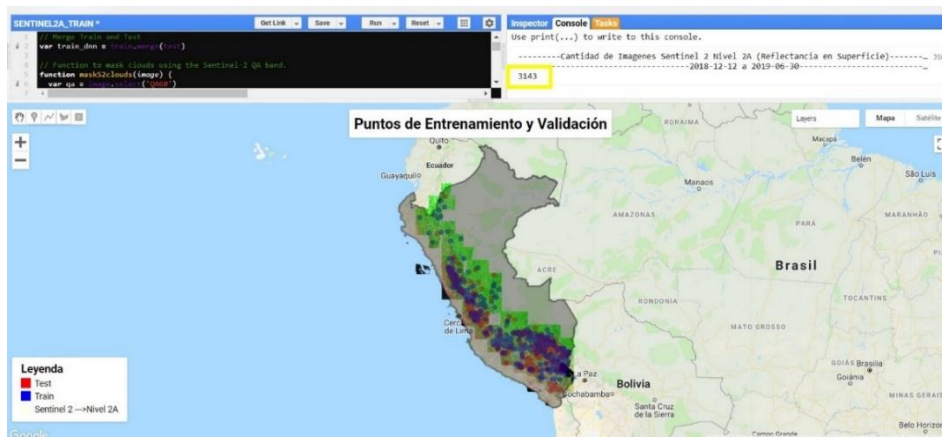


Figura N° 23 Recolectión de puntos de entrenamiento y Validación.

Fuente: (Google, 2015).

Visualizamos las firmas espectrales de la población de muestras, se utilizó la herramienta de QGIS, Random Selection within subsets. Se seleccionó 4 muestras aleatorias de firmas espectrales para Glaciar y No Glaciar.

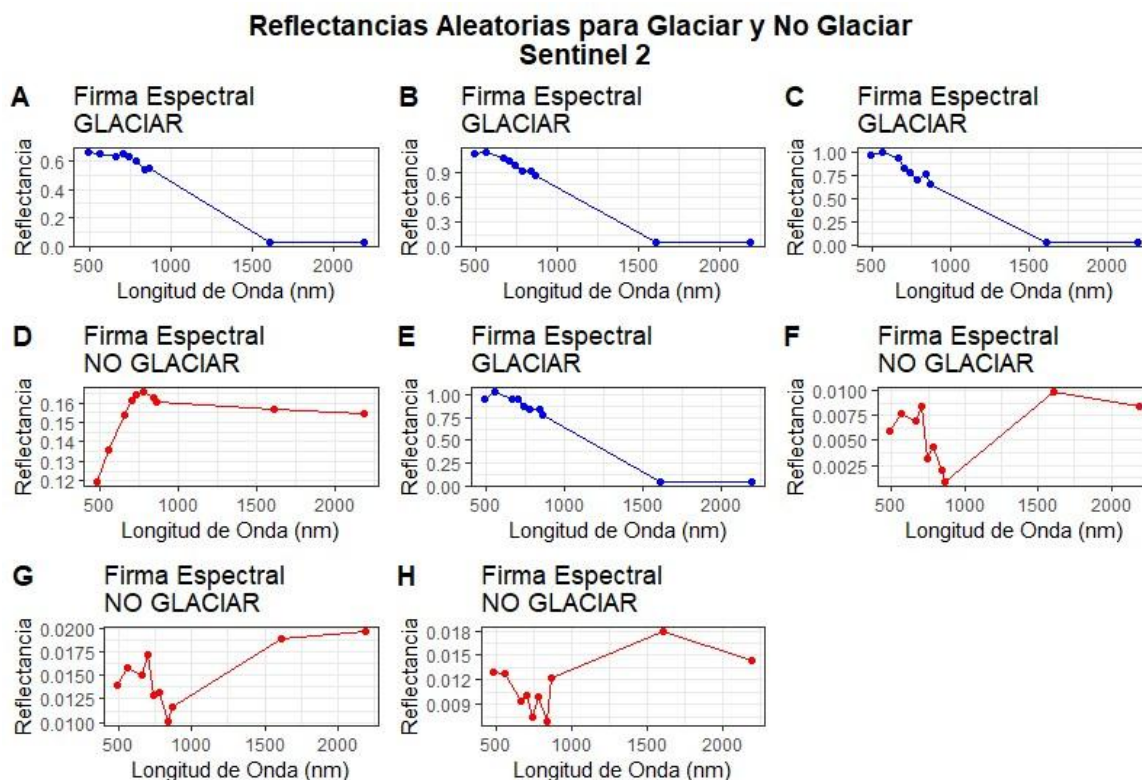


Figura N° 24 Firmas Espectrales Aleatorias del Dataset Entrenamiento.

Fuente: (Propio, 2019).

En la figura N°24 se observa las reflectancia aleatorias del Dataset de entrenamiento en las bandas ['SWIR2','SWIR1','RE4','NIR','RE3','RE2','RE1','R','G','B'], se observa además los glaciares tiene una alta reflectancia y es muy difícil que pueda confundirse con otros materiales de la superficie terrestre.

En la fase de recolección de Dataset para las pruebas estadísticas, se escogieron 12 escenas de TILE_ID "T19LBE" que están fuera de las fechas de entrenamiento, se escogieron imágenes desde el servidor Copernicus de la ESA <https://scihub.copernicus.eu/dhus/#/home>, para la descarga de imágenes se tomaron los siguientes criterios: Imágenes con menos nubosidad, Imágenes que estén en nivel 1C para ambos sensores y en mi área de estudio hay imágenes en nivel 2A desde la fecha

2018/12/16 hasta el presente. Están imágenes ya están procesadas. Los niveles 1C tenemos que procesarlas con la librería Sen2Cor.

PRODUCTO	FECHA	TILE_ID	SATELITE	# SATELITE	SENSOR	NIVEL DE PROCESAMIENTO
S2A_MSIL1C_20160619T145732_N0204_R039_T19LBE_20160619T145846	19/06/2016	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2A_MSIL2A_20160729T145732_N0204_R039_T19LBE_20160729T145736	29/07/2016	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2B_MSIL2A_20170719T145729_N0205_R039_T19LBE_20170719T150054	19/07/2017	T19LBE	SENTINEL-2	B	MSI	2A (Sen2Cor)
S2A_MSIL2A_20170724T145731_N0205_R039_T19LBE_20170724T150005	24/07/2017	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2A_MSIL2A_20170813T145731_N0205_R039_T19LBE_20170813T145912	13/08/2017	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2A_MSIL2A_20171002T150621_N0205_R039_T19LBE_20171002T150618	2/10/2017	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2A_MSIL2A_20180530T145731_N0206_R039_T19LBE_20180530T182205	30/05/2018	T19LBE	SENTINEL-2	A	MSI	2A (Sen2Cor)
S2B_MSIL2A_20190709T145739_N0213_R039_T19LBE_20190709T183447	9/07/2019	T19LBE	SENTINEL-2	B	MSI	2A (Descargadas)
S2A_MSIL2A_20190803T145731_N0213_R039_T19LBE_20190803T191158	3/08/2019	T19LBE	SENTINEL-2	A	MSI	2A (Descargadas)
S2B_MSIL2A_20190808T145729_N0213_R039_T19LBE_20190808T200436	8/08/2019	T19LBE	SENTINEL-2	B	MSI	2A (Descargadas)
S2A_MSIL2A_20190813T145731_N0213_R039_T19LBE_20190813T185638	13/08/2019	T19LBE	SENTINEL-2	A	MSI	2A (Descargadas)
S2B_MSIL2A_20190818T145729_N0213_R039_T19LBE_20190818T183609	18/08/2019	T19LBE	SENTINEL-2	B	MSI	2A (Descargadas)

Tabla N° 10 Imágenes Utilizadas para las pruebas estadísticas.

Fuente: (Propio, 2019).

Necesitamos tener los datos limpio de nubes, aerosoles, vapor de agua que afectan en gran medida la calidad de la imagen se realizó el Filtró de nubes, necesitamos remover las nubes de las imágenes y reemplazar por imágenes sin nubes, y tener imágenes limpias y obtener la mediana de toda la colección de imágenes. A la composición se le extrae la máscara de nieve el script esta publicado en el grupo de Google Earth Engine Developers del cual se hizo algunas modificaciones según las necesidades de nuestro problema, el código está en el siguiente link habilitado públicamente:

<https://code.earthengine.google.com/fb70aa2d3f65e83b88c456327e71ade3>.

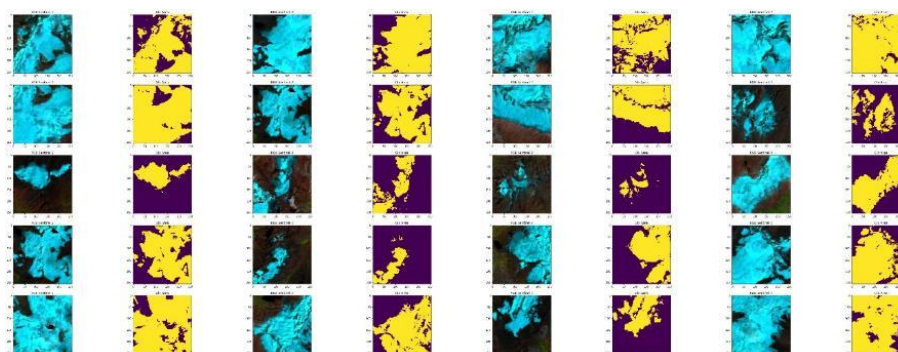


Figura N° 25 Imagen RGB y Su Target Correspondiente.

Fuente: (Propio, 2019).

En la figura N°25 se observa las imágenes y sus respectivos objetivo o target.

3.3.5. Procedimientos

3.3.5.1. Metodología para el Objetivo General

Los procedimientos están descritos detalladamente en la figura N° 26, es necesario resaltar que hay cuatros pasos principales como obtener área glaciár por el algoritmo NDSI>04, tomar puntos GPS en campo para validación, entrenar el modelo y finalmente realizar predicciones con las 12 escenas de prueba.

