

# Sea Ice Concentration Estimation Techniques Using Machine Learning

An end-to-end workflow for estimating concentration maps from SAR images



*Prepared by:*

**Stefan Dominicus**

**DMNSTE001**

Department of Electrical Engineering

University of Cape Town

*Prepared for:*

**Prof. Amit Mishra**

Department of Electrical Engineering

University of Cape Town

**October 2019**

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering.

## **Key Words:**

Sea Ice, Neural Networks, Remote Sensing, Southern Ocean, Marginal Ice Zone

## Declaration

---

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This final year project report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Name: Stefan Dominicus

---

Signature:  Date: 13 October 2019

---

## Terms of Reference

---

During the course of this project the student is expected to gain the requisite understanding of polar sea ice and the reasons for developing more accurate models describing its dynamics. Machine learning techniques should then be investigated and applied to the problem of estimating sea ice characteristics using remotely sensed data. Representatives from the South African Weather Service will be available for consultation and their objectives should be considered when making design choices. The final deliverables will be defined through discussion with these representatives, and this process should be documented and included in the final report.

A note on project work conducted prior to the official start date at the beginning of the second semester of 2019:

*"I met with Prof. Mishra to discuss the project offer on the 25th of February. Following that meeting, the project was granted to me on the 28th of February. My first meeting with representatives from the South African Weather Service took place on the 8th of March. We discussed their expectations coming into the project, and they provided some direction to resources which could help us understand their needs. I then focused on two things: introductory machine learning tutorials on Kaggle, and familiarising myself with PolarView (one of the sources of information suggested by WSA). The machine learning tutorials soon became irrelevant once the ML module in EEE4114F began, and PolarView was quickly abandoned as a source of data, since they merely present current information, and do not provide access to historical data. Any further efforts were put on hold when uncertainty around the topic allocation process arose. My work on the project only resumed when the topic allocation was confirmed (which was the same time as all other students)."*

## Acknowledgements

---

Thank you, Christo, and Marc, for your assistance and guidance over the course of this project. I appreciate the time you took to meet and discuss all aspects of the investigation as it unfolded. Your insights were very helpful.

**Christo Rautenbach**

Chief Marine Scientist at South African Weather Service

**Marc De Vos**

Senior Marine Scientist at South African Weather Service

## Abstract

---

Sea ice concentration is an important metric used to characterise polar sea ice behaviour. Understanding this behaviour and accurately representing it is of critical importance for climate science research, and also has important uses in the context of maritime navigation. Current models used to estimate concentration maps are derived from passive microwave radar satellite data, but these instruments offer poor spatial resolutions, and are susceptible to interference due to atmospheric variations. Additionally, these models are usually calibrated for Arctic conditions, and evidence from recent in situ observations in the Southern Ocean suggests that their estimates are significantly biased and are not reliable in the Antarctic marginal ice zone. An end-to-end workflow for generating learned concentration estimation models from synthetic aperture radar data, trained on existing passive microwave data, is presented here. A novel objective function was introduced to account for uncertainty in the passive microwave measurements, which can be extended to account for arbitrary sources of error in the training data, and a recent set of in situ observations was used to evaluate the reliability of the chosen passive microwave concentration estimation model. Google Colaboratory was used as the development platform, and all notebooks, training data, and trained models are available on GitHub.

*GitHub Repository Link:*

[https://github.com/stefandominicu/FYP\\_ML\\_SIC](https://github.com/stefandominicu/FYP_ML_SIC)

# Table of Contents

---

Declaration .....	i
Terms of Reference .....	ii
Acknowledgements.....	iii
Abstract.....	iv
Table of Contents .....	v
List of Figures.....	vii
List of Tables.....	xi
1. Introduction.....	1
Background .....	1
Objectives.....	2
Problems to be Investigated.....	2
Purpose of Study.....	2
Scope & Limitations.....	3
Plan of Development .....	3
2. Literature Review .....	5
Existing Numerical Methods .....	5
Challenges Developing New Models .....	6
Use of Remote Sensed Data .....	7
Current State of Affairs .....	7
Application of Deep Learning .....	8
3. Requirement Analysis & ATPs .....	10
User Requirements .....	10
Functional Requirements .....	10
Specifications.....	11
Acceptance Tests.....	13
4. Computational Setup.....	15
5. Data Pre-Processing .....	16
Creating a Labelled Dataset.....	16
Define Parameters.....	16
Download SENTINEL-1 Images .....	16
Download OSI SAF Concentration Charts .....	18
Generate Label Patches .....	18
Selective Filtering .....	22
Batching & Data Augmentation .....	23
Summary of Pre-Processing Pipeline.....	25
Summary of Datasets Used .....	26
6. Neural Networks.....	27
Neural Network Architectures.....	27
Convolutional Neural Network .....	27
U-Net Adapted .....	28

