Sea ice classification using a CNN-Transformer hybrid and AutoIce challenge dataset

- -

Master's Thesis

Malthe Aaholm Esbensen

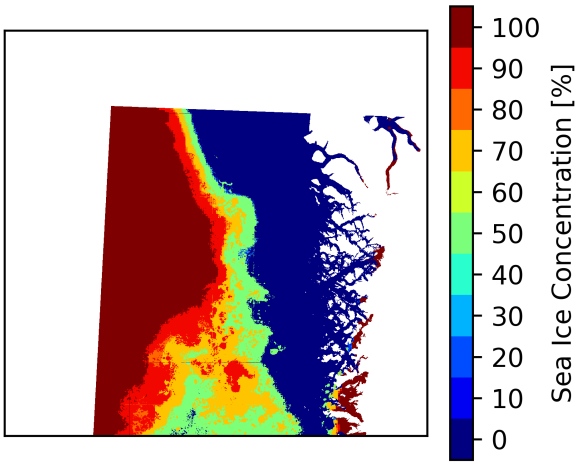

Figure 1: Predicted sea ice concentration

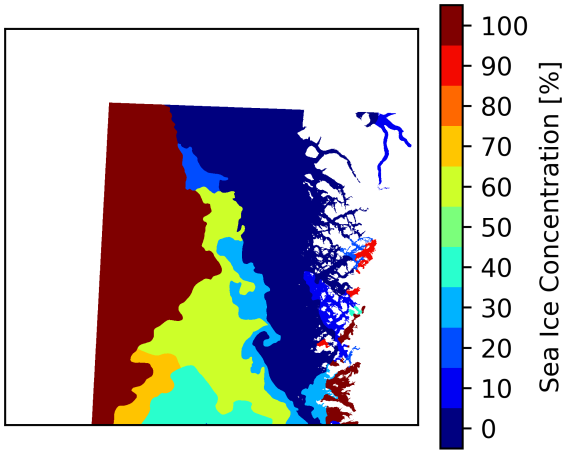

Figure 2: Ground truth for sea ice concentration

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Sea ice classification using CNN-Transformer hybrid and AutoIce challenge dataset

Theme:
Remote Sensing and Deep Learning

Project Period:
Spring Semester 2023

Project Group:
1049B

Participant(s):
Malthe Aaholm Esbensen

Supervisor(s):
Mark Philip Philipsen

Copies: 1

Page Numbers: 50

Date of Completion:
June 1, 2023

Abstract:

This master's thesis is presenting a novel application of the deep learning model TransUNet for segmenting sea ice into three charts. Sea Ice Concentration, Stage of Development, and floe sizes being the output charts of the model. Sea ice plays a critical role in global climate systems, and it is therefore beneficial to have precise measurements of the sea ice extent. Additionally, the shipping industry will benefit from having near real-time updates on the sea ice extent, as it will make navigation in the arctic regions more predictable. To address these challenges, TransUNet, a CNN-Transformer hybrid offering state-of-the-art segmentation due to its local and global awareness, is deployed in the context of sea ice classification. The developed model is selected by training different TransUNets that differs from each other by having changes in the configuration of the transformer. It is found that the number of layers and the patch size has a large impact on performance, and thus the best performing model is selected for further training(R^2 score is used for SIC and $F1$ score is used for SOD and FLOE). The training was ended after 120 epochs, and the combined validation score topped at 92.15%. The results from testing show that the model performs in line with state-of-the-art with a combined score of 86.22%. Additionally, an R^2 score of 86.91% was achieved on SIC, seeing an improvement of 0.57% compared to previous work. This thesis proves the viability of deploying TransUNets as a semantic segmentation method in remote sensing for predicting sea ice charts.

Contents

Preface

This thesis is made as a completion of the master education in Vision, Graphics, and Interactive Systems (VGIS). The undersigned has a bachelor degree in Robotics from Aalborg University and this thesis is the product of the master period, which is the last part of the VGIS study at Aalborg University, Electronics and IT.
Several persons or organizations have contributed with academical or practical support of this thesis. I would like to thank my supervisor Mark Philip Philipsen for his time and support throughout the master period. Additionally, I would like to thank Tore and Till at DMI for their time and expertise into the topic of mapping the arctic sea ice and how deep learning has previously been deployed in the area.

Reading Manual

This report is written for a one-page format. It is intended to be read digitally, which may be required for some of the figures, as the reader may need the capability to zoom in to read the details. Furthermore this project also has a online GitHub repository attached to it. This is used to store the developed software which is open and accessible on GitHub through the following URL: `https://github.com/MaltheEsbensen/TransUNet-seaice`

Aalborg University, June 1, 2023

Malthe Aaholm Esbensen
mesben18@student.aau.dk

Chapter 1

Introduction

Sea ice is frozen ocean water, which grows, melts, and forms in the ocean of the arctic regions. Both local and global climate is affected by the sea ice. With fluctuating temperatures, sea ice extent also varies and can be categorised from multi-year ice to young first-year ice.[1] Sea ice is a crucial component for Earth's ability to regulate it's climate by reflecting sunlight back into space and thereby act as an insulator between the ocean and atmosphere.
The arctic sea ice is affected by several weather conditions. Given an imbalance in these weather conditions from year to year, can result in an increase or decrease in natural occurring phenomenons, such as fluctuations in the area occupied by sea ice. This phenomenon can be decomposed into trend, seasonality, cyclycity and noise. A change to the extent of sea ice can affect humans, ecosystems and wildlife.[2]

Several vulnerable species are depending on the sea ice as they use the ice floes for resting while breeding and hunting. Furthermore, algae and phytoplankton will develop under the floes, and thus the lowest level of the food chain is decreased if the sea ice extent is decreased. This will create a domino effect through the food chain and thus decrease the higher hierarchies.[3]

The negative consequences can be critical for the species living in the arctic, but the positive consequences of a receding ice edge are that new shipping routes emerges. Median prediction is that by 2034 there will be an ice free Arctic route in September, and thus a shipping route over Asia and Europe will emerge. This will decrease the length of the shipping routes from China to Europe by more than 7000 km.[4]

In order to help scientists and policymakers to develop strategies for protecting wildlife as well as guiding ships through the arctic ocean, it is important to monitor and classify the sea ice extent.

Initial Problem Formulation

How can satellite data be used to classify total sea ice concentration (SIC), stage of development (SOD) and floe size (FLOE)?

Chapter 2

Problem Analysis

The following sections seeks to analyse and answer the initial problem formulation by firstly getting an understanding of what arctic sea ice is and what terminology is used to classify sea ice development. After having an understanding of how sea ice is classified, deep learning-based semantic segmentation will be researched, which will give an overview of the state-of-the-art with-in the topic, as well as an analysis of the strengths and weaknesses of the different use cases where the models have been deployed. Lastly, the dataset for the AutoIce challenge will be analysed in order to understand the strengths and weaknesses of using the dataset.

2.1 Arctic Sea Ice

Cavalieri and Parkinson[5] have looked at fluctuations and trends in Arctic sea ice from 1979 to 2010 utilizing passive microwave data from satellites. Both the extent and the area of the Arctic sea ice have decreased, according to the research, with the multiyear ice component (MYI) indicating a bigger decline than the first-year ice component (FYI). According to the authors, the average ice area declined by about 4.5% per decade.

The greatest sea ice losses occurred in the summer and fall, with September seeing the worst losses. According to the research, there is substantial interannual variation in sea ice area, with some years indicating increases. The study also uncovers regional differences in sea ice patterns, with the Beaufort, Chukchi, East Siberian, Laptev, and Kara Seas experiencing the most marked decreases. These areas can be seen in Figure 2.1

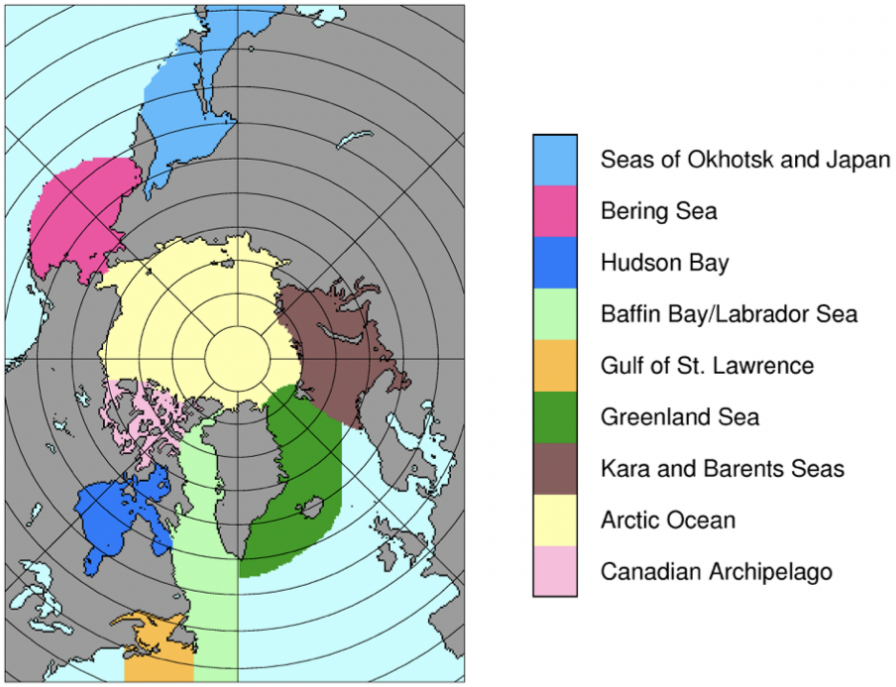

Figure 2.1: Location map with different regions highlighted. [5]

Cavalieri and Parkinson's research highlights the significant changes in Arctic sea ice over the 32-year period and emphasizes the importance of monitoring and comprehending the processes underlying these changes in order to more accurately predict future trends and potential effects on the environment, ecosystems, and human activities in the area.
The importance of having sea ice is the sea ice-albedo feedback mechanism, which is when sunlight is reflected back into space by the ice and snow. This means that the energy of the sun rays is not absorbed in the water and thus does not increase the ocean temperature as much as if the ice had not been there. [6]

However, a positive effect of the decreasing sea ice extent is that new shipping routes emerges. The time and distance of the southern sea route going from China through the Suez canal to Europe can be reduced by one third when the northern sea route will open. This will result in reduced shipping cost and faster delivery times. [7]
In order to monitor when the sea route opens as well as helping ships navigating in the arctic ocean, maps of the sea ice has to be made in near real time, which leads to the next section.

2.2 Sea Ice Charts

Since 1933, the Arctic and Antarctic Research Institute (AARI) has produced sea ice charts of the Northern Sea Route (NSR) as it was a goal to develop the NSR as a regularly operating

transport system.[8] The sea ice charts were originally hand-written, but the process of making the ice charts has over time been digitized. Throughout the years, several methods for obtaining the data for producing the sea ice charts has been used, as seen in Figure 2.2. In the beginning of the time series, the central arctic regions were not covered, as well as there were only recordings of the summer months.

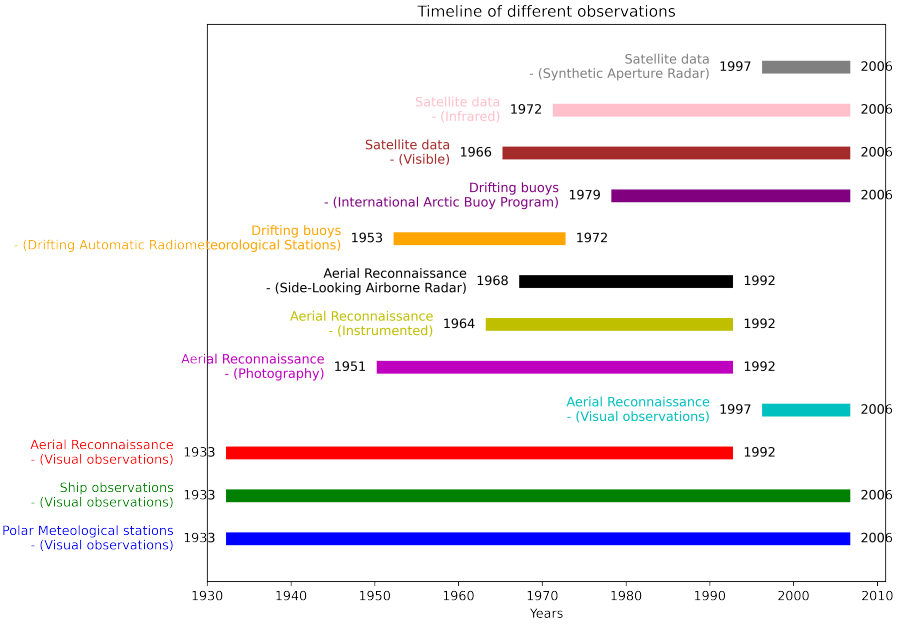

Figure 2.2: A timeline of different observation methods for tracking sea ice.

2.2.1 Sea Ice Classification

The sea ice is currently being classified in three different charts, as seen in Figure 2.3; Sea Ice Concentration (SIC), Stage of Development (SOD) and form of ice (FLOE).[9]

The chart parameters for sea ice is as follows:

- "SIC": The total SIC is the area's percentage of sea ice to open water, SIC is divided into 11 discrete 10 percent-bin classes with percentages ranging from 0% (open water) to 100% (fully covered sea ice).
- "SOD": The SOD can also be thought of as the type of sea ice, which serves as a representation for the sea ice's thickness, or how simple it is to cross. The parameter has 5 classes, 0 being open-water. There are five categories of ice are: new ice, young ice, thin first-year ice, thick first-year ice, and multi-year ice.
- "FLOE": The floe size has six parameters that describe how big or continuous the sea ice chunks are: Open water is represented by 0, cake ice by 1, small floe by 2, medium floe

by 3, big floe by 4, vast floe by 5, and bergs by 6, which are different types of icebergs and glacier ice.

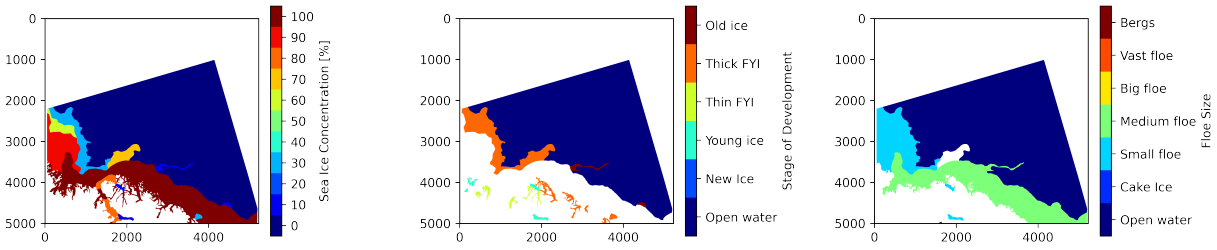

Figure 2.3: Ground truths of SIC, SOD, and FLOE.