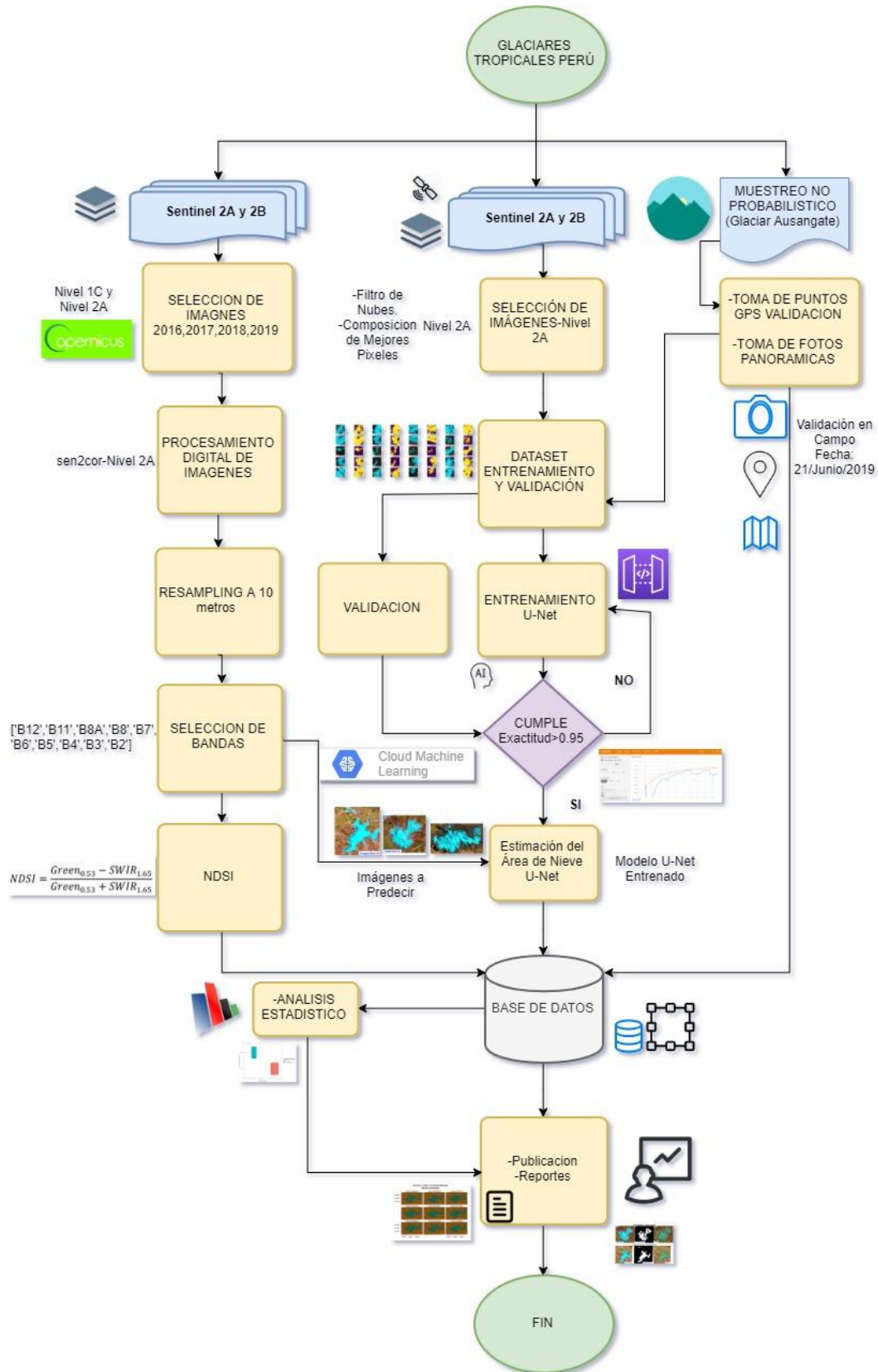


Figura N° 26 Flujo de Trabajo.

Fuente: (Propio, 2019).

3.3.5.2. Metodología para el Objetivo Especifico 2

En esta sección se muestran todos los procedimientos realizados para entrenar y validar el modelo.

3.3.5.3. Entrenamiento de la Red U-Net

Para dar a conocer los pasos que se siguió tengo que dar créditos a los creadores del siguiente código, el presente código está inspirado en la presentación Google Earth Engine User Summit 2018 presentado por (Brown & Clinton, 2018) y ejemplos publicados por la API Python por (Google, 2019), y el repositorio GitHub de (Aybar Camacho & Yali Samaniego, 2019) y todo el código se ha modificado según a las necesidades de mis problemas en este caso segmentar glaciares. El procedimiento completo está en el siguiente link: https://github.com/percyelbis/glx_U-Net está habilitado para todo el público, de las cuales pasare a detallar los puntos importantes.

Primero: Para implementar la red U-Net e interactuar con el servicio de GEE, se necesita instalar librerías como Tensorflow es una biblioteca de software de código abierto para computación numérica de alto rendimiento. Su arquitectura flexible permite una fácil implementación de cómputo en una variedad de plataformas (CPU, GPU, TPU) y desde computadoras de escritorio hasta clústeres de servidores a dispositivos móviles y de vanguardia.(Abadi et al., 2016). Instalamos la API de GEE. Como se muestran en las siguientes líneas de código Python 3.6.8.

```
!pip install -q tf-nightly-2.0-preview # tensorflow 2.0
!pip install earthengine-api==0.1.175 # earthengine API
!pip install -U -q PyDrive # Interacción con Google drive
# Cargar la extensión TensorBoard.
%load_ext tensorboard
```

Segundo: Tenemos que autenticarnos con los servicios de Google Cloud Storage (GCS) y GEE, esto para poder interactuar con los servicios. Las siguientes líneas de código muestran como autenticarse con los servicios de GCS y GEE.

```
# GCS y Google Drive.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

```
# GEE.
!earthengine authenticate
```

Tercero: Definimos la función maskS2clouds para que nos devuelvan imágenes enmascaradas y escaladas. Las muestras se estandarizaron, lo cual significa que todas estén e en [0 a 1] esto para que computacionalmente sea más rápido el proceso, se muestra las siguientes líneas de código:

```
# Funcion
'''
* Function to mask clouds using the Sentinel-2 QA band.
* Referencia : https://developers.google.com/earth-
engine/datasets/catalog/COPERNICUS_S2.
* Parametros : Image Sentinel-2 --->>> Nivel 2A.
* Return : Devuelve los datos enmascarados y escalados, sin las ba
ndas de control de calidad.
'''
def maskS2clouds(img):
    # Pixel QA band.
    qa = img.select('QA60')
    # Bits 10 and 11 are clouds and cirrus, respectively.
    cloudBitMask = (1 << 10)
    cirrusBitMask = (1 << 11)
    # Get the pixel QA band.

    # Both flags should be set to zero, indicating clear conditions.
    mask = qa.bitwiseAnd(cloudBitMask).eq(0) \
        .And(qa.bitwiseAnd(cirrusBitMask).eq(0))
    return img.updateMask(mask) \
        .divide(10000) \
        .copyProperties(img, ["system:time_start"])
```

Cuarto: Definir el Dataset de entrenamiento con las siguientes líneas de código.

```
#---Datos de Entrenamiento/Validacion.
# Importar dataset de Entrenamiento y Validacion.
train_glaciar = ee.FeatureCollection('users/torchi_12/train_dataset
_2019')
test_glaciar = ee.FeatureCollection('users/torchi_12/test_dataset_2
019')

# Display the train/test dataset
db_glx = train_glaciar.merge(test_glaciar)
center = db_glx.geometry().centroid().getInfo()['coordinates']
center.reverse()
#---Crear un Diccionario.
```

```

dicc = {'train': train_glaciar.draw(**{'color': 'FF0000', 'strokeWidth': 5}).getMapId(),
        'test' : test_glaciar.draw(**{'color': '0000FF', 'strokeWidth': 5}).getMapId(),
        }

#---Dataset Sentinel 2
'''
Sentinel-2 MSI: MultiSpectral Instrument, Level-2A
Dataset Availability : 2017-03-28T00:00:00 - Present
Referencia : https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2_SR
'''
# Seleccionar Bandas.
bands = ['B12', 'B11', 'B8A', 'B8', 'B7', 'B6', 'B5', 'B4', 'B3', 'B2']
# Mar 28, 2017 - Jul 11, 2019 Se tiene el Nivel 2A.
# En el area de entrenamiento solo se tiene desde 2018-12-13 hasta 2019-06-29.
s2_sr = ee.ImageCollection("COPERNICUS/S2_SR")\
        .filterBounds(db_glx)\
        .filterDate('2018-12-12', '2019-06-30')\
        .map(maskS2clouds)\
        .median()

# Load target.
target_glx = ee.Image('users/torchi_12/target_glx_s2').eq(1).rename('target')

# Rename Bands.
s2_msi = s2_sr.select(bands).rename(['SWIR2', 'SWIR1', 'RE4', 'NIR', 'RE3', 'RE2', 'RE1', 'R', 'G', 'B'])
# Add Bands s2_ndsi, target.
s2 = s2_msi.addBands(target_glx)

```

Bandas que se utilizaron en esta investigación se muestran la siguiente tabla N°11

Número de banda	S2A		S2B		Resolución espacial (m)
	Longitud de onda central (nm)	Ancho de banda (nm)	Longitud de onda central (nm)	Ancho de banda (nm)	
2	492.4	66	492.1	66	10
3	559.8	36	559	36	10
4	664.6	31	664.9	31	10
5	704.1	15	703.8	16	20
6	740.5	15	739.1	15	20
7	782.8	20	779.7	20	20
8	832.8	106	832.9	106	10
8a	864.7	21	864	22	20
11	1613.7	91	1610.4	94	20
12	2202.4	175	2185.7	185	20

Tabla N° 11 Bandas que se utilizaron en el entrenamiento.

Fuente: (European Space Agency, 2015a).

Quinto: Exportar la vecindad de pixeles con la función llamada `ee.Image.neighborhoodToArray`, la palabra reservada `ee.Image.neighborhoodToArray` de GEE, Convierte la vecindad de cada píxel en una imagen escalar en una matriz 2D. Los ejes 0 y 1 de la matriz de salida corresponden a los ejes Y y X de la imagen, respectivamente. La imagen de salida tendrá tantas bandas como la entrada; cada banda de salida tiene la misma máscara que la banda de entrada correspondiente. La huella y los metadatos de la imagen de entrada se conservan. (Google Developers, 2015).

