SNOW COVER DETECTION OVER FORESTED AND MOUNTAINOUS REGIONS FROM REMOTE SENSING IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SADETTİN ÖZEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

AUGUST 2024

Approval of the thesis:

**SNOW COVER DETECTION
OVER FORESTED AND MOUNTAINOUS REGIONS
FROM REMOTE SENSING IMAGERY
USING CONVOLUTIONAL NEURAL NETWORKS**

submitted by **SADETTİN ÖZEN** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Geodetic and Geographic Information Technologies, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, **Graduate School of Natural and Applied Sciences** _____

Prof. Dr. Sevda Zuhal Akyürek
Head of the Department, **Geodetic and Geographic Information Technologies** _____

Prof. Dr. Sevda Zuhal Akyürek
Supervisor, **Geodetic and Geographic Information Technologies, METU** _____

Assoc. Prof. Dr. Semih Kuter
Co-Supervisor, **Forest Engineering, Çankırı Karatekin University** _____

Examining Committee Members:

Prof. Dr. Mehmet Lütfi Süzen
Geological Engineering, METU _____

Prof. Dr. Sevda Zuhal Akyürek
Civil Engineering, METU _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Assoc. Prof. Dr. Semih Kuter
Forest Engineering., Çankırı Karatekin University _____

Assist. Prof. Dr. Murat Durmaz
Geomatics Engineering, Hacettepe University _____

Date: 28.08.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Sadettin Özen

Signature :

ABSTRACT

SNOW COVER DETECTION OVER FORESTED AND MOUNTAINOUS REGIONS FROM REMOTE SENSING IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

Özen, Sadettin
Master of Science, Geodetic and Geographic Information Technologies
Supervisor: Prof. Dr. Sevda Zuhal Akyürek
Co-Supervisor: Assist. Prof. Dr. Semih Kuter

August 2024, 91 pages

Studies on hydrology, ecology, environment, and climate greatly benefit from the monitoring of snow cover in mountainous and forested areas. In such varied terrains, traditional snow cover detection approaches frequently struggle with coverage and precision. On the other hand, deep learning methods have revolutionized remote sensing and provide a viable method for accurate and effective snow cover identification. This work offers a comprehensive exploration of using deep learning techniques for this purpose, in particular, using Sentinel-2 multispectral imagery. Utilizing cutting-edge deep learning methods, our investigation shows significant gains in performance. As an example, our model performs well on a test set with temporal variability, yielding a dice score of 0.805. Moreover, the model obtains a dice score of 0.928 on another test set, demonstrating its capacity to precisely define the extent of snow cover. This study also explores the transfer learning strategies, which minimize the reliance on labeled data by fine-tuning pre-trained models on large datasets. Overall, using deep learning techniques and customized band

combinations of Sentinel-2 data, this research represents a major increase in remote sensing capabilities for snow cover identification in complex and heterogeneous forested and mountainous environments.

Keywords: Sentinel-2, Remote Sensing, Classification, Deep Learning, Snow

ÖZ

UZAKTAN ALGILAMA GÖRÜNTÜLERİ KULLANARAK ORMANLIK VE DAĞLIK BÖLGELERDE KAR ÖRTÜSÜ TESPİTİ İÇİN EVRİŞİMLİ SİNİR AĞLARI KULLANIMI

Özen, Sadettin
Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri
Tez Yöneticisi: Prof. Dr. Sevda Zuhal Akyürek
Ortak Tez Yöneticisi: Doç. Dr. Semih Kuter

Ağustos 2024, 91 sayfa

Hidroloji, ekoloji, çevre ve iklim ile ilgili çalışmalarda dağlık ve ormanlık alanlardaki kar örtüsünün izlenmesinden büyük fayda sağlanmaktadır. Bu kadar çeşitli arazi yapılarında, geleneksel kar örtüsü tespiti yaklaşımları sıklıkla kapsam ve kesinlik bakımından sıkıntılar doğurmaktadır. Diğer yandan, derin öğrenme yöntemleri uzaktan algılamada devrim yaratmış ve kar örtüsünün doğru ve etkili bir şekilde belirlenmesi için geçerli bir yöntem sağlamıştır. Bu çalışma, özellikle Sentinel-2 multispektral görüntülerini kullanarak, derin öğrenme tekniklerinin kullanılmasına ilişkin kapsamlı bir araştırma sunmaktadır. En son derin öğrenme yöntemlerini kullanan araştırmamız, performansta önemli kazanımlar olduğunu göstermektedir. Örnek olarak modelimiz zamansal değişkenliğe sahip bir test setinde 0,805'lik bir dice puanı vermektedir. Üstelik model, başka bir test setinde 0,928'lik bir dice puanı elde ederek, kar örtüsünün kapsamını tam olarak tanımlama kapasitesini ortaya koymaktadır. Ayrıca, büyük veri kümelerinde önceden eğitilmiş modellere ince ayar yaparak etiketli verilere olan bağımlılığı en aza indiren transfer öğrenme stratejileri de denenmiştir. Genel olarak, Sentinel-2 verilerinde derin

öğrenme teknikleri ve özelleştirilmiş bant kombinasyonları kullanan bu araştırma, karmaşık ve heterojen ormanlık ve dağlık alanlarda kar örtüsünün belirlenmesine yönelik uzaktan algılama yeteneklerinde büyük bir artışı temsil etmektedir.

Anahtar Kelimeler: Sentinel-2, Uzaktan Algılama, Sınıflandırma, Derin Öğrenme, Kar

To my family

Ayşe & Mehmet Özen

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Prof. Dr. Sevda Zuhal Akyürek and co-supervisor Assoc. Prof. Dr. Semih Kuter for their guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank the examining committee, Prof. Dr. Lütfi Süzen, Prof. Dr. Pınar Karagöz and Assist. Prof. Dr. Murat Durmaz for their suggestions and comments.

I am thankful to my dear mother Ayşe Özen and my dear father Mehmet Özen for their support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
ASPP	Atrous Spatial Pyramid Pooling
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DL	Deep Learning
DPN	Dual Path Network
FPN	Feature Pyramid Networks
ML	Machine Learning
MODIS	Moderate Resolution Imaging Spectroradiometer
NDSI	Normalized Difference Snow Index
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
PAN	Path Aggregation Network
PSPNet	Pyramid Scene Parsing Network
SCE	Snow Cover Extent
SMAP	Soil Moisture Active Passive
SMOS	Soil Moisture and Ocean Salinity
SNOMAP	Snow Mapping
SWE	Snow Water Equivalent
ViT	Vision Transformer

CHAPTER 1

INTRODUCTION

Comprehending the dynamics of snow cover in complex terrains is crucial for evaluating the water availability and understanding how ecosystems are impacted by climate change (Akyürek and Şorman, 2002). The hydrological cycle depends critically on snow cover, especially in areas with high latitudes and altitudes. Precise snow cover detection and monitoring are essential for a number of purposes, such as water resource management, flood forecasting, agriculture, and ecological research.

Snow information can be extracted from in-situ point measurements or air-borne/space-borne remote sensing observations. Most in-situ snow measurements are still performed using traditional laborious standardized techniques: digging snow pits and measuring manually the density, temperature, hardness, and other quantities. While these techniques are very robust and straightforward, they are very expensive for larger areas or time spans, prone to human errors and biases, and do not provide all requested quantities or provide only qualitative information of snow parameters. Threshold-based algorithms applied to remote sensing data are also commonly used in retrieving snow information. But these approaches are not without difficulties, particularly in densely forested areas and mountainous terrains (Dozier, 1989; Hall et al., 1995). The drawbacks of conventional methods originate from their dependence on human interpretation, which can lead to subjectivity and mistakes, especially in places with complex topography and vegetation.

High-altitude and high-latitude regions' terrestrial ecosystems depend heavily on the seasonal snowfall to maintain their water balance (Hao et al., 2022; Rittger et al., 2020). Snow pack acts as a natural reservoir, keeping rivers, lakes, and groundwater supplies alive by retaining water during the winter and releasing it gradually during

the melting season. Snow cover affects local climate, biodiversity, and human activity in addition to hydrological processes.

When the sky is clear, optical remote sensing techniques are frequently utilized to identify snow cover, especially over flat areas without dense vegetation (Hall et al., 1995). To distinguish snow from other forms of land cover, these methods take advantage of the special spectral characteristics of snow through the interaction of electromagnetic radiation with the snow surface. However, determining the spatiotemporal distribution of snow becomes more difficult in mountainous areas with rough topography and varied weather conditions (Hao et al., 2022). Furthermore, extensive forest canopy blocks optical sensors' view of ground snow, making snow cover monitoring even more difficult (Hall et al., 1995; Klein et al., 1998).

Boreal forests span approximately 19% of the seasonally snow-covered regions in the Northern Hemisphere, highlighting the critical need for accurate snow cover monitoring in these mountainous and forested areas (Rutter et al., 2009). Precise snow cover detection techniques specific to forested landscapes are necessary because of the significant differences in snow accumulation, timing, and rate of melting between forested and non-forested locations. Moreover, snow cover dynamics in forested areas directly influence streamflow patterns, soil moisture content, and water availability, leading to significant hydrological implications.

The use of satellite imagery as the main source of data for monitoring the extent of snow cover has completely changed the field (Drusch et al., 2012). Researchers can monitor snow cover changes at regional and global scales using multispectral images from satellites like Sentinel-2, which offer worldwide coverage and high spatial resolution. The combination of satellite imagery and machine learning (ML) algorithms has provided new alternatives for snow cover detection and monitoring.

The accuracy and efficiency of snow cover identification from satellite data have improved recently thanks to ML algorithms, especially deep learning (DL) techniques (LeCun et al., 2015). With its ability to learn intricate patterns and

characteristics from sizable datasets, Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have become powerful tools for image analysis (Krizhevsky et al., 2012; Dosovitskiy et al., 2020). These DL algorithms have proven to be quite effective in detecting snow cover, even over rough terrain with different environmental factors.

This work aims to advance previous studies by thoroughly evaluating the use of DL approaches for snow cover detection in mountainous and forested areas using Sentinel-2 multispectral images. By analyzing and comparing various DL architectures and methodologies, this study seeks to enhance our understanding of the benefits and limitations of different algorithms for snow cover identification. Additionally, it explores the scalability and generalizability of DL models across diverse geographical locations and climatic conditions, aiming to improve the accuracy and reliability of snow cover monitoring systems.

The remainder of the thesis has the following structure: Chapter 2 reviews recent and relevant literature. Chapter 3 provides a detailed list of the supplies and techniques used in the thesis, describing the selected research fields, the input data and their combinations, the methods used to gather the input data, and the outcomes. Chapter 4 presents and discusses the results. Finally, Chapter 5 concludes the thesis by summarizing the work and suggesting potential directions for future research.

CHAPTER 2

LITERATURE REVIEW

Accurate and timely snow cover mapping, particularly in complex terrains such as forests and mountainous regions, is crucial for hydrological modeling, disaster management, and climate change studies. Traditional methods for snow cover detection often face challenges in these environments due to factors like shadowing, vegetation cover, and terrain complexity. Recent advancements in remote sensing and DL have offered promising opportunities to address these limitations. This literature review discusses existing research on snow cover detection using CNNs, with a particular focus on forested and mountainous areas. By examining previous studies, this review aims to identify the state-of-the-art techniques, challenges, and potential avenues for improvement in this field.

2.1 Importance of Snow Detection in Forested and Mountainous Areas

Snow cover plays a crucial role in Earth's water cycles and global climate, particularly within the cryosphere. Approximately 31% of the Earth's total land area and 40% of its land area in the Northern Hemisphere are covered by seasonal snow annually (Breen et al., 2023; Hall et al., 1995). High intra- and inter-annual changes in the spatial extent of snow cover are known to occur. Recent research indicates a global decline in snow cover, largely attributed to climate change (Pulliainen et al., 2020; Y. Wang et al., 2018). However, regional variations in these snow cover changes are significant (Brown and Mote, 2009).

Understanding the Earth's energy balance and the function that snow plays in the climate system depends heavily on the monitoring and mapping of snow cover on a global scale. Strong albedo of snow allows it to reflect a significant amount of solar

radiation back into space (Kuter et al., 2018; Tekeli et al., 2005). This feature is essential for controlling the Earth's surface's patterns of heating and cooling. Indeed, snow cover has the biggest influence on the surface energy balance of all the land surface categories (Chen et al., 2021). Changes in snow cover can have a substantial impact on the Earth's surface energy balance, which can lead to modifications in the climate and associated feedback processes. Thus, changes in the amount and duration of snow cover have a significant impact on the energy balance of the Earth and the global climate (Takala et al., 2011). Our knowledge of these processes and how they affect the Earth's climate system can be improved by precisely monitoring and mapping the amount of snow cover.

It is important to detect snow in mountainous and forested locations for several reasons. Snowmelt plays a major role in river flows in these areas, which affects downstream hydroelectric power generation, agriculture, and water supplies. These locations are important water reservoirs. For studying climate change, conducting ecological research, and managing natural disasters like avalanches, precise data on snow cover is essential. Studies like those by Mote et al. (2018) emphasizes the critical role of snowpack in water supply and highlight the severe implications of their decline. Understanding snowfall dynamics in these areas aids in predicting water availability and optimizing resource management.

Moreover, the critical role of snow cover in alpine hydrology marks the necessity for accurate snow monitoring, especially in the context of climate change impacts. Hammond et al. (2018) contributes significant insights by analyzing snow zone characteristics and persistence patterns over a 15-year span. This data is crucial for agricultural planning, water resource management, and assessing potential impacts of global warming on snowpack dynamics and water availability.

2.2 Traditional Methods for Snow Detection

There are two types of snow cover parameters which are snow cover extent (SCE) and snow water equivalent (SWE). SCE provides information about the spatial distribution of snow, while SWE quantifies the amount of water stored within the snowpack (Pulliainen and Hallikainen, 2001).

There are two primary methods to map SCE: binary and fractional snow cover mapping. Binary mapping simply categorizes each image pixel as either snow-covered or snow-free. While easier to process, it lacks accuracy in areas with partial snow cover. Fractional snow cover mapping, on the other hand, estimates the proportion of snow within a pixel, offering a more precise and detailed representation of snow distribution (Metsämäki et al., 2015).

Passive microwave sensors, such as those on satellites like soil moisture active passive (SMAP) and soil moisture and ocean salinity (SMOS), measure microwave energy emitted from the Earth's surface, including snow. By analyzing changes in brightness temperature as snow depth and water content vary, SWE can be estimated. Active microwave sensors, like radar systems, transmit microwave pulses and measure the backscattered signal. This allows for more precise measurements of snow depth, which, combined with other information like snow density, can contribute to SWE estimation. While passive microwave excels in large-scale monitoring, active microwave offers advantages in areas with complex terrain or dense vegetation.

Review of Dietz et al. (2012) focuses primarily on passive remote sensing methods for snow monitoring. Optical and microwave sensors are discussed in detail, with their effectiveness varying based on factors like cloud cover, terrain, and snow conditions. The authors emphasize the importance of these methods for large-scale snow cover mapping but also acknowledge their limitations in terms of accuracy and data accessibility.

Review of Dozier and Painter (2004) discusses the application of multispectral and hyperspectral remote sensing for characterizing alpine snow properties. They emphasize the potential of these techniques to estimate crucial parameters such as snow cover extent, albedo, grain size, liquid water content, and temperature. While these methods offer valuable insights into snowpack dynamics, the authors also highlight challenges like atmospheric interference, topographic effects, and the complexity of snow spectral signatures. Despite these limitations, the study underscores the importance of remote sensing for understanding snowpack evolution in mountainous regions.

Review of Frei et al. (2012) provides a comprehensive assessment of global satellite-derived snow products. The authors evaluate the strengths and limitations of various satellite sensors and retrieval algorithms for estimating snow cover extent and snow water equivalent. While passive microwave sensors offer advantages in cloud-cover conditions, their accuracy can be compromised by complex terrain and vegetation. Optical sensors, on the other hand, provide high spatial resolution but are susceptible to cloud interference. The review highlights the importance of these satellite products for hydrological modeling, climate studies, and water resource management. However, it also emphasizes the need for continued research to improve retrieval accuracy and consistency across different regions and snow conditions.

The study of Hall et al. (1995) pioneered the use of Moderate Resolution Imaging Spectroradiometer (MODIS) data for global snow cover mapping. This groundbreaking research laid the foundation for developing algorithms capable of identifying snow-covered areas at a 500-meter spatial resolution. By utilizing spectral characteristics and decision rules, the authors introduced the snow mapping (SNOMAP) algorithm, demonstrating promising results in accurately detecting snow cover. While the study presented a significant advancement in snow remote sensing, it also highlighted challenges associated with cloud cover, forest canopy, and the complexity of snow spectral signatures, which can impact the accuracy of snow cover mapping.

Rosenthal and Dozier (1996) initiated the use of Landsat Thematic Mapper data to accurately map snow cover in mountainous regions at a subpixel level. This innovative approach addressed the limitations of traditional pixel-based methods, allowing for more precise snow cover estimations in complex terrain. By employing advanced image processing techniques, the authors developed a method to determine the proportion of snow within each pixel, providing valuable information for hydrological modeling and water resource management. While this study demonstrated the potential of Landsat data for detailed snow mapping, challenges related to cloud cover, topographic effects, and variations in snow spectral properties still limited the accuracy of the derived snow cover estimates.

Klein et al. (1998) addressed a critical challenge in snow cover mapping: the impact of forest canopies. Recognizing the limitations of traditional methods in forested areas, the authors developed a canopy reflectance model to improve snow cover detection accuracy. By incorporating the effects of forest structure and snow properties, their model enhanced the interpretation of satellite data. This research demonstrated the significance of accounting for forest influences when mapping snow cover, especially in regions with extensive forest cover. While the study provided valuable insights, further advancements in canopy modeling and the availability of higher resolution data are necessary to fully address the complexities of snow cover mapping in forested environments.

Salomonson and Appel (2004) introduced a novel approach to estimate fractional snow cover using MODIS data. They developed the Normalized Difference Snow Index (NDSI) to quantify the proportion of snow within a pixel. By correlating NDSI values with ground-based snow observations, the authors demonstrated the potential of this method to improve upon traditional binary snow cover maps. While the NDSI showed promising results in estimating fractional snow cover, the accuracy was influenced by factors such as cloud cover, terrain, and land cover. This research contributed significantly to the advancement of snow remote sensing by providing a more nuanced representation of snow cover distribution.