DenseNet Adapted .....	29
Summary of Models Used .....	29
Training & Evaluation .....	30
Objective Function .....	30
Swath Stitching Artefacts .....	31
7. Targeted Predictions .....	32
Pass Direction Error.....	32
Copernicus Pre-Processing of SAR Images .....	32
Evidence of Mislabelling .....	34
Workaround Solution.....	37
Navigational Support .....	38
Evaluation of Existing Models .....	40
8. Results.....	41
Training & Evaluation .....	41
Performance Metrics .....	41
Loss Trajectories .....	42
Visual Inspection.....	43
Validity of Existing Models.....	43
Acceptance Tests.....	44
9. Discussion.....	45
Spatio-Temporal Resolution.....	45
Learning vs Numerical Models .....	45
Arbitrary Loss Definition.....	46
Data Augmentation .....	46
Multi-Stage Training .....	46
Reliability of Existing Models .....	47
User Requirements .....	47
10. Conclusions.....	48
References .....	49
Appendix A – Neural Network Loss Trajectories .....	51
CNN Training Progression .....	51
U-Net Training Progression .....	52
DenseNet Training Progression .....	53
Appendix B – Neural Network Concentration Estimates .....	54
Using Arctic Training Data .....	54
Using Antarctic Training Data .....	55
Using Hybrid Training Scheme .....	56
Appendix C – In Situ Observation Comparisons.....	57
EBE Faculty: Assessment of Ethics in Research Projects .....	59

## List of Figures

---

Figure 1: A detailed plan of development showing the overall structure of this document and clearly dividing the discussion in each chapter into definition, design, implementation, and interpretation phases. The emphasis in this diagram is on the content of each section. The order and purpose of main chapters is included in above.	.....4
Figure 2: Pixel coordinate interpolation scheme. This diagram illustrates how the OData ‘footprint’ attribute is structured, and how it was used to generate a grid space interpolation of WGS84 coordinates, allowing the real-world location of each pixel in a SAR image to be determined and used for further processing.....19	
Figure 3: South Sandwich Islands pixel coordinate interpolation validation test results for two SAR images acquired in January 2018. The original SAR image is shown first, followed by the resampled image, which forces a square aspect ratio. The proximity mask is shown third, where each pixel is masked if it’s interpolated coordinates are not within a radius of 10km from the known coordinates of any of the islands in the chain. This mask is then applied to the resampled image, and shown fourth. The results show each island is clearly visible within the corresponding mask bubble, and therefore the coordinate interpolation algorithm is deemed to be accurate and reliable enough to use for the remainder of the patch label generation process. ....20	
Figure 4: OSI-401-b concentration chart illustration, with an arbitrary SAR image footprint overlaid to show the significance of first reducing the chart size to focus only on the portion of the concentration chart which is near to the SAR image. This significantly reduces the computational cost of finding the nearest concentration sample for each pixel in the SAR image. ....21	
Figure 5: Patch label validation, showing an example image-label set. The resampled SAR image is shown on the left, the concentration patch label in the middle, and the uncertainty patch label on the right. By visual inspection, it is clear that the features present in the SAR image are also reflected in both patch labels. This validates the label generation procedure, and also shows that the generated labels are at a higher resolution than the chart provided by the OSI-401-b product. The pixel spacing in this example is 3.26km x 3.43km. ....22	
Figure 6: Pixel-wise concentration distribution. In this figure, the resampled SAR image, concentration patch label and uncertainty patch label are shown for two representative samples in the dataset. The histogram on the right shows the distribution of concentration values present in the concentration patch label for each sample. The first sample (top) is clearly more “interesting” in terms of concentration features, since it includes a section of ice edge, while the second sample (bottom) contains almost no variation, with most of the concentration values being greater than 80%. A variance threshold was used to filter all samples, ensuring only “interesting” were included. ....23	

Figure 7: Median filter (5x5) applied to concentration and uncertainty labels. Both unfiltered and filtered patch labels are shown, so that the effect can be visually inspected. The median filter does well at smoothing the patch labels without blurring the image, and also doesn't introduce any concentration values which were not already present in the vicinity of each pixel to begin with.....	24
Figure 8: Summary of the data pre-processing pipeline developed for this investigation. This graphic is a summary of the detailed discussion over the previous sections on dataset management. The pipeline was divided into four major stages, and this graphic illustrates how each stage links together, as well as how data moves through each stage in the pipeline.....	25
Figure 9: An illustration of the sequential FCNN architecture. This example contains two hidden layers, and illustrates how the input SAR image data is propagated through the network, and how the number of convolutional filters is increased at each layer. ....	27
Figure 10: A diagram illustrating the original U-Net architecture. With each additional layer, the number of filters is increased, but feature maps are also passed across to layers closer to the output, which allows for better gradient propagation and feature reuse [8]. ..	28
Figure 11: A diagram illustrating the original DenseNet architecture. Each dense block consists of densely connected convolutional layers. Although it was originally designed for classification, the structure was easily modified to predict the concentration maps needed in this investigation [7]. ..	29
Figure 12: Python definition for the uncertainty weighted mean absolute error function used as the objective function for all models in this investigation. The customised Sequence class passes both concentration and uncertainty patches through as the target output. The loss function separates the two layers so that each can be used independently in calculating the loss for each sample's estimation. ....	31
Figure 13: Satellite tracker website used to show the path of SENTINEL-1A over the surface of the Earth. Each portion of the orbit is labelled as either ASCENDING or DESCENDING, as indicated. [13] .....	32
Figure 14: SENTINEL-1 pixel-wise image structure convention, showing how the pixels are ordered in both acquisition and display, thereby explaining the vertical flip which must be accounted for in pre-processing.....	33
Figure 15: A diagram explaining the pre-processing steps applied to SENTINEL-1 Quicklook images, based on pass direction. Only the final output is accessible for download, once all pre-processing steps are complete. The locally applied rotation is also shown, to illustrate how each pass direction needs to be rotated in a different direction in order for the features in the image to align correctly in the Earth's reference frame.....	33
Figure 16: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only DESCENDING products were included. A	