2.2.2 Remote Sensing

As stated above, satellite data has been used since 1966, of which the Advanced Microwave Scanning Radiometer (AMSR) and Synthetic Aperture Radar (SAR) are of interest, as AMSR and SAR are the sensors used for remote sensing of sea ice concentration.[10]

Before going in depth with the two sensor types and their data, it is beneficial to first delve into the electromagnetic spectrum and electromagnetic radiation and its uses.

Electromagnetic Radiation

The continuous range of wavelengths known as the electromagnetic spectrum include a variety of electromagnetic waves, as seen in Figure 2.4.[11] Depending on the wavelength and frequency of the waves, the EM spectrum is often divided into portions, each of which has distinct characteristics and uses. [12]

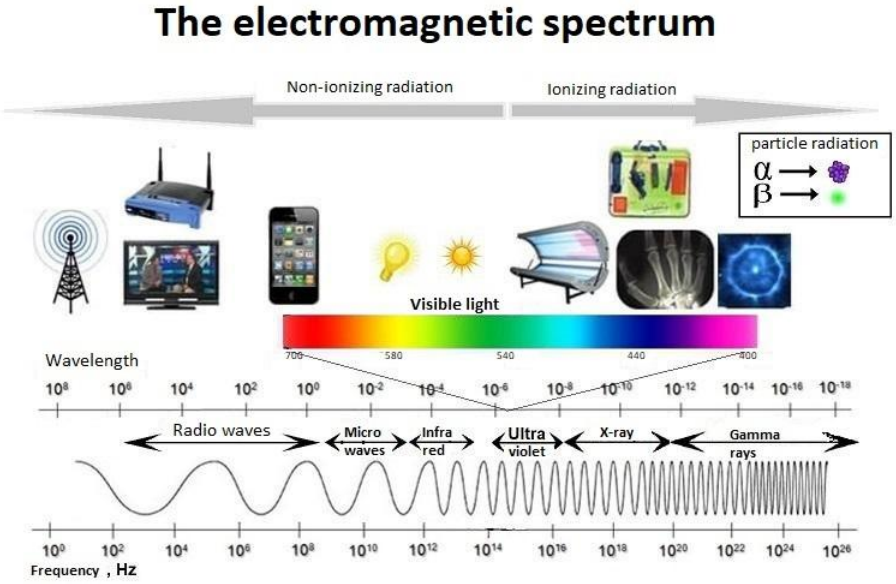

Figure 2.4: The electromagnetic spectrum visualized.[13]

Electromagnetic radiation is the term used to describe the emission and propagation of energy in the form of electromagnetic waves. This transfer of energy occurs when electrons, which are electrically charged particles, oscillate or accelerate.[14] The uses of radiation is depending on the radiation's strength and frequency.

Microwave radiation is a region of the electromagnetic spectrum that is often defined by wavelengths between 1 millimeter and 1 meter and frequencies between 300 MHz and 300 GHz, which makes it possible to "see through" the clouds, as the longer wavelengths of microwaves can penetrate through cloud cover. [15] [16] Microwave remote sensing is the practice of using microwave radiation to gather information about the Earth's surface and atmosphere. Microwave remote sensing is separated into passive and active microwave remote sensing, this difference can be seen in Figure 2.5. [17]

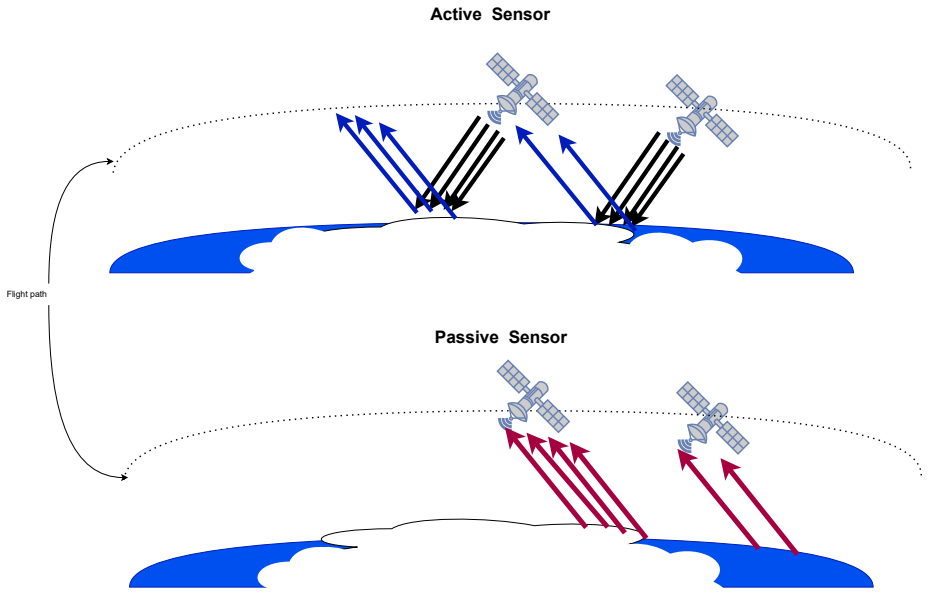

Figure 2.5: Figure showing the difference between active and passive radar.

The practice of detecting naturally existing microwave radiation from the Earth's surface and atmosphere is known as passive microwave remote sensing. [16] This radiation can provide insight into information on a number of geophysical factors, such as sea surface temperature, and atmospheric water vapor content. The Earth's microwave radiation is continually observed by passive microwave sensors, which are carried onboard satellites, such as GCOM-W. [18]

On the other hand, active microwave remote sensing involves emitting microwaves towards the Earth's surface and then observing the backscattered signals, using radars.[19] This makes it possible to identify characteristics such as different snow cover products and surface roughness. Active microwave remote sensing devices like radars provide high-resolution data in most situation, and radar remote sensing instrument consists of both a transmitter and a receiver of energy at the wavelength of interest, such as the Sentinel-1.[20][21] As a result, radars are excellent for keeping track of dynamic processes and surface characteristics.

Advanced Microwave Scanning Radiometer

The passive microwave instrument, developed by Japan Aerospace Exploration Agency (JAXA), was launched in 2012 and is a part of the Global Change Observation Mission-Water (GCOM-W) satellite mission. The GCOM-W mission is used to capture observations of a range of water cycles such as precipitation, sea ice concentration, soil moisture and sea surface temperature, and was developed in order to give scientists an improved knowledge of water cycles and climate change.[22]

The AMSR2 measures properties of the Earth's surface and atmosphere which is captured by measuring microwave radiation from 7 frequency bands betweeen 6.9 to 89 GHz, of which both horizontal and vertical polarization is recorded.[23] AMSR2 has a Spatial resolution that is dependant on the frequency which are:

- 6.925GHz (35kmx61km)
- 7.3GHz (35kmx61km)
- 10.65GHz (34kmx41km)
- 18.7GHz (14kmx22km)
- 23.8GHz (15kmx26km)
- 36.5GHz (7kmx12km)
- 89.0GHz (3kmx5km)

The AMSR2 channels gives the brightness temperature, which is measured in the unit $Wm^{-2}sr^{-1}Hz^{-1}$, which is watt per square metre per steradian per hertz. [24][25]

Brightness temperature is used as a proxy for the intensity of the microwave radiation, in the context of remote sensing, as brightness temperature is usaually measured in kelvin. However, in the microwave spectrum the two are closely related as seen in the equations 2.1 to 2.3.

$$I_\nu = \frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{kT}} - 1} \tag{2.1}$$

$$T_b = \frac{h\nu}{k}\ln^{-1}\left(1 + \frac{2h\nu^3}{I_\nu c^2}\right) \tag{2.2}$$

When $h\nu << kT$, Rayleigh-Jeans approximation can be used, as can be seen in Figure 2.6, and thus T_b can be written as

$$T_b = \frac{I_\nu c^2}{2k\nu^2} \tag{2.3}$$

where, h is Planck's constant, v is the frequency of interest, c is the speed of light, k is the Boltzman constant, T is the temperature in kelvin, I_v is the intensity of radiation for the frequency of interest, and T_b is the brightness temperature.

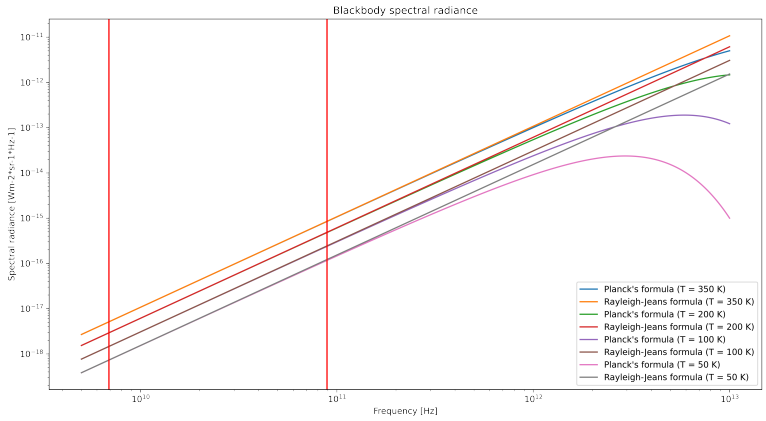

Figure 2.6: Planck's law and Rayleigh-Jeans approximation in the relevant frequencies. The red vertical lines shows the interval from 6.9 GHz to 89 GHz, as these are within the range of the AMSR2 data.

Synthetic Aperture Radar

Synthetic Aperature Radar (SAR) is a technique used in remote sensing that utilizes radar to capture images of the Earth's surface. SAR works by transmitting a microwave beam from a satellite or airplane and then measure the backscatter signals received from the ground. If the signal is weak, meaning the sensor is not receiving much backscatter, the surface is smooth, and the stronger the signal is the rougher the surface is. These signals can then be processed and create a high-resolution image of the Earth's surface. [26]

The method artificially extents the antenna aperture by capturing the backscatter as the satelitte moves along the flight path, which results in a high-resolution image. This is due to the sensor capturing several backscattring signals from the same location from different positions, which stitched together increases the resolution of the image.[26][27]

There are different modes for SAR, common modes are spotlight, Extra Wide Swath Mode, and Stripmap. Spotlight mode has the radar beam fixed on a area and then tilting the beam as the satellite moves along the flight path, which results in smaller coverage, but higher resolution. Extra Wide Swath Mode and Stripmap on the other hand, covers a much larger area, where the major difference between the two is that Extra Wide Swath Mode allows the satellite to cover a larger area compared to Stripmap, by electronically steering the radar antenna to capture multiple sub-swaths in a series of bursts. The data is then combined to create an extra wide image of the Earth's surface. [28] Lastly, Stripmap is the conventional SAR mode which has the radar antenna fixed to receive the backscatter from the narrow swath as the satellite moves along the flight path.

Table 2.1: Pixel spacing of Extra Wide Swath Mode and Stripmap Mode in ground range distance.[29]

Mode	Pixel spacing	
	Range	Azimuth
Full resolution Level-1 GRD		
Stripmap	3.5m	3.5m
High resolution Level-1 GRD		
Stripmap	10m	10m
Extra Wide Swath	25m	25m
Medium resolution Level-1 GRD		
Stripmap	40m	40m
Extra Wide Swath	40m	40m

The greatest advantage of using Extra Wide Swath mode over Stripmap is that it covers a larger area, which is beneficial for monitoring natural phenomena happening at a large scale. In Table 2.1, it can be seen that there is a trade-off between coverage and Pixel Spacing. Pixel Spacing is the distance between adjacent pixels in an image measured in metres in ground range distance for GRD products.[29]

The satellite mission, Sentinel-1, launched by the European Space Agency (ESA) is a part of the Copernicus Programme for Earth observation. The mission features two satellites, Sentinel-1A and Sentinel-1B, that are utilizing C-band SAR instruments which are designed for monitoring various land and ocean phenomenas, such as land use changes, agriculture, forestry, and marine environment monitoring.[20]

The Sentinel-1 satellites can operate in four different modes, as seen in Figure 2.7, Stripmap, Interferometric Wide Swath, Extra Wide Swath, and Wave[30]. As seen in Table 2.1, these modes provide trade-offs between pixel spacing and coverage, which allows versatile usage for different applications. Interferometric Wide Swath and Extra Wide Swath are both based on TOPSAR, which combines sub-swaths, and lastly, Wave is a spotlight mode. [20][31]

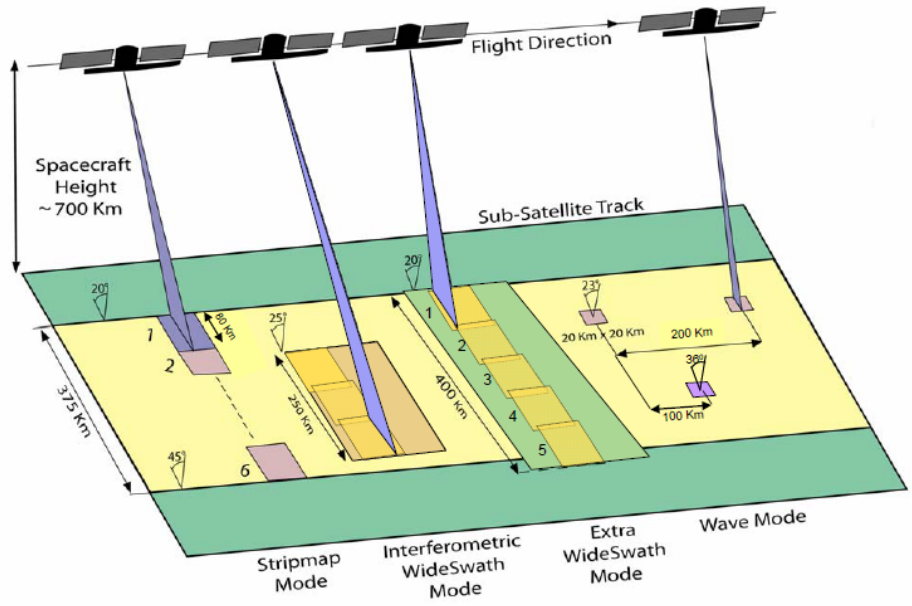

Figure 2.7: The acquisition modes of Sentinel-1 satellite. [32]

2.3 Deep Learning-Based Semantic Segmentation

Fully Convolutional Networks (FCN), seen in Figure 2.8, were introduced by Long et al.[33], and was the first deep learning architecture that could produce dense pixel-wise predictions for semantic segmentation. FCN differs from a conventional CNN by not flattening the data before fully connected layers, and instead have convolutional layers, which allows the output to be the same size as the original input image. The network was trained on annotated images to predict the class of each pixel, allowing for accurate semantic segmentation, and achieved state of the art performance on the PASCAL dataset.