Traditional snow detection techniques often involve the use of optical remote sensing and snow indices such as the NDSI. Salminen et al. (2009) discusses the challenges of using reflectance-based methods where there are snow-free surfaces and forest canopy. Lv and Pomeroy (2019) points out that due to canopy interception, it can be challenging to distinguish between areas with and without snow. These methods require careful calibration and they are often limited by cloud cover and varying illumination conditions.

Because of the complex interactions that exist between snow and the forest canopy, traditional methods of detecting snow typically prove inadequate in forested areas. With a technique designed for dense coniferous forests, Wang et al. (2015) showed an increased accuracy in snow cover mapping. A different study, Ma and Zhang (2022), suggests a dynamic thresholding strategy to improve the extraction of snow cover in mountainous regions.

Additionally, Hou et al. (2020) investigates the application of artificial neural networks to accessible image pairs to enhance the precision of MODIS fractional snow cover mapping. This demonstrates the drawbacks of conventional techniques and the promise of cutting-edge algorithms for improving the detection of snow cover.

Other significant contributions include Wang et al. (2020), which presents an adaptive algorithm to improve snow identification accuracy in forested regions, and X. Wang et al. (2018), which proposes a multi-index technique for more accurate snow cover mapping in complex terrains.

2.3 Snow Cover Mapping with Machine and Deep Learning

Recent developments in DL and ML algorithms have greatly enhanced snow cover mapping in mountainous and forested areas. Haseeb Azizi et al. (2024) leverages the power of ML to estimate fractional snow cover in the complex terrain of the Hindu Kush Mountains. By combining MODIS and Landsat data, they developed a robust

model that effectively addresses the challenges posed by varying snow conditions, topography, and land cover. Their approach demonstrates the potential of ML to improve the accuracy of snow cover mapping compared to traditional methods. The study contributes significantly to understanding snow dynamics in this critical region, providing valuable information for water resource management, climate change studies, and disaster risk assessment. However, further validation and application of the model to other mountainous regions are necessary to assess its generalizability.

DL algorithms can be used to collect comprehensive snow cover data with more accuracy and complexity. Wang et al. (2024) study introduce a novel approach to estimate fractional snow cover using a Deep Feature Snow Index. This method leverages the power of DL to extract informative features from remote sensing data, significantly enhancing the accuracy of snow cover estimation compared to traditional methods. By capturing complex relationships between spectral information and snow cover conditions, the proposed index provides a more robust and reliable assessment of fractional snow cover. While the study demonstrates promising results, further research is needed to evaluate its performance in diverse geographical and climatic conditions.

Ma et al. (2023) provide a novel DL architecture, UCTNet, for accurate snow coverage mapping using Sentinel-2 satellite imagery. This approach effectively addresses the challenging task of differentiating snow from clouds, a common issue in remote sensing.

Further, Zheng et al. (2021) present an alternative method for distinguishing between clouds and snow, improving the reliability of snow cover maps. This encoder-decoder architecture enhances segmentation accuracy by effectively learning and differentiating cloud and snow features.

Applications for snow cover mapping that incorporate ML and DL are numerous. Wang et al. (2022) process Sentinel-2 data using ML algorithms, which results in improved snow cover identification accuracy over traditional techniques. Yin et al.

(2022) have refined snow cover maps even further by enhancing cloud and snow recognition in remote sensing imagery by an enhanced Unet3+ model.

The creation of ML-powered automatic snow mappers is another noteworthy achievement. Snow detection efficiency and accuracy can be improved by automation, as Wang et al. (2021) show. In order to process big datasets fast and reliably, this method minimizes manual interaction and makes use of ML techniques.

Furthermore, Z. Wang et al. (2022) greatly improve snow cover detection by combining sophisticated DL models for cloud and snow identification in remotely-sensed images. Similar to this, Zhan et al. (2017) enhance the ability to distinguish between snow and clouds in satellite imagery by using a deep convolutional network.

All things considered, the use of ML and DL techniques in snow cover mapping is a major improvement over traditional approaches. These technologies provide more reliable and precise snow detection capabilities, overcoming many of the drawbacks of traditional methods and offering insightful information for studies related to climate change, ecology, and water resources management.

2.4 Background on Segmentation Networks

Semantic segmentation networks are a class of neural network architectures engineered to classify images at the pixel level, aiming to assign each pixel to a particular semantic class or category. In this sub-section, the segmentation networks used in this study are briefly explained.

2.4.1 U-Net

CNNs like U-Net are designed specifically for biomedical image segmentation applications. U-Net, which was created by Ronneberger et al. in 2015, is well known for its ability to segment pictures even with a small amount of training data. The

network architecture allows for accurate pixel-wise segmentation by having an expanding path after a contracting path.

A sequence of convolutional and max-pooling layers are used by the network in the contracting route to extract contextual information and shrink the input image's spatial dimensions. This approach preserves computational efficiency while making high-level feature extraction easier. Furthermore, to maintain fine-grained information during up sampling, skip connections are added between matching layers in the contracting and expanding routes.

Up sample layers in the expanded route gradually raise the feature maps' spatial resolution. Convolutional and concatenation procedures are combined in each up sampling step to combine feature maps from the contracting route with those from the corresponding up sampling layer. Through the integration of high-resolution features from previous layers, lost spatial information from down sampling are partially recovered.

Additionally, U-Net has a symmetric design, meaning that in the contracting path, the number of feature channels decreases gradually, and in the expanding path, it increases proportionately. Precise segmentation is made possible by this symmetry, which permits smooth information transfer between the encoding and decoding phases. Furthermore, a pixel-wise sigmoid activation function is used in the network's last layer to create probability maps that show the chance of each pixel being a part of the segmented object.

Cell segmentation, organ delineation, and medical picture analysis are just a few of the biomedical image segmentation tasks in which U-Net has shown outstanding performance. In the field of biomedical imaging, its capacity to provide precise segmentation findings from a little amount of annotated data has made it a popular option. Moreover, U-Net's modular design facilitates simple modification and expansion, making it an adaptable framework for tackling a range of segmentation problems.

2.4.2 U-Net++

Zhou et al. (2020) proposed U-Net++ as an update of the original U-Net design with the goal of adding dense and layered skip routes to improve segmentation speed even more. U-Net++, which builds on the success of U-Net, has a unique design with many layered skip routes, which helps the model better collect multiscale contextual information.

Dense skip connections, as opposed to conventional skip connections, are used in U-Net++ to enable a more thorough integration of features at various sizes. Improved feature representation and accurate segmentation are the result of these extensive skip connections, which let information move not just across related layers but also across other levels of abstraction.

U-Net++'s layered skip paths are made to record hierarchical characteristics at various sizes. With layered skip connections across network layers, U-Net++ is able to make better use of contextual information at both the local and global levels, which improves the model's capacity to segment objects with varied sizes and forms.

Moreover, attention gates—a unique attention mechanism introduced by U-Net++—adaptively modify the information flow inside the network. By selectively highlighting valuable traits and suppressing irrelevant ones, these attention gates help the model concentrate on prominent regions and enhance segmentation performance.

Overall, U-Net++'s adaptable architecture and modular design make it a valuable tool for a variety of segmentation tasks.

2.4.3 DeepLabV3

DeepLabV3 is a state-of-the-art CNN architecture developed by Google researchers for semantic image segmentation tasks. Introduced by Chen et al. in 2017, DeepLabV3 builds upon its predecessors, incorporating several key innovations to

improve segmentation accuracy, especially for objects with fine details and intricate boundaries.

The use of atrous convolution, sometimes referred to as dilated convolution, across the network is one of DeepLabV3's standout characteristics. The network can effectively capture multi-scale contextual information thanks to atrous convolution, all without requiring more parameters or compromising computational speed. The receptive range of convolutional filters may be effectively expanded by DeepLabV3 utilizing varying dilation rates, allowing the model to include contextual information from a wider spatial environment.

Moreover, DeepLabV3 includes a potent version of atrous spatial pyramid pooling (ASPP) that makes use of several atrous convolutional layers operating in parallel at varying dilation rates. The ASPP module improves the network's capacity to separate objects of different sizes and scales by allowing it to record multi-scale features at various resolutions. To further enhance segmentation performance, DeepLabV3 incorporates global average pooling to capture global context.

In addition, DeepLabV3 presents ASPP, an efficient method for capturing multi-scale features that makes use of several concurrent atrous convolutional layers with varying dilation rates. With the help of this module, the network can segment objects with greater accuracy, regardless of their size or scale.

Scene understanding, object detection, and image parsing are just a few of the semantic segmentation tasks in computer vision that DeepLabV3's ability to capture multi-scale contextual information and generate detailed segmentation masks makes it a versatile and powerful tool for.

2.4.4 DeepLabV3+

Chen et al. (2018) presented DeepLabV3+, an enhanced version of the DeepLabV3 architecture. Building on DeepLabV3's advantages, DeepLabV3+ adds more modules and optimizations to improve semantic picture segmentation performance,

especially when it comes to enhancing boundary delineation and capturing fine details.

The addition of a decoder module to DeepLabV3+ is one of its primary advances; it enhances the segmentation outcomes that the encoder produces. Several convolutional layers and up-sampled feature maps are used by this decoder module to recover spatial features and improve the segmentation boundaries. Through integrating the output of the decoder with high-resolution feature maps from the encoder, DeepLabV3+ can generate segmentation masks that are more accurate in terms of localization.

Feature pyramid pooling is a method that DeepLabV3+ introduces to enable smooth integration between the encoder and decoder modules. With the help of this approach, the decoder can access rich contextual information at several resolutions by aggregating multi-scale characteristics from distinct encoder levels. DeepLabV3+ can handle objects of different sizes and scales by combining features from numerous scales, so that segmentation performance is improved on a variety of datasets and circumstances.

Because of its design, DeepLabV3+ is a helpful tool for semantic image segmentation in a variety of applications, such as augmented reality, autonomous driving, and medical imaging. It can handle objects with varying scales and dimensions, manage intricate features, and enhance segmentation boundaries.

2.4.5 MA-Net

MA-Net, short for Multi-Attention Network, is a novel CNN architecture introduced for semantic image segmentation tasks. Developed by Fan et al. in 2020, MA-Net presents a unique approach by leveraging multiple attention mechanisms to capture both global and local contextual information effectively.

Integrating several attention modules, each intended to capture unique elements of the input image, is the fundamental component of MA-Net. These modules of

attention comprise boundary refinement, local contextual, and global contextual attention. By capturing global context information, the network can gain a more comprehensive picture of the entire scenario thanks to the global contextual attention mechanism. Local structures and tiny details within the image are the main targets of the local contextual attention mechanism, on the other hand. The accuracy with which the model can distinguish object borders is further improved by the boundary refinement attention module.

MA-Net has a hierarchical architecture with several tiers for aggregating and extracting features. To extract hierarchical features while selectively attending to pertinent spatial regions, the network combines convolutional and attentional operations at each level. Because of its hierarchical processing, MA-Net can effectively capture contextual information at several scales, which makes it easier to accurately classify objects at different resolutions and scales.

Moreover, MA-Net presents a novel fusion method that synergistically mixes the outputs of various attention modules. Global context and fine-grained details are combined in MA-Net's feature representations, which are created by combining the global, local, and boundary refinement attention maps. This fusion technique successfully integrates complementary input from several attention mechanisms, improving the segmentation performance of the model.

MA-Net demonstrates promise in semantic image segmentation tasks due to its hierarchical architecture and fusion strategy, which enable the aggregation of both local and global contextual information. Its modular design and attention-based architecture also facilitate further optimization and customization for specific segmentation challenges in real-world applications.

2.4.6 LinkNet

LinkNet is a CNN architecture designed for semantic image segmentation tasks, introduced by Chaurasia and Culurciello (2017). It presents an efficient and

straightforward architecture that achieves competitive segmentation performance while minimizing computational complexity.

Utilizing skip connections—a technique made popular by the U-Net architecture—is the fundamental idea behind LinkNet. In contrast to U-Net, LinkNet utilizes a symmetric encoder-decoder architecture that includes residual connections. The information can move across several levels of abstraction with the help of these residual connections, sometimes referred to as "links," which join comparable encoder and decoder blocks.

In LinkNet's encoder, hierarchical features are extracted while the input image's spatial resolution is gradually decreased through a succession of convolutional blocks and down sampling layers. A residual connection connects each encoder block to its matching decoder block, enabling the decoder to retrieve feature maps from previous network phases.

Up sampling layers are used to progressively recover the spatial resolution of the feature maps throughout the decoder stage. By combining high-resolution encoder characteristics with up-sampled features, the decoder blocks—which are made up of convolutional layers and residual links—allow the network to improve segmentation outcomes.

LinkNet's computational efficiency is one of its main benefits. Unlike previous architectures, LinkNet provides reliable segmentation results with fewer parameters by carefully choosing when to use skip connections and residual connections. LinkNet is an excellent choice for real-time applications and devices with limited resources because of its efficiency.

For semantic image segmentation tasks, LinkNet is a practical and effective solution because of its effectiveness, efficiency, and ease of use. This is especially true when real-time performance is critical or computational resources are limited.

2.4.7 Feature Pyramid Network

A CNN design called Feature Pyramid Networks (FPN) was put forth by Lin et al. (2016). For tasks, like object detection and semantic segmentation, FPN tackles the difficulty of identifying items at various scales in an image.

The key idea behind FPN is to build a feature pyramid from a single CNN. This pyramid consists of feature maps at multiple scales, each capturing different levels of semantic information. FPN achieves this by adding a top-down pathway to a backbone network.

Feature maps from higher pyramid levels are routinely sampled up and merged with feature maps from lower levels in FPN's top-down approach. A rich hierarchical representation with features at varying scales at each level is produced by this technique. Thus, more reliable and accurate object detection and segmentation are made possible by the feature pyramid that is produced, which maintains both high-level semantics and fine-grained details.

Enhancing object detection systems' performance is one of FPN's main advantages, especially when it comes to handling objects of various scales. Feature part numbering (FPN) facilitates the efficient localization and classification of objects of different sizes in an image by offering a multi-scale feature representation.

Furthermore, FPN has been widely adopted in other computer vision tasks beyond object detection, including semantic segmentation. By incorporating FPN into segmentation architectures, researchers have achieved state-of-the-art results by leveraging multi-scale features for more accurate pixel-level predictions.

Overall, FPN's approach to addressing scale variance in object detection and segmentation tasks represents a significant breakthrough in the field of computer vision.

2.4.8 Path Aggregation Network

Liu et al. (2018) presented the Path Aggregation Network (PAN), a CNN architecture intended to overcome the drawbacks of conventional feature pyramid techniques for semantic segmentation tasks. With the use of a unique path aggregation module introduced by PAN, features from several scales may be combined in a hierarchical fashion to enable more efficient integration of multi-scale data for precise segmentation.

Using both top-down and bottom-up paths to effectively capture multi-scale characteristics is the fundamental concept of PAN. While the bottom-up approach up samples feature maps to preserve fine-grained details, the top-down pathway uses down sampling to extract high-level semantic information from feature maps. Subsequently, PAN utilizes a path aggregation module to combine elements from both pathways, allowing the network to obtain contextual data at various sizes.

Adaptive convolutional kernels are employed in the path aggregation module to merge features from the top-down and bottom-up pathways. These kernels efficiently aggregate multi-scale information while enabling PAN to selectively focus on pertinent spatial regions by dynamically adjusting their receptive fields in response to the input feature maps. More accurate segmentation outcomes are produced by the network's increased capacity to handle objects of different sizes and scales thanks to this adaptive fusion technique.

Additionally, prior to aggregation, PAN offers a feature alignment module for aligning feature maps from various sizes. By guaranteeing that features with comparable semantic meanings are matched spatially, this alignment promotes more coherent feature fusion and enhances segmentation efficiency. To further improve the network's capacity to attend to pertinent spatial contexts, PAN also includes a spatial attention mechanism that highlights informative regions in the feature maps.

PAN demonstrates robust capabilities in semantic segmentation tasks, leveraging its capacity to capture multi-scale contextual information and efficiently integrate

features from diverse pathways. With a novel path aggregation module, PAN enhances the integration of multi-scale features, marking a significant advancement in semantic segmentation systems. Its hierarchical design, adaptive fusion processes, and attention mechanisms make PAN particularly effective for tasks such as scene parsing, object detection, and overall image understanding within computer vision.

2.4.9 Pyramid Scene Parsing Network

Pyramid Scene Parsing Network (PSPNet) is a CNN architecture proposed by Zhao et al. (2016), specifically designed for the task of semantic image segmentation. PSPNet introduces a novel pyramid pooling module that captures multi-scale contextual information effectively, enabling the network to make more informed pixel-wise predictions.

The pyramid pooling module, which combines characteristics from several pyramid levels to capture context at various scales, is the central component of PSPNet. To integrate the pooled features, convolutional layers are applied after pooling processes at different spatial resolutions. PSPNet's segmentation capabilities are improved by efficiently capturing both local details and global context through the pooling of features at different scales.

PSPNet employs a deep CNN backbone to extract hierarchical features from the input image. These features are then fed into the pyramid pooling module, which enriches them with multi-scale contextual information. The resulting feature maps are further processed through convolutional layers to produce segmentation masks at the output.

PSPNet is distinguished by its ability to capture context efficiently without significantly increasing computational complexity. By employing pyramid pooling, PSPNet achieves a balanced approach between computational efficiency and receptive field size, enabling it to handle objects of varying sizes and scales effectively. Furthermore, PSPNet enhances segmentation quality through a fully

connected conditional random field (CRF) for post-processing, which considers both pixel-level features and pairwise relationships between neighboring pixels. This refinement step ensures smoother and more coherent segmentation results. PSPNet thus stands out as a powerful tool for semantic image segmentation tasks in computer vision, leveraging its capacity for multi-scale contextual information and robust design principles.

2.5 Background on Encoder Structures

DL frameworks require encoder architectures to perform tasks like image recognition, natural language processing, and model generation. These architectures are critical because they make it possible to transform incoming data into a high-dimensional, succinct representation that captures significant patterns and characteristics. Encoder architectures help with more effective processing and information extraction by reducing the dimensionality of the input data. Moreover, the representations produced by encoder architectures frequently have beneficial characteristics like latent features or semantic meaning, which can be utilized in a variety of further tasks like generation, regression, or classification. Encoder architectures are fundamental components of DL models that enable effective learning and representation of complex input.