preview of one product is included (bottom), showing that the footprint and Quicklook are in agreement [10]. ..	34
Figure 17: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only ASCENDING products were included. However the trajectories inferred from the collection of footprints suggests that these are actually DESCENDING products. A preview of one product is included (bottom), showing that the footprint and Quicklook are not in agreement, but in fact the Quicklook is rotated by 180 degrees [10]. ..	35
Figure 18: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only ASCENDING products were included, and the number of results per page increased to show more footprints on the map. It appears that the selected products are correctly labelled as ASCENDING, while the rest are actually DESCENDING products [10].....	35
Figure 19: A diagram explaining what happens during pre-processing if a product's pass direction is mislabelled, based on the pre-processing steps discussed above. If this were to happen, it is clear that the Quicklook image would appear rotated by 180 degrees. ....	36
Figure 20: A diagram explaining the convention governing the order in which the footprint points are listed in the OData 'footprint' attribute. This can be leveraged to determine the satellite's actual orientation at acquisition time, without relying on the 'pass direction' attribute.....	37
Figure 21: This interface was developed to allow a user to search for the most recent SAR data available for their location, or for historical data if needed, and then to view the estimates produced by a number of models side-by-side. The map on the left offers location context, and includes the footprint of each SAR measurement. ....	38
Figure 22: Illustration of an arbitrary SAR image footprint projected onto a stereographic map. The angles which the top and bottom edges make with the horizontal are defined as $\theta_1$ and $\theta_2$ respectively, as shown in the diagram. These angles were calculated using trigonometry and the known coordinates of each corner of the footprint polygon. The two angles were averaged to determine the required degree of rotation. ....	39
Figure 23: This comparison interface shows images taken in the MIZ and compares them with the OSI-401-b estimates for the same day and location, with the exact location indicated with crosshairs. ....	40
Figure 24: Loss trajectories for DenseNet, covering all training strategies. The divergence of test results from the training curves is noteworthy, and warrants further discussion.....	42
Figure 25: Two examples of how each trained model was visually evaluated using a pre-selected sample. The top row shows a DenseNet_N_N estimate, and the bottom row shows a UNet_NS_S estimate. The significance of these particular concentration estimates will be discussed further below. ....	43

Figure 26: In situ observations from the MIZ compared with PMW-derived concentration estimates for the same day and location. Each chart was generated using the evaluation tool developed for this investigation, which was presented in the previous section. ....	44
Figure 27: CNN loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets. ....	51
Figure 28: U-Net loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets. ....	52
Figure 29: DenseNet loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets. ....	53
Figure 30: A set of concentration estimates generated using Northern hemisphere trained models. ....	54
Figure 31: A set of concentration estimates generated using Southern hemisphere trained models. ....	55
Figure 32: A set of concentration estimates generated using models trained partially on each hemisphere's data. ....	56
Figure 33: A set of in situ observation comparisons generated using the comparison tool developed for use in this investigation. The PMW-derived concentration estimate shown is from the same day as each image, and the location is indicated on the concentration chart with crosshairs. ....	58

## List of Tables

---

Table 1: User requirements table, showing how the task has been divided into individual requirements. The purpose of this table is to define the main objectives, as agreed upon with the client, in order to focus the engineering efforts in the investigation process and achieve a disireable end product. ....	10
Table 2: Functional requirements table, showing how each user requirement has been analysed in terms of distinct functions. This allows the high-level user requirements to be defined according to a number of lower-level functional elements which must all be met in order to satisfy each user requirement. ....	11
Table 3: Specifications table, showing the design specifications derived from the functional requirements above. Each of the specifications listed is specific enough to be directly implemented, or used in the implementation of a larger process. Specifications are defined to meet the problem's low-level detail-specific requirements.....	13
Table 4: Acceptance test procedures table, showing the tests which will be used to determine if the user requirements have been met successfully. Each test is focused on assessing a particular specification, or aspect of a specification, but together the complete set of tests will indirectly asses the efficacy of the proposed solution in terms of the higher-level functional and user requirements. ....	14
Table 5: Summary of batch sets used in this investigation. Each batch set has a unique name, and is created from a parent dataset as described in the previous section. The table also shows the total number of images acquired in the parent dataset, and the number of images selected by the concentration variance threshold filter.....	26
Table 6: Summary of neural network models used in this investigation. The details and important differences are shown in the table, and more details on each model can be found in the preceding sections.....	30
Table 7: Summary of results obtained from each training strategy for each model type under investigation. Multiple sets of training data indicate that the model was partially trained on the first set, and then trained to completion using the second set.....	41
Table 8: Acceptance test procedure outcome table, showing the outcome of each test, and justifying each result in reference to work done or results shown in the course of this investigation. ....	44