```
selectors = ['SWIR2', 'SWIR1', 'RE4', 'NIR', 'RE3', 'RE2', 'RE1', 'R', 'G', 'B', 'target']
train_filenames = saveCNN_batch(s2, train_glaciar, 128, 10, 'trainUNET', selectors, folder='UNET', bucket='tesis_dataset')
test_filenames = saveCNN_batch(s2, test_glaciar, 128, 10, 'testUNET', selectors, folder='UNET', bucket='tesis_dataset')
```

Guardar el Dataset de entrenamiento y Validación en formato .TFRecords en GCS.

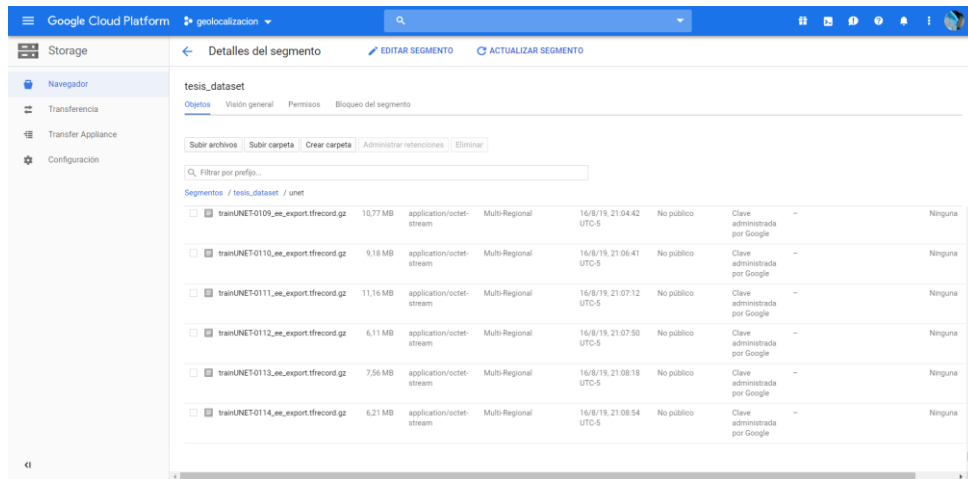


Figura N° 27 Almacenamiento de Dataset en Google Cloud Storage

Fuente: (Google Cloud, 2014).

Séptimo: Definir la forma de la Imagen, es decir la dimensión de la imagen, se definió de la siguiente forma 256x256x10 cada pixel con un valor entre 0 a 1 es decir las reflectancia normalizadas $[0,1]$, teniendo como neuronas 655360 en la capa de entrada la red. El modelo devolverá una predicción (y') entre 0 y 1 es decir no glaciar y si es glaciar respectivamente.

Al respecto de las épocas es el número de pasadas en los datos, es algo empírico para esta investigación se utilizó 20 después de muchas pruebas y error.

```
# Definir la Forma de la Imagen de Entrada.
IMG_SHAPE = (256, 256, 10)
EPOCHS = 20
```

Octavo: En TensorFlow, ya están implementado los algoritmos de optimización y además esta con Keras no tenemos que implementar desde cero ya existen herramientas muy poderosas. Creamos el modelo U-Net con Keras. Se creó un modelo de CNN con:

- 5 capas de codificador: Se compone CONV, Normalización por Lote (BN) y ReLu seguidas de un Max Pool, cada Max Pool reducirá la resolución espacial de nuestro mapa de características en un factor de 2.
- 5 capas de decodificador: Se compone de una muestreo ascendente (UpSampling 2D), CONV, BN y ReLu, luego concatenamos el mapa de características del mismo tamaño en el lado del decodificador

- 1 capa de salida se compone de la operación CONV final que realiza una convolución a lo largo de las bandas para cada pixel individual, genera nuestra mascara de segmentación en escalas de grises.

- El diseño que se realizo está inspirado en la publicación de este artículo científico, (Ronneberger et al., 2015) en este caso lo aplicaremos a distinguir que las imágenes que ingresamos con sus respectivas bandas y predijo si es glaciar o no es glaciar.

Los paso para construir la red U-Net son explicadas detalladamente en el marco teórico en la figura N°.15

Capa de Entrada: La imagen de 256x256x10, es decir esta capa no tiene nada que aprender entonces los parámetros es = 0

Capa Convolutional (CONV): Indica la convolutional de la capa. En la primera capa ocurre un filtro de una matriz de 3x3 y aplicar la función de activación ReLu. Aquí es donde CNN aprende.

Capa de Agrupación (POOL): usamos el submuestreo Max Pooling explicado cómo funciona en el marco teórico, en esta investigación va a reducir en ancho = 2 y alto = 2 y esto tiene parámetros = 0.

Batch Normalization (BN): Indica la normalización por lotes en la capa.

Capa Totalmente Conectada (FC): En Esta capas tiene el mayor número de parámetros, cada neurona está conectada a todas las demás neuronas.

Definimos los parámetros del modelo con las siguientes líneas.

```
from tensorflow.keras import layers

def conv_block(input_tensor, num_filters):
    encoder = layers.Conv2D(num_filters, (3, 3), padding='same')(input_tensor) # Los filtros son 3x3--->Capa de Convolucion
    encoder = layers.BatchNormalization()(encoder) # Capa de Normalizacion
    encoder = layers.Activation('relu')(encoder) # Capa de Activacion
    encoder = layers.Conv2D(num_filters, (3, 3), padding='same')(encoder)
    encoder = layers.BatchNormalization()(encoder)
```



```

encoder = layers.Activation('relu')(encoder)
return encoder

def encoder_block(input_tensor, num_filters):
    encoder = conv_block(input_tensor, num_filters)
    encoder_pool = layers.MaxPooling2D((2, 2), strides=(2, 2))(encoder)

    return encoder_pool, encoder

def decoder_block(input_tensor, concat_tensor, num_filters):
    decoder = layers.Conv2DTranspose(num_filters, (2, 2), strides=(2, 2), padding='same')(input_tensor)
    decoder = layers.concatenate([concat_tensor, decoder], axis=-1)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    decoder = layers.Conv2D(num_filters, (3, 3), padding='same')(decoder)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    decoder = layers.Conv2D(num_filters, (3, 3), padding='same')(decoder)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    return decoder

inputs = layers.Input(shape=IMG_SHAPE)
# 256
encoder0_pool, encoder0 = encoder_block(inputs, 32)
# 128
encoder1_pool, encoder1 = encoder_block(encoder0_pool, 64)
# 64
encoder2_pool, encoder2 = encoder_block(encoder1_pool, 128)
# 32
encoder3_pool, encoder3 = encoder_block(encoder2_pool, 256)
# 16
encoder4_pool, encoder4 = encoder_block(encoder3_pool, 512)
# 8
center = conv_block(encoder4_pool, 1024)
# center
decoder4 = decoder_block(center, encoder4, 512)
# 16
decoder3 = decoder_block(decoder4, encoder3, 256)
# 32
decoder2 = decoder_block(decoder3, encoder2, 128)
# 64
decoder1 = decoder_block(decoder2, encoder1, 64)

```

```
# 128
decoder0 = decoder_block(decoder1, encoder0, 32)
# 256
outputs = layers.Conv2D(1, (1, 1), activation='sigmoid')(decoder0)
```

Definimos el modelo.