2.5.1 Mobile Vision Transformer

Mehta and Rastegari (2021, 2022) introduced Mobile Vision Transformer (MobileViT) to address the problem of creating precise and lightweight computer vision models for mobile devices. Conventional CNNs are effective for mobile applications, but they are not very good at capturing aspects of an image on a global scale. On the other hand, although they are computationally costly, ViTs are excellent in processing information globally.

A new layer called MobileViT fills the gap between CNNs and ViTs. This layer uses transformers to provide a global processing method in place of the convolutions' built-in local processing. Compared to CNN and ViT-based architectures, MobileViT can achieve competitive accuracy with a substantially smaller number of parameters thanks to this innovation. All things considered, MobileViT presents a viable method for creating strong yet lightweight vision models for mobile devices.

2.5.2 EfficientNet

Tan and Le (2019) revolutionized the CNN architecture by introducing the EfficientNet, which achieves state-of-the-art performance with noticeably fewer parameters and computational resources than conventional models. Its novel compound scaling technique—which consistently modifies the network's depth, width, and resolution to attain maximum performance under a range of resource constraints—is the cause of this accomplishment. EfficientNet maintains competitive accuracy even in the face of hardware restrictions by carefully balancing model size and performance.

The architecture of EfficientNet combines performance-preserving efficient construction elements, like squeeze-and-excitation modules and inverted residual blocks, to optimize computing efficiency. By utilizing methods, like depth-wise separable convolutions and channel-wise attention mechanisms, these blocks efficiently capture informative information with fewer parameters and computations. EfficientNet's ability to perform well in a range of computer vision tasks, such as object detection, semantic segmentation, and image classification, is a result of its comprehensive approach and sophisticated scaling strategy that strikes a balance between network depth, width, and resolution. This combination makes EfficientNet a highly adaptable and potent solution for real-world deployment on scarce devices and platforms. It offers different variants (B0 to B7) for different capacity requirements.

2.5.3 Multi-Axis Vision Transformer

Introduced by Tu et al. (2022), the Multi-Axis Vision Transformer (MaxViT) is a unique combination of CNNs and ViTs that are optimized for vision tasks. Because of its hybrid architecture, which smoothly combines traditional CNN layers with ViT-style self-attention blocks, MaxViT is able to effectively capture both local and global information. With its distinct adaptive max-pooling mechanism and progressive training approach, MaxViT maintains computational efficiency while achieving remarkable results on many vision tasks. Using the advantages of both methods for reliable representation learning and task-specific refinement, this strategy alternates between pre-training with ViT-like structures and fine-tuning with the integrated MaxViT model.

2.5.4 MobileNet

A breakthrough in CNN design, MobileNet was unveiled by Howard et al. (2017) and it is especially designed for use on mobile and embedded devices. Fundamentally, MobileNet uses depth-wise separable convolutions to strike a compromise between computational performance and model size. Because of this innovative architectural design, MobileNet performs competitively even in contexts with limited resources because it requires a substantially lower number of parameters and computations.

The main advantage of MobileNet for applications with low memory and computing power is its compact model size and great accuracy. MobileNet effectively captures channel-wise and geographical information using depth-wise separable convolutions, which lower computing costs and facilitate fast feature extraction. Furthermore, MobileNet's design allows customers to modify the model's size and complexity to meet individual deployment requirements.

2.5.5 Dual Path Networks

Y. Chen et al. (2017) introduced the Dual Path Network (DPN), a CNN design. Each layer in the architecture has a dual route design that combines more intricate transformations with conventional convolutional paths to improve feature representation learning. Utilizing two attentions and a variety of feature extraction pathways, DPN maintains computational efficiency while achieving cutting-edge results on several benchmark datasets, including ImageNet. The novel architecture of DPN renders it an effective instrument for problems related to semantic segmentation, object identification, and image classification.

2.5.6 Inception

With its effective design and exceptional performance in image recognition tasks, Inception, introduced by Szegedy et al. (2015), revolutionized the CNN architectures. Inception modules, which have concurrent convolutional procedures with different kernel sizes, are at the center of it. This design makes it possible for Inception to efficiently capture multi-scale information, which improves its capacity to learn detailed representations of incoming images. By optimizing feature extraction capabilities while decreasing parameters and computations, Inception sets a new benchmark in neural network design by finding a compromise between model complexity and computational efficiency. The primary breakthrough of Inception is its capacity to reduce dimensionality and make effective use of parallel convolutional pathways to maximize processing resources.

2.5.7 DenseNet

A ground-breaking CNN architecture known for its effective feature reuse and dense connection patterns, DenseNet was proposed by Huang et al. (2017). DenseNet uses dense blocks as its fundamental building block, in which each layer is connected to

all other layers in the block. Because of this extensive connectivity, learning efficiency and representation capacities are increased as well as feature reuse and gradient flow throughout the network are improved.

DenseNet's primary novelty is its dense connection structure, which allows for deep feature extraction while reducing problems associated with vanishing gradients. DenseNet promotes feature reuse across several network depths by densely linking layers within each block, which enables the model to successfully capture complex patterns and representations. DenseNet also presents the idea of growth rates, which control how many feature maps are produced by each layer inside the dense blocks to strike a compromise between computing efficiency and model complexity.

2.5.8 ResNet

He et al. 's (2016) proposed ResNet, short for Residual Network transformed CNN architectures by introducing the idea of residual learning. ResNet uses residual blocks at its core, where each block has shortcut connections that go around one or more layers. By reducing the vanishing gradient issue, this clever design allows the network to learn residual mappings, which facilitates the training of deeper models.

ResNet's primary innovation resides in its residual connections, which facilitate direct information flow via shortcut links. Training of incredibly deep networks with hundreds or thousands of layers is made easier by ResNet's ability to omit some levels and guarantee the smooth propagation of gradients during training. Enhancing the model's accuracy and convergence speed overall, this architectural innovation also makes it possible to train deeper networks.

CHAPTER 3

MATERIALS AND METHODS

This section outlines the procedures employed in this study, complemented by a detailed flowchart illustrated in Figure 3.1, which visually summarizes the methodology utilized for accurate detection of snow cover across diverse and challenging landscapes.

3.1 Satellite Imagery and Pre-processing

3.1.1 Sentinel-2 Imagery

In this research, 9 distinct Sentinel-2 images over 6 different tile locations were used. All the data were sourced from the Sentinel Hub EO Browser (https://www.sentinel-hub.com/explore/eobrowser/). Sentinel-2 tiles consist of 13 spectral bands, where B1 and B10 have a pixel size of 60 meters, B5, B6, B7, B8A, B11, and B12 have a pixel size of 20 meters, and B2, B3, B4, and B8 have a pixel size of 10 meters. B8A was excluded from our analysis due to redundancy with B8, which offers higher spatial resolution. Bands with resolutions other than 10 meters were resampled to match the 10-meter resolution using nearest neighbor resampling.

Table 3.1 provides comprehensive details on the Sentinel-2 data, while Figure 3.2 illustrates their respective locations. The selection of tile locations encompassed Alps, Tatra Mountains, Rocky Mountains, and Kackar Mountains for their forested and mountainous terrain. Additionally, the Tibetan Plateau and Saskatchewan were chosen for their abundance of water bodies.

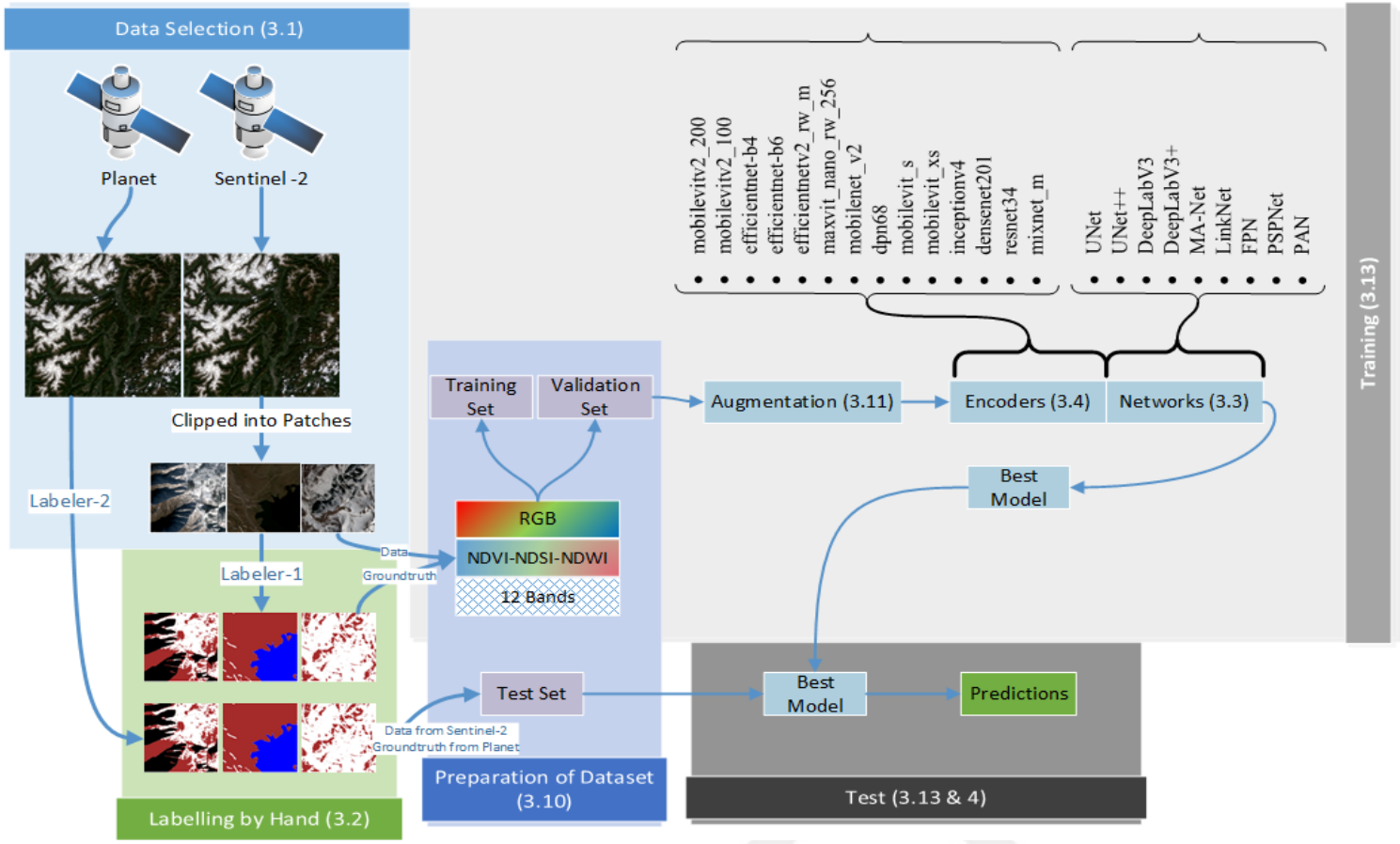


Figure 3.1 The workflow: data selection, preparation of dataset, model training and testing.

Table 3.1 The details of Sentinel-2 tiles used in the study.

Sentinel-2 Tile S/N	Location	Tile Number	Date
1	Tatra Mountains	T34UCV	29.11.2018
2	Alps	T32TPS	05.12.2018
3			13.06.2019
4	Kackar Mountains	T37TFE	19.03.2018
5			08.04.2018
6	Tibetan Plateau	T45SXR	16.07.2023
7			08.12.2022
8	Rocky Mountains	T11TNL	11.01.2023
9	Saskatchewan	T13UFA	19.05.2023

Although there are publicly available datasets from prior studies like Nambiar et al. (2022), Wang et al. (2022), Lu et al. (2022), Zhang et al. (2021), and Wu et al. (2021), they couldn't be utilized in our study due to misalignment in class definitions and variations in image sizes and band combinations, which do not match the requirements of our research.

A common practice for splitting of the dataset is 80% for training, 20% for validation and test (Goodfellow et al., 2016). In total, 689 patches of 256 × 256 pixel (equivalent to 2.56 × 2.56 km^2 area) were extracted from 9 tiles. Patches were split into 588 patches for training, 61 patches for validation, 40 patches for test sets. Distribution of patches among tiles are given in Table 3.2. Python code for extracting patches from images can be seen in Appendix F.

Handling extremely large images can pose challenges for CNN training due to computational limitations. Splitting an image into smaller patches addresses these issues effectively. Firstly, it significantly reduces the memory footprint required for processing, making it feasible to train the model on available hardware. Secondly, processing smaller patches is computationally faster, accelerating the training process.

Beyond efficiency, creating patches from a single image substantially increases the dataset size. This data augmentation enhances the model's ability to learn diverse features and improves its generalization performance. Additionally, each patch captures specific local details of the image, allowing the CNN to focus on and extract relevant patterns from different regions.

Most CNN architectures are designed to process fixed-size input images. Dividing the large image into patches ensures compatibility with the network's input requirements. Furthermore, the hierarchical structure of CNNs, where early layers extract local features, aligns well with the patch-based approach.

Table 3.2 Distribution of patches among the tiles.

Sentinel-2 Tile	**Number of Patches**		
S/N	**Training**	**Validation**	**Test**
1	134	4	1
2	147	7	0
3	85	8	8
4	90	11	10
5	61	21	6
6	20	0	5
8	15	4	5
9	36	6	5

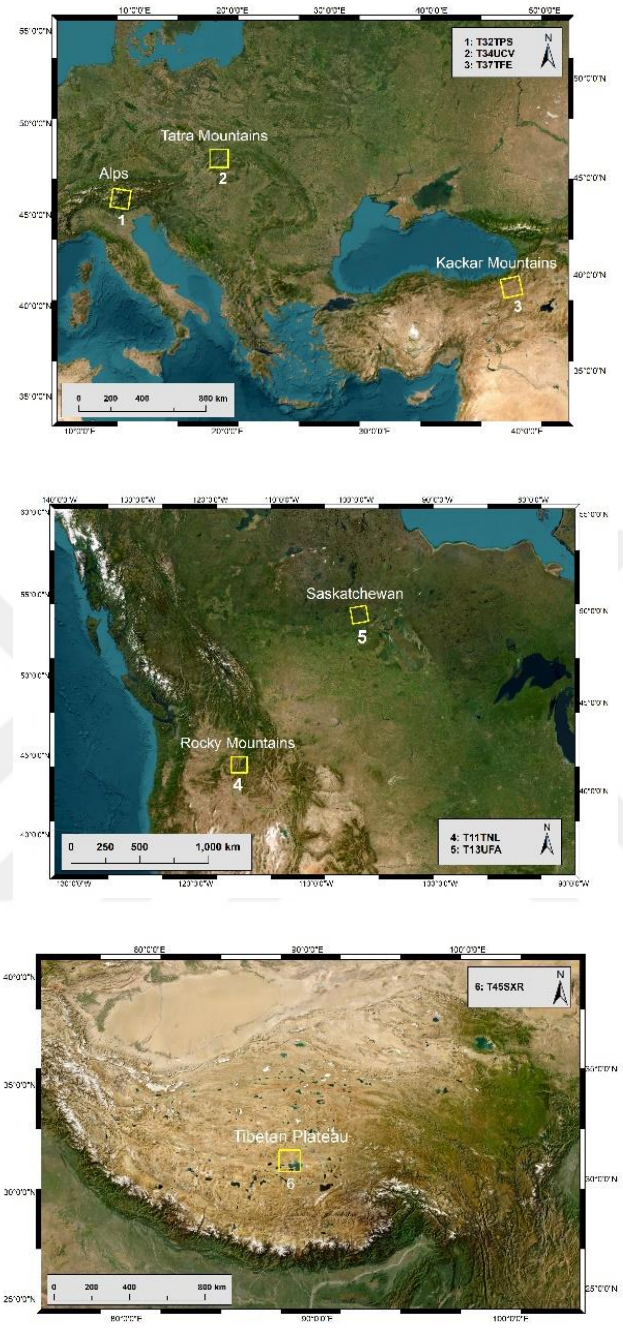
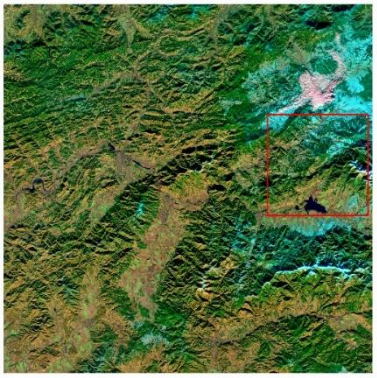


Figure 3.2 The locations of the Sentinel-2 tiles.

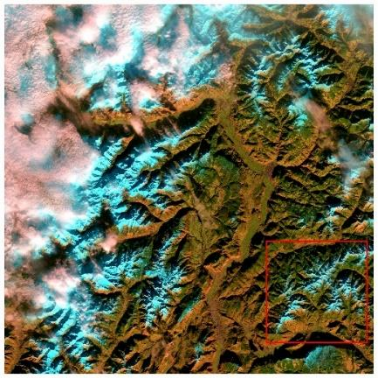
3.1.2 Planet Imagery

Orthorectified and atmospherically corrected PlanetScope orthotile surface reflectance imagery, acquired from the Dove-R and SuperDove sensors via the Education and Research Standard program, was used as an independent validation dataset in this study. These PlanetScope orthotiles offer a resolution of 3.125 meters and consist of four bands: blue, green, red, and near-infrared, closely matching Sentinel-2 bands 2, 3, 4, and 8. This facilitates the examination and comparison of our DL-based snow mapping approach.

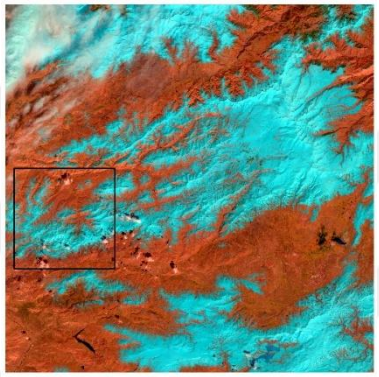
The evaluation phase included five PlanetScope images, each covering an area of 10,000 by 10,000 pixels (equivalent to 30 by 30 kilometers). These images were classified into four distinct categories—water, snow, land, and cloud—by expert groups, ensuring an impartial assessment of the model's performance on previously unobserved data. Figure 3.3 illustrates the locations of the 30 km × 30 km PlanetScope images corresponding to each Sentinel-2 image.