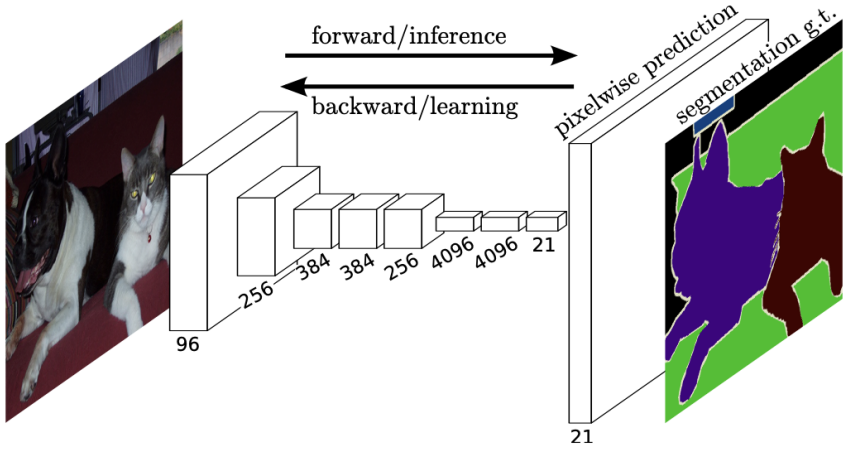

Figure 2.8: The model architecture of FCNs.[33]

SegNet, seen in Figure 2.9, was proposed by Badrinarayanan et al. [34] and introduced the encoder-decoder architecture for semantic segmentation. The encoder part of the network performs feature extraction, much like a regular backbone of a CNN. However, the decoder part upsamples the feature maps and the final output is a pixel-wise prediction. SegNet uses max-pooling in the backbone, and have a corresponding number of upsampling steps, to maintain the spatial information in the input image.

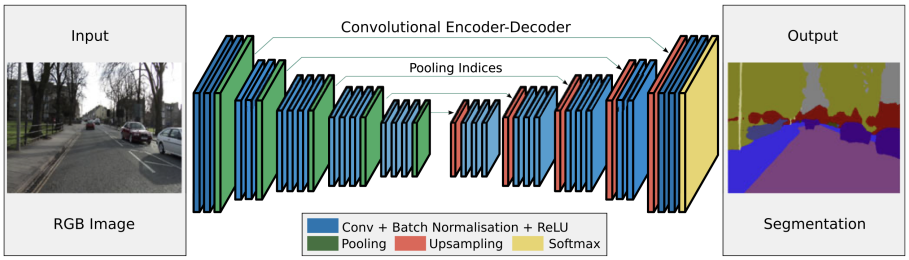

Figure 2.9: The model architecture of SegNet. [34]

The following models will only be described briefly in this section, as they will be further explained in chapter 3.

UNet introduced by Ronneberger et al.[35] build ontop of the SegNet, by maintaining the encoder-decoder architecture, but introducing skip connections by appending a copy of the feature maps from the encoder layers to the decoder layers. This allows for enhancing the segmentation performance and keep the spatial information.

Vision Transformers (ViTs) were enhanced by Dosovitskiy et al.[36], ViTs are a type of trans-

former network, which are widely used in Natural Language Processing, but has been adopted to image classification. Transformers have been tried to replace the CNN backbone of of already existing encoder-decoder architectures for semantic segmentation. The transformer encoder produces a set of features that can be used for pixel-wise prediction using a multi layer perceptron.

TransUNet presented by Chen et al.[37] and is an example of a U-Net architecture, where the CNN backbone has been replaced by a CNN-Transformer Hybrid. TransUNet has achieved state-of-the-art performance on the datasets Synapse multi-organ segmentation dataset and Automated cardiac diagnosis challenge. A key adjustment compared to the original UNet is that the receptive field is enlarged, allowing the TransUNet to extract long-range features as well, which improves the accuracy performance in applications where such features are present.

2.4 Semantic Segmentation for Sea Ice

AI4SeaIce is a continuous project by DMI[38], which develops models for classifying sea ice using remote sensing data. The AI4SeaIce model uses a U-Net architecture with skip connections to capture both local and global information in SAR imagery, which is critical for accurate sea ice concentration charting. The AI4SeaIce model is a continuation of their work on fusing SAR imagery from sential-1 and AMSR2 data.[10] Their latest work involves using a UNet with [16, 32, 64, 64] filters for their layers in the model. Additionally, data augmentation has been used, to create more variation in the data, as well as batch normalization layers to improve generalization.
The paper demonstrates how sea ice concentration can be classified using remote sensing data. The AI4SeaIce model classifies sea ice concentration using a scale from 0 to 100% with 10 step increments of 10%. The ground truths are provided by sea ice chart form DMI. R^2-score is used to evaluate the results, as R^2-score does not penalize 10% misclassifications as much as a 100% misclassifications, these give an R^2-score on test data between 69.61% to 86.34%. Additionally, they investigate several parameters for finetuning.

Another model was introduced by Zhang et. al in 2022[39], which was trained to classify the stage of development into 4 classes using the SAR images HV-polarization,VV-polarization, and double-polarization. The test accuracy is found from three scenes, and get accuracies of 95.37%, 95.66%, and 95.85%.

A model trained on the same dataset as AI4SeaIce, called E-MPSPNet[40], seeks to classify sea ice concentration < 1/10 as Open Water, ice concentration 1–3/10 as Very Open Drift Ice, sea ice concentration 4–6/10 as Open Drift Ice, and concentration of 7–8/10 as Close Drift Ice. The proposed model in the paper achieved an accuracy of 94.2%, F-score of 0.930, and MIoU of 0.892.

2.5 Dataset for Autoice Challenge

The AI4Arctic Sea Ice Challenge Dataset[9] is a gathering of SAR, AMSR2, ERA5. SAR, seen in Figure 2.11, and AMSR, seen in Figure 2.12, has already been investigated, and will therefore not be described in this section. ERA5, seen in Figure 2.10 is a dataset consisting of global atmospheric reanalysis created by the European Centre for Medium-Range Weather Forecasts (ECMWF) and is providing environmental variables. ERA5 features various variables such as air temperature, skin temperature, wind speed and direction, precipitation, and humidity. The ERA5 dataset is covering the entirety of Earth's surface, with a resolution of 0.25 degrees (31 km).

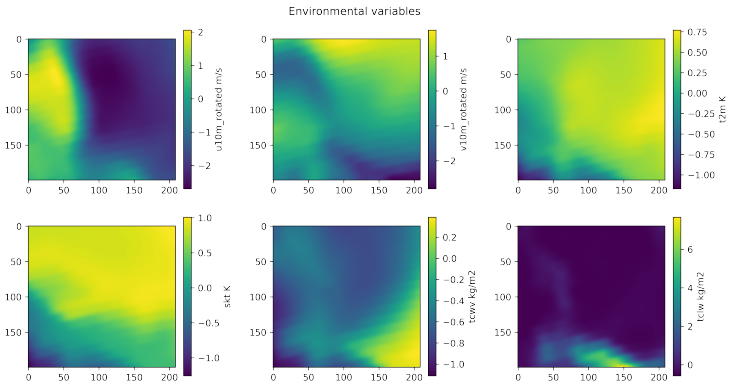

Figure 2.10: Environmental variables.

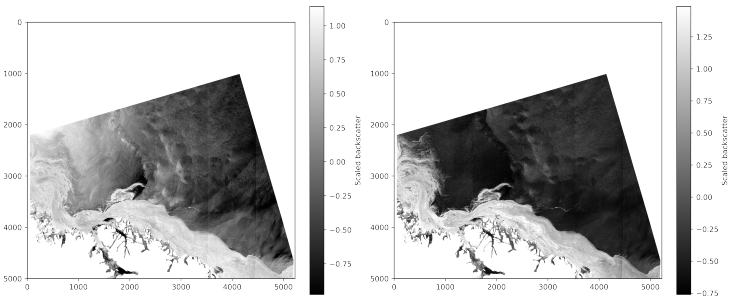

Figure 2.11: SAR imagery

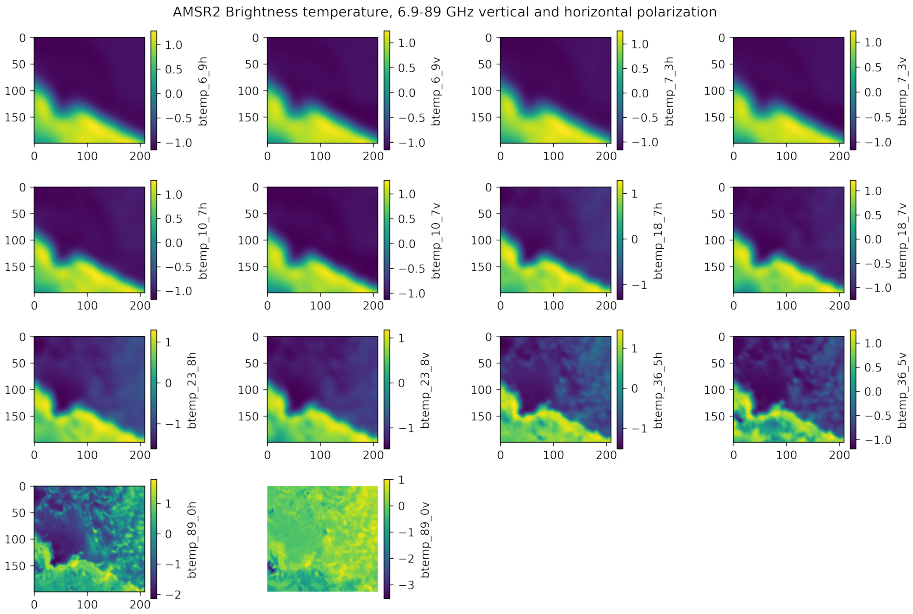

Figure 2.12: AMSR2 channels.

The spatial resolution is 400 square kilometres and originally the SAR images are 10000x10000 pixels, but is downsampled to 5000x5000 pixels. AMSR2 is given in a lower resolution, to compensate for the coarser resolution from AMSR2, the data is resampled to every 50x50 pixel of the corresponding SAR pixel. The same applies to the ERA5 data, which has been resampled in the same manner. The AMSR2 data is already contained in the netCDF files, and therefore retreived along the SAR images. However, the ERA5 data are gathered with the smallest difference in time to the data from sentinel-1.

The seasonal variation in number of samples taken at each month is seen in Figure 2.13, and shows that there are more samples from August through October than the rest of the months. This can cause problems when generalising the model to the rest of the year, as the sea ice extent is generally lower in these months compared to the rest of the year.

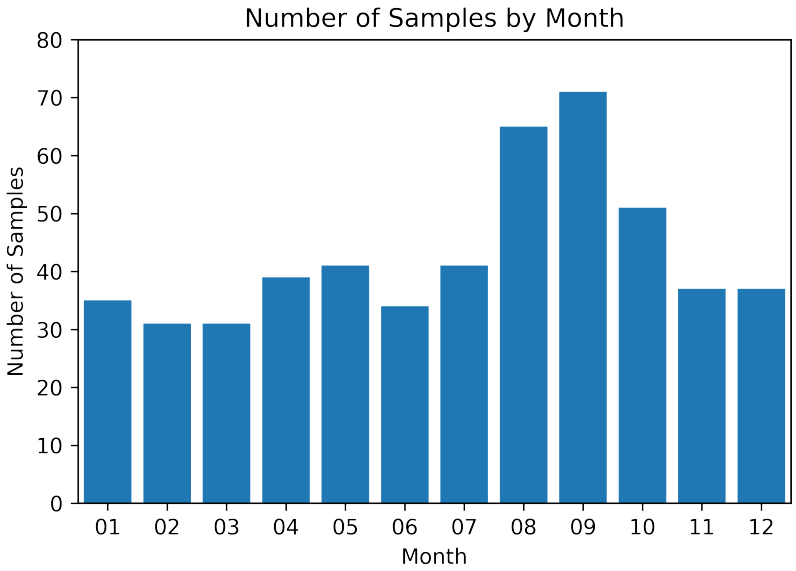

Figure 2.13: Number of samples from different months.

Chapter 3

Methods

This section seeks to investigate the method chosen for classifying sea ice, by describing the components of the latest advances in semantic segmentation. Firstly, by describing the overall architecture of TransUNet, as well as the reasoning for implementing the model for classifying sea ice. Then the section transition into decomposing the model into the key elements of the TransUNet architecture, which are the UNet and Vision Transformer.

3.1 TransUNet

TransUNet is as stated in section 2.3, a novel model for image segmentation, which combines the benefits from a UNet and Vision Transformers. The CNN-Transformer Hybrid, seen in Figure 3.1, integrates the benefits of a UNet to extract the short-range relations in an image and the global attention of a Vision transformer to extract the long-range relations. [37]

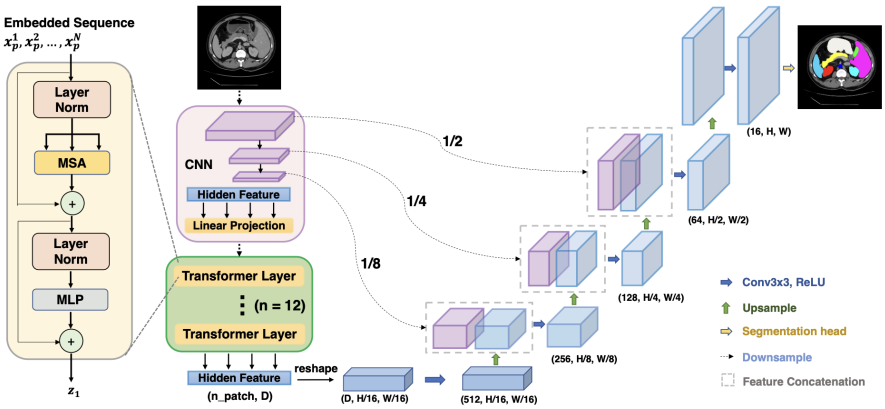

Figure 3.1: The architecture of the TransUNet model. [37]

UNet is a symmetric U-shaped encoder-decoder network, thereby the name, which includes

skip-connections in order to enhance the detail of the segmented image output. Due to the localised feature extraction of convolutional operations of the CNNs, UNets typically sees challenges with modelling the long-range relations in an image.