# 1. Introduction

---

## Background

Sea ice concentration (SIC) is an important metric used to quantify polar sea ice dynamics. Sea ice exists at the ocean-atmosphere interface and plays an important role in the exchange of energy between these two mediums. Coupled climate models, used to predict, and analyse Earth climate system evolution, are currently heavily reliant on numerical models describing the interplay of influential factors in order to produce accurate results. However, these numerical models are notoriously difficult to develop fully since the process of parameterising each phenomenon requires extensive observational datasets. These validated datasets are far more prevalent in the Arctic region compared to the Antarctic and Southern Ocean, and as a result, the derived models tend to be biased towards our understanding of Arctic conditions. Despite limited in situ observations from the Antarctic, there is sufficient evidence that the Arctic and Antarctic environments differ significantly, and that the naïve application of Arctic-derived models to the Antarctic, and the larger Southern Ocean, produces unreliable results. While it may be possible to calibrate these models to give more accurate Antarctic estimates, the lack of consistency in existing observational data gathered from the region makes it difficult to assimilate these measurements into any kind of calibration process.

Satellite remote sensing platforms have proved very useful in generating an abundance of observational data. However, the interpretation of satellite data is still reliant to some extent on an underlying numerical understanding of the sensed environment, which has limited the extent to which satellite data can be utilised. Current models rely on passive microwave (PMW) radar data and are widely used, particularly in the Arctic. Unfortunately, PMW instrument measurements are sensitive to atmospheric water content, and so tend to be less reliable during the intense polar cyclones experienced in the Southern Ocean. The resolution of PMW is also poor and introduces uncertainty in the order of kilometres which makes these models unsuitable for any analysis tasks where high spatial resolution is required.

Deep learning has been proposed as a tool which may allow a wider variety of satellite measurement types to be used in characterising metrics such as SIC in polar regions. Each measurement type offers its own advantages in terms of resolution, sensitivity to atmospheric conditions, timeliness, etc, and so it would be advantageous to have a set of interpretation models available for each characteristic of interest, allowing multiple interpretations to be assimilated into a single model, thus describing each metric more accurately and reliably. Work in this field has most often relied on manually interpreted measurement sets for training data, but very recently, methods for training on automatically generated label information derived from existing PMW interpretations models have shown promise and have the advantage of removing reliance on manually interpreted datasets which are not available in all situations.

# Objectives

## Problems to be Investigated

The primary focus of this investigation was to develop machine learning models able to estimate SIC using only remote sensed satellite imagery. This required the identification of a suitable remote sensed measurement type which could offer improvements in spatio-temporal resolution compared to existing PMW measurements, and the design of a method for automatically generating SIC label information for each measurement to be used in training. An important question when designing the training process was whether or not any known inaccuracies or uncertainties in the PMW data could be accounted for so as to avoid learning the same flaws in each new model. The outcome of this line of investigation was a user interface tool demonstrating the proposed workflow for acquiring new remote measurements, generating a series of SIC predictions, and serving the results alongside additional location context to aid visual interpretation.

The secondary focus was to design a comparison tool able to present in situ observations from the Southern Ocean marginal ice zone alongside PMW-derived SIC estimates. This involved identifying a suitable set of in situ observations to use for comparison, ensuring that a variety of SIC conditions were included in the set of observations. The images were incorporated into a display interface, alongside the corresponding SIC estimate, and additional location context. This interface allowed the sea ice conditions in each image to be visually compared to the existing model's estimates.

## Purpose of Study

The importance of accurately modelling polar sea ice was alluded to in the background discussion and is also covered in detail in the literature review presented in the following section. Deep learning has been shown to be a very powerful image processing tool in almost all fields of science in which it is applied, but the task remains to develop the specific implementation required in the context of SIC estimation. Although there have been many advances in transfer learning and deep learning architectures which generalise well to a wide range of applications, the process of applying deep learning techniques to SIC estimation remains non-trivial, especially considering the myriad of remote sensed data sources and interpretation methods available. An investigation in this regard would be a valuable step towards developing a better understanding of the dynamics driving changes in polar sea ice coverage, and by extension may improve the performance of existing coupled climate models.

Regarding the validity of PMW-derived estimates in the Southern Ocean, it is important to understand what the models are capable of and where they become unreliable. These products are so widely used that it is easy to imagine how discrepancies in the accuracy of SIC estimates could negatively affect the results of any research based on these results. However, with the limited availability of validated data for correctly calibrating these models for the Antarctic region, any tool which simplifies the process of comparing in situ observations to estimates from PMW-derived models is desirable and may offer insight into the shortfalls of existing models.