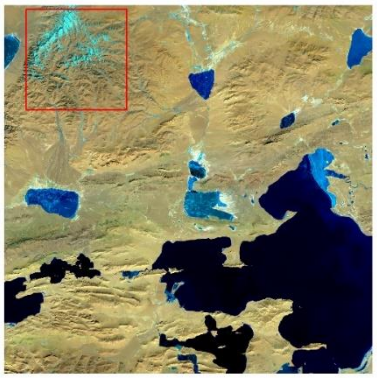
T34UCV (29.11.2018)



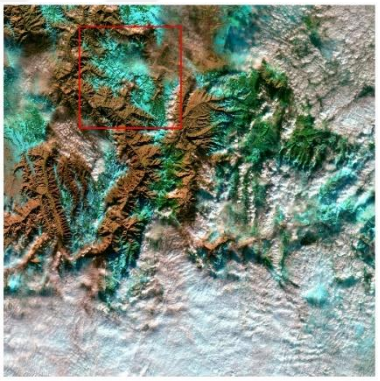
T32TPS (05.12.2018)



T37TFE (19.03.2018)



T45SXR (08.12.2022)

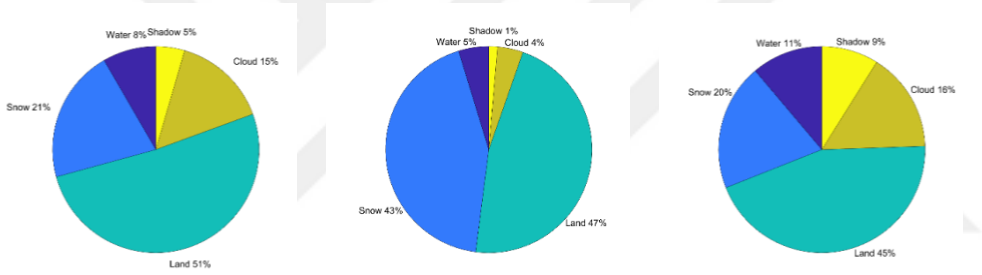


T11TNL (11.01.2023)

Figure 3.3 Locations of the PlanetScope images in Sentinel-2 tiles.

3.2 Data Labelling

During this phase, each pixel within 689 image patches extracted from Sentinel-2 data underwent classification into one of five categories: water, snow, land, cloud, or shadow. This classification process relied on RGB images generated in the preprocessing stage. Specifically, pixels covered by snow were categorized as snow, areas occupied by water bodies were labeled as water, and regions obscured by clouds were labelled as cloud-covered. Additionally, shadowy areas created by clouds or mountains were identified as shadows. Any pixels that remained unlabeled were categorized as land. In cases where there was slight shadow, but the underlying pixels were distinguishable, accurate labeling was applied. The distribution of pixel categories among the training, validation, and test sets is illustrated in Figure 3.4.



Training Set Validation Set Test Set

Figure 3.4 Pixel category distribution of image patches from Sentinel-2.

The labeling process was carried out utilizing the Image Labeler application of MATLAB 2022a (The MathWorks Inc., 2022). This application provides tools such as superpixel, smart polygon, and flood fill, which facilitated the labeling task. Examples of labeled image patches and their corresponding masks can be seen in Figure 3.5.

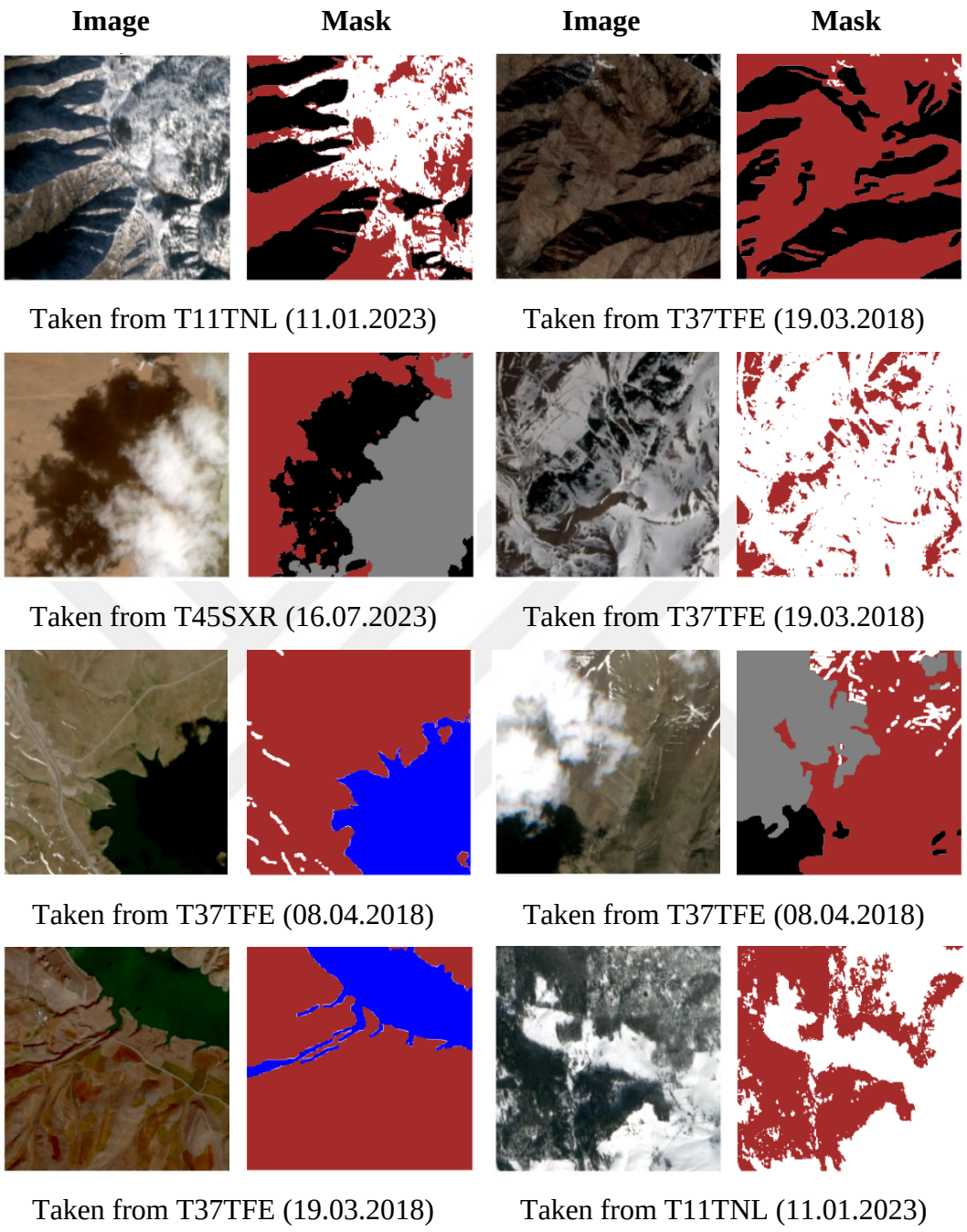


Figure 3.5 Image patches and corresponding masks (Blue: Water, White: Snow, Brown: Land, Grey: Cloud, Black: Shadow).

3.3 Transfer Learning

ML's transfer learning paradigm is using information from one problem's solution to enhance learning on a related but different problem. Because training large models from scratch requires a considerable amount of labeled data and computational resources, transfer learning is frequently used in the context of DL. Rather, pre-trained models are refined on smaller, task-specific datasets. These models are frequently furnished with encoder backbones.

Transferring knowledge from the pre-trained model's learnt features to the new task—usually through feature extraction or fine-tuning—is the fundamental idea of transfer learning. To enable the pre-trained model to adjust its learnt representations to the new task, fine-tuning entails adjusting the weights of the model during training on the new dataset. However, in feature extraction, the learned representations are extracted and fed into a new classifier that has been specially trained for the task at hand, using the pre-trained model as a fixed feature extractor.

Transfer learning facilitates faster convergence, greater generalization, and enhanced performance, especially when labeled data is scarce, by utilizing pre-trained models with strong encoder backbones. Furthermore, transfer learning makes it easier to implement DL models in actual settings where gathering a lot of labeled data could be costly or impracticable.

Pre-trained weights are for RGB channels. For 3 index bands, weights are used as is such that weights of red channel to NDVI, green to NDSI, blue to NDWI. For 12 channels, weights are repeated four times and then scaled.

The list of networks and encoders are given in Table 3.3. Network-Encoder combinations used in this study are given in Table 3.4. Almost all applicable combinations were tried.

Table 3.3 List of networks and encoders used in this study.

No	***Network***	***Encoder***
1	UNet	mobilevitv2_200
2	UNet++	mobilevitv2_100
3	DeepLabV3	efficientnet-b4
4	DeepLabV3+	efficientnet-b6
5	MA-Net	efficientnetv2_rw_m
6	LinkNet	maxvit_nano_rw_256
7	FPN	mobilenet_v2
8	PAN	dpn68
9	PSPNet	mobilevit_s
10		mobilevit_xs
11		inceptionv4
12		densenet201
13		resnet34

Table 3.4 Network-Encoder combinations used in this study.

	Networks								
Encoders	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	✓	✓	×	×	✓	✓	✓	×	×
2	✓	✓	✓	×	×	×	×	×	×
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	×	×	×	✓	×	×	✓
5	✓	×	×	×	×	×	×	×	×
6	✓	✓	×	×	✓	✓	✓	×	×
7	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	×	✓	×	×	×	×	×	×	×
10	×	✓	×	×	×	×	×	×	×
11	✓	✓	×	×	×	✓	✓	×	✓
12	✓	✓	×	×	×	✓	✓	×	✓
13	✓	✓	✓	✓	✓	✓	✓	✓	✓

3.4 Evaluation Metrics

3.4.1 Dice Score

Dice score is utilized as evaluation metric in this study. The similarity between the ground truth and the projected segmentation is measured. For multiclass segmentation, the Dice score can be extended to multiple classes. For *N* classes, the Dice score for each class *i* is determined using the formula:

$$Dice_i = \frac{2x|A_i \cap B_i|}{|A_i| + |B_i|} \quad \text{(Eq 3.1)}$$

where A_i and B_i are the sets of pixels corresponding to the ith class in the predicted and ground truth segmentations, respectively.

An overall performance statistic for multiclass segmentation can be obtained by calculating the average Dice score for each class. There are several ways to calculate the average, including by utilizing the macro-, micro- and weighted-averages.

To calculate the Micro-average Dice score, the Dice score is first determined for each class independently, and then the average is computed taking into account all of the pixels. This is also known as the global Dice score.

$$Dice_{micro} = \frac{2x \sum_{i=1}^{N} |A_i \cap B_i|}{\sum_{i=1}^{N} |A_i| + \sum_{i=1}^{N} |B_i|} \quad \text{(Eq 3.2)}$$

To calculate the macro-average dice score, the dice score for each class is determined independently, and the average for all classes is then determined. This is also known as the per-class Dice score.

$$Dice_{macro} = \frac{1}{N} \sum_{i=1}^{N} Dice_i \quad \text{(Eq 3.3)}$$

Weighted Dice score computes the Dice score for each class and then averages them, weighted by the number of true instances in each class.

$$Dice_{weighted} = \frac{1}{total\ true\ instances} \sum_{i=1}^{N} true\ instances_i x Dice_i \quad \text{(Eq 3.4)}$$

Macro average dice score calculation is used during training in this study.

3.4.2 Accuracy

Accuracy was also calculated, but since very similar results were obtained with dice score, the values were not shared for simplification but calculated for every training experiment. It is calculated using the formula:

$$Accuracy = \frac{Number\ of\ Correctly\ Classified\ Pixels}{Number\ of\ Total\ Pixels} \quad \text{(Eq 3.5)}$$

3.5 Activation Function

Activation functions are the nonlinear changes made to the neural network layer outputs. Neural networks may identify complex connections in the data by including complexity and nonlinearity. The activation function Softmax is used in this work. The formula that defines the softmax function is as follows:

$$Softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad \text{(Eq 3.6)}$$

Here, e^{x_i} is the exponential of the *i*th element, and $\sum_j e^{x_j}$ is the sum of exponentials across all elements in the vector.

3.6 Loss Functions

In a ML model, the difference between anticipated and actual values is quantified by a mathematical function called a loss function. It functions as a gauge of the model's effectiveness, showing how well the model predicts the desired variable. When training a ML model, the objective is to minimize this loss function by varying the model's parameters to bring the predictions as close to the real values as feasible.

The difference between the true and projected probability distributions is measured by Cross Entropy Loss, sometimes referred to as log loss. It can be stated mathematically as:

$$L_{CE} = -\sum_i y_i \log p_i \quad \text{(Eq 3.7)}$$

where y_i is the ground truth and p_i is the predicted probability for class *i*.

Focal Loss (Lin et al., 2017) introduces a focusing parameter (γ) and a modifying factor (α) to address the issue of class imbalance. The following is the formulation:

$$L_{FL} = -\sum_i \alpha_i (1 - p_i)^{\gamma} \log p_i \quad \text{(Eq 3.8)}$$

Dice Loss assesses the degree of spatial overlap between segmentation masks that are expected and those that are grounded in reality (Milletari et al., 2016). Double the intersection divided by the total of the ground truth and expected volumes is its definition. It has the following mathematical representation:

$$L_{Dice} = 1 - \frac{2\sum_i p_i y_i}{\sum_i p_i^2 + \sum_i y_i^2} \quad \text{(Eq 3.9)}$$

3.7 Experiment Platform

This study's experiments were conducted on a Windows 10 64-bit machine equipped with Intel Core i7-8750H CPU, 16 GB of RAM, 512 GB of SSD, 1 TB mechanical hard drive, and Nvidia GeForce RTX 2070 GPU with 8 GB of VRAM. The Python 3.8.10 (Python Software Foundation, 2021), PyTorch 2.0.1 (PyTorch Software, 2023), CUDA 11.8 (NVIDIA Corporation, 2023), and Segmentation Models Pytorch 0.3.3 (Iakubovskii, 2019) were utilized in the construction of the experiment setting.

3.8 Input Variables Fed into the Networks

The study uses three distinct predictors variable combinations. The first combination includes the NDVI, NDSI, and NDWI. The second one is composed of RGB bands (B4, B3, and B2), whereas the last combination uses 12 Sentinel-2 bands (All bands except B8A). Sentinel-2 band designations are given Table 3.5. While NDSI reveals the presence of snow and ice, NDVI evaluates the existence and health of vegetation. On the other hand, NDWI evaluates an area's water distribution and content. Python code for generating normalized band indices can be seen in Appendix G. Those indexes are computed using the following formulas:

$$NDVI = \frac{B8 - B4}{B8 + B4}, \qquad NDSI = \frac{B3 - B11}{B3 + B11}, \qquad NDWI = \frac{B3 - B8}{B3 + B8}, \quad \text{(Eq3.10)}$$

Table 3.5 Sentinel-2 spectral bands.

Sentinel-2 Bands	***Central Wavelength (µm)***	***Resolution (m)***
B1 – Coastal Aerosol	0.443	60
B2 – Blue	0.490	10
B3 – Green	0.560	10
B4 – Red	0.665	10
B5 – Vegetation Red Edge	0.705	20
B6 – Vegetation Red Edge	0.740	20
B7 – Vegetation Red Edge	0.783	20
B8 – NIR	0.842	10
B8A – Vegetation Red Edge	0.865	20
B9 – Water Vapor	0.945	60
B10 – SWIR – Cirrus	1.375	60
B11 – SWIR	1.610	20
B12 – SWIR	2.190	20

3.9 Data Augmentation

One key method in the field of ML is data augmentation, which allows datasets to be improved by making different adjustments to preexisting data samples. Usually, these adjustments involve flipping, scaling, and rotation. Data augmentation dramatically increases the quantity and diversity of the dataset by applying these transformations, which enriches the information that the learning algorithm has access to.

Data augmentation is justified by the fact that it can expose the model to a wider range of variances seen in the data. The model learns to recognize patterns and characteristics in a greater variety of scenarios by introducing various changes.

Better generalization is thus encouraged by this exposure, making it possible for the model to function better on unknown or distorted data.

Flipping both horizontally and vertically was used to show the effectiveness of data augmentation in a study by Shorten and Khoshgoftaar (2019). The dataset was enriched even though the augmentation techniques used in this study were not as complex as more involved modifications like rotation and scaling. Even though the applied augmentations are simple, their influence on model performance highlights how powerful even little augmentation strategies can be in improving the robustness and generalization capabilities of ML models.

3.10 Optimizers

To minimize the error or loss function during model training, a wide range of optimization strategies were investigated in the study. Of them, Adam (Kingma and Ba, 2014) was a flexible option that combined momentum and RMSprop methods for effective parameter space navigation. A basic optimizer called Stochastic Gradient Descent (SGD) offers a computationally efficient method by iteratively updating model parameters using gradients calculated from subsets of training data. An Adam version called Adamax uses the infinity norm for adaptive learning rates and works well for models whose gradients are sparse (Kingma and Ba, 2014). Developed by Tieleman (2012), RMSprop improves convergence on non-convex optimization problems by adjusting learning rates according on recent gradient magnitudes. Zeiler (2012) proposed Adadelta, which further improves RMSprop by doing away with the necessity for human global learning rate setting. Rprop was first presented by Riedmiller and Braun (1993). It promotes faster convergence and noise resilience by independently modifying the learning rates for each parameter based on gradient indications. By incorporating weight decay directly into Adam, Loshchilov and Hutter's (2017) AdamW solves the issues brought on by high weight decay values. Dozat (2015) introduced NAdam; it is a merging of Adam and Nesterov momentum that helps to accelerate convergence and navigate acute

minima. In order to increase stability and generalization, particularly with large minibatches, Liu et al. (2019) presented RAdam, which corrects Adam's adaptive learning rate. Lastly, Adagrad introduced by Duchi et al. (2011), adapts learning rates based on historical gradients, effectively handling sparse data by emphasizing infrequently updated parameters. Each optimizer was rigorously evaluated to discern its efficacy within the specific context of the study.

3.11 Training Process

Each model undergoes 150 training epochs. Weights that yield the best dice score on the validation set are saved during training. The test set is used to determine the model's performance. Python scripts for training, testing and related functions can be seen in Appendix A, B, C, D and E. The dice scores found in Table 3.6, Table 3.7, Table 3.8, Table 3.9, and Table 3.10 are part of the exam set. To decide whether to utilize augmentation or transfer learning, as well as to establish hyperparameters such kernel size, learning rate, batch size, and optimizer, several tests are first conducted.