On the other hand, transformers rely on attention mechanisms, which allows it to extract global contextual information of the image. This feature can be an advantage in some domains, where the context of the target is important, such as Natural Language Processing. In image segmentation the design of transformers, which solely models global context and treats the input as 1D sequences results in low resolution features, and thereby lacks the ability to extract detailed local information. This limits the transformers to tasks that does not need precise spatial localisation.

TransUNet addresses the weaknesses of both models by combining them to make a hybrid model, that takes the benefits from both of them such that it can segment based on global context encoded by the transformer and local features extracted by the CNNs. The architecture consists of a UNet architecture where the encoder is consisting of CNN layers with maxpooling and the final layer is a Vision Transformer. The decoder upsamples the encoded features and for each layer the corresponding feature map from the CNN encoder is concatenated to get the skip-connections.

3.1.1 UNet

UNet, seen in Figure 3.2, is a CNN architecture designed for biomedical image segmentation, and was first introduced in the paper "UNet: Convolutional Networks for Biomedical Image Segmentation" by Ronneberger et al. in 2015. The name "UNet" originates from the shape of the model. The original UNet model uses the following number of feature channels [64, 128, 256, 512, 1024].[35]

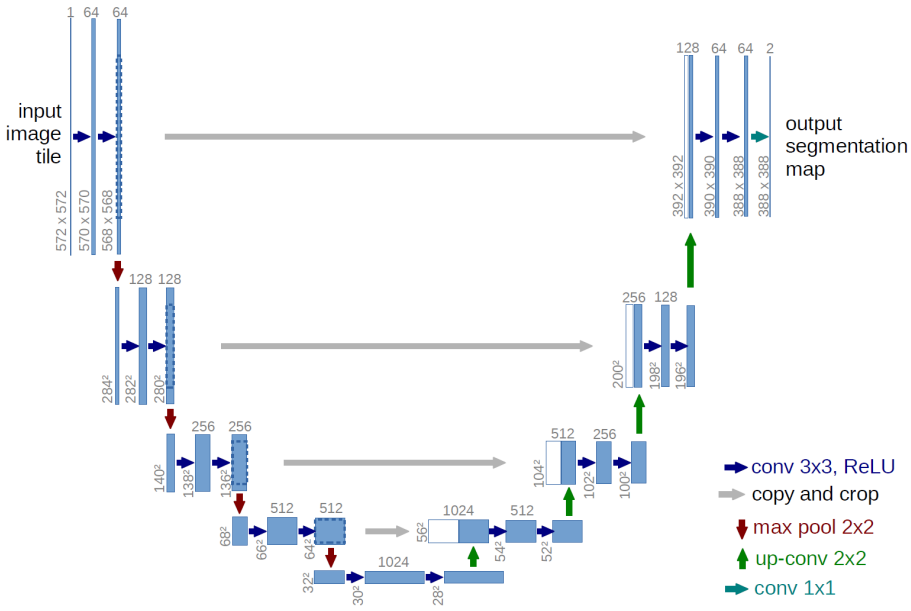

Figure 3.2: The architecture of the original UNet model. [35]

The UNet consists of the following components:

1. Encoder (Contracting Path)

The first part of a UNet consists of an encoder part that follows the structure of a typical CNN. It has a repetitive pattern of two $3x3$ unpadded convolutions followed by a rectified linear unit (ReLU) and a $2x2$ max pooling operation with stride two. The max pooling causes the input channels/feature maps to be downsampled in each layer, and thus capture context in the image. As the network becomes deeper, the number of feature maps are doubled after each max pooling operation.

2. Decoder (Expanding Path)

The second part of the UNet is the decoder, which is localizing the segments in the image. The decoder follows the same structure as the encoder in a reverse manner, meaning that the decoder consists of upsampling-step that first upsamples the feature maps and a $2x2$ convolution that halves the number of feature maps. This is followed by a concatenation of feature channels from the corresponding encoder layer. The upsampling is followed by two $3x3$ convolutions, where the first halves the feature channels and the second maintains the same amount of channels, each followed a ReLU.

3. Skip Connections

The UNet is able to segment based on both short-range and to some extent long-range features. This is due to the skip connections, which was the novelty at the time of publication of the paper. The presence of skip connections means, that that the output of each encoder

layer is connected and concatenated to the corresponding layer in the decoder. This allows the decoder to obtain information from earlier layers in the encoder, which would otherwise become missing, and as thus help reconstruct the spatial information that was lost during the downsampling steps in the encoder.

4. Output Layer

The output layer of the UNet is a $1x1$ convolution that that maps the feature maps to the number of classes that is being segmented.

3.1.2 Vision Transformer

Vision Transformer (ViT)[36], seen in Figure 3.3, was a novel approach to image classification at the time of publishing the paper, which adapts the Transformer architecture normally used for natural language processing, and applies the concept to images.

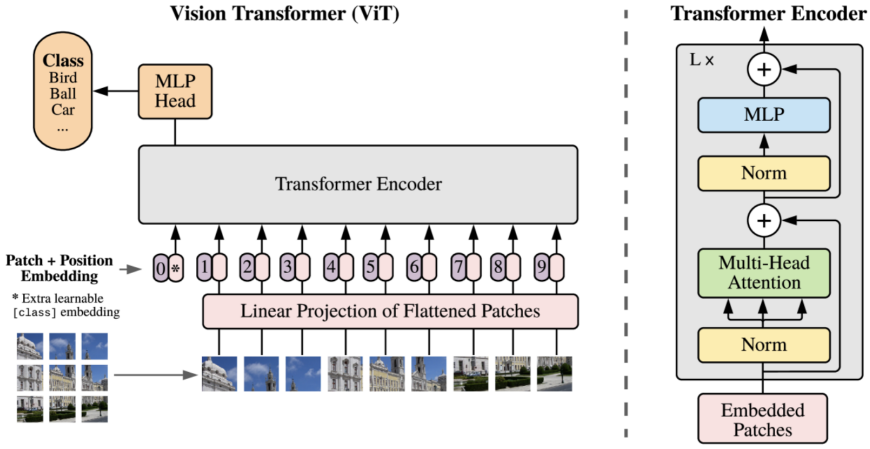

Figure 3.3: The architecture of the original ViT model.

The ViT consists of the following components:

1. Patch Embedding

The first thing that occurs in the ViT is that the image in divided into patches of fixed sizes. Afterwards, each patch is flattened into 1D vectors of size P. As an example, an RGB image divided into 16x16 patches will result in a (N, P) matrix where N is the number of patches and $P = 16 \cdot 16 \cdot 3$, as each color channel is treated individually.

The flattened patches are all transformed into a D dimensional vector by appliyng a linear projection. This is done by multipling the flattened patches with a trainable embedding tensor of shape $(P^2 \cdot channels, D)$. The tensor will learn to project each flattened patch to dimension D linearly. This results in N embedded patches of shape $(1, D)$.

3. Positional Encoding To retain the spatial information, the position of the patches are encoded using positional encoding. The positional encoding tensor is added to the patch embeddings, which learns 1D positional information for the patches.

4. Transformer Encoder

The patch embeddings, with positional encodings, Z_0, is the first input to the L stack of transformer encoders, where L is the number of transformer encoder layers. Each layer takes an input represented as $(N+1, D)$, and the output is of the same shape. The transformer encoder is made of layers of blocks, which consists of a multi-head attention mechanism and a multi layer perceptron, each leaded by a normalization.

4.a Self-Attention Mechanism

The importance of different patches are weighted in the self -attention mechanism when the individual patches are being processed. For each patch, the patch's compatibility with the other patches is determined by transforming the patch embeddings into three spaces; the "query" space, Q, the "key" space, K, and the "value" space, V, by applying different linear transformations.

The query space is a set of query vectors, one for each patch. The query vectors determine the compatibility of each patch, with respect the patch which is currently being processed. The query vectors are used to score the relevance or compatibility of each input with respect to the input being processed.

The key space is a set of key vectors, one for each patch. The key vectors are used together with the query vectors to calculate an attention score, in order to determine the relevance of the patch being processed.

The value space is a set of value vectors, one for each patch. The value vectors are a representation of the actual patches' content. The model uses the calculated attention scores to get a weighted sum of the value vectors, which gives more importance to the value vectors with high attention scores. This weighted sum is the output of the attention mechanism.

If WQ, WK, and WV denotes the weight matrices for these transformations, the transformed vectors are computed as $Q = X_{PE} \cdot WQ, K = X_{PE} \cdot WK$, and $V = X_{PE} \cdot WV$, where X_{PE} are the patch embeddings. Then the attention score, A is computed by $A = softmax(Q \cdot K^T / sqrt(D))$. The output, O, of the self-attention mechanism is the weighted sum for each patch, calculated as $O = A \cdot V$.

4.b Multi-Head Self-Attention

In the paper, a multi-head self attention mechanism is used, which differs from the regular self attention mechanism by stacking the self attention mechanisms. The weighted sum for each head is then concatenated, and the weighted sum for each patch is computed by:

For each head:

$$Q_h = X_{PE} * WQ_h \tag{3.1}$$
$$K_h = X_{PE} * WK_h \tag{3.2}$$
$$V_h = X_{PE} * WV_h \tag{3.3}$$
$$A_h = softmax(Q_h * K_h^T / sqrt(D)) \tag{3.4}$$
$$O_h = A_h * V_h \tag{3.5}$$

Concatenate all O_h along the last dimension:

$$O_{concat} = concat(O_1, ..., O_H) \tag{3.6}$$

Final output:

$$O = O_{concat} * WO \tag{3.7}$$

5. Classification Head

The output from the transformer encoder is a sequence that represents the patches, and can be viewed as a form of [cls] token, and is used as a global image representation. This sequence of patch representations is used to compute the output class probabilities by passing it through a multi layer perceptron.

Extra step for TransUNet: Reshape

In order to make the transformer compatible with the TransUNet, an extra step is introduced, which reshapes the (N, D)-dimensional classification output into a $(D, H/16, W/16)$-dimensional feature map.

3.2 Implementation

The model developed for classifying sea ice, seen in Figure 3.4, has several changes. Most notable is the tiling which is done due to limitations in compute power. Additionally, the number of feature channels have been changed to [16, 32, 64, 128] due to two reasons; firstly, the proposed UNet from AI4Arctic has a reduced number of feature channels in each layer and achieves a good performance. Secondly, due to compute power as well, it was heuristically determined that the used number of feature channels were a good trade-off between the model's accuracy and the computational cost.

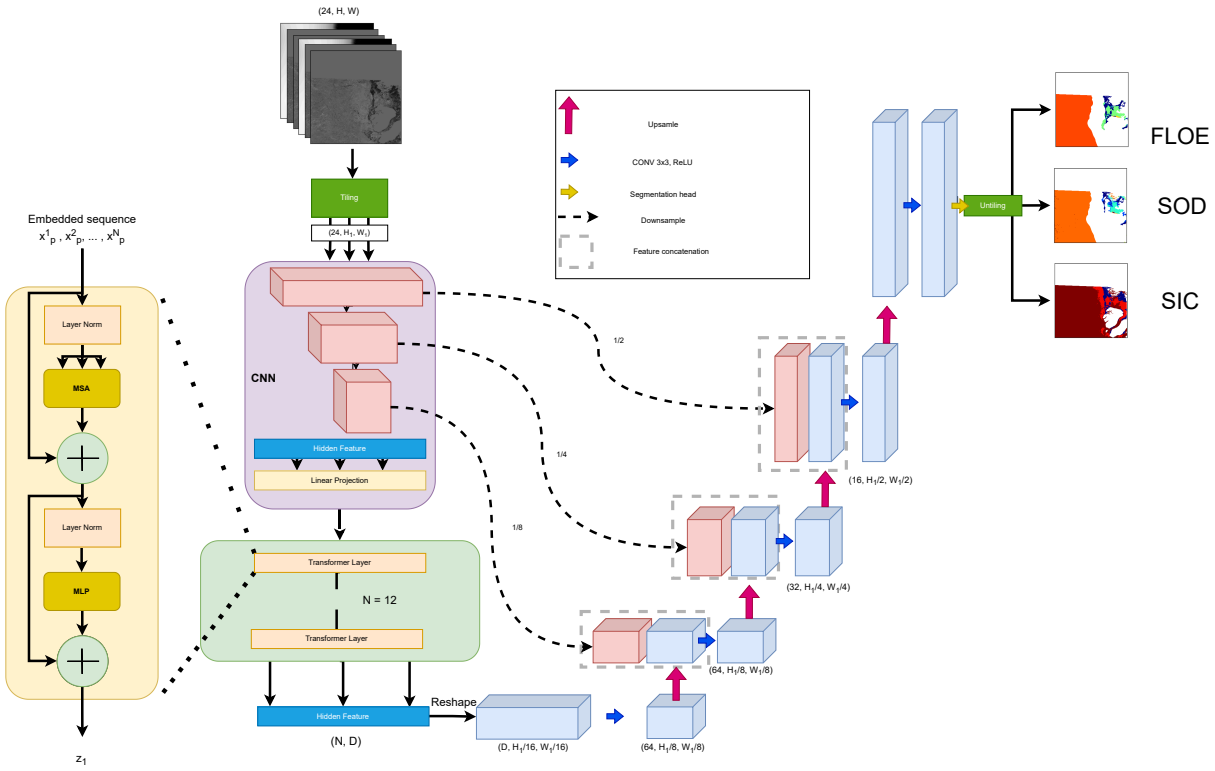

Figure 3.4: The architecture of the TransUNet developed for classifying sea ice.

A summary of the model can be found in Appendix A, where the input shape is set to [24, 512, 512], as this is the patch size used for training.

3.2.1 Tiling, Padding, and Stitching

As the segmentation done on all the data during inference is computationally heavy, the input is cropped into smaller tiles of size [24, 512, 512]. This approach has a drawback, as the input size is not necessarily divisible by 512, which means the last tile in each column and row is not of size [24, 512, 512] but smaller, and thus causes problems in the transformer. To overcome this issue the tiles that are not of size [24, 512, 512] is padded before the transformer encoder. The reason for not padding it before the convolution encoders is because the ratio between padding and real data is smaller after the convolutions and thus the features will not be based on the padding as much as if the ratio had been larger. The last step of the model during inference is to stitch the tiles together. This approach can leave artefacts in the segmentation output, such as misclassifications which can manifest as straight lines.