```
from tensorflow.keras import models
model = models.Model(inputs=[inputs], outputs=[outputs])
model.summary()
```

Operación de cada capa se muestra en la siguiente tabla N°12

Capa	(Tipo)	Forma de Salida	# Parametros	Conectado a
input_1	(InputLayer)	256x256x10	0	
conv2d	(Conv2D)	256x256x32	2912	input_1[0][0]
batch_normalization	(BatchNorma	256x256x32	128	conv2d[0][0]
activation	(Activation)	256x256x32	0	batch_normalization[0][0]
conv2d_1	(Conv2D)	256x256x32	9248	activation[0][0]
batch_normalization_1	(BatchNor	256x256x32	128	conv2d_1[0][0]
activation_1	(Activation)	256x256x32	0	batch_normalization_1[0][0]
max_pooling2d	(MaxPooling2D)	128x128x32	0	activation_1[0][0]
conv2d_2	(Conv2D)	128x128x64	18496	max_pooling2d[0][0]
batch_normalization_2	(BatchNor	128x128x64	256	conv2d_2[0][0]
activation_2	(Activation)	128x128x64	0	batch_normalization_2[0][0]
conv2d_3	(Conv2D)	128x128x64	36928	activation_2[0][0]
batch_normalization_3	(BatchNor	128x128x64	256	conv2d_3[0][0]
activation_3	(Activation)	128x128x64	0	batch_normalization_3[0][0]
max_pooling2d_1	(MaxPooling2D)	64x64x64	0	activation_3[0][0]
conv2d_4	(Conv2D)	64x64x128	73856	max_pooling2d_1[0][0]
batch_normalization_4	(BatchNor	64x64x128	512	conv2d_4[0][0]
activation_4	(Activation)	64x64x128	0	batch_normalization_4[0][0]
conv2d_5	(Conv2D)	64x64x128	147584	activation_4[0][0]
batch_normalization_5	(BatchNor	64x64x128	512	conv2d_5[0][0]
activation_5	(Activation)	64x64x128	0	batch_normalization_5[0][0]
max_pooling2d_2	(MaxPooling2D)	32x32x128	0	activation_5[0][0]
conv2d_6	(Conv2D)	32x32x256	295168	max_pooling2d_2[0][0]
batch_normalization_6	(BatchNor	32x32x256	1024	conv2d_6[0][0]
activation_6	(Activation)	32x32x256	0	batch_normalization_6[0][0]
conv2d_7	(Conv2D)	32x32x256	590080	activation_6[0][0]
batch_normalization_7	(BatchNor	32x32x256	1024	conv2d_7[0][0]
activation_7	(Activation)	32x32x256	0	batch_normalization_7[0][0]
max_pooling2d_3	(MaxPooling2D)	16x16x256	0	activation_7[0][0]
conv2d_8	(Conv2D)	16x16x512	1180160	max_pooling2d_3[0][0]
batch_normalization_8	(BatchNor	16x16x512	2048	conv2d_8[0][0]
activation_8	(Activation)	16x16x512	0	batch_normalization_8[0][0]
conv2d_9	(Conv2D)	16x16x512	2359808	activation_8[0][0]
batch_normalization_9	(BatchNor	16x16x512	2048	conv2d_9[0][0]
activation_9	(Activation)	16x16x512	0	batch_normalization_9[0][0]
max_pooling2d_4	(MaxPooling2D)	8x8x512	0	activation_9[0][0]
conv2d_10	(Conv2D)	8x8x1024	4719616	max_pooling2d_4[0][0]
batch_normalization_10	(BatchNo	8x8x1024	4096	conv2d_10[0][0]
activation_10	(Activation)	8x8x1024	0	batch_normalization_10[0][0]
conv2d_11	(Conv2D)	8x8x1024	9438208	activation_10[0][0]
batch_normalization_11	(BatchNo	8x8x1024	4096	conv2d_11[0][0]
activation_11	(Activation)	8x8x1024	0	batch_normalization_11[0][0]
conv2d_transpose	(Conv2DTranspo	16x16x512	2097664	activation_11[0][0]
concatenate	(Concatenate)	16x16x1024	0	conv2d_transpose[0][0]
batch_normalization_12	(BatchNo	16x16x1024	4096	concatenate[0][0]
activation_12	(Activation)	16x16x1024	0	batch_normalization_12[0][0]
conv2d_12	(Conv2D)	16x16x512	4719104	activation_12[0][0]
batch_normalization_13	(BatchNo	16x16x512	2048	conv2d_12[0][0]
activation_13	(Activation)	16x16x512	0	batch_normalization_13[0][0]
conv2d_13	(Conv2D)	16x16x512	2359808	activation_13[0][0]
batch_normalization_14	(BatchNo	16x16x512	2048	conv2d_13[0][0]

activation_14	(Activation)	16x16x512	0	batch_normalization_14[0][0]
conv2d_transpose_1	(Conv2DTrans	32x32x256	524544	activation_14[0][0]
concatenate_1	(Concatenate)	32x32x512	0	activation_7[0][0]
batch_normalization_15	(BatchNo	32x32x512	2048	conv2d_transpose_1[0][0]
activation_15	(Activation)	32x32x512	0	concatenate_1[0][0]
conv2d_14	(Conv2D)	32x32x256	1179904	batch_normalization_15[0][0]
batch_normalization_16	(BatchNo	32x32x256	1024	activation_15[0][0]
activation_16	(Activation)	32x32x256	0	conv2d_14[0][0]
conv2d_15	(Conv2D)	32x32x256	590080	batch_normalization_16[0][0]
batch_normalization_17	(BatchNo	32x32x256	1024	activation_16[0][0]
activation_17	(Activation)	32x32x256	0	conv2d_15[0][0]
conv2d_transpose_2	(Conv2DTrans	64x64x128	131200	batch_normalization_17[0][0]
concatenate_2	(Concatenate)	64x64x256	0	activation_17[0][0]
batch_normalization_18	(BatchNo	64x64x256	1024	activation_5[0][0]
activation_18	(Activation)	64x64x256	0	conv2d_transpose_2[0][0]
conv2d_16	(Conv2D)	64x64x128	295040	concatenate_2[0][0]
batch_normalization_19	(BatchNo	64x64x128	512	batch_normalization_18[0][0]
activation_19	(Activation)	64x64x128	0	activation_18[0][0]
conv2d_17	(Conv2D)	64x64x128	147584	conv2d_16[0][0]
batch_normalization_20	(BatchNo	64x64x128	512	batch_normalization_19[0][0]
activation_20	(Activation)	64x64x128	0	activation_19[0][0]
conv2d_transpose_3	(Conv2DTrans	128x128x64	32832	conv2d_17[0][0]
concatenate_3	(Concatenate)	128x128x128	0	batch_normalization_20[0][0]
batch_normalization_21	(BatchNo	128x128x128	512	activation_20[0][0]
activation_21	(Activation)	128x128x128	0	activation_3[0][0]
conv2d_18	(Conv2D)	128x128x64	73792	conv2d_transpose_3[0][0]
batch_normalization_22	(BatchNo	128x128x64	256	concatenate_3[0][0]
activation_22	(Activation)	128x128x64	0	batch_normalization_21[0][0]
conv2d_19	(Conv2D)	128x128x64	36928	activation_21[0][0]
batch_normalization_23	(BatchNo	128x128x64	256	conv2d_18[0][0]
activation_23	(Activation)	128x128x64	0	batch_normalization_22[0][0]
conv2d_transpose_4	(Conv2DTrans	256x256x32	8224	activation_22[0][0]
concatenate_4	(Concatenate)	256x256x64	0	conv2d_19[0][0]
batch_normalization_24	(BatchNo	256x256x64	256	batch_normalization_23[0][0]
activation_24	(Activation)	256x256x64	0	activation_23[0][0]
conv2d_20	(Conv2D)	256x256x32	18464	activation_1[0][0]
batch_normalization_25	(BatchNo	256x256x32	128	conv2d_transpose_4[0][0]
activation_25	(Activation)	256x256x32	0	concatenate_4[0][0]
conv2d_21	(Conv2D)	256x256x32	9248	batch_normalization_24[0][0]
batch_normalization_26	(BatchNo	256x256x32	128	activation_24[0][0]
activation_26	(Activation)	256x256x32	0	conv2d_20[0][0]
conv2d_22	(Conv2D)	256x256x1	33	conv2d_21[0][0]
=====				batch_normalization_25[0][0]
Total de Parametros:				activation_25[0][0]
				activation_26[0][0]
Parametros Entrenables:				
Parametros No Entrenables:				

Tabla N° 12 Definición y Operación en cada Capa.

Fuente: (Propio, 2019).

Como resumen el total de parámetros es la suma de todos los parámetros, parámetros no entrenables son aquellos que no se actualizarán y optimizaran durante el entrenamiento.

Conv2d es la operación de la convolución definido como 3x3.

Noveno: La métrica utilizadas para evaluar la puntuación de nuestro entrenamiento fue el coeficiente dice, y entropía cruzada.

```
from tensorflow.keras import losses

def dice_coeff(y_true, y_pred):
    smooth = 1.
    # Flatten
    y_true_f = tf.reshape(y_true, [-1])
    y_pred_f = tf.reshape(y_pred, [-1])
    intersection = tf.reduce_sum(y_true_f * y_pred_f)
    score = (2. * intersection + smooth) / (tf.reduce_sum(y_true_f)
    + tf.reduce_sum(y_pred_f) + smooth)
    return score

def dice_loss(y_true, y_pred):
    loss = 1 - dice_coeff(y_true, y_pred)
    return loss

def bce_dice_loss(y_true, y_pred):
    loss = losses.binary_crossentropy(y_true, y_pred) + dice_loss(y_t
rue, y_pred)
    return loss
```

Seguidamente nuestras pérdidas personalizadas lo compilamos con Keras para entrenar el modelo. Utilizamos el optimizador Adam es el más usado en el mundo de las redes neuronales profundas, el optimizador Adam me permite determinar la función de costo para los peso, las pérdidas que se definió en las anteriores líneas y finalmente tenemos las métricas no va dar la precisiones. El algoritmo de optimización Adam es muy adecuado para problemas que son grandes en términos de datos y/o parámetros. El método también es adecuado para objetivos no estacionarios y problemas con gradientes muy ruidosos y/o dispersos. Resultados empíricos demuestran que Adam funciona bien en la práctica y se compara favorablemente con otros métodos de optimización.(Kingma & Ba, 2014). Adam es conocido por converger rápido durante el entrenamiento y es muy usado en la comunidad de aprendizaje automático y ha resuelto muchos problemas.

```
# Compilamos el Modelo.
model.compile(optimizer='adam', loss=bce_dice_loss, metrics=[dice_1
oss, 'accuracy'])
```

Decimo: Seguidamente lo entrenamos el modelo con las siguientes líneas de código. Todos los procedimientos descritos se realizan en la nube. El Batch size o tamaño del lote nos indica el número de observaciones después de actualizar los pesos en nuestro caso vamos a tomar como 10. El modelo entrenado se guardó en el siguiente link https://drive.google.com/open?id=10_cMA9KHgkXnS1hUuxem4IVV_vDRLZ1a

```
#### Entrenamiento Opcional ya se tiene un modelo pre-
entrenado guardado en google drive.####
N_train = 1883
# Validacion
N_test = 1470