## Scope & Limitations

This investigation was concerned only with sea ice *concentration* estimation and did not consider other sea ice metrics such as drift or thickness, although the importance of other factors is discussed in the literature review. Training data consisted of a collection of remotely sensed images, and sea ice concentration labels dynamically generated for each input image. Only one source of input images was used, and only one existing PMW-derived SIC model was used to generate each concentration label. The concentration information derived from PMW data were assumed to be accurate for the purposes of training and were not re-processed in any way. These choices are discussed in the Specifications section below. Two datasets were generated which covered the Arctic Circle and the Antarctic Circle for a period of six months in the beginning of 2019. No other data was used when training each neural network model. Three well-known neural network models typically used in image processing were used. Some slight modifications were needed to make each architecture suitable for this application, but the focus of the investigation was on tuning the hyperparameters of these three architectures to attain satisfactory results, and not to introduce any novel architectures.

This investigation was conducted as a final year project for a BSc (Eng) Mechatronics degree over a period of three months, in conjunction with course work. Progress was primarily hindered by time constraints, particularly during the initial design and development stages of the project. Additionally, the use of Google Colaboratory for development and Google Drive for storage and data management presented a number of challenges which stalled overall progress once large datasets were introduced. These issues were ultimately overcome and are discussed in detail in the Data Pre-Processing section below. Nevertheless, these challenges limited the total time spent on neural network development, which could certainly have yielded improved results with more time.

## Plan of Development

A review of relevant literature is presented in the following section, with the intention of demonstrating the importance of sea modelling for climate science and justifying the need for improvements in the field, particularly in the Southern hemisphere. The Requirements Analysis section defines the problems to be addressed in this investigation and ensures that the expectations put forward by the South African Weather Service are properly incorporated into each design decision. An overview of the computational setup is described next, which is followed by the Data Pre-Processing section, which presents a thorough discussion on each stage of the processing pipeline developed to handle all data needs for this investigation. Focus then shifts to neural networks, and all details related to architecture, objective functions, training schemes, and challenges are discussed. The section on Targeted Results explains the development of interfaces used to display estimation results to a user, demonstrating the proposed workflow for navigational support and evaluating existing models. The results of each training scheme are then presented, followed by the Discussion which highlights

important results and draws on the theory discussed in the literature review to put the results in context. Finally, the Conclusion mentions the most important results and offers some suggestions for further research in this direction. Figure 1 presents a more detailed overview of the content of this document, and shows how the structure is divided into definition, design, implementation, and interpretation phases.