Chapter 4

Training and Testing

4.1 Training

A Macbook pro with an M1-chip and 8GB ram has been used for training the implemented model. The model selection was done on a 60GB ram cloud CPU, provided by DMI. However, due to the AI4SeaIce competition coming to an end, the access to the cloud CPU expired. This means, that the models in the model selection do not utilize the tiling and stitching which is implemented in the final model, but due to time limitations the model selection process was not possible to recreate with the updated model.

4.1.1 Limitations

Due to limitations in compute power, the data is first split into tiles in order to process a smaller patch at a time, and then after the semantic segmentation the outputs are stitched back together. This will make the receptive field smaller and thus there is a possibility the model is less accurate.
Due to time constraints there is a limitation to the different configurations of the transformer in the model that can be tested, before selecting the best option.

4.1.2 Train Options

Dataset Split

The dataset contains 512 samples, these are then spilt into subsets for training, validation and testing and the ration is seen in Table 4.1. The subsets are randomly selected, but a seed is used in order to have the dataset be divided into the same subsets every time a new model is being trained.

Table 4.1: The dataset split.

Dataset split		
Train	Validation	Test
462	30	20

In Figure 4.1, the number of samples for each month for the validation and testing subset can be seen. The split is not even and the uneven split was first discovered late in the project, which is why the spilt was kept as it is. The uneven split can potentially create bias in the model and will not generalise as well as if the split had been more even.

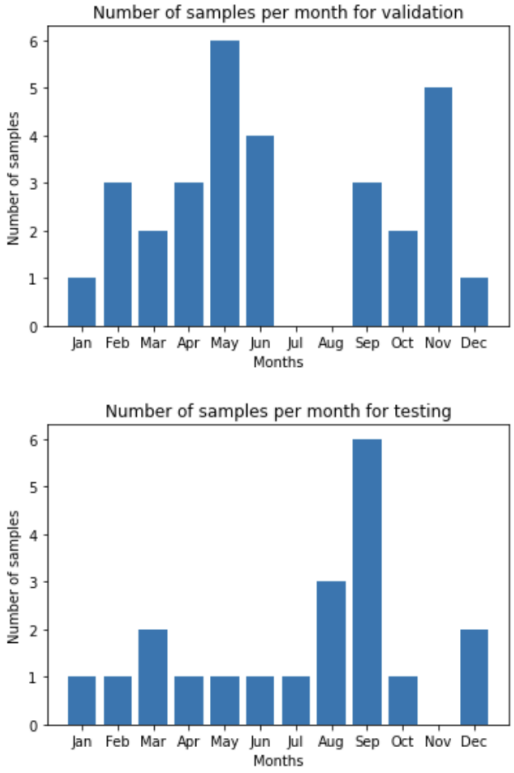

Figure 4.1: The months the samples for validation and testing is taken from.

Hyper Parameters

The hyper parameters have been heuristically chosen to be the ones seen in Table 4.2, as these were found to be a good trade-off between training time and convergence rate of the training loss.

Table 4.2: Hyper parameters for training.

Hyper Parameters					
Batch size	Patch size	Learning rate		Epochs	Epoch length
24	512	Epoch	lr	120	100
		1-39	0.001		
		40-64	0.0001		
		65-69	0.00001		
		70-120	0.0001		

4.1.3 Optimizer

For optimization of the loss function the Adam optimizer is used, which is an algorithm for stochastic gradient descent-based optimization. For the learning rate a manual form of a scheduled learner is being used, as can be seen in Table 4.2.

4.1.4 Evaluation Metrics

Training

Cross Entropy Loss
During training, cross entropy loss[41] is being used as the loss function for updating weights and biases. The input has to be of size $(minibatch, C, d_1, d_2, ..., d_K)$, and the loss function is defined as:

$$\ell(x, y) = L = \{l_1, ..., l_N\}^T, \qquad l_n = -\sum_{c=1}^{C} w_c \frac{exp(x_{n,c})}{\sum_{i=1}^{C} exp(x_{n,i})} y_{n,c} \tag{4.1}$$

where x is the input, y is the target, w is the weight, C is the number of classes, and N spans the minibatch dimension as well as $d1, ..., dk$ for the K-dimensional case (K=2 in this use case) [41]. If reduction is not 'none', but set to 'mean' as is the case during the training in this project, then;

$$\ell(x, y) = \frac{\sum_{n=1}^{N} l_n}{N} \tag{4.2}$$

Validation

R^2 score
The R^2 score is defined in equation 4.3. R^2 score is also known as coefficient of determination, and is a measure of similarity between two sets of data, which is useful in the case of SIC, as it is a continuous value from 0% to 100%. This means that predicting 50% is better than predicting

0%, when the ground truth is 60%. This similarity, or lack of, between classes should not be penalized equally which is why the R^2 score is used for evaluating SIC.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{4.3}$$

where $\hat{y}_i$ is the predicted value of the i-th sample, y_i is the corresponding true value for total n samples, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(\epsilon)^2$.[42]

$F1$ score
$F1$ score is the harmonic mean between precision and recall and represents both precision and recall in one metric. The highest score is 1 and indicates perfect precision and recall, with 0 being the lowest and is found if either precision or recall is zero. Precision is defined as $Precision = \frac{tp}{tp+fp}$ and recall is defined as $Recall = \frac{tp}{tp+fn}$. The $F1$ score is then found using equation 4.4.[43]

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2tp}{2tp + fp + fn} \tag{4.4}$$

The $F1$ score is used to evaluate SOD and FLOE over accuracy, as there is an uneven class distribution.

Combined Score
The combined score is the weighted average of SIC and SOD being weighted 2 and FLOE being weighted 1. This score is used to determine which model is the best, and the best model is updated in case a higher combined score is reached after validation.

4.1.5 Model Selection

The primary contribution of the work done in this project lies in the addition of the transformer to the network, therefore the transformer configuration will be the determining factor during model selection. The selection will be based on different configurations of the transformer, and training of different configurations will be stopped after 20 epochs to evaluate the best model. The parameters that will be investigated are the transformer layers and patch sizes used in the transformer.

The model selection will be based on the combined score obtained on the validation subset, as it would already have knowledge of the test subset, in case the model selection was based on the combined score obtained on the test subset. The results can be seen in Table 4.3, and from this the selected model is the model featuring 12 layers and uses patch sizes of 16x16.

Table 4.3: Different configurations for the developed model together with the scores after 20 epochs.

Model configurations					
Layers	Patch size	SIC	SOD	FLOE	Combined
4	16	82,545%	77,930%	71,328%	79,624%
6	8	80,028%	66,119%	68,259%	74,373%
8	4	79,133%	67,973%	68,274%	74,393%
8	8	85,244%	78,660%	73,525%	81,689%
12	16	92,003%	84,838%	86,256%	89,135%

4.1.6 Validation Results

The results of the training is seen in Figure 4.2 (a better overview of the results can be found in Appedix B). It is seen that the training and validation losses converges quickly, followed by a slight linear decrease. The same applies to the R^2 score of SIC, which also converges to a steady state at around 92%-93%. The $F1$ scores for SOD and FLOE has a positive trend and thus the combined score increases together with those.

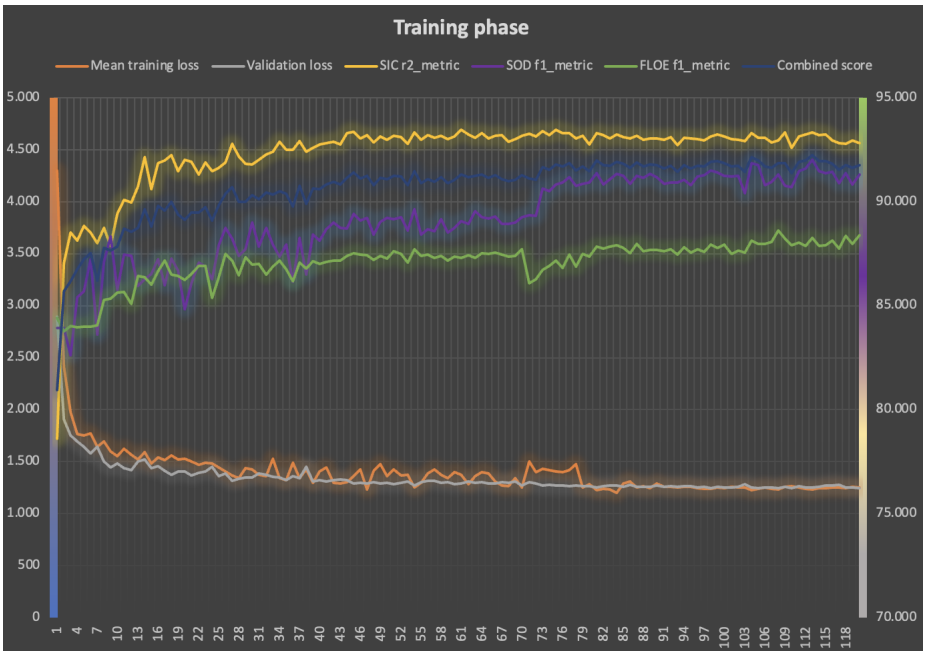

Figure 4.2: The months the samples for validation and testing is taken from.

There is a disruption at the 70th epoch which is most likely due to the learning rate being changed to 10^{-4} from 10^{-5}, it had both a negative and positive impact, as the $F1$ score for FLOE dropped but the $F1$ score for SOD increased.

4.2 Testing

4.2.1 Results

The results from testing can be seen in Table 4.4, and shows that it performs almost as well on FLOE as SIC and SOD, which is unexpected, as the model performs a bit worse on the FLOE $F1$ score during validation. Another unexpected finding is that the score for SIC and SOD is decreased significantly from the validation scores to the test scores.

Table 4.4: Test results

	SIC	SOD	FLOE	Combined
TransUNet SeaIce	86.91	85.34	85.20	86.22

Looking at the confusion matrices, seen in Figure 4.3 to 4.5, of the predictions it can also be seen that there are biases in the predictions. For SIC it can be seen that 0% and 100% is over represented in SIC predictions. For SOD "Open Water" and "Thick FYI" is over represented in the predictions. Lastly, for FLOE it is seen that "Open Water" and "Vast Floe" is over represented in the predictions. It generally shows the same pattern, namely that the model generally overestimates or underestimates in its predictions for the charts, and that it sees difficulties with the intermediate classes.

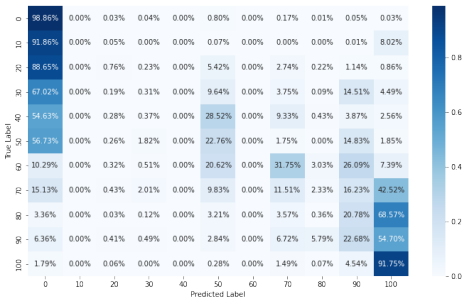

Figure 4.3: Confusion matrix of SIC segmentation.

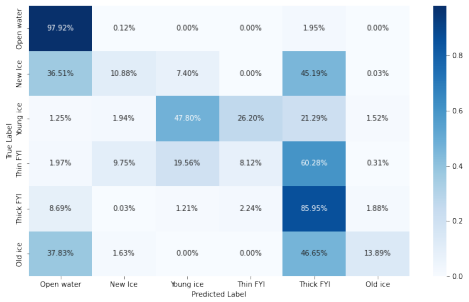

Figure 4.4: Confusion matrix of SOD segmentation.

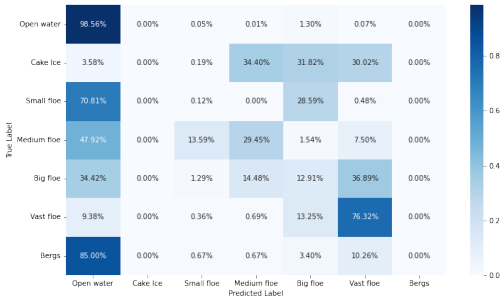

Figure 4.5: Confusion matrix of FLOE segmentation.

Test Examples

To get a better understanding of why the model performs as it does, three examples of inference will be analysed.

The first is seen in Figure 4.6, and shows a sample scene that has a high score and segments the sea ice in all charts rather well, with minor segments being misclassified as open water, and some intermediate misclassifications.

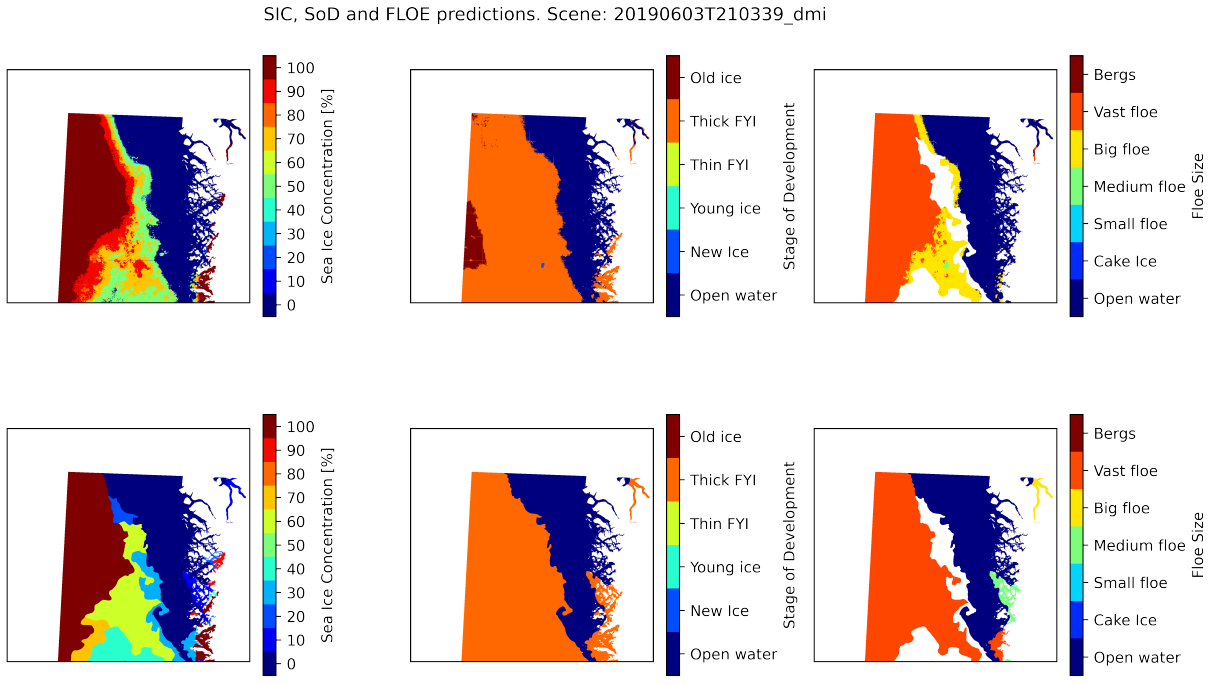

Figure 4.6: An example of a scene with a high score for the segmentation.