# Tamaño de Lote.
batch_size = 10

# Train the model.
history = model.fit(train_dba,
                    validation_data=test_dba,
                    steps_per_epoch= int(np.ceil(N_train / float(ba
tch_size))),
                    validation_steps= int(np.ceil(N_test / float(ba
tch_size))),
                    epochs=EPOCHS,
                    callbacks=[tensorboard_callback, es, mcp])
```

En el artículo de (Raina, Madhavan, & Ng, 2009). Menciona que entrenar con GPU da una velocidad de entrenamiento 70 veces más rápido que hacerlo con una CPU multinúcleo. Este proceso de entrenamiento tomo un tiempo aproximado de ~11 horas.

Tiempo de Entrenamiento	GPU
Entrenamiento	1883 muestras
Validacion	1470 muestras
Epocas	20
Tiempo por Epoca Promedio (s)	2000.4
Tiempo Total (Horas)	11

Tabla N° 13 Tiempo de entrenamiento.

Fuente: (Propio, 2019)

ARGUMENTOS	VALOR
Tamaño de la Muestra	256x256x10 pixeles
Entrenamiento	1883 muestras
Validacion	1470 muestras
Epocas	20
Batch_size	10
Loss Function	Cross entropy o Entropia Cruzada, Coeficiente Dice.
Optimizador	Adam
Funcion de Activacion por Capa	ReLU
Funcion de Activacion del Modelo	Sigmoide
Numero de Canales o Bandas	10
Image_size (Numero de Pixeles)	256

Tabla N° 14 Resumen de los argumentos Utilizados para el Modelo U-net.

Fuente: (Propio, 2019).

Undécimo: Guardamos el modelo pre-entrenada, durante el entrenamiento donde se calcularon los pesos óptimos, estos pesos determinar el funcionamiento de la red. El modelo puede ser usado a futuro no hay necesidad de volver a entrenar, el modelo Glx_Unet.h5 puede ser usado en cualquier glaciar con imágenes Sentinel 2 el link está habilitado públicamente:

https://drive.google.com/open?id=1_1M9_zziF753YSVWEx3nKFZx0uJJtgIk

Duodécimo: La precisión del entrenamiento es cercana a 0.9789 y la precisión de la validación es cercana a 0.9788186. Podemos visualizar la curva de aprendizaje en la librería TensorBoard. Nuestro Dataset de validación son 1470 muestras.

RESULTADOS					
Modelo	Accuracy	Loss	Dice_loss	Epocas	Tiempo (Horas)
U-Net	0.9788	0.0839	0.0286	20	~11

Tabla N° 15 Resultados de la Precisión del Modelo.

Fuente: (Propio, 2019).

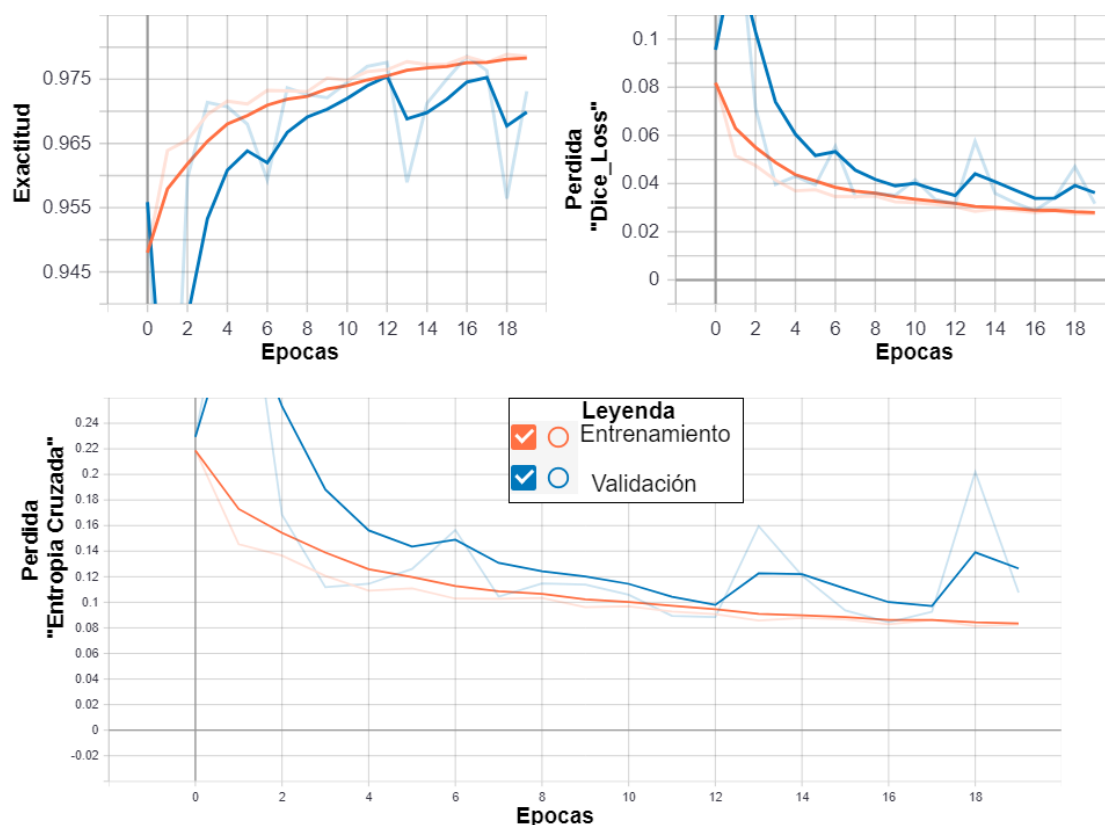


Figura N° 28 Curva de Aprendizaje Durante el Entrenamiento.

Fuente: (Propio, 2019).

Interpretación: Puede verse que la función de costo disminuye. Muestra que los parámetros están aprendiendo. Pero sin embargo se puede ver que el modelo se puede entrenar aún más. Lo ideal es que las precisiones aumenten y las pérdidas bajen en el tiempo, el sobreajuste mayormente se da cuando hay pocos datos y forzamos a darle muchos datos más.

Decimotercero: Realizar las Predicciones, se tiene que insertar la ID de la Imagen en la variable s2.

```
# Area Glaciar a Predecir.
xmin,ymin,xmax,ymax = [-71.2895576800648882,-13.8284791303437160, -
71.1695138486407046,-13.7602836898167542]

# Qgis Extension
#-71.2895576800648882,-13.8284791303437160 : -71.1695138486407046,-
13.7602836898167542---> Ausangate
#-70.8966675314120351,-13.9960341865535920 : -70.7458947766221655,-
13.8373832645521784---> Quelccaya
```



```

#-77.6893128297523248,-9.1858434409387861 : -77.5221655263183607,-
9.0342021137613369---> Huascarán
#-73.04705802581952,-50.4798111720673,-72.73120109222577,-
50.310848619751894--->Perito
# Passing a rectangle (prediction area) to Earth Engine
ausangate = ee.Geometry.Rectangle([xmin, ymin, xmax, ymax])
# Desde GEE
s2 = ee.Image('COPERNICUS/S2_SR/20190917T145729_20190917T150535_T19
LBE')
# Normalizando 0-1
s2_msi = s2.multiply(0.0001)
# Renombrar bandas.
s2p = s2_msi.select(bands).rename(['SWIR2', 'SWIR1', 'RE4', 'NIR', 'RE3
', 'RE2', 'RE1', 'R', 'G', 'B'])
# Extraer la Fecha.
date = ee.Date(s2.get('system:time_start'))
fechasat= date.format('YMd').getInfo()
print(fechasat)

```

Las escenas segmentadas se puede observar en la siguiente figura N°29 el modelo nos arroja probabilidades en cada pixel, entonces tenemos que darle un umbral $Glx > 0.5$.

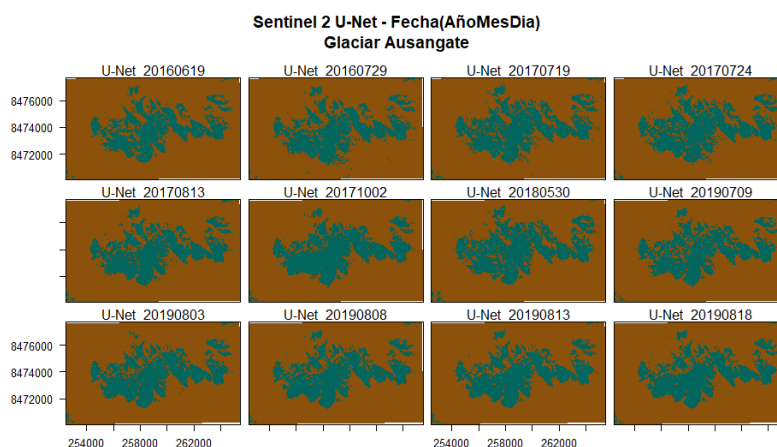


Figura N° 29 Estimación de área glaciar Ausangate.

Fuente: (Propio, 2019).

El procedimiento que se detalló es prueba error y ver cuál es el mejor que modelo que se ajusta a nuestras necesidades. Es muy difícil saber de antemano qué enfoque funcionará mejor para un nuevo problema. Incluso los investigadores experimentados en aprendizaje automático generalmente probarán muchas docenas de ideas antes de descubrir algo satisfactorio.(Ng, 2018).

3.3.5.4. Metodología para el Objetivo Especifico 3

La Metodología para el objetivo específico 3 es analizar las diferencias que existen entre los dos métodos NDSI y U-Net.

3.3.5.5. Calculo de NDSI

Las imágenes tiene que estar en reflectancia de superficie es decir en Nivel 2A procesado con la librería Sen2Cor, seguidamente se realiza el mapeo de la cobertura glaciara. El NDSI el algoritmo definido por (Dozier, 1989), el resultado de la imagen tiene valores de -1 a 1 es decir [-1,1], se tiene que segmentar definiendo un umbral $NDSI > 0.4$.

$$NDSI = \frac{Green_{0.53} - SWIR_{1.65}}{Green_{0.53} + SWIR_{1.65}}$$

Las bandas que se utilizaran para Sentinel 2 son: SWIR 1 = B11 y Green = B3, la banda 11 tiene que hacerse un re muestreo a 10 metros. Los resultados obtenidos aplicando la expresión matemática y el umbral de 0.4 en el software SNAP 6.0.9. En las figuras N°30 se observa el mapeo con $NDSI > 0.4$ en las 12 escenas de prueba.

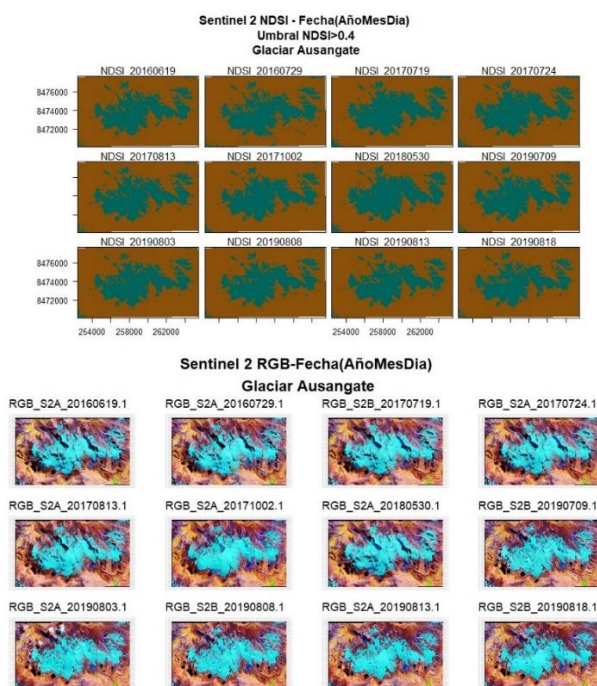


Figura N° 30 Área glaciara por el Algoritmos $NDSI > 0.4$ en Diferentes Fechas.

Fuente: (Propio, 2019).

IV. RESULTADOS Y DISCUSIÓN

4.1. Prueba T-Student para Muestras Relacionadas

✓ Para el objetivo general

Se registró las estimaciones de área glaciar Ausangate en 12 escenas, con lo dos métodos de cartografiado el $NDSI > 0.4$ y la Arquitectura U-Net, las áreas registradas se muestran en la siguiente tabla N°16.

PRODUCTO	FECHA	AREA NDSI (km2)	AREA U-Net (Km2)
S2A_MSIL1C_20160619T145732_N0204_R039_T19LBE_20160619T145846	19/06/2016	26.109	19.186
S2A_MSIL2A_20160729T145732_N0204_R039_T19LBE_20160729T145736	29/07/2016	28.351	21.219
S2B_MSIL2A_20170719T145729_N0205_R039_T19LBE_20170719T150054	19/07/2017	29.911	22.735
S2A_MSIL2A_20170724T145731_N0205_R039_T19LBE_20170724T150005	24/07/2017	29.426	22.670
S2A_MSIL2A_20170813T145731_N0205_R039_T19LBE_20170813T145912	13/08/2017	27.440	21.918
S2A_MSIL2A_20171002T150621_N0205_R039_T19LBE_20171002T150618	2/10/2017	26.928	22.862
S2A_MSIL2A_20180530T145731_N0206_R039_T19LBE_20180530T182205	30/05/2018	29.739	23.492
S2B_MSIL2A_20190709T145739_N0213_R039_T19LBE_20190709T183447	9/07/2019	27.339	21.070
S2A_MSIL2A_20190803T145731_N0213_R039_T19LBE_20190803T191158	3/08/2019	27.065	21.995
S2B_MSIL2A_20190808T145729_N0213_R039_T19LBE_20190808T200436	8/08/2019	26.798	21.749
S2A_MSIL2A_20190813T145731_N0213_R039_T19LBE_20190813T185638	13/08/2019	25.947	21.307
S2B_MSIL2A_20190818T145729_N0213_R039_T19LBE_20190818T183609	18/08/2019	25.636	21.087

Tabla N° 16 Áreas Obtenidas por el Método $NDSI > 0.4$ y U-Net

Fuente: (Propio, 2019).

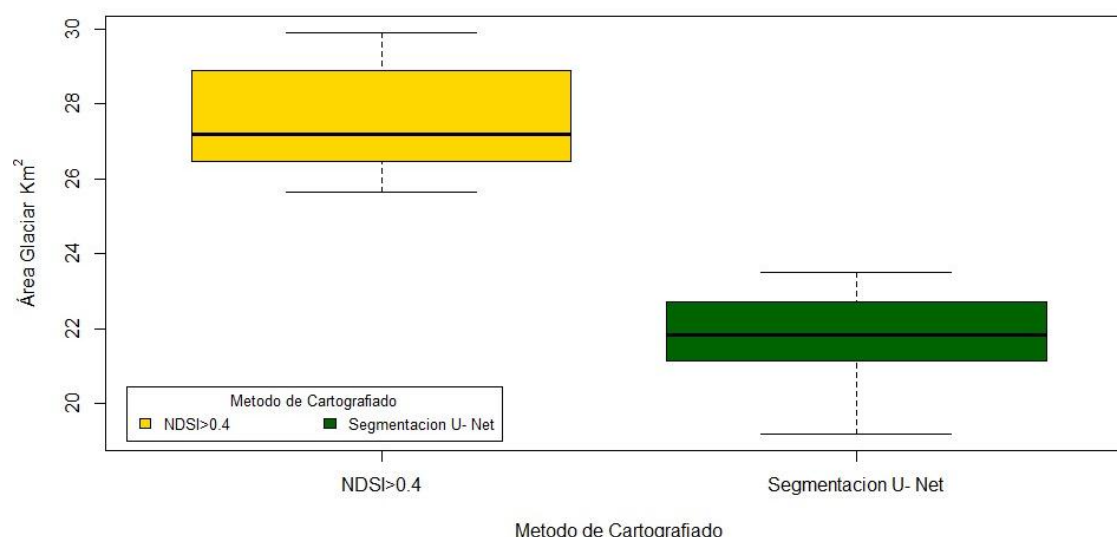


Figura N° 31 Diagrama de Cajas.

Fuente: (Propio, 2019)

En la figura N°31 Podemos observar que la mediana, no es igual en ambos métodos de cartografiado. Para ver la distribución normal se realizara la prueba de normalidad de Shapiro Wilk en el software estadístico r statics.

Hipótesis Nula (Ho): Normalidad.

Hipótesis Alterna (Ha): No hay Normalidad.