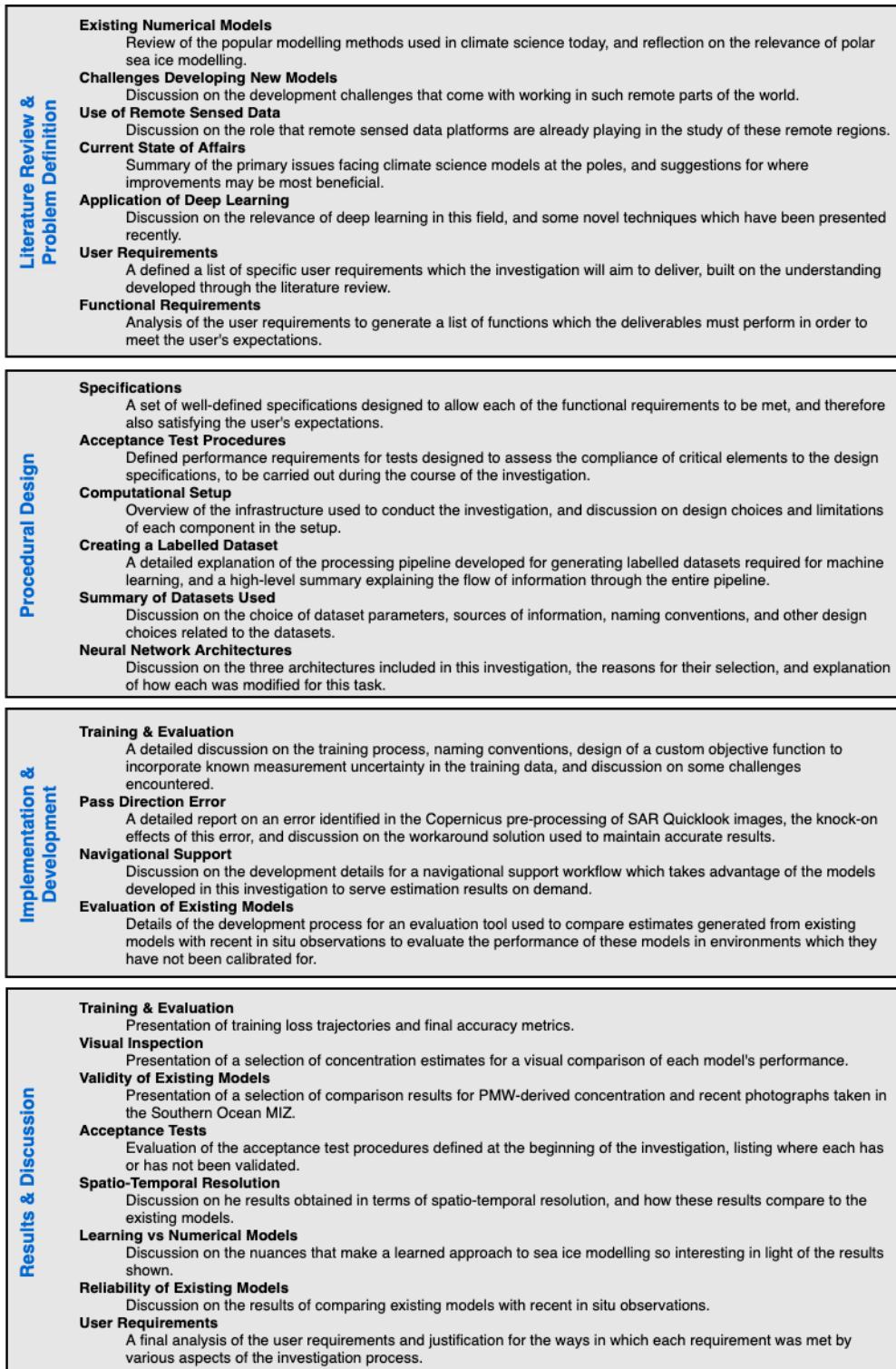


Figure 1: A detailed plan of development showing the overall structure of this document and clearly dividing the discussion in each chapter into definition, design, implementation, and interpretation phases. The emphasis in this diagram is on the content of each section. The order and purpose of main chapters is included in above.

## 2. Literature Review

---

Polar sea ice coverage, and the phenomena which drive changes in this coverage, form a critical link in the Earth's climate system. Sea ice is located right at the ocean-atmosphere interface and is thus intricately linked to climate system evolution [1]. Sea ice coverage is characterised by a number of important variables which have each been tied to particular aspects of climate system evolution, and it is clear that understanding the interplay between these variables is an important requirement for accurately modelling large-scale climate trends [2]. While the majority of the available literature is focused on understanding these trends for the purpose of improving climate science models, it is also worth noting that some work has been dedicated to sea ice modelling for the purpose of mapping ice coverage in shipping channels, which is of particular importance in the more land-constricted regions of the Arctic [3].

### Existing Numerical Methods

Due to the complexity of the sea ice system, a single unified model of the sea ice characteristic behaviour does not yet exist. Instead, each research effort tends to focus on a select subset of characteristics and aims to further our understanding of the mechanisms involved. For example, an investigation of the interaction between sea ice drift and ocean currents [4], and one providing the first in situ observations of sea ice drift behaviour during a polar cyclone [5]. In another study also investigating sea ice drift, ice edge information was included in the analysis, where the ice edge is defined as the point where the ice concentration first exceeds 15% [2]. While this metric has been a traditional choice for quantifying ice coverage, recent investigations in the Southern Ocean have chosen to focus on the much broader Marginal Ice Zone (MIZ), which is defined as the region for which the sea ice concentration lies between 15% and 80%. The MIZ is interesting because it includes a wide range of ice types, and varying concentration levels. Studies of this dynamic region in the Antarctic have presented many interesting observations which often contradict with what is expected based on our understanding of sea ice dynamics derived from Arctic observations. This suggests that our understanding of the system dynamics in the Antarctic is not yet complete, while more attention has been given to understanding the Arctic environment [5].

These, and other, contrasting observations have spawned a range of new research projects aiming to develop a deeper understanding of the dynamics at play in the Antarctic sea ice system. The reason for this is that coupled climate models, which are used to model the interaction of various Earth systems, require each phenomenon to be quantified and parameterised so that it can be simulated. Only then can the complex interaction between multiple feedback mechanisms be analysed. In the case of sea ice, the most important aspects to understand are the thermodynamic and mechanical properties of each type of ice. However, these characteristics are not well-understood for all ice types at present, and can only be crudely modelled, which leads to significant inaccuracies in the model output [6]. For example, so-called 'pancake ice' floes account for the bulk of the seasonal Antarctic ice mass, and yet the thermodynamic and

mechanical properties of this ice type are not well-known. This means that numerical models are not able to accurately capture a large portion of the climate system behaviour, which results in significant biases in the model output [1].