The second example is seen in Figure 4.7, and shows a sample scene where the sea ice is underestimated in the three charts, and some sea ice being misclassified as open water.

Though the two first examples have performance issues with regards to the intermediate classes, the sea ice edge is generally segmented well.

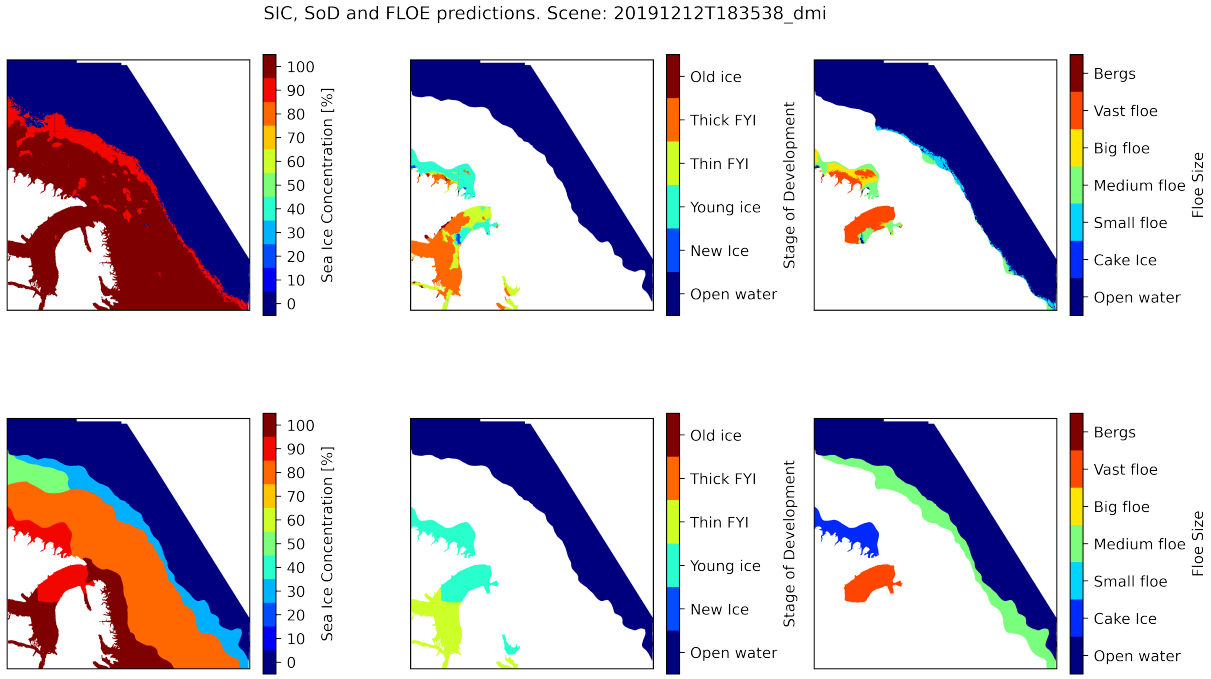

Figure 4.7: An example of a scene where the model underestimates the SIC and SOD but overestimates FLOE in some area and underestimates FLOE in other areas.

The last example can be seen in Figure 4.8, which shows an extreme case of poor performance with regards to the segmentation. This might be due to the nature of the ground truth labelled data, which is sea ice charts made using large polygons, whereas the TransUNet makes pixel-wise predictions for classifying the sea ice. Meaning that the segmented sea ice charts can potentially be more accurate than the "ground-truth" data. However, this needs to be verified by a sea ice expert in order to be sure which is more accurate.

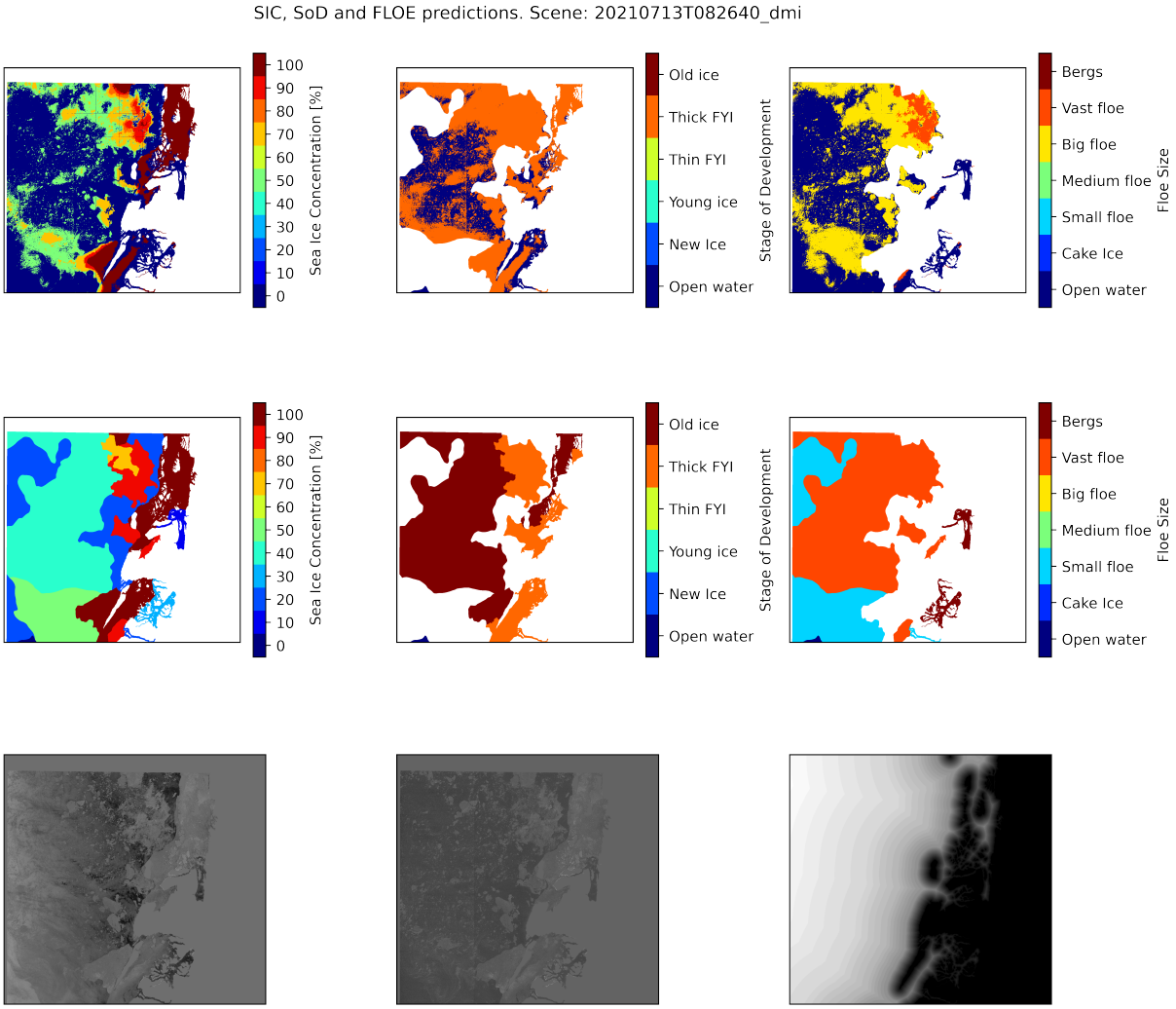

Figure 4.8: An example of a scene with a low score for the segmentation.

4.2.2 AutoIce Challenge Results

The results cannot be directly compared, as the test data used to test the TransUNet is not the same as the test data used in the challenge. However, it can give an idea of how the TransUNet generally compares to other models.

The distribution of months for the AutoIce challenge test set can be seen in Figure 4.9, and differs from the test data used on the TransUNetSeaIce. As it is having most samples taken from July, it could potentially be a problem for the TransUNetSeaIce as samples from July is not present in the validation subset.

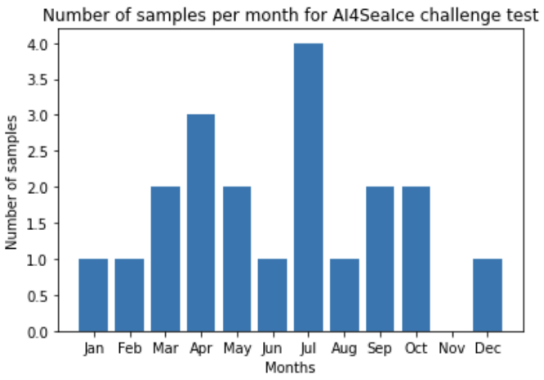

Figure 4.9: The months the samples for AutoIce challenge test dataset is taken from.

As seen in Table 4.5, it is shown that the TransUNetSeaIce achieves a good performance compared to the other contestants. TransUNetSeaIce scores the highest $F1$ score for FLOE, but falls short on the other charts compared to the team Major who won the competition. They used the baseline model, which is a standard UNet[44]. They contributed to the advances by investigating different hyperparameters, and by looking into the dataset in order to have the validation subset closesly resemble the test subset by using the same regions and have the two scenes of a region being close in time.

Table 4.5: Comparison of scores of the top contestants of the AutoIce Challenge.

Teams	SIC (R^2)	SOD ($F1$)	FLOE ($F1$)	Combined score
Major	**92.02**	**88.61**	70.70	**86.39**
TransUNet SeaIce	86.91	85.34	**85.21**	86.22
PWGSN	89.70	76.94	79.12	82.48
crissy	85.34	80.26	74.66	81.17

As seen in Table 4.5, not one solution is the best, but they have different strengths. One solution is overall performing better than others' solution, which does not necessarily mean that the solution is predicting all charts than the others.

Chapter 5

Discussion

5.1 Comparison to Related Work and AutoIce challenge winner

The AI4SeaIce UNet developed by DMI managed to get an R^2 score for SIC laying in the range of 69.61% to 86.34%, whereas the TransUNetSeaIce performs better, as an R^2 score of 86.91% was acheived for SIC.

Compared to the model trained by Major, the TransUNet falls short during testing. However, it should be noted that Major found validation scenes that were from the same regions with timestamps close to the data in the test set. This will cause the model to fit well to the validation set, and thus also fit to the test set as the two are closely correlated and might develop a model that is not generalisable.

5.2 Future Work

For future work it could prove beneficial to try more model configurations, as those investigated are heuristically found. So more possible solutions to the problem could give a broader insight into what proves to work rather than hypothesising the best configurations to test. This is a time consuming task, that would either require more processing power or more time to carry out the task.

Another improvement to the work carried out in this project would be to go more into detail with the data, and split the data into training, validation, and testing in a way that emphasizes on having an equal distribution of the months the samples are taken from. Another solution could be to investigate the class distribution of the data, and then make the data split based on the distribution of classes in the samples, such that it is not biased towards classifying the segments as open water or 100% ice, as for the case of SIC in the TransUNet developed in this project.

As an alternative to using tiling, padding, and stitching it could potentially lead to a better model if the semantic segmentation was done directly on the whole scene instead of one tile of

the scene at a time. Though it did not manifest itself significantly, it still had an influence on the outcome during inference, and thus it could be interesting to see how the model performs without the extra step of tiling and stitching, as the receptive field will be increased.
Lastly, in order to be able to directly compare the model to the other contestants of the AutoIce challenge, the reference data for the test set for by the challenge should be used to test the model once the reference data has been published and processed for testing.

5.3 Conclusion

This thesis set out to answer the following hypothesis ***How can satellite data be used to classify total sea ice concentration (SIC), stage of development (SOD) and floe size (FLOE)?***

To accomplish this, different deep learning algorithms for semantic segmentation was investigated and it was hypothesised that the TransUNet would be a good alternative to the work carried out by DMI. The TransUNet was trained using different configurations for the transformer and the best model after 20 epochs was selected. The model with 12 layers and using 16x16 patches in the transformer was chosen and trained for an additional 100 epochs. The best scores during validation was 93.29% R^2 score for SIC, 91.84% $F1$ score for SOD, 88.14% $F1$ score for FLOE, and 92.14% combined. For testing the results were found to be 86.91% for SIC, 85.34% for SOD, 85.20% for FLOE, and 86.22% combined, which was not enough to win the AutoIce challenge. However, it did prove to be better than the UNet proposed by DMI for segmenting the sea ice concentration by 0.57%.

Bibliography

[1] Delphine Lannuzel et al. "The future of Arctic sea-ice biogeochemistry and ice-associated ecosystems". In: *Nature Climate Change* 10.11 (2020), pp. 983–992.

[2] Julienne C Stroeve et al. "The Arctic's rapidly shrinking sea ice cover: a research synthesis". In: *Climatic change* 110.3-4 (2012), p. 1005.

[3] Eddy Bekkers, Joseph F Francois, and Hugo Rojas-Romagosa. "Melting ice caps and the economic impact of opening the Northern Sea Route". In: *The Economic Journal* 128.610 (2018), pp. 1095–1127.

[4] Eric Post et al. "Ecological consequences of sea-ice decline". In: *science* 341.6145 (2013), pp. 519–524.

[5] Donald J Cavalieri and Claire L Parkinson. "Arctic sea ice variability and trends, 1979–2010". In: *The Cryosphere* 6.4 (2012), pp. 881–889.

[6] Judith A Curry, Julie L Schramm, and Elizabeth E Ebert. "Sea ice-albedo climate feedback mechanism". In: *Journal of Climate* 8.2 (1995), pp. 240–247.

[7] Eddy Bekkers, Joseph F Francois, and Hugo Rojas-Romagosa. "Melting ice caps and the economic impact of opening the Northern Sea Route". In: *The Economic Journal* 128.610 (2018), pp. 1095–1127.

[8] Andrew R Mahoney et al. "Observed sea ice extent in the Russian Arctic, 1933–2006". In: *Journal of Geophysical Research: Oceans* 113.C11 (2008).

[9] A. Stokholm (DTU/ESA) N. Longépé (ESA) M. B. Kreiner (DMI). "AI4Arctic Sea Ice Challenge Dataset User Manual". In: (2022).

[10] David Malmgren-Hansen et al. "A convolutional neural network architecture for Sentinel-1 and AMSR2 data fusion". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.3 (2020), pp. 1890–1902.