A data augmentation probability of $p = 0.5$ and transfer learning were chosen based on the improved dice scores observed on the test set (cf. Table 3.6). Different kernel sizes were tested in the second experiment, with no significant increase in dice scores noted despite longer training times caused by increased computational demands. A 3×3 kernel size was selected based on these observations (cf. Table 3.7). In the third trial, various learning rates were tested, and an optimal learning rate of 10^{-4} was identified (cf. Table 3.8).

Table 3.6 Experiment to determine augmentation and transfer learning (Aug: Augmentation, HF: Horizontal Flip, VF: Vertical Flip, p: probability, Pr-Tr: Pretrained Weights, ksize: Kernel Size, LF: Loss Function, LR: Learning Rate, CE: Cross Entropy).

Aug	***Input***	***Network***	***Encoder***	***Pr-Tr***	***ksize***	***LF***	***LR***	***Batch Size***	***Optimizer***	***Dice Score***
-	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.90890
HF(p=0.3) VF(p=0.3)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.90633
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.91195
-	RGB	U-Net	-	-	3x3	CE	1e-4	8	Adam	0.87697
HF(p=0.3) VF(p=0.3)	RGB	U-Net	-	-	3x3	CE	1e-4	8	Adam	0.87921
HF(p=0.5) VF(p=0.5)	RGB	U-Net	-	-	3x3	CE	1e-4	8	Adam	0.88590

Table 3.7 Experiment to determine kernel size.

Aug	***Input***	***Network***	***Encoder***	***Pr-Tr***	***ksize***	***LF***	***LR***	***Batch Size***	***Optimizer***	***Dice Score***
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.91195
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	5x5	CE	1e-4	8	Adam	0.91407
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	7x7	CE	1e-4	8	Adam	0.91106

Table 3.8 Experiment to determine learning rate.

Aug	***Input***	***Network***	***Encoder***	***Pr-Tr***	***ksize***	***LF***	***LR***	***Batch Size***	***Optimizer***	***Dice Score***
-	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-5	8	Adam	0.88690
-	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.90890
-	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-3	8	Adam	0.85845

Different batch sizes were experimented, and a batch size of eight was selected based on experiment 4, as it yielded the highest dice score (cf. Table 3.9). Various optimizers were tested in experiment 5, with the Adam optimizer being chosen for its superior performance (cf. Table 3.10).

Table 3.9 Experiment to determine batch size.

Aug	***Input***	***Network***	***Encoder***	***Pr-Tr***	***ksize***	***LF***	***LR***	***Batch Size***	***Optimizer***	***Dice Score***
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	4	Adam	0.90952
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	8	Adam	0.91195
HF(p=0.5) VF(p=0.5)	RGB	U-Net	Resnet34	imagenet	3x3	CE	1e-4	16	Adam	0.90105

Table 3.10 Experiment to determine optimizer (en-b4: efficientnet-b4).

Aug	***Input***	***Network***	***Encoder***	***Pr-Tr***	***ksize***	***LF***	***LR***	***Batch Size***	***Optimizer***	***Dice Score***
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Adam	0.92754
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	SGD	0.91288
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Adamax	0.91076
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	RMSProp	0.92101
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Adadelta	0.92171
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Rprop	0.90582
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	AdamW	0.92326
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Nadam	0.91929
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Radam	0.92101
HF(p=0.5) VF(p=0.5)	12 Bands	U-Net	en-b4	imagenet	3x3	CE	1e-4	8	Adagrad	0.90112

Other training experiments are based on Table 3.4. The findings of the best experiment will be included in the results section. Loss and dice scores during training, which corresponds to the first fifteen experiments in terms of test dice score will be provided.

CHAPTER 4

RESULTS AND DISCUSSIONS

This section presents a comprehensive analysis of the proposed CNN models for snow cover detection in forested and mountainous regions. The efficieny of various classification approaches, band combinations, segmentation networks, and loss functions is evaluated in detail. Additionally, the performance of the optimal model on an independent Planet test dataset is assessed. The chapter concludes with an in-depth examination of the model's efficiency in differentiating snow-covered areas within complex terrains.

4.1 Classification Results

Table 4.1 provides information on the first fifteen trials based on test dice scores. Figure 4.1 displays the dice score on the validation set and the loss experienced during the best model's training.

Table 4.1 The details of the first 15 experiments in terms of test dice score (3 Index is NDVI, NDSI, NDWI, Pr-Tr is imagenet, ksize is 3x3, Aug is HF(p=0.5) VF(p=0.5), LR is 1e-4, Batch size is 8, Optimizer is Adam)

S/N	***Input***	***Network***	***Encoder***	***LF***	***Dice Score***
1	3 Index	U-Net++	mobilevitv2_200	Focal	0.92814
2	12 Bands	U-Net	efficientnet-b4	Focal	0.92754
3	3 Index	U-Net	maxvit_nano_rw_256	Focal	0.92706
4	12 Bands	U-Net++	mobilenet_v2	Focal	0.92687
5	12 Bands	U-Net++	dpn68	Cross Entropy	0.92583
6	3 Index	U-Net++	maxvit_nano_rw_256	Focal	0.92583
7	12 Bands	U-Net++	efficientnet-b6	Cross Entropy	0.92576
8	12 Bands	U-Net++	efficientnet-b6	Focal	0.92568
9	12 Bands	U-Net++	efficientnet-b4	Cross Entropy	0.92563
10	12 Bands	U-Net	efficientnet-b4	Dice	0.92547
11	12 Bands	U-Net	mobilenet_v2	Cross Entropy	0.92471
12	12 Bands	U-Net++	mobilenet_v2	Dice	0.92447
13	3 Index	U-Net++	mobilenet_s	Focal	0.92433
14	12 Bands	U-Net++	inceptionv4	Cross Entropy	0.92402
15	12 Bands	U-Net	efficientnet-b6	Dice	0.92400

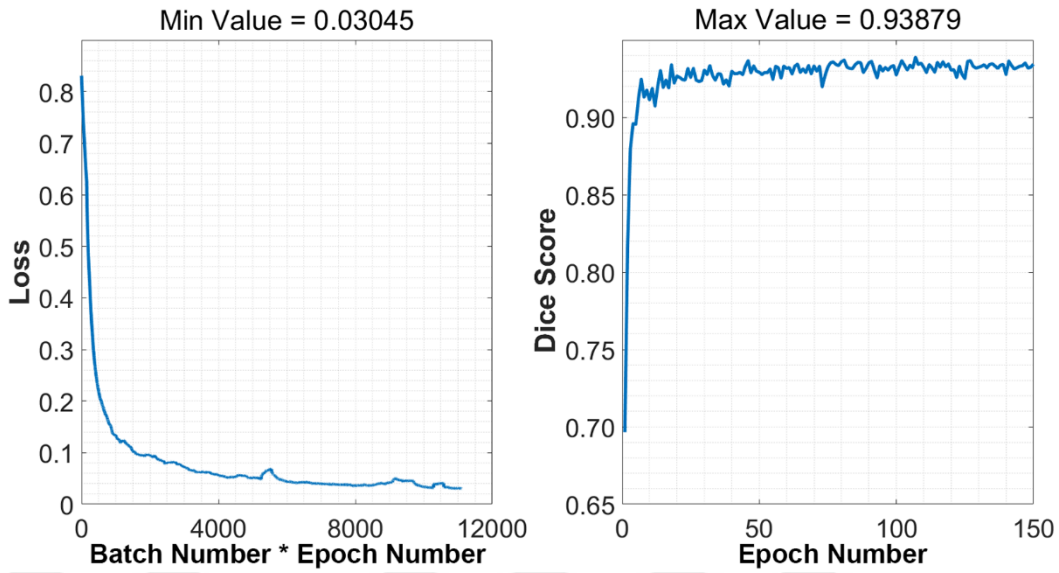


Figure 4.1 Dice score on validation set and loss during the training of the best model (Total batch number is obtained by dividing the number of test images (i.e., 588) to batch size (i.e., 8) and rounding it up.).

Figure 4.2 shows the best model's predictions on the Sentinel-2 test set (input: NDSI, NDVI, NDWI, network: U-Net++, focus loss). Figure 4.3 displays the outcomes of the PlanetScope test set. Table 4.2 provides dice scores of each class and Table 4.3 provides overall dice scores for the PlanetScope test set. Dice scores were determined after resampling the Planet photos to the Sentinel-2 resolution using the closest neighbor approach, as the resolution of the Sentinel-2 and Planet images differs.

Table 4.2 Overall dice scores of each class for Planet test images

Classes	*Water*	*Snow*	*Land*	*Cloud*
Dice Score	0.270	0.695	0.848	0

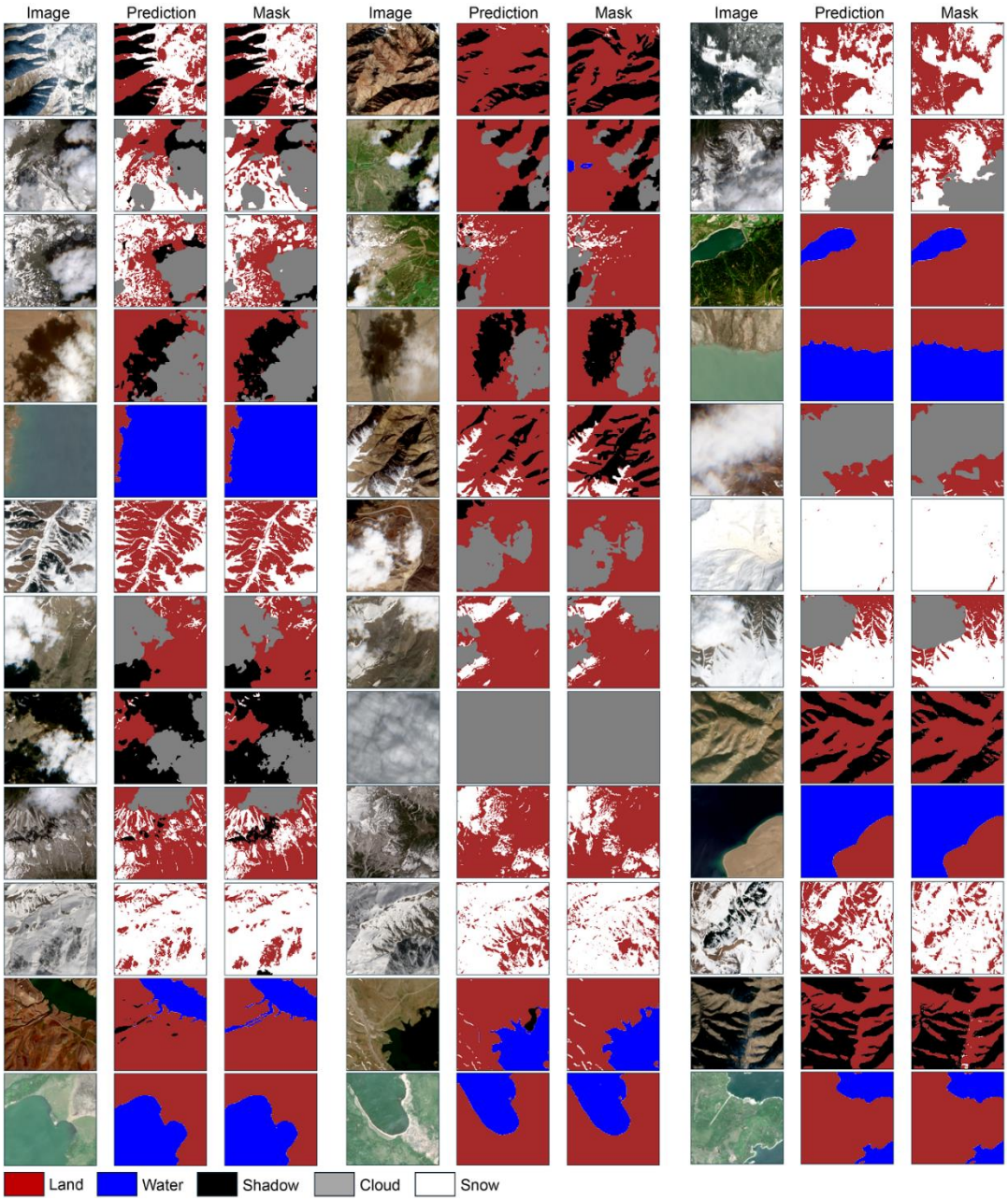


Figure 4.2 Predictions of the best model on Sentinel-2 test set (Image: RGB real color composite; Prediction: Image classified by the model; Mask: Pre-labeled image by experts).

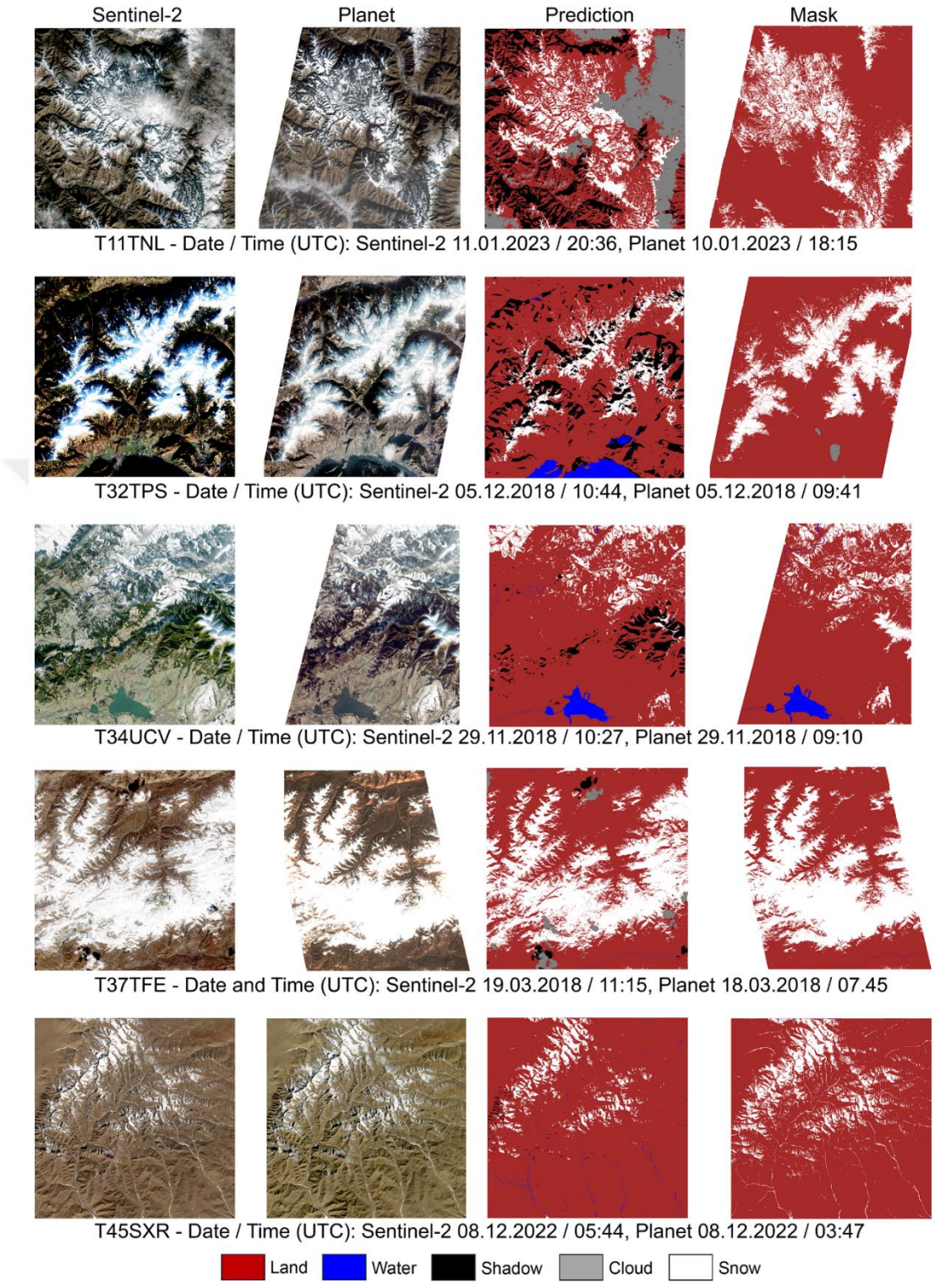


Figure 4.3 Predictions of the best model on PlanetScope test set (Mask is the classified map obtained from the corresponding Planet images by an expert.).

Table 4.3 Dice scores for Planet test images.

Tile	*Dice Score – All Classes*		*Dice Score – Excluded Shadow Class*	
	Weighted	***Micro***	***Weighted***	***Micro***
T45SXR	0.929	0.934	0.931	0.938
T34UCV	0.775	0.761	0.798	0.806
T32TPS	0.749	0.684	0.829	0.816
T11TNL	0.639	0.524	0.700	0.610
T37TFE	0.762	0.752	0.765	0.759
Overall	**0.771**	**0.731**	**0.805**	**0.786**

Only 256 × 256-pixel wide images, or patches, are compatible with our concept. We separated the Sentinel-2 tile into patches, classed each patch, and then put the patches back together to get the classifications for the full tile. We, however, found that this method results in incorrect classifications, particularly along the edges of the patches where one patch ends and other begins. We employed a different strategy in place of this direct one, moving the patch window 16 pixels as opposed to 256. After obtaining overlapping categorized images, we determined the final classifications by majority vote. This is how the classification findings shown in Figure 4.3 were obtained. Figure 4.4 illustrates how these two methods differ from each other.

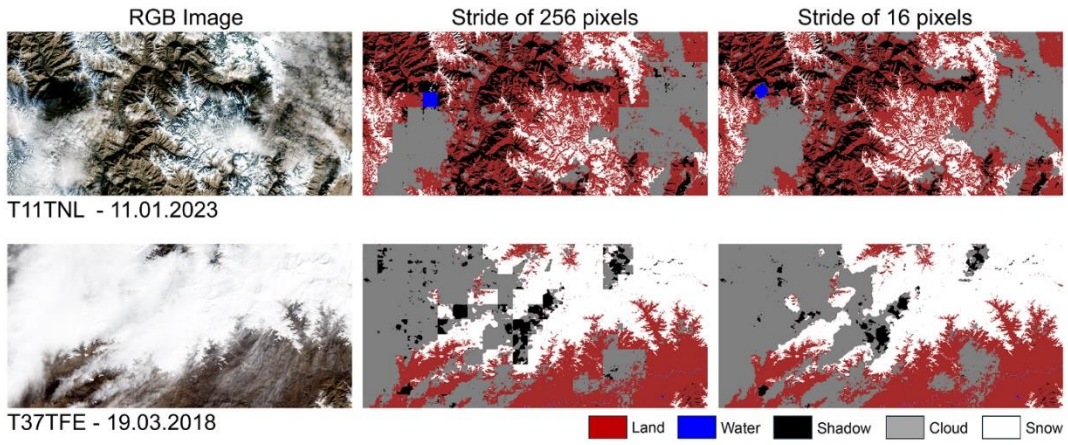


Figure 4.4 The resulting classifications for stride of 256 and 16 pixels.