## Challenges Developing New Models

There are a few factors which make this a difficult problem to solve. First, consider that much of what is known in the Arctic in terms of climate dynamics, and sea ice in particular, has been shown to be of little use in the Antarctic region due to significant differences in the underlying driving forces influencing sea ice dynamics in each region, and there is currently no consensus on what factors may be the primary driving forces of sea ice dynamics in the Antarctic [2]. Additionally, much of the current understanding in this field regarding sea ice characteristics was derived from old Arctic sea ice, which is of questionable relevance to the present day Arctic environment, let alone the Antarctic [6]. In this regard, it has been suggested that the more severe storm conditions are to blame for many of the observed differences in the Antarctic, the thinking being that the more powerful storm conditions observed in the Antarctic are enough to change the way sea ice in the MIZ behaves in terms of melting and consolidation, and that this drives knock-on effects which can explain many of the observed discrepancies between Antarctic measurements and existing Arctic-based models. In fact, the effect of polar storms on the sea ice dynamics is believed to be so significant, and so different from how storm conditions force changes in the Arctic, that the very definition of the MIZ is has been called into question since it does not reasonably apply to the Antarctic observations. Unfortunately, these storm forces are also not well-understood at present and are not well-represented in existing climate models [5].

The second, and perhaps more critical factor hindering the development of accurate Antarctic sea ice models is the chronic lack of validated data. Without comprehensive datasets describing the sea ice behaviour in the Antarctic region, it is nearly impossible to develop accurate parameterisations for each ice type (recall the importance of accurate parameterisations describing the sea ice dynamics for use in coupled climate models). While more in situ measurement data is slowly becoming available, the need still exceeds what existing datasets have to offer. While the accuracy of existing models may be sufficient for some long-term synoptic analysis, coupled models are still not able to capture large scale sea ice coverage features due to incomplete parameterisations of the system dynamics [6]. Additionally, the relatively low availability of in situ measurements combined with the increased number of sea ice types found in the Antarctic has led to greater inconsistency in reported parameters. The variability of conditions in the Antarctic simply cannot be captured in the few existing measurements, and so parameters such as drag coefficients tend to vary greatly based on the location where the observations were made and the conditions at the time. Without additional measurements it is difficult to correlate the data and identify trends which may offer insight into the underlying dynamics. The variability of reported

results also makes it challenging to assimilate new information into existing models due to discrepancies between measurements [1].

## Use of Remote Sensed Data

It is at this point, confronted with the need for more data and the difficulty of actually obtaining additional in situ measurements, that satellite remote sensing must be discussed. Perhaps the biggest obstacle when using satellite data is that some understanding of the sensed environment is still required in order to properly interpret the data provided by the remote platform [4]. Interpretation models have been developed for the Arctic region, aided by the superior availability of validation data. As before, these models can be transferred for use in the Antarctic, but will tend to overlook many of the features and dynamics which are unique to the this region, and are thus only suitable for synoptic analysis [5]. This inability to accurately capture the higher frequency sea ice dynamics has been shown to result in underestimation of various parameters, such as maximum drift speed, by over 20%. This renders the model all but useless when a high temporal resolution is desired [1]. So, with the limited availability of in situ observations to validate satellite estimates, the accuracy of analysis techniques is once again dependant on models developed for the Arctic. As a result, these datasets are coarse in both spatial and temporal resolution and introduce uncertainty in the order of kilometres due to this granularity, which makes them unsuitable for short-term, or small-scale Antarctic analysis [2].

Another factor which has to some extent precluded the use of satellite observations in Antarctic sea ice analysis is that so much of the characteristic behaviour takes place during intense stormy conditions, where clouds would obscure the scene from view by a satellite [5]. Even if passive microwave (PMW) instruments are used, the increased water and water vapour content in the atmosphere can significantly degrade the quality and accuracy of the reported information [3]. However, a promising alternative is to use Synthetic Aperture Radar (SAR) instruments which are not affected by cloud cover or sunlight and are much less sensitive to atmospheric water vapour content. The use of SAR platforms also presents the advantage of increased spatial resolution [6]. This improvement in spatial resolution will allow the interpretive models being developed to achieve greater intricacy [1].

## Current State of Affairs

A reasonable summary to this point is the following: obtaining in situ measurements is costly and subject to significant variability; the available in situ datasets are insufficient to derive accurate dynamics parameterisations for use in coupled climate models, however they do present significant evidence of the differences between the Arctic and Antarctic environments; due to this lack of understanding, numerical models cannot provide accurate high spatio-temporal resolution analysis; numerical models tend to be suited towards the Arctic environment, and lack the flexibility to be easily transferred to the Antarctic; satellite instruments can provide an abundance of data compared to the existing in situ measurements, but the lack of validation data has limited the

effective spatio-temporal resolution of the interpretive models used to extract useful information from the satellite data; and finally the high granularity in current satellite data models results in high levels of uncertainty spanning many kilometres. The remainder of this review will address some recent developments in dealing with these limitations, stemming from the desire for an improved modelling approach which is not reliant existing numerical models, can offer improved spatio-temporal resolutions, is flexible and can be adapted to suit the desired region, and which takes into account the uncertainty present in any datasets used so as not to be hindered by the high variability of some existing measurements, as discussed above. The most comprehensive effort found in addressing these desires is discussed next in detail.