```
> # Pruebas de Normalidad Shapiro Wilk, comparar con la teórica
> # Ho: Hay Normalidad
> # Ha: No Hay Normalidad
> shapiro.test(ndsi)
```

```
Shapiro-Wilk normality test
```

```
data: ndsi
W = 0.91025, p-value = 0.2149 # Se acepta la Ho
```

```
> shapiro.test(unet)
```

```
Shapiro-Wilk normality test
```

```
data: unet
W = 0.93761, p-value = 0.4677 # Se acepta la Ho
```

p-value = 0.2149 para el área estimado por NDSI.

p-value = 0.4677 para el área estimado por U-Net.

Entonces se acepta la Ho, indica que hay normalidad. Y aplicamos la prueba estadística T-Student para muestras relacionadas o pareadas. Para determinar si existen diferencias significativas, las muestras de prueba son inferiores a 30, se tiene un tamaño muestral de $n = 12$ escenas.

Hipótesis Estadísticas:

Ho: Área NDSI = Área segmentación U-Net.

Ha: Área NDSI \neq Área segmentación U-Net.

Nivel de significancia: $\alpha = 0.05$.

Tomar Muestra: $n = 12$.

Prueba Estadística: T –Student pareada para dos colas.

```
> t.test(ndsi, unet, mu = 0, alt = "two.sided", paired = T,
conf.level = 0.95)