4.2 Optimal Band Combination

Upon conducting comprehensive experiments utilizing three unique band configurations (namely, NDSI-NDVI-NDWI, RGB, and all 12 bands), we achieved the average dice scores, which are presented in Table 4.4. The NDSI-NDVI-NDWI outperformed RGB and the 12-band set with a dice score of 0.917. This result highlights the value of spectral indices in distinguishing certain traits related to snow, vegetation, and water bodies. The network was able to differentiate between different land cover categories thanks to these indices, which led to more precise segmentation findings.

Table 4.4 Average dice scores of 3 different band combinations based on Sentinel-2 test set.

Band Combination	Average Dice Score Based on Sentinel-2 Test Set
NDSI-NDVI-NDWI	0.917
12 Bands	0.910
RGB	0.896

While RGB bands provide an accurate representation of color, they may not have the spectral depth needed to distinguish minute changes across various land cover groups. RGB bands are still a strong option for some segmentation jobs, though, especially if they need to be easily comprehended and shown, as seen by the relatively high dice score.

The 12-band set did not yield the highest dice score, which was unexpected given its wide spectral coverage. Greater information is obtained by adding more bands, although dimensionality and feature redundancy problems are nonetheless present. In this case, the combination of NDSI, NDVI, and NDWI performed somewhat better, indicating that choosing band combinations based on the specific task at hand may be preferable to employing the entire spectral range freely. This outcome is

consistent with Wang et al. (2022) findings, which showed that B2, B11, B4, and B9 constitute the best spectral band combination for U-Net.

4.3 Comparison of Segmentation Networks

Using Sentinel-2 imagery, this study evaluated the performance of several segmentation networks on semantic segmentation tasks. Three indices were used as the input for the evaluation of nine distinct networks.

U-Net++ was the best performer with a dice score of 0.921. Significant variations in segmentation performance between networks are seen by the data in Table 4.5. The well-known U-Net architecture has been upgraded with U-Net++, which adds additional skip connections and multi-scale feature fusion algorithms to enhance its capacity to extract finer features and contextual information. This greater performance is an indication of the improved segmentation accuracy that comes from architectural changes. Apart from its competitive performance, U-Net's simplicity was compared to topologies with more complexity.

Table 4.5 Average dice scores of 9 different segmentation networks based on Sentinel-2 test set.

Segmentation Network	Average Dice Score based on Sentinel-2 Test Set
U-Net++	0.921
U-Net	0.913
DeepLabV3+	0.912
Linknet	0.912
FPN	0.909
PAN	0.903
DeepLabV3	0.896
MA-Net	0.894
PSPnet	0.890

Simplified architectures like U-Net are still competitive and more usable for real-world applications, even though more complex architectures like U-Net++ and DeepLabV3+ provide better performance in some circumstances.

4.4 Comparison of Loss Functions

This study also investigated how different loss functions affect the performance of semantic segmentation, as illustrated in Table 4.6. The average dice score was slightly lower when comparing focal loss to cross entropy loss and dice loss. However, focal loss yielded the best results, which is noteworthy since it emphasizes how well this approach works to correct class imbalance and how important it is to focus on examples that are challenging to classify. Focal loss dynamically adjusts the weighting of each pixel based on the difficulty of its classification, thereby focusing more on hard-to-classify pixels and mitigating the class imbalance issue

Table 4.6 Average dice scores of 3 different loss functions based on Sentinel-2 test set.

Loss Function	***Average Dice Score Based on Sentinel-2 Test Set***
Dice	0.914
Cross Entropy	0.913
Focal	0.908

The performance of each loss function can differ based on optimization parameters and network architecture, as demonstrated by the reported differences between average dice scores and actual segmentation performance.

4.5 Performance On Planet Test Set

The tile T45SXR, which contains only land and snow classes, demonstrated the highest accuracy based on the findings from the PlanetScope test set. Both images

were captured on December 8, 2022, at different times—05:44 UTC for Sentinel-2 and 03:47 UTC for Planet. The acquisition date for tile T34UCV is consistent. Misclassification in the Sentinel data is primarily caused by shadow classes. When the shadow class is excluded, the aggregate micro dice score improves from 0.761 to 0.806. In the DL method, the land shadow on tile T32TPS is misclassified as water, leading to low accuracy. Notable improvement is observed for tile T32TPS when the shadow class is removed from accuracy metric computations. However, for tile T11TNL, excluding the shadow classes does not significantly increase the dice score, resulting in the lowest accuracy. The acquisition times for this tile differ: Sentinel-2 at 20:36 UTC on January 10, 2023, and Planet at 18:15 UTC on the same date. Due to snowfall, the area covered in snow has increased.

Based on the ERA5-Land reanalysis product (Muñoz-Sabater et al., 2021) minimum, maximum, and average temperatures as well as cumulative precipitation values, it can be observed that there was precipitation on July 11 between 2 and 4 am in the form of snowfall (Figure 4.5), with the planet data from the previous day being free of this newly fallen snow.

To reduce the impact of the missing shadow class in the planet dataset on the dice score, synthetic images with artificial shadows added to areas that are not shadowed or non-shaded images of the same area with shadows could be used for training. With this technique the model could learn what is underneath the shadow hence there would not be a need for shadow class.

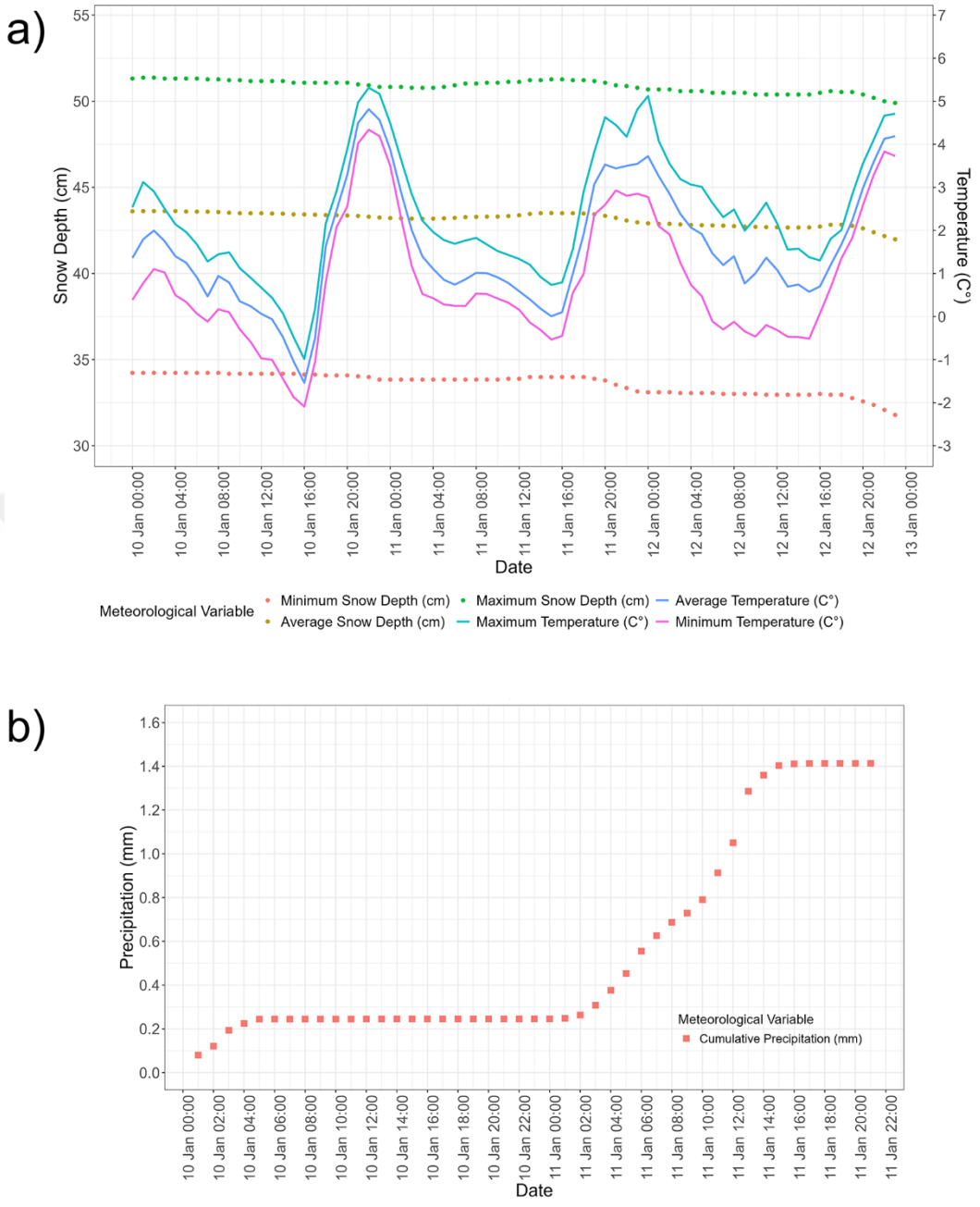


Figure 4.5 a) Snowfall and temperature patterns, and b) cumulative precipitation retrieved from ERA5-Land reanalysis product between the acquisition times of Sentinel-2 and Planet images for T11TNL tile.

For T37TFE tile, the snow classification in Planet tile is greater than the snow classes in Sentinel-2 tile. Furthermore, the acquisition times for this tile are also different (19.03.2018 11:15 UTC for Sentinel-2, 18.03.2018 07:45 UTC for Planet). Sentinel data is one day behind Planet data. When the temperature data for those dates are compared, it is found that melting is occurring at an increasing rate, while the snow depth values from the ERA5-Land reanalysis product demonstrate a decrease (see Figure 4.6). This explains why this tile's accuracy was low.

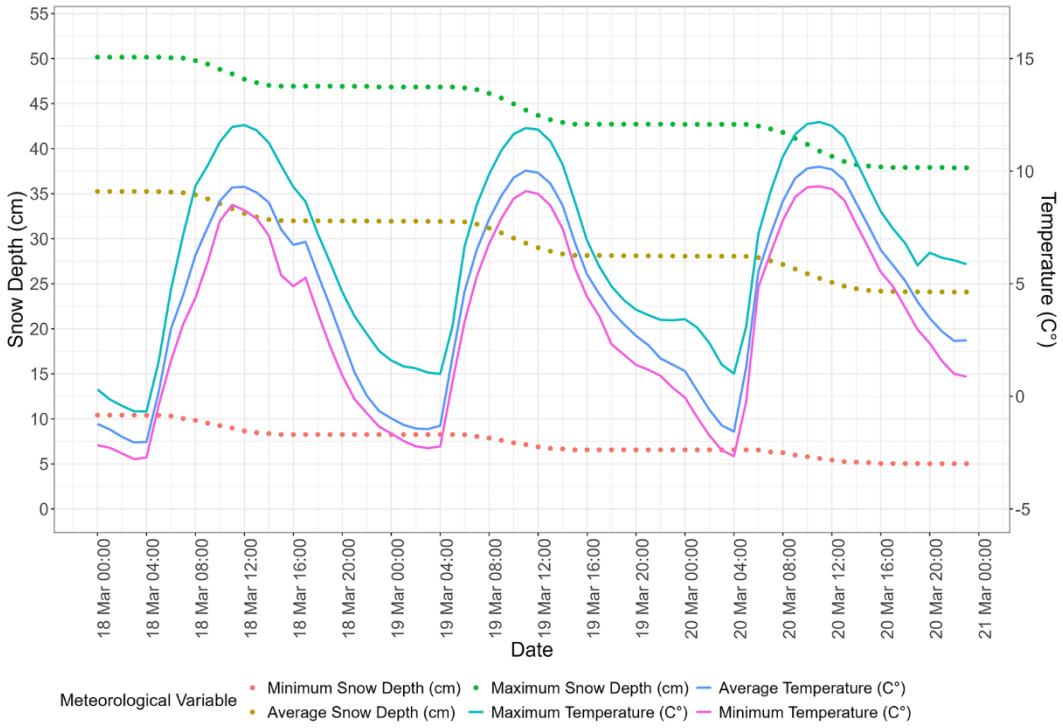


Figure 4.6 Temperature increases between the acquisition times of Sentinel-2 and Planet images for T37TFE tile retrieved from ERA5-Land reanalysis product.

4.6 Efficiency in Forested and Mountainous Regions

Using Sentinel-2 satellite images, our study demonstrates the effectiveness of our segmentation network in defining land cover in both mountainous and forested regions (cf. Figure 4.7 and Table 4.7). While the network classification maps the snow in the open regions of the forest covered area, the planet data displays the snow

over the area covered by the forest. As demonstrated by high dice score metrics and visually consistent segmentation results, our network performs robustly in forested environments where dense canopy cover presents obstacles. In mountainous areas with variable topography and heterogeneous spectral light, our network efficiently and accurately detects patterns of land cover. The numerical results that are displayed, along with the dice score, confirm how effective our method is. Additionally, qualitative evaluations using the resulting images confirm that the segmentation borders are clearly defined.

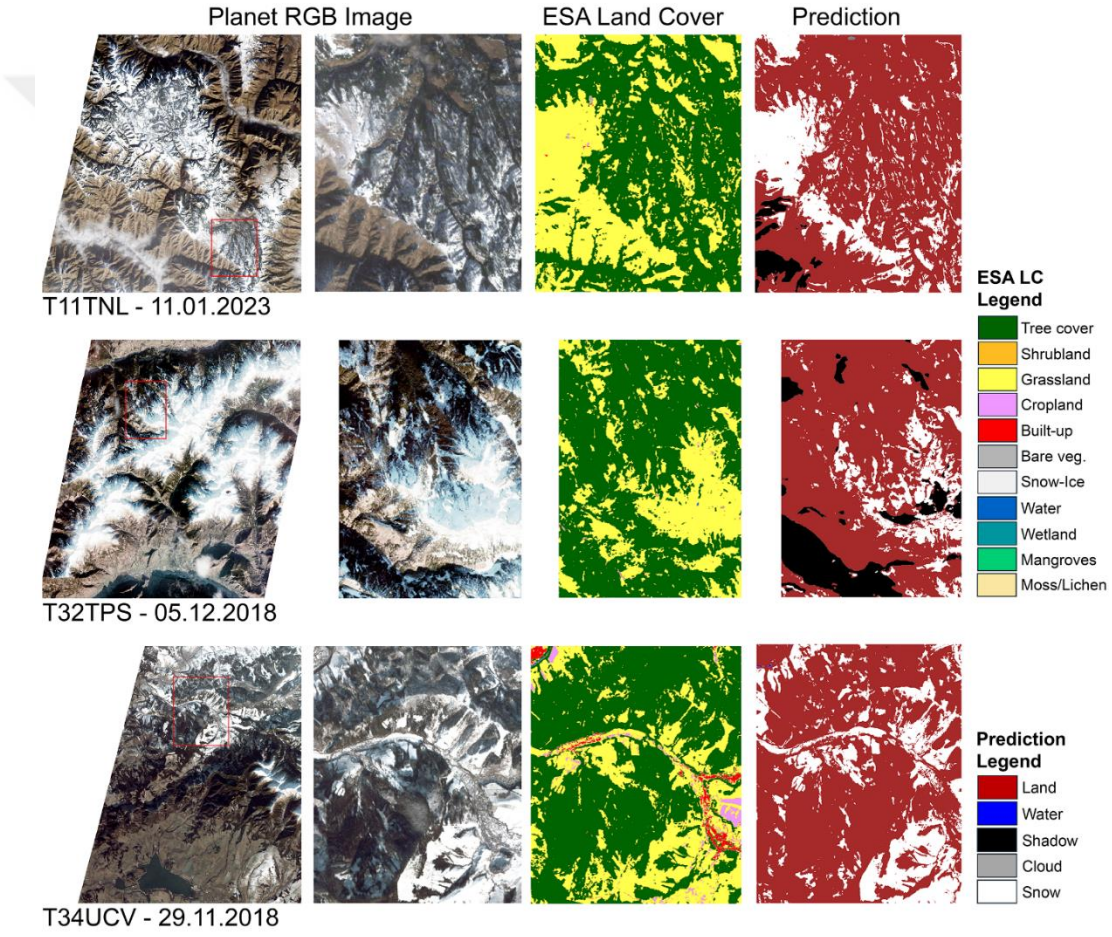


Figure 4.7 The network efficiency on forested and mountainous areas.

The reason for the relatively low dice score for T11TNL is the previously mentioned snowfall (cf. Figure 4.5).

Table 4.7 Dice scores for forested and mountainous areas.

Tile	***Dice Score – All Classes***		***Dice Score – Excluded Shadow Class***	
	Weighted	***Micro***	***Weighted***	***Micro***
T11TNL	0.750	0.745	0.763	0.771
T32TPS	0.808	0.751	0.872	0.869
T34UCV	0.823	0.805	0.843	0.856
Overall	**0.793**	**0.767**	**0.826**	**0.832**

CHAPTER 5

CONCLUSIONS

Utilizing Sentinel-2 data and DL techniques, our study provides an in-depth analysis of snow cover detection in mountainous and forested regions. We meticulously tested various band combinations and segmentation networks to generate accurate and reliable results, particularly tailored to the challenges posed by these complex environments.

Our findings emphasize the critical role of spectral indices, such as NDVI, NDSI, and NDWI, in enhancing the segmentation accuracy. These indices outperform RGB and all 12 bands by a small margin. The use of these indices underscores the importance of leveraging spectral diversity to improve detection capabilities in challenging terrains. By combining these indices, we were able to better differentiate between snow, vegetation, and other land cover types, thereby improving the overall accuracy of our snow cover maps.

Among the segmentation networks, U-Net++ emerged as the most effective in our study, underscoring the importance of architectural modifications in enhancing segmentation accuracy, especially in areas with dense vegetation and variable topography. While deeper architectures like DeepLabV3+ are adept at capturing larger contextual features, simpler architectures like U-Net remain competitive and practical for real-world applications, exhibiting strong performance in both mountainous and forest areas.