[11] Bahaa EA Saleh and Malvin Carl Teich. *Fundamentals of photonics*. john Wiley & sons, 2019.

[12] Sindhuja Sankaran and Reza Ehsani. "Introduction to the electromagnetic spectrum". In: *Imaging with Electromagnetic Spectrum: Applications in Food and Agriculture* (2014), pp. 1–15.

[13] University of Bergen. *The Electromagnetic Spectrum*. Accessed: 2023-05-29. 2022. URL: https://www.uib.no/en/hms-portalen/75292/electromagnetic-spectrum.

[14] John David Jackson. "Classical electrodynamics third edition". In: (1998).

[15] E.G. Njoku. "Passive microwave remote sensing of the earth from space—A review". In: *Proceedings of the IEEE* 70.7 (1982), pp. 728–750. DOI: 10.1109/PROC.1982.12380.

[16] David Long and Fawwaz Ulaby. *Microwave radar and radiometric remote sensing*. Artech, 2015.

[17] Fawwaz T Ulaby, Richard K Moore, and Adrian K Fung. "Microwave remote sensing: Active and passive. Volume 3-From theory to applications". In: (1986).

[18] Keiji Imaoka et al. "Global Change Observation Mission (GCOM) for Monitoring Carbon, Water Cycles, and Climate Change". In: *Proceedings of the IEEE* 98.5 (2010), pp. 717–734. DOI: 10.1109/JPROC.2009.2036869.

[19] Merrill Ivan Skolnik. "Introduction to radar systems". In: *New York* (1980).

[20] Ramon Torres et al. "GMES Sentinel-1 mission". In: *Remote sensing of environment* 120 (2012), pp. 9–24.

[21] John Alan Richards et al. *Remote sensing with imaging radar*. Vol. 1. Springer, 2009.

[22] Keiji Imaoka et al. "Status of AMSR2 instrument on GCOM-W1". In: *Earth observing missions and sensors: development, implementation, and characterization II*. Vol. 8528. SPIE. 2012, pp. 201–206.

[23] JAXA. "AMSR2 Level 1A product format specification". In: (2014). URL: https://gportal.jaxa.jp/gpr/assets/mng_upload/GCOM-W/AMSR2_Level1_Product_Format_EN.pdf.

[24] Kebiao Mao et al. "A physics-based statistical algorithm for retrieving land surface temperature from AMSR-E passive microwave data". In: *Science in China Series D: Earth Sciences* 50.7 (2007), pp. 1115–1120.

[25] George B Rybicki and Alan P Lightman. *Radiative processes in astrophysics*. John Wiley & Sons, 1991.

[26] John C Curlander and Robert N McDonough. *Synthetic aperture radar*. Vol. 11. Wiley, New York, 1991.

[27] Chris Oliver and Shaun Quegan. *Understanding synthetic aperture radar images*. SciTech Publishing, 2004.

[28] Francesco De Zan and A Monti Guarnieri. "TOPSAR: Terrain observation by progressive scans". In: *IEEE transactions on geoscience and remote sensing* 44.9 (2006), pp. 2352–2360.

[29] *User Guides - Sentinel-1 SAR - Level-1 Ground Range Detected - Sentinel Online - Sentinel Online*. URL: https://copernicus.eu/user-guides/sentinel-1-sar/resolutions/level-1-ground-range-detected.

[30] *User Guides - Sentinel-1 SAR - Acquisition Modes - Sentinel Online - Sentinel Online*. URL: https://copernicus.eu/user-guides/sentinel-1-sar/acquisition-modes.

[31] *Sentinel-1 - Overview - Sentinel Online - Sentinel Online*. https://copernicus.eu/missions/sentinel-1/overview.

[32] European Space Agency. *Sentinel-1 SAR Acquisition Modes*. Accessed: 2023-05-29. URL: https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar/sar-instrument/acquisition-modes.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[34] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[36] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[37] Jieneng Chen et al. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. 2021. arXiv: 2102.04306 [cs.CV].

[38] Andreas Stokholm et al. "AI4SeaIce: Toward Solving Ambiguous SAR Textures in Convolutional Neural Networks for Automatic Sea Ice Concentration Charting". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–13. DOI: 10.1109/TGRS.2022.3149323.

[39] Jiande Zhang et al. "An Improved Sea Ice Classification Algorithm with Gaofen-3 Dual-Polarization SAR Data Based on Deep Convolutional Neural Networks". In: *Remote Sensing* 14.4 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14040906. URL: https://www.mdpi.com/2072-4292/14/4/906.

[40] Wei Song et al. "E-MPSPNet: Ice & Water SAR Scene Segmentation Based on Multi-Scale Semantic Features and Edge Supervision". In: *Remote Sensing* 14.22 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14225753. URL: https://www.mdpi.com/2072-4292/14/22/5753.

[41] URL: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[42] Scikit learn developers. *Model evaluation: quantifying the quality of predictions*. [Online; accessed 21-May-2023]. 2023. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score.

[43] Scikit learn developers. *Model evaluation: quantifying the quality of predictions*. [Online; accessed 21-May-2023]. 2023. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics.

[44] PWGSN Major. “AutoICE challenge winners’ event”. In: AutoICE challenge winners’ event for top-performing teams, 2023.

Appendix A

Model Summary

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 512, 512]	3,472
BatchNorm2d-2	[-1, 16, 512, 512]	32
ReLU-3	[-1, 16, 512, 512]	0
Conv2d-4	[-1, 16, 512, 512]	2,320
BatchNorm2d-5	[-1, 16, 512, 512]	32
ReLU-6	[-1, 16, 512, 512]	0
MaxPool2d-7	[-1, 16, 257, 257]	0
Conv2d-8	[-1, 32, 257, 257]	4,640
BatchNorm2d-9	[-1, 32, 257, 257]	64
ReLU-10	[-1, 32, 257, 257]	0
Conv2d-11	[-1, 32, 257, 257]	9,248
BatchNorm2d-12	[-1, 32, 257, 257]	64
ReLU-13	[-1, 32, 257, 257]	0
MaxPool2d-14	[-1, 32, 129, 129]	0
Conv2d-15	[-1, 64, 129, 129]	18,496
BatchNorm2d-16	[-1, 64, 129, 129]	128
ReLU-17	[-1, 64, 129, 129]	0
Conv2d-18	[-1, 64, 129, 129]	36,928
BatchNorm2d-19	[-1, 64, 129, 129]	128
ReLU-20	[-1, 64, 129, 129]	0
MaxPool2d-21	[-1, 64, 65, 65]	0
Conv2d-22	[-1, 128, 65, 65]	73,856
BatchNorm2d-23	[-1, 128, 65, 65]	256
ReLU-24	[-1, 128, 65, 65]	0
Conv2d-25	[-1, 128, 65, 65]	147,584

```
      BatchNorm2d-26          [-1, 128, 65, 65]             256
             ReLU-27          [-1, 128, 65, 65]               0
        MaxPool2d-28          [-1, 128, 33, 33]               0
           Conv2d-29            [-1, 128, 2, 2]       4,194,432
          Flatten-30               [-1, 128, 4]               0
   PatchEmbedding-31               [-1, 4, 128]               0
        LayerNorm-32               [-1, 4, 128]             256
MultiheadAttention-33  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-34               [-1, 4, 128]               0
        LayerNorm-35               [-1, 4, 128]             256
           Linear-36               [-1, 4, 512]          66,048
             GELU-37               [-1, 4, 512]               0
           Linear-38               [-1, 4, 128]          65,664
          Dropout-39               [-1, 4, 128]               0
         ViTBlock-40               [-1, 4, 128]               0
        LayerNorm-41               [-1, 4, 128]             256
MultiheadAttention-42  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-43               [-1, 4, 128]               0
        LayerNorm-44               [-1, 4, 128]             256
           Linear-45               [-1, 4, 512]          66,048
             GELU-46               [-1, 4, 512]               0
           Linear-47               [-1, 4, 128]          65,664
          Dropout-48               [-1, 4, 128]               0
         ViTBlock-49               [-1, 4, 128]               0
        LayerNorm-50               [-1, 4, 128]             256
MultiheadAttention-51  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-52               [-1, 4, 128]               0
        LayerNorm-53               [-1, 4, 128]             256
           Linear-54               [-1, 4, 512]          66,048
             GELU-55               [-1, 4, 512]               0
           Linear-56               [-1, 4, 128]          65,664
          Dropout-57               [-1, 4, 128]               0
         ViTBlock-58               [-1, 4, 128]               0
        LayerNorm-59               [-1, 4, 128]             256
MultiheadAttention-60  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-61               [-1, 4, 128]               0
        LayerNorm-62               [-1, 4, 128]             256
           Linear-63               [-1, 4, 512]          66,048
             GELU-64               [-1, 4, 512]               0
           Linear-65               [-1, 4, 128]          65,664
          Dropout-66               [-1, 4, 128]               0
```

```
         ViTBlock-67               [-1, 4, 128]               0
        LayerNorm-68               [-1, 4, 128]             256
MultiheadAttention-69  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-70               [-1, 4, 128]               0
        LayerNorm-71               [-1, 4, 128]             256
           Linear-72               [-1, 4, 512]          66,048
             GELU-73               [-1, 4, 512]               0
           Linear-74               [-1, 4, 128]          65,664
          Dropout-75               [-1, 4, 128]               0
         ViTBlock-76               [-1, 4, 128]               0
        LayerNorm-77               [-1, 4, 128]             256
MultiheadAttention-78  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-79               [-1, 4, 128]               0
        LayerNorm-80               [-1, 4, 128]             256
           Linear-81               [-1, 4, 512]          66,048
             GELU-82               [-1, 4, 512]               0
           Linear-83               [-1, 4, 128]          65,664
          Dropout-84               [-1, 4, 128]               0
         ViTBlock-85               [-1, 4, 128]               0
        LayerNorm-86               [-1, 4, 128]             256
MultiheadAttention-87  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-88               [-1, 4, 128]               0
        LayerNorm-89               [-1, 4, 128]             256
           Linear-90               [-1, 4, 512]          66,048
             GELU-91               [-1, 4, 512]               0
           Linear-92               [-1, 4, 128]          65,664
          Dropout-93               [-1, 4, 128]               0
         ViTBlock-94               [-1, 4, 128]               0
        LayerNorm-95               [-1, 4, 128]             256
MultiheadAttention-96  [[-1, 4, 128], [-1, 2, 2]]               0
          Dropout-97               [-1, 4, 128]               0
        LayerNorm-98               [-1, 4, 128]             256
           Linear-99               [-1, 4, 512]          66,048
            GELU-100               [-1, 4, 512]               0
          Linear-101               [-1, 4, 128]          65,664
         Dropout-102               [-1, 4, 128]               0
        ViTBlock-103               [-1, 4, 128]               0
       LayerNorm-104               [-1, 4, 128]             256
MultiheadAttention-105  [[-1, 4, 128], [-1, 2, 2]]               0
         Dropout-106               [-1, 4, 128]               0
       LayerNorm-107               [-1, 4, 128]             256
```

```
          Linear-108               [-1, 4, 512]          66,048
            GELU-109               [-1, 4, 512]               0
          Linear-110               [-1, 4, 128]          65,664
         Dropout-111               [-1, 4, 128]               0
        ViTBlock-112               [-1, 4, 128]               0
       LayerNorm-113               [-1, 4, 128]             256
MultiheadAttention-114  [[-1, 4, 128], [-1, 2, 2]]               0
         Dropout-115               [-1, 4, 128]               0
       LayerNorm-116               [-1, 4, 128]             256
          Linear-117               [-1, 4, 512]          66,048
            GELU-118               [-1, 4, 512]               0
          Linear-119               [-1, 4, 128]          65,664
         Dropout-120               [-1, 4, 128]               0
        ViTBlock-121               [-1, 4, 128]               0
       LayerNorm-122               [-1, 4, 128]             256
MultiheadAttention-123  [[-1, 4, 128], [-1, 2, 2]]               0
         Dropout-124               [-1, 4, 128]               0
       LayerNorm-125               [-1, 4, 128]             256
          Linear-126               [-1, 4, 512]          66,048
            GELU-127               [-1, 4, 512]               0
          Linear-128               [-1, 4, 128]          65,664
         Dropout-129               [-1, 4, 128]               0
        ViTBlock-130               [-1, 4, 128]               0
       LayerNorm-131               [-1, 4, 128]             256
MultiheadAttention-132  [[-1, 4, 128], [-1, 2, 2]]               0
         Dropout-133               [-1, 4, 128]               0
       LayerNorm-134               [-1, 4, 128]             256
          Linear-135               [-1, 4, 512]          66,048
            GELU-136               [-1, 4, 512]               0
          Linear-137               [-1, 4, 128]          65,664
         Dropout-138               [-1, 4, 128]               0
       ViTBlock1-139            [-1, 128, 2, 2]               0
          Conv2d-140             [-1, 64, 8, 8]          73,792
     BatchNorm2d-141             [-1, 64, 8, 8]             128
            ReLU-142             [-1, 64, 8, 8]               0
        Upsample-143           [-1, 64, 16, 16]               0
          Conv2d-144           [-1, 64, 65, 65]          73,792
     BatchNorm2d-145           [-1, 64, 65, 65]             128
            ReLU-146           [-1, 64, 65, 65]               0
        Upsample-147         [-1, 64, 130, 130]               0
          Conv2d-148         [-1, 32, 130, 130]          18,464
```

```
     BatchNorm2d-149         [-1, 32, 130, 130]              64
            ReLU-150         [-1, 32, 130, 130]               0
          Conv2d-151         [-1, 32, 129, 129]          18,464
     BatchNorm2d-152         [-1, 32, 129, 129]              64
            ReLU-153         [-1, 32, 129, 129]               0
        Upsample-154         [-1, 32, 258, 258]               0
          Conv2d-155         [-1, 16, 258, 258]           4,624
     BatchNorm2d-156         [-1, 16, 258, 258]              32
            ReLU-157         [-1, 16, 258, 258]               0
          Conv2d-158         [-1, 16, 257, 257]           4,624
     BatchNorm2d-159         [-1, 16, 257, 257]              32
            ReLU-160         [-1, 16, 257, 257]               0
        Upsample-161         [-1, 16, 514, 514]               0
          Conv2d-162         [-1, 16, 514, 514]           2,320
     BatchNorm2d-163         [-1, 16, 514, 514]              32
            ReLU-164         [-1, 16, 514, 514]               0
          Conv2d-165         [-1, 12, 512, 512]             204
      FeatureMap-166         [-1, 12, 512, 512]               0
          Conv2d-167         [-1, 11, 512, 512]             187
      FeatureMap-168         [-1, 11, 512, 512]               0
          Conv2d-169          [-1, 7, 512, 512]             119
      FeatureMap-170          [-1, 7, 512, 512]               0
================================================================
Total params: 6,275,694
Trainable params: 6,275,694
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 24.00
Forward/backward pass size (MB): 731.04
Params size (MB): 23.94
Estimated Total Size (MB): 778.98
----------------------------------------------------------------
```