Paired t-test

data: ndsi and unet
t = 18.184, df = 11, p-value = 1.48e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 5.083229 6.483204
sample estimates:
mean of the differences
      5.783217
```

Cálculo del valor p: $p\text{-value} = 1.48e-09$.

Interpretación, si el p valor es menor o igual que 0.05, entonces rechazamos la hipótesis nula.

Conclusión: Hay una diferencia significativa en la media de las áreas tomadas en diferentes fechas en el glaciar Ausangate. Por lo cual se concluye que el umbral estándar de 0.4 tiene limitaciones y siempre tiene que intervenir la eliminación manual de polígonos de lagunas glaciares, El modelo U-Net demostró ser robusto al mapear.

✓ **Para los objetivos específicos:**

- Se entrenó el modelo U-net con 1883 muestras, 1470 muestras para validación y se utilizaron 12 escenas para predecir toda esta estas muestras se recolectaron en la plataforma GEE, Demostrando GEE ser una plataforma geoespacial potente, veloz y eficiente en el procesamiento y análisis de imágenes satelitales.

- En el proceso de Entrenamiento y validación se lograron precisiones del 97.89% y 97.88%.

- La diferencias de estimación de áreas por los dos métodos $NDSI > 0.4$ y U-Net se puede observar visualmente ciertas diferencias como se muestras en Figura N°32. Con un p valor de $1.48e-09$.

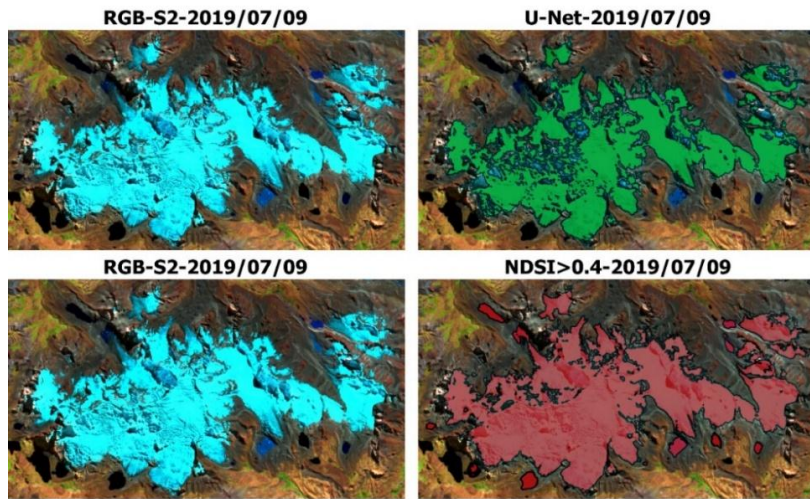


Figura N° 32 Comparación U-Net y NDSI Fecha 2019/07/09.

Fuente: (Propio, 2019).

- En la Figura N°33 se observa que al aplicar un modelo predictivo este puede incurrir en predicciones erróneas. Exactamente en el grafico A el cual está sobreestimando, en el grafico B se observa el modelo U-Net.

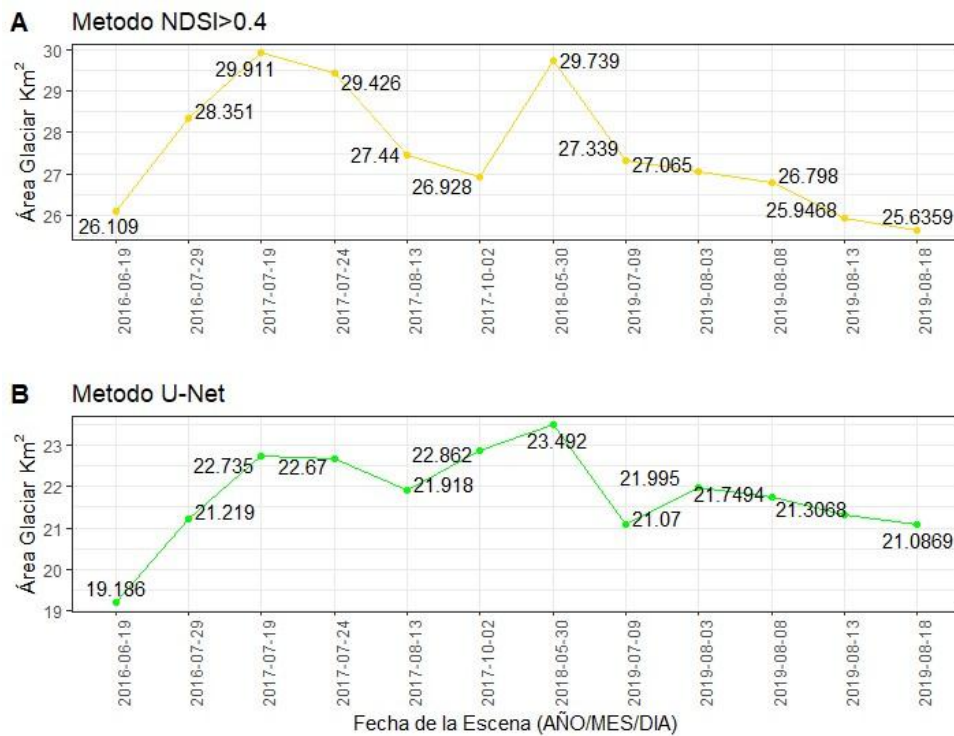


Figura N° 33 Serie de Tiempo en las 12 escenas en el Ausangate.

Fuente: (Propio, 2019).

4.2. Discusión

✓ Para El Objetivo General

- Al Aplicar el Modelo U-Net en el Cartografiado de glaciares fue muy exitoso y se puede utilizar para cualquier glaciar donde se quiera monitorear, realizar estudios hidrológicos. En el artículo de (Mohajerani et al., 2019) donde realizó un mapeo continuo y preciso de los frentes de nacimiento de glaciares es esencial para monitorear y comprender los rápidos cambios de glaciares en la Antártida y Groenlandia, esto surgió principalmente por el problema de la digitalización manual lenta y laboriosa como solución se desarrolló herramientas que se basa en la segmentación de imágenes semánticas utilizando redes neuronales convolucionales (CNN) con una arquitectura U-Net modificada.

✓ Para Los Objetivos Específicos

- Se entrenó el modelo U-net con 1883 muestras y 1470 muestras para validación toda esta estas muestras se recolectaron en la plataforma GEE se etiquetó con valores de 1 = Glaciar y 0 = No Glaciar. En la Investigación de (Mohajerani et al., 2019) se utilizó la arquitectura U-Net donde se recolectó un conjunto de datos de imágenes y sus correspondientes frentes de parto determinados manualmente con imágenes Landsat 5, 7 y 8.

- En el proceso de entrenamiento y validación se lograron precisiones del 97.89% y 97.88%, El modelo U-Net se entrenó durante 20 épocas el tiempo aproximado es ~11 horas. Estos resultados guardan una relación con lo que sostiene (Mohajerani et al., 2019) donde el entrenamiento del modelo U-Net durante 54 épocas, conduce a una precisión del 92.4% en el conjunto de entrenamiento y del 93.6% en el conjunto de validación en el mapeo de frentes de nacimiento de glaciares. En otra investigación que ha sido aplicado a segmentar tipos de bosques escala regional con el modelo U-Net donde se logró una precisión general mayor al 95%. (Wagner et al., 2019).

- A partir de los resultados obtenidos donde se aceptó la hipótesis alternativa que establece que hay diferencias en el mapeo de glaciares con los métodos NDSI y U-Net demostrando U-Net superioridad al NDSI, guarda una relación con lo que sostiene. (Härer

et al., 2018), que menciona que el umbral estándar 0.4 es cuestionable escala local existe incertidumbres la investigación propone un modelo polinomial cuadrático calibrado. Otra investigación respecto al NDSI menciona que es sensible a cuerpos de agua y sombras. (Sibandze et al., 2014). Respecto al modelo U-Net en el artículo de (Mohajerani et al., 2019). Encontró que la red neuronal entrenada supera significativamente las técnicas de detección de bordes comunes y se puede utilizar para mapear continuamente los frentes de hielo de parto con una variedad de productos de datos.

El modelo entrenado en esta investigación se puede utilizar para cualquier glaciar, se experimentó con algunos glaciares muy representativos del Perú como se muestra en la figura N°34. Los resultados se pueden visualizar y descargar en formato .tif, en el siguiente link habilitado públicamente: https://percyelbis.github.io/glx_unet/

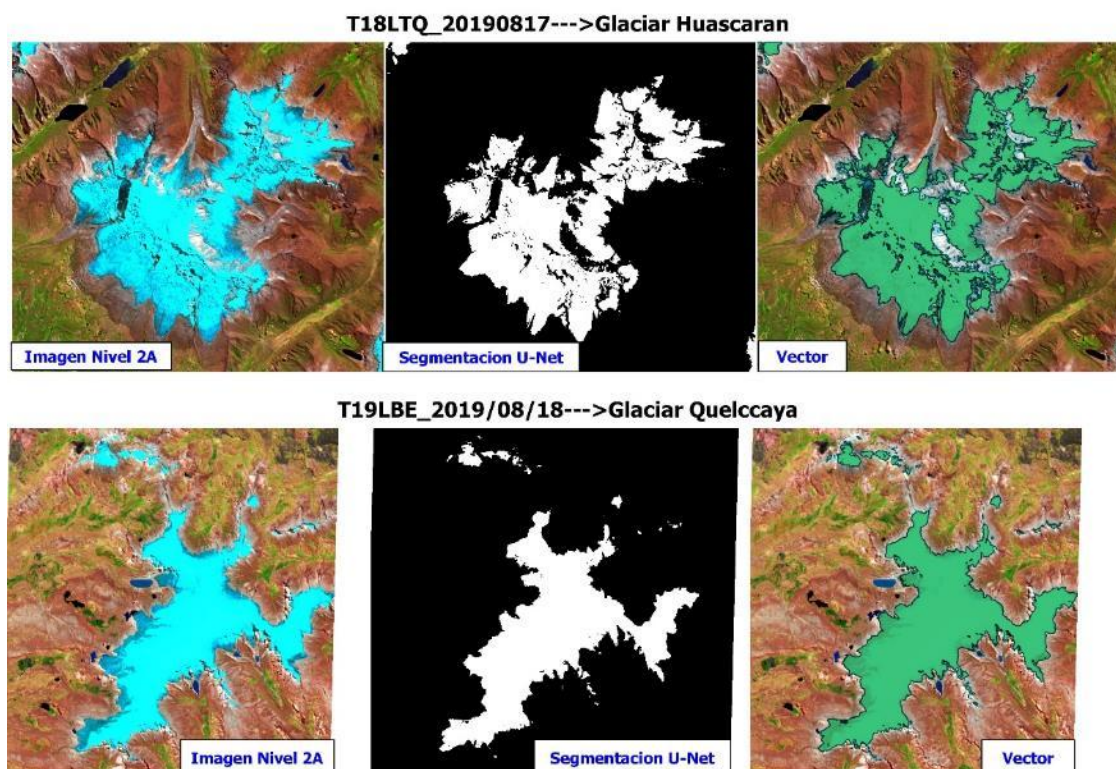


Figura N° 34 Prueba en algunos glaciares importantes del Perú.

Fuente: (Propio, 2019).

V. CONCLUSIONES

El modelo U-Net estimó área glaciar satisfactoriamente, obteniendo precisiones de 97.88%, es importante mencionar que la investigación ha demostrado que las redes neuronales convolucionales U-Net se pueden usar para cartografiado de superficie glaciar y demostró tener un buen rendimiento frente al método tradicional NDSI, el modelo es capaz de alcanzar una digitalización manual similar a un experto humano, podemos afirmar que el modelo puede ser utilizado para cualquier glaciar.

La recolección del conjunto de muestras de entrenamiento y validación se realizó con éxito en la plataforma Google Earth Engine y con sus respectivas etiquetas el cual ha tenido mucha importancia a hora de entrenar y validar el modelo.

El modelo se entrenó aproximadamente por ~11 horas, se validó con la métrica Coeficiente Dice con el fin de medir la calidad de la segmentación obteniendo una pérdida 0.0286 y las predicciones se realizaron con 12 escenas además se realizaron pruebas en diferentes glaciares prestigiosos del Perú en el Glaciar Quelccaya y el Glaciar Huascarán demostrando que el mapeo es muy bueno.

En los análisis de comparación con el método convencional NDSI, el modelo U-Net demostró ser más robusto obteniendo las siguientes áreas promedio de las 12 escenas de 27.557 y 21.774 km² respectivamente. En el método NDSI se observó que las elevaciones más bajas donde están las lagunas glaciares son las más vulnerables a que sea mapeado por este motivo el área se sobrestima.

VI. RECOMENDACIONES

Se recomienda experimentar otras arquitecturas como SegNet para segmentar coberturas más complejas como lagunas, pajonal, bofedales, morrenas y otros.

Para futuros trabajos en la comunidad glaciológica del Perú se recomienda desarrollar nuevos métodos de cartografiados para segmentar áreas glaciares en Imágenes Ópticas y utilizar Imágenes Radar para un monitoreo constante sin distorsiones atmosféricas.

Para un futuro trabajo de Investigación se recomienda tomar en cuenta la corrección Topográfica o por Iluminación, para mejorar la segmentación en los glaciares de valle donde las pendientes son pronunciadas.

Se recomienda utilizar la plataforma de Inteligencia Artificial de Google para proyectos grandes como inventariar glaciares y otras coberturas a nivel regional o nivel mundial.

VII. REFERENCIAS

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Recuperado de <http://arxiv.org/abs/1603.04467>
- Albert, T. H. (2002). Evaluation of Remote Sensing Techniques for Ice-Area Classification Applied to the Tropical Quelccaya Ice Cap, Peru. *Polar Geography*, 26(3), 210-226. <https://doi.org/10.1080/789610193>
- Almeida, T. de, & Meneses, P. R. (2012). *Introdução ao Processamento de Imagens de Sensoriamento Remoto*. Brasilia. <https://doi.org/10.1590/S1413-294X2002000200020>
- Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., ... Rabkin, A. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50. <https://doi.org/10.1145/1721654.1721672>
- Autoridad Nacional del Agua. (2014). *Inventario Nacional de Glaciares y Lagunas*. Huaraz. Recuperado de <http://www.ana.gob.pe/media/981508/glaciares.pdf>
- Avery Bick, I., Mullet, B., & Wang, D. (2018). Deep Learning and Deforestation.
- Berzal, F. (2018). *Redes Neuronales & Deep Learning*. Recuperado de <https://www.amazon.es/Redes-Neuronales-Deep-Learning-Edición/dp/1731314337>
- Bhardwaj, A., Di, W., & Wei, J. (2018). *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*.
- Burns, P., & Nolin, A. (2014). Using atmospherically-corrected Landsat imagery to measure glacier area change in the Cordillera Blanca, Peru from 1987 to 2010. *Remote Sensing of Environment*, 140, 165-178. <https://doi.org/10.1016/J.RSE.2013.08.026>
- Chollet, F. (2017). *Deep Learning with Python*. United States of America. Recuperado de <http://faculty.neu.edu.cn/yury/AAI/Textbook/Deep Learning with Python.pdf>

- Chuvieco, E. (1995). *Fundamentos de Teledetección Espacial*. España.
- Dozier, J. (1989). *Spectral Signature of Alpine Snow Cover from the Landsat Thematic Mapper* (Vol. 28). Recuperado de <https://pdfs.semanticscholar.org/564e/a7dae0b418df486d97f51be934660442c4e7.pdf>
- Florez, O. U. (2018). *Diccionario Inglés-Español de Términos Técnicos en Inteligencia Artificial (IA)*. San Francisco: GitHub Repository. Recuperado de https://github.com/capitalone/AI_Dictionary_English_Spanish
- Geewax, J. (2018). *Google Cloud PlatForm In Action*. Recuperado de www.manning.com/books/google-cloud-platform-in-action
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Recuperado de <https://www.deeplearningbook.org/>
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18-27. <https://doi.org/10.1016/J.RSE.2017.06.031>