Our study also explored the impact of various loss functions on segmentation performance, with focal loss proving particularly effective in addressing class imbalance in mountainous and forested environments. Class imbalance is a common issue in remote sensing applications, where the minority class (e.g., snow cover) is often overshadowed by the majority class (e.g., vegetation or bare ground).

In summary, our research advances methods for detecting snow cover in mountainous and forested regions by providing insights into the most effective ways to integrate DL techniques with Sentinel-2 imagery and to accurately and efficiently segment land cover. The robustness of our method for environmental monitoring and management is validated by both qualitative and quantitative metrics, highlighting the potential of DL approaches to significantly enhance environmental monitoring capabilities and inform decision-making processes.

Our results demonstrate the feasibility of using CNNs for snow cover mapping, achieving promising classification accuracies across different experimental setups. However, a significant challenge encountered in this study is the lack of a suitable reference benchmark test dataset for assessing classification accuracy. Unlike many image classification tasks where a wide range of benchmark test sets are readily available, the absence of standardized datasets in snow cover mapping remains a critical issue.

Moreover, the application of transfer learning techniques could improve the generalization of our models to different geographic regions and climatic conditions.

Despite the spectral differences, the lower-level features learned by the model, such as edge detection and texture patterns, can still be relevant. For instance, boundaries between different land cover types (e.g., water vs. land) might still be recognizable by the network. However, the higher-level features may need fine-tuning to adapt to the specific characteristics of NDWI, NDSI, and NDVI.

In addition to methodological advancements, our study underscores the importance of interdisciplinary collaboration in addressing complex environmental challenges. By combining expertise from remote sensing, computer science, and environmental science, we can develop more comprehensive and effective solutions for snow cover detection and monitoring.

REFERENCES

Akyürek, Z., Şorman, A.Ü., 2002. Monitoring snow-covered areas using NOAA-AVHRR data in the eastern part of Turkey. Hydrological Sciences Journal 47, 243–252. https://doi.org/10.1080/02626660209492927

Breen, C., Vuyovich, C., Odden, J., Hall, D., Prugh, L., 2023. Evaluating MODIS snow products using an extensive wildlife camera network. Remote Sens Environ 295, 113648. https://doi.org/10.1016/j.rse.2023.113648

Brown, R.D., Mote, P.W., 2009. The Response of Northern Hemisphere Snow Cover to a Changing Climate*. J Clim 22, 2124–2145. https://doi.org/10.1175/2008JCLI2665.1

Chaurasia, A., Culurciello, E., 2017. LinkNet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, pp. 1–4. https://doi.org/10.1109/VCIP.2017.8305148

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking Atrous Convolution for Semantic Image Segmentation.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.

Chen, X., Yang, Y., Yin, C., 2021. Contribution of Changes in Snow Cover Extent to Shortwave Radiation Perturbations at the Top of the Atmosphere over the Northern Hemisphere during 2000–2019. Remote Sens (Basel) 13, 4938. https://doi.org/10.3390/rs13234938

Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J., 2017. Dual path networks. Adv Neural Inf Process Syst 30.

Dietz, A.J., Kuenzer, C., Gessner, U., Dech, S., 2012. Remote sensing of snow – a review of available methods. Int J Remote Sens 33, 4094–4134. https://doi.org/10.1080/01431161.2011.640964

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Dozat, T., 2015. CS229 Machine Learning Final Project Report: Predicting Diabetes with Machine Learning.

Dozier, J., 1989. Spectral signature of alpine snow cover from the landsat thematic mapper. Remote Sens Environ 28, 9–22. https://doi.org/10.1016/0034-4257(89)90101-6

Dozier, J., Painter, T.H., 2004. Multispectral and Hyperspectral Remote Sensing of Alpine Snow Properties. Annu Rev Earth Planet Sci 32, 465–494. https://doi.org/10.1146/annurev.earth.32.101802.120404

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., Bargellini, P., 2012. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. Remote Sens Environ 120, 25–36. https://doi.org/10.1016/j.rse.2011.11.026

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research 12.

Fan, T., Wang, G., Li, Y., Wang, H., 2020. MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation. IEEE Access 8, 179656–179665. https://doi.org/10.1109/ACCESS.2020.3025372

Frei, A., Tedesco, M., Lee, S., Foster, J., Hall, D.K., Kelly, R., Robinson, D.A., 2012. A review of global satellite-derived snow products. Advances in Space Research 50, 1007–1029. https://doi.org/10.1016/j.asr.2011.12.021

Goodfellow, I., Bengio, Y., Courville, A., 2016. Machine Learning Basics in Deep Learning.

Hall, D.K., Riggs, G.A., Salomonson, V. V., 1995. Development of methods for mapping global snow cover using moderate resolution imaging spectroradiometer data. Remote Sens Environ 54, 127–140. https://doi.org/10.1016/0034-4257(95)00137-P

Hammond, J.C., Saavedra, F.A., Kampf, S.K., 2018. Global snow zone maps and trends in snow persistence 2001–2016. International Journal of Climatology 38, 4369–4383. https://doi.org/10.1002/joc.5674

Hao, X., Huang, G., Zheng, Z., Sun, X., Ji, W., Zhao, H., Wang, J., Li, H., Wang, X., 2022. Development and validation of a new MODIS snow-cover-extent product over China. Hydrol Earth Syst Sci 26, 1937–1952. https://doi.org/10.5194/hess-26-1937-2022

Haseeb Azizi, A., Akhtar, F., Kusche, J., Tischbein, B., Borgemeister, C., Agumba Oluoch, W., 2024. Machine learning-based estimation of fractional snow cover in the Hindukush Mountains using MODIS and Landsat data. J Hydrol (Amst) 638, 131579. https://doi.org/10.1016/j.jhydrol.2024.131579

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hou, J., Huang, C., Zhang, Y., Guo, J., 2020. On the Value of Available MODIS and Landsat8 OLI Image Pairs for MODIS Fractional Snow Cover Mapping Based on an Artificial Neural Network. IEEE Transactions on Geoscience and Remote Sensing 58, 4319–4334. https://doi.org/10.1109/TGRS.2019.2963075

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708.

Iakubovskii, P., 2019. Segmentation Models Pytorch.

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization.

Klein, A.G., Hall, D.K., Riggs, G.A., 1998. Improving snow cover mapping in forests through the use of a canopy reflectance model. Hydrol Process 12, 1723–1744. https://doi.org/10.1002/(SICI)1099-1085(199808/09)12:10/11<1723::AID-HYP691>3.0.CO;2-2

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25.

Kuter, S., Akyurek, Z., Weber, G.-W., 2018. Retrieval of fractional snow covered area from MODIS data by multivariate adaptive regression splines. Remote Sens Environ 205, 236–252. https://doi.org/10.1016/j.rse.2017.11.021

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2016. Feature Pyramid Networks for Object Detection.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J., 2019. On the Variance of the Adaptive Learning Rate and Beyond.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path Aggregation Network for Instance Segmentation.

Loshchilov, I., Hutter, F., 2017. Decoupled Weight Decay Regularization.

Lu, Y., James, T., Schillaci, C., Lipani, A., 2022. Snow detection in alpine regions with Convolutional Neural Networks: discriminating snow from cold clouds and water body. GIsci Remote Sens 59, 1321–1343. https://doi.org/10.1080/15481603.2022.2112391

Lv, Z., Pomeroy, J.W., 2019. Detecting intercepted snow on mountain needleleaf forest canopies using satellite remote sensing. Remote Sens Environ 231, 111222. https://doi.org/10.1016/j.rse.2019.111222

Ma, J., Shen, H., Cai, Y., Zhang, T., Su, J., Chen, W.-H., Li, J., 2023. UCTNet with Dual-Flow Architecture: Snow Coverage Mapping with Sentinel-2 Satellite Imagery. Remote Sens (Basel) 15, 4213. https://doi.org/10.3390/rs15174213

Ma, Y., Zhang, Y., 2022. Improved on Snow Cover Extraction in Mountainous Areas Based on Multi-factor NDSI Dynamic Threshold. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B3-2022, 771–778. https://doi.org/10.5194/isprs-archives-XLIII-B3-2022-771-2022

Mehta, S., Rastegari, M., 2022. Separable Self-attention for Mobile Vision Transformers.

Mehta, S., Rastegari, M., 2021. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer.

Metsämäki, S., Pulliainen, J., Salminen, M., Luojus, K., Wiesmann, A., Solberg, R., Böttcher, K., Hiltunen, M., Ripper, E., 2015. Introduction to GlobSnow Snow Extent products with considerations for accuracy assessment. Remote Sens Environ 156, 96–108. https://doi.org/10.1016/j.rse.2014.09.018

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, in: 2016 Fourth International Conference on 3D Vision (3DV). IEEE, pp. 565–571. https://doi.org/10.1109/3DV.2016.79

Mote, P.W., Li, S., Lettenmaier, D.P., Xiao, M., Engel, R., 2018. Dramatic declines in snowpack in the western US. NPJ Clim Atmos Sci 1, 2. https://doi.org/10.1038/s41612-018-0012-1

Muñoz-Sabater, J., Dutra, E., Agustí-Panareda, A., Albergel, C., Arduini, G., Balsamo, G., Boussetta, S., Choulga, M., Harrigan, S., Hersbach, H., 2021. ERA5-Land: A state-of-the-art global reanalysis dataset for land applications. Earth Syst Sci Data 13, 4349–4383.

Nambiar, K.G., Morgenshtern, V.I., Hochreuther, P., Seehaus, T., Braun, M.H., 2022. A Self-Trained Model for Cloud, Shadow and Snow Detection in Sentinel-2 Images of Snow- and Ice-Covered Regions. Remote Sens (Basel) 14, 1825. https://doi.org/10.3390/rs14081825

NVIDIA Corporation, 2023. CUDA Toolkit (version 11.8) .

Pulliainen, J., Hallikainen, M., 2001. Retrieval of Regional Snow Water Equivalent from Space-Borne Passive Microwave Observations. Remote Sens Environ 75, 76–85. https://doi.org/10.1016/S0034-4257(00)00157-7

Pulliainen, J., Luojus, K., Derksen, C., Mudryk, L., Lemmetyinen, J., Salminen, M., Ikonen, J., Takala, M., Cohen, J., Smolander, T., Norberg, J., 2020. Publisher Correction: Patterns and trends of Northern Hemisphere snow mass from 1980 to 2018. Nature 582, E18–E18. https://doi.org/10.1038/s41586-020-2416-4

Python Software Foundation, 2021. Python (version 3.8.10).

PyTorch Software, 2023. PyTorch (version 2.0.1) .

Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: IEEE International Conference on Neural Networks. IEEE, pp. 586–591. https://doi.org/10.1109/ICNN.1993.298623

Rittger, K., Raleigh, M.S., Dozier, J., Hill, A.F., Lutz, J.A., Painter, T.H., 2020. Canopy Adjustment and Improved Cloud Detection for Remotely Sensed Snow Cover Mapping. Water Resour Res 56. https://doi.org/10.1029/2019WR024914

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rosenthal, W., Dozier, J., 1996. Automated Mapping of Montane Snow Cover at Subpixel Resolution from the Landsat Thematic Mapper. Water Resour Res 32, 115–130. https://doi.org/10.1029/95WR02718

Rutter, N., Essery, R., Pomeroy, J., Altimir, N., Andreadis, K., Baker, I., Barr, A., Bartlett, P., Boone, A., Deng, H., Douville, H., Dutra, E., Elder, K., Ellis, C., Feng, X., Gelfan, A., Goodbody, A., Gusev, Y., Gustafsson, D., Hellström, R., Hirabayashi, Y., Hirota, T., Jonas, T., Koren, V., Kuragina, A., Lettenmaier, D., Li, W., Luce, C., Martin, E., Nasonova, O., Pumpanen, J., Pyles, R.D., Samuelsson, P., Sandells, M., Schädler, G., Shmakin, A., Smirnova, T.G., Stähli, M., Stöckli, R., Strasser, U., Su, H., Suzuki, K., Takata, K., Tanaka, K., Thompson, E., Vesala, T., Viterbo, P., Wiltshire, A., Xia, K., Xue, Y., Yamazaki, T., 2009. Evaluation of forest snow processes models (SnowMIP2). Journal of Geophysical Research: Atmospheres 114. https://doi.org/10.1029/2008JD011063

Salminen, M., Pulliainen, J., Metsämäki, S., Kontu, A., Suokanerva, H., 2009. The behaviour of snow and snow-free surface reflectance in boreal forests:

Implications to the performance of snow covered area monitoring. Remote Sens Environ 113, 907–918. https://doi.org/10.1016/j.rse.2008.12.008

Salomonson, V.V., Appel, I., 2004. Estimating fractional snow cover from MODIS using the normalized difference snow index. Remote Sens Environ 89, 351–360. https://doi.org/10.1016/j.rse.2003.10.016

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.

Takala, M., Luojus, K., Pulliainen, J., Derksen, C., Lemmetyinen, J., Kärnä, J.-P., Koskinen, J., Bojkov, B., 2011. Estimating northern hemisphere snow water equivalent for climate research through assimilation of space-borne radiometer data and ground-based measurements. Remote Sens Environ 115, 3517–3529. https://doi.org/10.1016/j.rse.2011.08.014

Tan, M., Le, Q., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR, pp. 6105–6114.

Tekeli, A.E., Akyürek, Z., Arda Şorman, A., Şensoy, A., Ünal Şorman, A., 2005. Using MODIS snow cover maps in modeling snowmelt runoff process in the eastern part of Turkey. Remote Sens Environ 97, 216–230. https://doi.org/10.1016/j.rse.2005.03.013

The MathWorks Inc., 2022. Computer Vision Toolbox version 10.2 (R2022a).

Tieleman, T., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4, 26.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y., 2022. MaxViT: Multi-axis Vision Transformer. pp. 459–479. https://doi.org/10.1007/978-3-031-20053-3_27

Wang, H., Zhang, L., Wang, L., He, J., Luo, H., 2021. An Automated Snow Mapper Powered by Machine Learning. Remote Sens (Basel) 13, 4826. https://doi.org/10.3390/rs13234826

Wang, X., Chen, S., Wang, J., 2020. An Adaptive Snow Identification Algorithm in the Forests of Northeast China. IEEE J Sel Top Appl Earth Obs Remote Sens 13, 5211–5222. https://doi.org/10.1109/JSTARS.2020.3020168

Wang, X., Wang, J., Che, T., Huang, X., Hao, X., Li, H., 2018. Snow Cover Mapping for Complex Mountainous Forested Environments Based on a Multi-Index Technique. IEEE J Sel Top Appl Earth Obs Remote Sens 11, 1433–1441. https://doi.org/10.1109/JSTARS.2018.2810094

Wang, X.-Y., Wang, J., Jiang, Z.-Y., Li, H.-Y., Hao, X.-H., 2015. An Effective Method for Snow-Cover Mapping of Dense Coniferous Forests in the Upper Heihe River Basin Using Landsat Operational Land Imager Data. Remote Sens (Basel) 7, 17246–17257. https://doi.org/10.3390/rs71215882

Wang, Y., Gu, L., Li, X., Fan, X., 2024. An Effective Fractional Snow Cover Estimation Method Using Deep Feature Snow Index. IEEE Geoscience and Remote Sensing Letters 21, 1–5. https://doi.org/10.1109/LGRS.2024.3365792

Wang, Y., Huang, X., Liang, H., Sun, Y., Feng, Q., Liang, T., 2018. Tracking Snow Variations in the Northern Hemisphere Using Multi-Source Remote Sensing Data (2000–2015). Remote Sens (Basel) 10, 136. https://doi.org/10.3390/rs10010136

Wang, Y., Su, J., Zhai, X., Meng, F., Liu, C., 2022. Snow Coverage Mapping by Learning from Sentinel-2 Satellite Multispectral Images via Machine Learning Algorithms. Remote Sens (Basel) 14, 782. https://doi.org/10.3390/rs14030782

Wang, Z., Fan, B., Tu, Z., Li, H., Chen, D., 2022. Cloud and Snow Identification Based on DeepLab V3+ and CRF Combined Model for GF-1 WFV Images. Remote Sens (Basel) 14, 4880. https://doi.org/10.3390/rs14194880

Wu, X., Shi, Z., Zou, Z., 2021. A geographic information-driven method and a new large scale dataset for remote sensing cloud/snow detection. ISPRS Journal of Photogrammetry and Remote Sensing 174, 87–104. https://doi.org/10.1016/j.isprsjprs.2021.01.023

Yin, M., Wang, P., Ni, C., Hao, W., 2022. Cloud and snow detection of remote sensing images based on improved Unet3+. Sci Rep 12, 14415. https://doi.org/10.1038/s41598-022-18812-6

Zeiler, M.D., 2012. ADADELTA: An Adaptive Learning Rate Method.

Zhan, Y., Wang, J., Shi, J., Cheng, G., Yao, L., Sun, W., 2017. Distinguishing Cloud and Snow in Satellite Images via Deep Convolutional Network. IEEE Geoscience and Remote Sensing Letters 14, 1785–1789. https://doi.org/10.1109/LGRS.2017.2735801

Zhang, G., Gao, X., Yang, Y., Wang, M., Ran, S., 2021. Controllably Deep Supervision and Multi-Scale Feature Fusion Network for Cloud and Snow Detection Based on Medium- and High-Resolution Imagery Dataset. Remote Sens (Basel) 13, 4805. https://doi.org/10.3390/rs13234805

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2016. Pyramid Scene Parsing Network.