Appendix B

Results

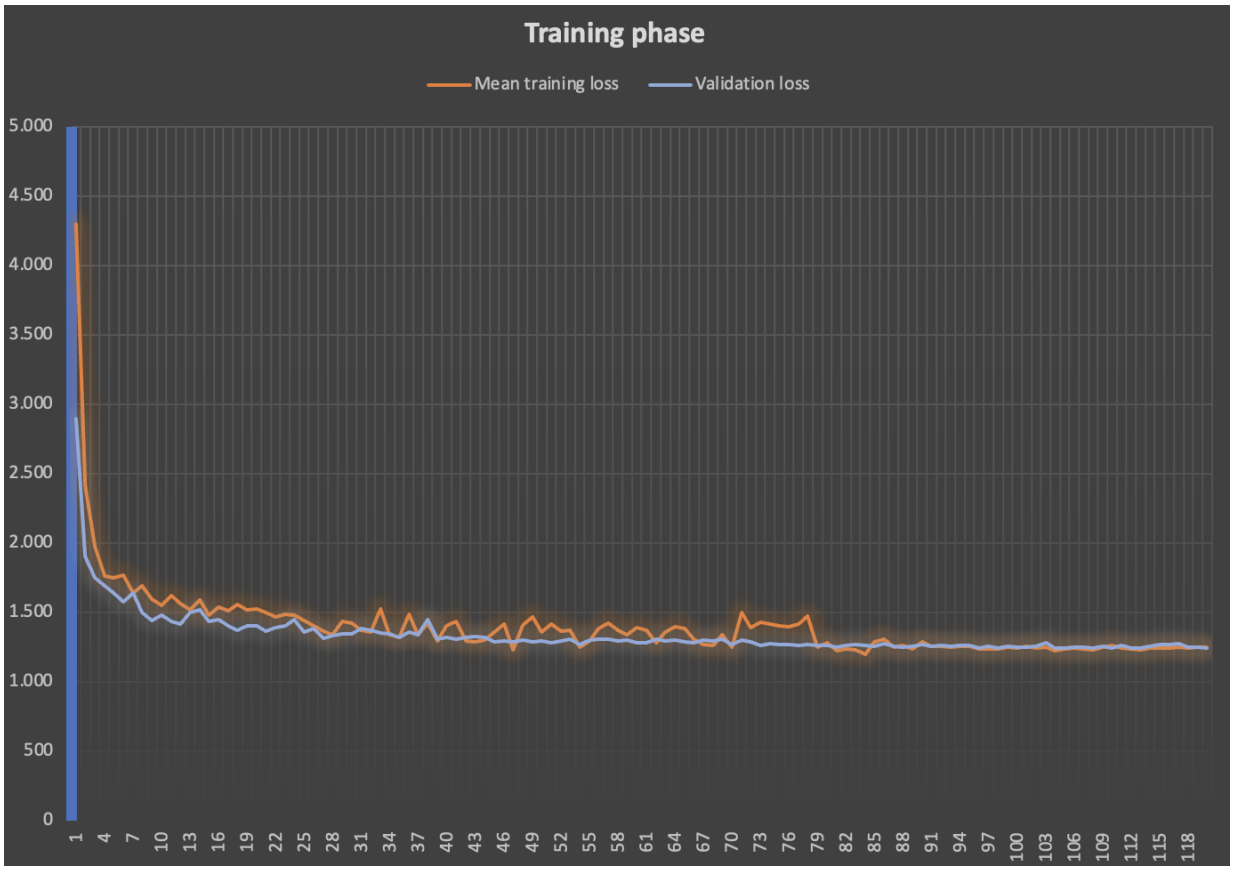

Figure B.1: Result of training and validation loss.

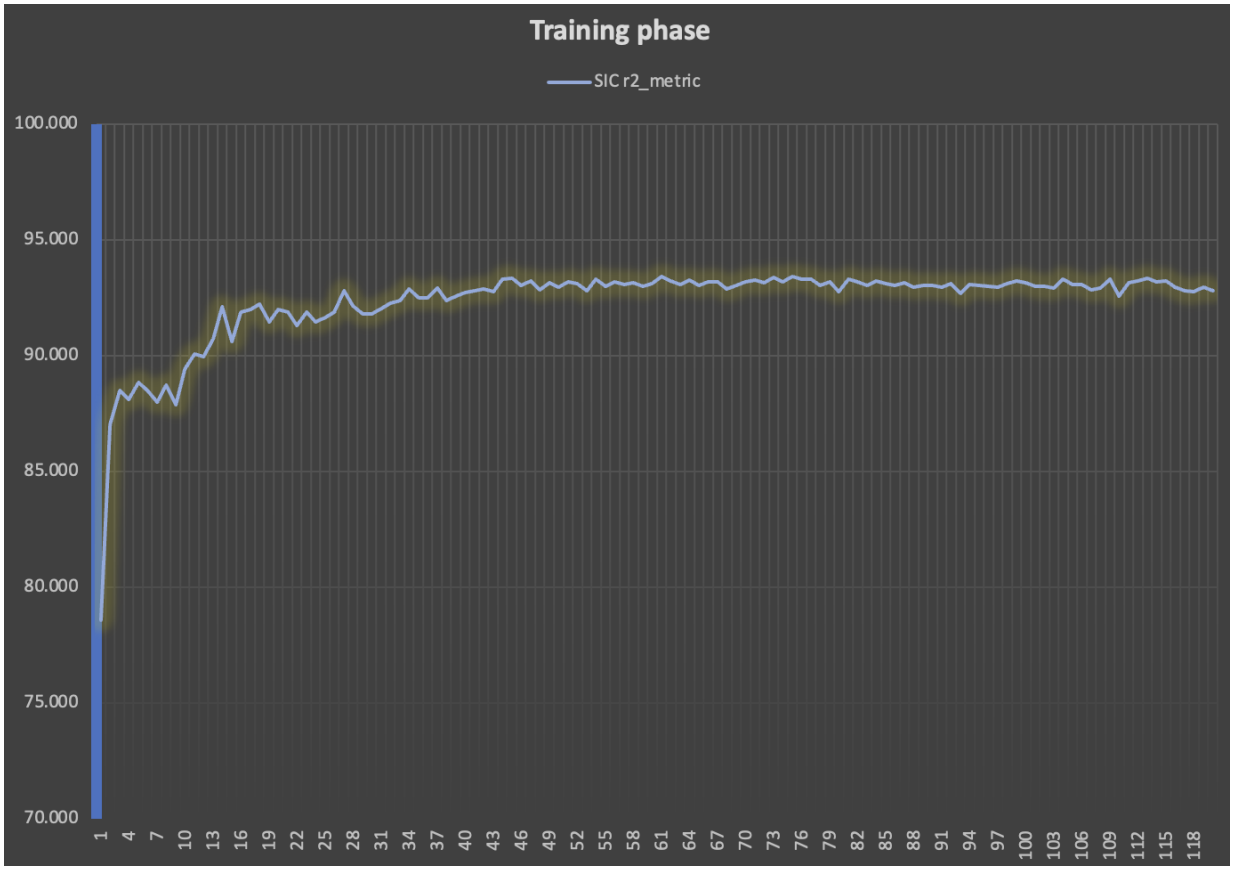

Figure B.2: Result of SIC

Figure B.3: Result of SOD.

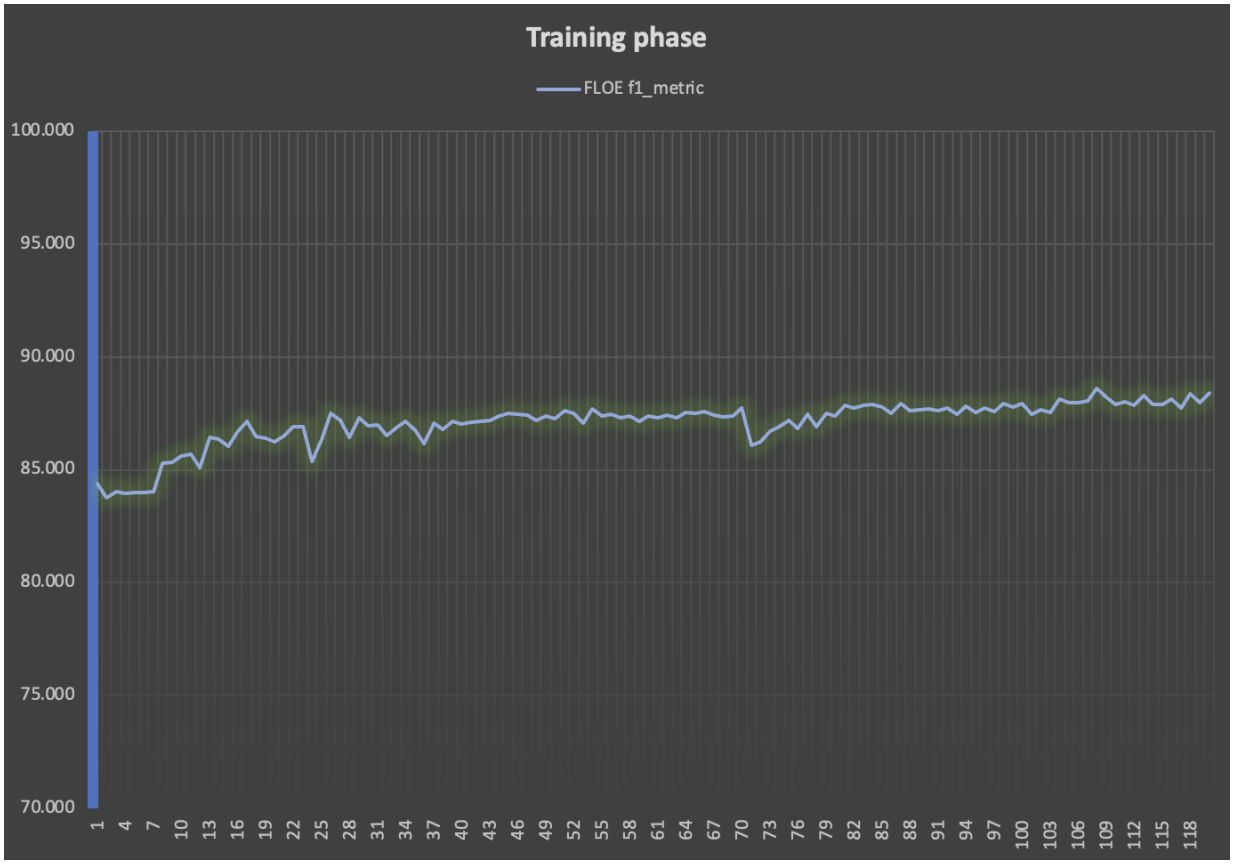

Figure B.4: Result of FLOE.

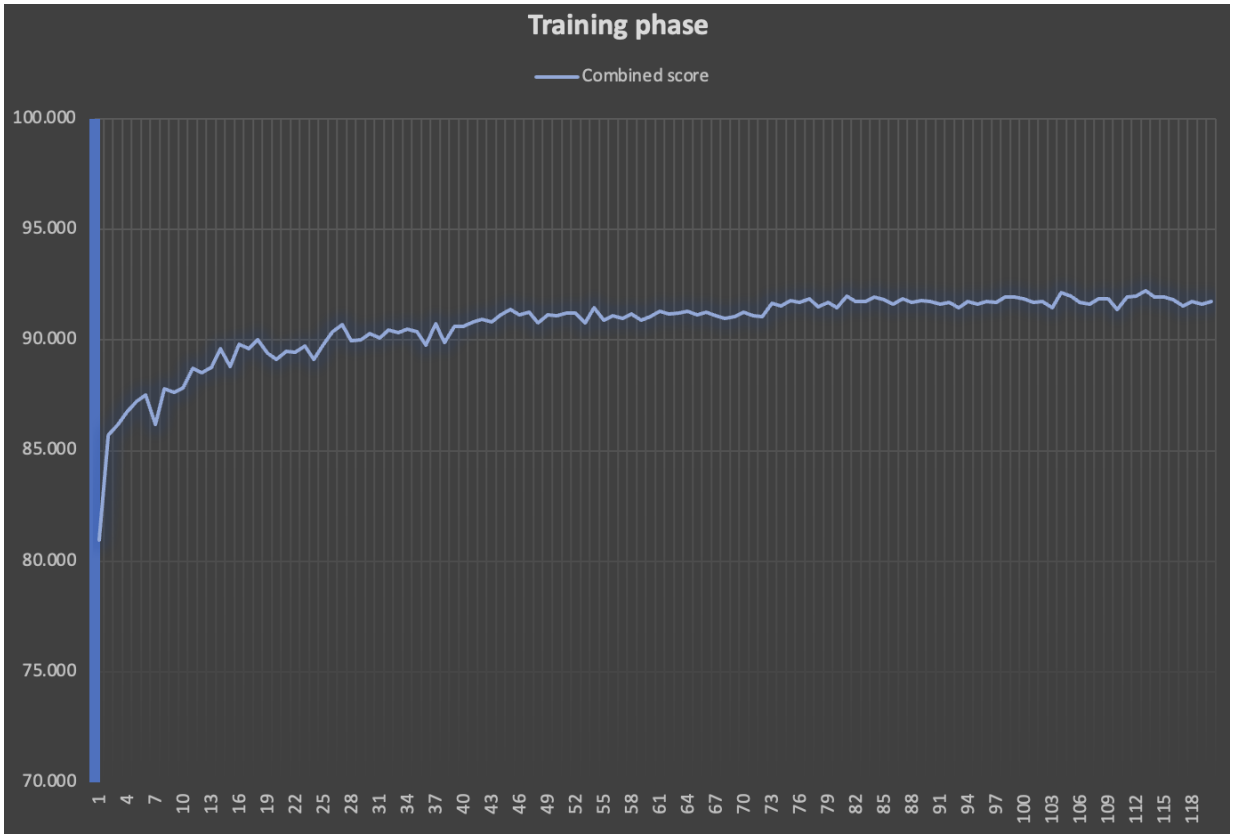

Figure B.5: Result of weighted average of SIC, SOD, and FLOE.