- Guardamino Soto, L. Y. (2014). *Análisis de la evolución de las lagunas de alta montaña en la Cordillera del Vilcabamba (Cusco y Apurímac) entre los años 1991-2014 mediante métodos de sensoramiento remoto y SIG*. Pontificia Universidad Católica del Perú. Recuperado de <http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/5896>
- Härer, S., Bernhardt, M., Siebers, M., & Schulz, K. (2018). On the need for a time-and location-dependent estimation of the NDSI threshold value for reducing existing uncertainties in snow cover maps at different scales. *The Cryosphere*, 12, 1629-1642. <https://doi.org/10.5194/tc-12-1629-2018>
- Hernandez Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). *Metodología de la Investigación* (Sexta). Mexico.

- Huddart, D., & Stott, T. (2010). *Earth Environments: Past, Present and Future*. Wiley-Blackwell. Recuperado de http://125.234.102.150:8080/dspace/handle/DNULIB_52011/5024
- INAIGEM. (2017). Manual Metodológico de Inventario Nacional de Glaciares. *Instituto Nacional de Investigación en Glaciares y Ecosistemas de Montaña*. Recuperado de <http://repositorio.inaigem.gob.pe/handle/INAIGEM/173>
- Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, 132, 679-688. <https://doi.org/10.1016/J.PROCS.2018.05.069>
- Ji, S., Zhang, C., Xu, A., Shi, Y., & Duan, Y. (2018). 3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images. *Remote Sensing*, 10(2), 75. <https://doi.org/10.3390/rs10010075>
- Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Recuperado de <http://arxiv.org/abs/1412.6980>
- Kodym, O., Michalšpaněl, M. M., & Herout, A. (2018). *Segmentation of Head and Neck Organs at Risk Using CNN with Batch Dice Loss*. Recuperado de <https://arxiv.org/pdf/1812.02427.pdf>
- Kokaly, R. F., Clark, R. N., Swayze, G. A., Livo, K. E., Hoefen, T. M., Pearson, N. C., ... Klein, A. J. (2017). *USGS Spectral Library Version 7. Data Series*. <https://doi.org/10.3133/ds1035>
- Kulkarni, A. V, Bahuguna, I. M., Rathore, B. P., Singh, S. K., Randhawa, S. S., Sood, R. K., & Dhar, S. (2007). *Glacial retreat in Himalaya using Indian Remote Sensing satellite data*. *CURRENT SCIENCE* (Vol. 92). Recuperado de http://sa.indiaenvironmentportal.org.in/files/glaciers_and_climate.pdf
- Kumar, R. (2010). *Research Methodology: A Step-by-Step Guide for Beginners* (Tercera). SAGE Publications Ltd.
- Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in

- remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166-177.
<https://doi.org/10.1016/J.ISPRSJPRS.2019.04.015>
- Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sensing*, 10(7), 1119.
<https://doi.org/10.3390/rs10071119>
- Malo, J., Tuia, D., Gómez-Chova, L., Camps-Valls, G., & Jiménez, S. (2011). *Remote Sensing Image Processing. Synthesis Lectures on Image, Video, and Multimedia Processing* (Vol. 5). <https://doi.org/10.2200/s00392ed1v01y201107ivm012>
- Mathworks. (2018). Introducing Deep Learning with MATLAB, 15.
<https://doi.org/10.1109/ISBI.2018.8363547>
- Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. Recuperado de <http://arxiv.org/abs/1606.04797>
- Mohajerani, Y., Wood, M., Velicogna, I., Rignot, E., Mohajerani, Y., Wood, M., ... Rignot, E. (2019). Detection of Glacier Calving Margins with Convolutional Neural Networks: A Case Study. *Remote Sensing*, 11(1), 74.
<https://doi.org/10.3390/rs11010074>
- National Aeronautics and Space Administration, S. M. D. (2010). Introduction to the Electromagnetic Spectrum. Recuperado 21 de julio de 2019, de https://science.nasa.gov/ems/01_intro
- Ng, A. (2018). Machine Learning Yearning. Recuperado de <https://www.deeplearning.ai/machine-learning-yearning/>
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press. Recuperado de <http://neuralnetworksanddeeplearning.com/index.html>
- Olaya, V. (2011). *Sistemas de Informacion Geografica* (Version 1).

- Postadjian, T., Le Bris, A., Sahbi, H., & Mallet, C. (2017). Investigating The Potential Of Deep Neural Networks For Large-Scale Classification Of Very High Resolution Satellite Images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1*, 183-190. <https://doi.org/10.5194/isprs-annals-IV-1-W1-183-2017>
- Racoviteanu, A. E., Paul, F., Raup, B., Khalsa, S. J. S., & Armstrong, R. (2009). Challenges and recommendations in mapping of glacier parameters from space: results of the 2008 Global Land Ice Measurements from Space (GLIMS) workshop, Boulder, Colorado, USA. *Annals of Glaciology*, 50(53), 53-69. <https://doi.org/10.3189/172756410790595804>
- Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. En *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09* (pp. 1-8). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1553374.1553486>
- Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9), 2352-2449. https://doi.org/10.1162/neco_a_00990
- Rius, R., & González, J. A. (2011). Free software: An opportunity for researchers. *Medicina Clinica*, 136(5), 205-206. <https://doi.org/10.1016/j.medcli.2010.05.011>
- Rivera Mantilla, H. (2005). *Geologia General*. Lima.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Recuperado de <http://arxiv.org/abs/1505.04597>
- Santaella, L. (2012). A tecnocultura atual e suas tendencias futuras. *Signo y Pensamiento*, 31, 30-43.
- Sibandze, P., Mhangara, P., Odindi, J., & Kganyago, M. (2014). A comparison of Normalised Difference Snow Index (NDSI) and Normalised Difference Principal Component Snow Index (NDPCSI) techniques in distinguishing snow from related land cover types. *South African Journal of Geomatics*, 3(2), 197.

<https://doi.org/10.4314/sajg.v3i2.6>

- Stoian, A., Poulain, V., Inglada, J., Poughon, V., & Derksen, D. (2019). Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems. *Remote Sensing*, 11(17), 1986. <https://doi.org/10.3390/rs11171986>
- Tempfli, K., Kerle, N., Huurneman, G. C., Janssen, L. L. F., Bakker, W. H., Van Der Meer, F. D., ... Woldai, T. (2009). *Principles of Remote Sensing*.
- Tian, B. (2017). *GIS technology applications in environmental and earth sciences*. Taylor & Francis;CRC Press.
- Turpo cayo, E. Y. (2014). *Determinación del retroceso Glaciar de la Cordillera de la Raya mediante Imágenes Satelitales landsat ,1975-2013*. Universidad Nacional del Altiplano.
- Turpo Cayo, E. Y., & Maria Chaves, J. (2017). *Google Earth Engine como herramienta de detección de cambios espaciales e multitemporales: Caso glaciar Quelccaya, Perú. Anais do XVIII Simpósio Brasileiro de Sensoriamento Remoto -SBSR*. Feira de Santana. <https://doi.org/10.1017/CBO9781107415324.004>
- Veettil, B. K., Pereira, S. R., Wang, S., Valente, P. T., Grondona, A. E. B., Rondón, A. C. B., ... Simões, J. C. (2016). Un análisis comparativo del retroceso glaciar en los Andes Tropicales usando teledetección. *Investigación Geográfica*, 36, 3-36.
- Wagner, F. H., Sanchez, A., Tarabalka, Y., Lotte, R. G., Ferreira, M. P., Aidar, M. P. M., ... Arag~ Ao, E. O. C. (2019). Using the U-net convolutional network to map forest types and disturbance in the Atlantic rainforest with very high resolution images. <https://doi.org/10.1002/rse2.111>
- Wolfgang, E. (2017). *Introduction to Artificial Intelligence*.
- Yao, X., Liu, S., Han, L., Sun, M., & Zhao, L. (2018). Definition and classification system of glacial lake for inventory and hazards study. *Journal of Geographical Sciences*, 28(2), 193-205. <https://doi.org/10.1007/s11442-018-1467-z>

Zhang, L., Zhang, L., & Du, B. (2016). Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 22-40. <https://doi.org/10.1109/MGRS.2016.2540798>

Web Grafía

- Aybar Camacho, C., & Yali Samaniego, R. (2019). Google Earth Engine with Python. Recuperado 26 de julio de 2019, de <https://github.com/csaybar/EEwPython>
- Brown, C., & Clinton, N. (2018). EEUS 2018- Introduction to TensorFlow models. Recuperado 28 de julio de 2019, de <https://www.youtube.com/watch?v=w-1xfF0IaeU>
- European Space Agency. (2014). El programa Copérnico. Recuperado 22 de julio de 2019, de https://www.esa.int/esl/ESA_in_your_country/Spain/El_programa_Copernico
- European Space Agency. (2015a). Sentinel-2. Recuperado 22 de julio de 2019, de <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>
- European Space Agency. (2015b). *SENTINEL-2 User Handbook*. Recuperado de https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook
- European Space Agency, E. (2013). La dinámica de los glaciares. Recuperado 23 de julio de 2019, de https://www.esa.int/SPECIALS/Eduspace_Global_ES/SEM4OL22ECH_0.html
- Google. (2015). Platform Google Earth Engine. Recuperado 21 de julio de 2019, de <https://earthengine.google.com/platform/>
- Google. (2019). EarthEngine-API. Recuperado 26 de julio de 2019, de <https://github.com/google/earthengine-api/>
- Google Cloud. (2014). Documentación de Cloud Storage. Recuperado 22 de julio de 2019, de <https://cloud.google.com/storage/docs/>
- Google Developers. (2015). Google Earth Engine API. Recuperado 22 de julio de 2019, de <https://developers.google.com/earth-engine/>

ANEXOS

ANEXO A

Figura N° 35 Recolección de Puntos GPS para Validación.



ANEXO B

Script JavaScript utilizado para el target de nieve en las eco regiones Puna y Selva Alta.
El código completo está en el siguiente link habilitado públicamente.

<https://code.earthengine.google.com/fb70aa2d3f65e83b88c456327e71ade3>

La grafica completa para una mejor visualización está en el siguiente link habilitado públicamente.

https://drive.google.com/open?id=10_cMA9KHgkXnS1hUuxem4IVV_vDRLZ1a

Código de Programación completo del Modelo U-Net está en el siguiente link habilitado públicamente. https://github.com/percyelbis/glx_U-Net