Zheng, K., Li, J., Ding, L., Yang, J., Zhang, Xucheng, Zhang, Xun, 2021. Cloud and Snow Segmentation in Satellite Images Using an Encoder–Decoder Deep Convolutional Neural Networks. ISPRS Int J Geoinf 10, 462. https://doi.org/10.3390/ijgi10070462

Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2020. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. IEEE Trans Med Imaging 39, 1856–1867. https://doi.org/10.1109/TMI.2019.2959609

APPENDICES

A. Data Loading Function for Training, Validation and Test

```
import os
from PIL import Image
import tifffile as tif
from torch.utils.data import Dataset
import numpy as np

class SadettinDataset(Dataset):
    def __init__(self, image_dir, mask_dir, rgb_dir, transform=None):
        self.image_dir = image_dir
        self.mask_dir = mask_dir
        self.rgb_dir = rgb_dir
        self.transform = transform
        self.images = sorted(os.listdir(image_dir), key=len)
        self.masks = sorted(os.listdir(mask_dir), key=len)
        self.rgbs = sorted(os.listdir(rgb_dir), key=len)

    def __len__(self):
        return len(self.images)

    def __getitem__(self, index):
        img_path = os.path.join(self.image_dir, self.images[index])
        mask_path = os.path.join(self.mask_dir, self.masks[index])
        rgb_path = os.path.join(self.rgb_dir, self.rgbs[index])
        # PIL Image library can only read RGB
        # image = np.array(Image.open(img_path))
        image = tif.imread(img_path)
        mask = np.array(Image.open(mask_path))
        rgb = np.array(Image.open(rgb_path))

        # make augmentation
        if self.transform is not None:
            augmentations = self.transform(image=image, mask=mask, rgb=rgb)
            image = augmentations["image"]
            mask = augmentations["mask"]
            rgb = augmentations["rgb"]

        return image, mask, rgb
```

B. Utility Functions

```
import torch
import torchvision
from torchmetrics import Dice
from matplotlib.colors import ListedColormap
from torch.utils.data import DataLoader
import torch.nn as nn
import matplotlib.pyplot as plt
import numpy as np
import tifffile as tif
from data_load import SadettinDataset

def save_checkpoint(state, filename="checkpoint.pth.tar"):
    torch.save(state, filename)

def load_checkpoint(checkpoint, model):
    print("=> Loading checkpoint")
    model.load_state_dict(checkpoint["state_dict"])

def get_loaders(
    train_dir,
    train_maskdir,
    train_rgbdir,
    val_dir,
    val_maskdir,
    val_rgbdir,
    batch_size,
    train_transform,
    val_transform,
    num_workers,
    pin_memory=True,
):
    train_ds = SadettinDataset(
        image_dir=train_dir,
        mask_dir=train_maskdir,
        rgb_dir=train_rgbdir,
        transform=train_transform
    )

    train_loader = DataLoader(
        train_ds,
```

```
        batch_size=batch_size,
        pin_memory=pin_memory,
        num_workers=num_workers,
        persistent_workers=True,
        shuffle=True
    )

    val_ds = SadettinDataset(
        image_dir=val_dir,
        mask_dir=val_maskdir,
        rgb_dir=val_rgbdir,
        transform=val_transform
    )

    val_loader = DataLoader(
        val_ds,
        batch_size=batch_size,
        pin_memory=pin_memory,
        num_workers=num_workers,
        persistent_workers=True,
        shuffle=True
    )

    return train_loader, val_loader

def check_accuracy(loader, model, device="cuda"):
    num_correct = 0
    num_pixels = 0
    dice_score = 0
    model.eval()

    with torch.no_grad():
        for x, y, rgb in loader:
            x = x.to(device).float()
            y = y.to(device) - 1
            softmax = nn.Softmax(dim=1)
            preds = torch.argmax(softmax(model(x)),axis=1)
            # for accuracy calculations
            num_correct += (preds == y).sum()
            num_pixels += torch.numel(preds)
            # dice score
            dice = Dice(num_classes = 5, average='micro',
mdmc_average='global').to(device)
            dice_score += dice(preds, y)
```

```
    accuracy = (num_correct/num_pixels)*100
    dice_score = dice_score/len(loader)

    model.train()

    return accuracy.item(),dice_score.item()

def save_predictions_as_imgs(loader, model, epoch_num, folder, device="cuda"):
    model.eval()
    for idx, (x, y, rgb) in enumerate(loader):
        x = x.to(device=device).float()
        with torch.no_grad():
            softmax = nn.Softmax(dim=1)
            preds = torch.argmax(softmax(model(x)),axis=1).to('cpu')

        for i in range(len(x)):
            # img = np.transpose(np.array(x[i,:,:,:].to('cpu'), dtype=np.uint8),(1,2,0)) #
rgb
            # img = np.transpose(np.array(x[i,0:3,:,:].to('cpu'), dtype=np.uint8),(1,2,0))
# 12 bands
            img = np.transpose(np.array(rgb[i,:,:,:].to('cpu'), dtype=np.uint8),(1,2,0))
            pred = np.array(preds[i,:,:])
            mask = np.array(y[i,:,:]-1)

            cmap = ListedColormap(["blue", "white", "brown", "grey","black"])
            vmin=0
            vmax=4
            fig , ax =  plt.subplots(1, 3, figsize=(15, 15))
            ax[0].set_title('Image')
            ax[1].set_title('Prediction')
            ax[2].set_title('Mask')
            ax[0].axis("off")
            ax[1].axis("off")
            ax[2].axis("off")
            ax[0].imshow(img)
            ax[1].imshow(pred, cmap=cmap, vmin=vmin, vmax=vmax)
            ax[2].imshow(mask, cmap=cmap, vmin=vmin, vmax=vmax)

            plt.savefig(folder+"/Epoch_"+str(epoch_num)+"_"+"Batch_"+str(idx)+"_"
+str(i)+".png", bbox_inches="tight")
            plt.close()

    model.train()
```

```
def save_images_masks_as_imgs(loader, epoch_num, folder, device="cuda"):

    for idx, (x, y, rgb) in enumerate(loader):
        x = x.to(device=device).float()

        for i in range(len(x)):
            # img = np.transpose(np.array(x[i,:,:,:].to('cpu'), dtype=np.uint8),(1,2,0)) #
rgb
            # img = np.transpose(np.array(x[i,0:3,:,:].to('cpu'), dtype=np.uint8),(1,2,0))
# 12 bands
            img = np.transpose(np.array(rgb[i,:,:,:].to('cpu'), dtype=np.uint8),(1,2,0))
            mask = np.array(y[i,:,:]-1)

            cmap = ListedColormap(["blue", "white", "brown", "grey","black"])
            vmin=0
            vmax=4
            fig , ax =  plt.subplots(1, 2, figsize=(15, 15))
            ax[0].set_title('Image')
            ax[1].set_title('Mask')
            ax[0].axis("off")
            ax[1].axis("off")
            ax[0].imshow(img)
            ax[1].imshow(mask, cmap=cmap, vmin=vmin, vmax=vmax)

            plt.savefig(folder+"/Epoch_"+str(epoch_num)+"_"+"Batch_"+str(idx)+"_"
+str(i)+".png", bbox_inches="tight")
            plt.close()
```

C. Training Script

```
import torch
import albumentations as Alb
from model import encoder, encoder_weights, in_ch, out_ch
from albumentations.pytorch import ToTensorV2
from tqdm import tqdm
import torch.nn as nn
import segmentation_models_pytorch as smp
import torch.optim as optim
import scipy.io as sio
from utils import (
    load_checkpoint,
    save_checkpoint,
    get_loaders,
    check_accuracy,
    save_predictions_as_imgs)

# Hyperparameters
LEARNING_RATE = 1e-4
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
BATCH_SIZE = 8
NUM_EPOCHS = 150
CHECK_ACC_FREQ = 1 # check accuracy every CHECK_ACC_FREQ epoch
IMAGE_HEIGHT = 256
IMAGE_WIDTH = 256
PIN_MEMORY = True
NUM_WORKERS = 2
LOAD_MODEL = False
accuracy_init_lim = 88
dice_init_lim = 0.88

# Relative file paths
TRAIN_IMG_DIR = "../../Training_rgb/train_data_nd/"
TRAIN_MASK_DIR = "../../Training_rgb/train_masks/"
TRAIN_RGB_DIR = "../../Training_rgb/train_data/"
VAL_IMG_DIR = "../../Training_rgb/val_data_nd/"
VAL_MASK_DIR = "../../Training_rgb/val_masks/"
VAL_RGB_DIR = "../../Training_rgb/val_data/"

def train(loader, model, optimizer, loss_func, scaler, epoch):
    loss_history = []
```

```
    loop = tqdm(enumerate(loader), total=len(loader))
    for batch_index, (data, targets, rgb) in loop:
        data = data.to(device=DEVICE).float()
        targets = targets.to(device=DEVICE)
        targets = targets.type(torch.long)

        # forward
        with torch.cuda.amp.autocast():
            predictions = model(data)
            # -1 because pytorch is 0 based while classes are 1 based
            loss = loss_func(predictions, targets-1)

        # backward
        optimizer.zero_grad()
        scaler.scale(loss).backward()
        scaler.step(optimizer)
        scaler.update()

        # update loading bar
        loop.set_description(f"Epoch [{epoch+1}/{NUM_EPOCHS}]")
        loop.set_postfix(loss=loss.item())
        loss_history.append([loss.item()])

    return loss_history

def main():
    loss_history = []
    acc_history = []
    dice_history = []

    train_transform = Alb.Compose(
        [
            Alb.HorizontalFlip(p=0.5),
            Alb.VerticalFlip(p=0.5),
            ToTensorV2()
        ]
    )

    val_transforms = Alb.Compose(
        [
            Alb.HorizontalFlip(p=0.5),
            Alb.VerticalFlip(p=0.5),
            ToTensorV2()
```

```
        ],
        additional_targets={"rgb":"image"}
    )

    model = smp.UnetPlusPlus(
        encoder_name=encoder,
        encoder_weights=encoder_weights,
        in_channels=in_ch,
        classes=out_ch).to(DEVICE)

    # loss_func = nn.CrossEntropyLoss()
    # loss_func = smp.losses.DiceLoss(mode="multiclass")
    loss_func = smp.losses.FocalLoss(mode="multiclass")
    optimizer = optim.Adam(model.parameters(), lr=LEARNING_RATE)

    train_loader, val_loader = get_loaders(
        TRAIN_IMG_DIR,
        TRAIN_MASK_DIR,
        TRAIN_RGB_DIR,
        VAL_IMG_DIR,
        VAL_MASK_DIR,
        VAL_RGB_DIR,
        BATCH_SIZE,
        train_transform,
        val_transforms,
        NUM_WORKERS,
        PIN_MEMORY
    )

    if LOAD_MODEL:
        load_checkpoint(torch.load("checkpoint.pth.tar"), model)

    scaler = torch.cuda.amp.GradScaler()
    accuracy_best = accuracy_init_lim
    dice_score_best = dice_init_lim

    for epoch in range(NUM_EPOCHS):
        loss = train(train_loader, model, optimizer, loss_func, scaler, epoch)
        loss_history.extend(loss)

        if (epoch+1) % CHECK_ACC_FREQ == 0:
            # check accuracy and dice score
            accuracy, dice_score = check_accuracy(val_loader, model,
device=DEVICE)
```

```
            acc_history.append([accuracy])
            dice_history.append([dice_score])
            if accuracy > accuracy_best and dice_score > dice_score_best:
                # print examples to a folder
                # save_predictions_as_imgs(
                #     val_loader, model, epoch+1, folder="saved_images/validation",
device=DEVICE
                # )
                # save model
                checkpoint = {
                    "state_dict": model.state_dict()}
                save_checkpoint(checkpoint)
                accuracy_best = accuracy
                dice_score_best = dice_score
                print(f"Accuracy: %{accuracy:.3f}")
                print(f"Dice score: {dice_score:.5f}")

            sio.savemat('LossAccDice.mat', {'loss':loss_history,
                                            'accuracy':acc_history,
                                            'dice_score':dice_history})

if __name__ == "__main__":
    main()
```

D. Model Parameters

```
encoder = "tu-maxxvit_rmlp_small_rw_256" # change accordingly to the network
encoder_weights = "imagenet"
in_ch = 3 # change accordingly to input
out_ch = 5 # number of classes
```

E. Testing Script

```
import torch
import albumentations as Alb
import segmentation_models_pytorch as smp
from model import encoder, encoder_weights, in_ch, out_ch
from albumentations.pytorch import ToTensorV2
from torch.utils.data import DataLoader
from data_load import SadettinDataset
from utils import (
    load_checkpoint,
    get_loaders,
    check_accuracy,
    save_predictions_as_imgs)

TEST_IMG_DIR = "../../Training_rgb/test_data_nd/"
TEST_MASK_DIR = "../../Training_rgb/test_masks/"
RGB_DIR = "../../Training_rgb/test_data/"

batch_size = 1
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

model = smp.UnetPlusPlus(
    encoder_name=encoder,
    encoder_weights=encoder_weights,
    in_channels=in_ch,
    classes=out_ch).to(DEVICE)


test_transform = Alb.Compose(
    [
        ToTensorV2()
    ],
    additional_targets={"rgb":"image"}
)

test_ds = SadettinDataset(
    image_dir=TEST_IMG_DIR,
    mask_dir=TEST_MASK_DIR,
    rgb_dir=RGB_DIR,
    transform=test_transform
)
```

```
test_loader = DataLoader(
    test_ds,
    batch_size=batch_size,
    pin_memory=True,
    shuffle=False
)

checkpoint = torch.load("checkpoint.pth.tar")
model.load_state_dict(checkpoint["state_dict"])

accuracy, dice_score = check_accuracy(test_loader, model, device=DEVICE)
print(f"Accuracy: %{accuracy:.3f}")
print(f"Dice score: {dice_score:.5f}")
save_predictions_as_imgs(
                test_loader, model, 0, folder="saved_images/test", device=DEVICE
            )
```

F. Image Patch Generator Script

```
import tifffile as tif
import numpy as np

image =
tif.imread("S2B_MSIL2A_20230519T175919_N0509_R041_T13UFA_20230519
T205146_12bands_8bit.tif")

patch_size = 256
length = len(image)
num_patches = np.floor(length/patch_size)

num = 1
for i in range(255,int(num_patches*patch_size),256):
   for j in range(255,int(num_patches*patch_size),256):
      file_name =
'S2B_MSIL2A_20230519T175919_N0509_R041_T13UFA_20230519T205146_'
+ str(num) + '.tif'
      num += 1
      patch = image[i-255:i+1,j-255:j+1,:]
      tif.imwrite(file_name,patch,planarconfig='CONTIG')
```

G. NDVI, NDSI, NDWI Calculator Script

```
import os
import tifffile as tif
import numpy as np

folder_train_12b = "D:/Ders/Tez/Data/Training_rgb/train_data_12b/"
folder_test_12b = "D:/Ders/Tez/Data/Training_rgb/test_data_12b/"
folder_val_12b = "D:/Ders/Tez/Data/Training_rgb/val_data_12b/"

folder_train_nd = "D:/Ders/Tez/Data/Training_rgb/train_data_nd/"
folder_test_nd = "D:/Ders/Tez/Data/Training_rgb/test_data_nd/"
folder_val_nd = "D:/Ders/Tez/Data/Training_rgb/val_data_nd/"

files_in_folder_train_12b = os.listdir(folder_train_12b)
files_in_folder_test_12b = os.listdir(folder_test_12b)
files_in_folder_val_12b = os.listdir(folder_val_12b)

for i in files_in_folder_train_12b:
    image = tif.imread(folder_train_12b + i)
    b11 = np.float32(image[:,:,10])
    b8 = np.float32(image[:,:,7])
    b4 = np.float32(image[:,:,3])
    b3 = np.float32(image[:,:,2])
    # ndvi
    ndvi = (b8 - b4)/(b8 + b4 + 1e-6)
    # ndsi
    ndsi = (b3 - b11)/(b3 + b11 + 1e-6)
    # ndwi
    ndwi = (b3 - b8)/(b3 + b8 + 1e-6)
    # combine
    comb = np.dstack((ndvi, ndsi, ndwi))
    #write to file
    tif.imwrite(folder_train_nd+i,comb,planarconfig='CONTIG')

for i in files_in_folder_test_12b:
    image = tif.imread(folder_test_12b + i)
    b11 = np.float32(image[:,:,10])
    b8 = np.float32(image[:,:,7])
    b4 = np.float32(image[:,:,3])
    b3 = np.float32(image[:,:,2])
    # ndvi
    ndvi = (b8 - b4)/(b8 + b4 + 1e-6)
```

```
    # ndsi
    ndsi = (b3 - b11)/(b3 + b11 + 1e-6)
    # ndwi
    ndwi = (b3 - b8)/(b3 + b8 + 1e-6)
    # combine
    comb = np.dstack((ndvi, ndsi, ndwi))
    #write to file
    tif.imwrite(folder_test_nd+i,comb,planarconfig='CONTIG')

for i in files_in_folder_val_12b:
    image = tif.imread(folder_val_12b + i)
    b11 = np.float32(image[:,:,10])
    b8 = np.float32(image[:,:,7])
    b4 = np.float32(image[:,:,3])
    b3 = np.float32(image[:,:,2])
    # ndvi
    ndvi = (b8 - b4)/(b8 + b4 + 1e-6)
    # ndsi
    ndsi = (b3 - b11)/(b3 + b11 + 1e-6)
    # ndwi
    ndwi = (b3 - b8)/(b3 + b8 + 1e-6)
    # combine
    comb = np.dstack((ndvi, ndsi, ndwi))
    #write to file
    tif.imwrite(folder_val_nd+i,comb,planarconfig='CONTIG')
```

TEZ İZİN FORMU / THESIS PERMISSION FORM

ENSTİTÜ / INSTITUTE

Fen Bilimleri Enstitüsü / Graduate School of Natural and Applied Sciences [X]

Sosyal Bilimler Enstitüsü / Graduate School of Social Sciences [ ]

Uygulamalı Matematik Enstitüsü / Graduate School of Applied Mathematics [ ]

Enformatik Enstitüsü / Graduate School of Informatics [ ]

Deniz Bilimleri Enstitüsü / Graduate School of Marine Sciences [ ]

YAZARIN / AUTHOR

Soyadı / Surname : ÖZEN
Adı / Name : SADETTİN
Bölümü / Department : GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

TEZİN ADI / TITLE OF THE THESIS (**İngilizce** / English) : SNOW COVER DETECTION OVER FORESTED AND MOUNTAINOUS REGIONS FROM REMOTE SENSING IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

TEZİN TÜRÜ / DEGREE**:** **Yüksek Lisans** / Master [X] **Doktora** / PhD [ ]

1. **Tezin tamamı dünya çapında erişime açılacaktır. /** Release the entire work immediately for access worldwide. [X]
2. **Tez iki yıl süreyle erişime kapalı olacaktır.** / Secure the entire work for patent and/or proprietary purposes for a period of **two year. *** [ ]
3. **Tez altı ay süreyle erişime kapalı olacaktır.** / Secure the entire work for period of **six months**. ***** [ ]

***** *Enstitü Yönetim Kurulu Kararının basılı kopyası tezle birlikte kütüphaneye teslim edilecektir.*
A copy of the Decision of the Institute Administrative Committee will be delivered to the library together with the printed thesis.

Yazarın imzası / Signature ..... **Tarih** / Date .....