## Application of Deep Learning

A recent paper presented a novel method of sea ice concentration (SIC) estimation using deep learning [3]. While this was certainly not the first time deep learning techniques had been applied to the problem of SIC estimation from satellite data, the proposed method for obtaining training data is what sets this effort apart. As mentioned earlier, existing satellite-based models rely on data from passive microwave instruments which are sensitive to atmospheric water content and offer poor spatial resolutions, both of which are undesirable in the highly dynamic MIZ. Traditionally, when higher spatial resolution information is required, it can be obtained by the manual interpretation of SAR images, which also have the advantage of being less susceptible to interference from atmospheric water content. While this method is certainly able to produce accurate interpretations, it is very time consuming, and simply cannot keep up with the vast number of SAR images generated, which means that the data source is not being fully utilised.

While deep learning is indeed well-suited to this kind of high-volume data processing task, previous efforts have relied on manually annotated SAR images for training, which are generally only available in specific regions, and only describe SIC in discrete 10-20% concentration bands (not continuously valued concentration labels) [3]. Both of these are non-ideal for training deep learning models. For this reason, a method of automatically generating label information is desirable, and is exactly what has been proposed in [3]. SIC information generated from PMW data was used to generate concentration labels corresponding to a set of SAR images. These image-label pairs were then used to train a deep neural network, based on the DenseNet architecture [7].

It is worth discussing some of the challenges, design choices, and outcomes presented in [3], as they were a major influence on the path taken in this investigation. Firstly, the size of SAR images (around 10,000 x 10,000 pixels) is prohibitively large to be used directly in neural networks and was addressed by breaking each image into smaller patches which were each processed individually. Secondly, the difference in resolution between the SAR images and the existing SIC data necessitated the use of an interpolation and resampling scheme to generate a concentration label corresponding to the footprint of each SAR image. Thirdly, not all of the available image patches were used for training - only those which contained a representative range of SIC values

were included in the training dataset. In terms of neural network architecture, the popular DenseNet structure was used with simple modifications made to the output to allow for single value regression, rather than classification for which the architecture is already well-known. This kind of architecture reuse dramatically reduces the amount of neural network development required, and benefits from progress made in other domains of deep learning. Lastly, this approach presented evidence that models trained in this way were able to predict SIC features at a higher resolution than the data with which they were trained, and also suggested that some models would inherit the flaws and inaccuracies of the PMW data used for training, while others were able to avoid these flaws, and actually produce more accurate predictions than the data with which they were trained [3].

With the success of a modified DenseNet architecture clearly demonstrated, one must also consider other architectures which may exhibit similar benefits. Much of the novelty in the original DenseNet structure was in the way it utilised skip connections to aid in gradient propagation during training, making larger networks more efficient to train [7]. One of the most well-known architectures to also leverage skip connections is U-Net [8], which was introduced one year prior to DenseNet. Where DenseNet was designed for classification, U-Net was designed for image segmentation tasks, and its encoder-decoder type structure is naturally suited to producing outputs of a similar dimension to the input. This is particularly interesting in this context, where the improved resolution of SAR images is perhaps one of the most important factors in trying to generate higher spatial resolution SIC maps. Additionally, the original U-Net implementation also used sample weighting maps in the training process, which served to emphasise particular features, and cause the network to learn those features above others [8]. This method of training manipulation by error-weighting presents some interesting opportunities in terms of allowing the training process to account for known inaccuracies in the training data and may play an important role in allowing learned models to avoid inheriting the flaws present in the models used to generate training data.

### 3. Requirement Analysis & ATPs

---

The literature review presented in the previous section discussed many of the most prominent issues facing sea ice modelling and devoted some attention to the state of affairs in this domain with regards to the application of deep learning techniques, which is main focus of this investigation. This section serves to consolidate the key points raised in the prior discussion, in order to formulate a set of engineering requirements to inform the efforts of the remaining investigation.

#### User Requirements

The user requirements define a set of high-level objectives or deliverables which will be of use to the end user. This investigation was supported by representatives from South African Weather Service ('the client'). The details of their requirements from this investigation were set out in the early stages of problem definition and revised a number of times throughout the design process as milestones were reached or as alternative avenues of research became apparent. Table 1 below summarises the agreed upon user requirements.

ID	Details
U1	Provide sea ice concentration estimates for an arbitrary region of interest, at an improved spatio-temporal resolution compared to existing models.
U2	Compare estimates with existing sea ice concentration models for both Arctic and Antarctic regions.
U3	Account for the inherent inaccuracies in existing models, so that these inaccuracies are not transferred.
U4	Compare predictions from existing models used with in situ observations in the Antarctic.

*Table 1: User requirements table, showing how the task has been divided into individual requirements. The purpose of this table is to define the main objectives, as agreed upon with the client, in order to focus the engineering efforts in the investigation process and achieve a desirable end product.*

These requirements draw heavily on the discussion presented in the literature review, which mentions many of the difficulties with existing SIC models. Spatial-temporal resolution was identified as the primary aspect to be improved, as this would allow for a significant reduction in uncertainty compared to existing models. The remaining objectives are focused on validating any new results, and evaluating the accuracy of existing models in the Antarctic environment by comparing predictions with in situ observations.

#### Functional Requirements

The discussion on functional requirements aims to assess the user requirements and derive a set of more specific engineering functions which are required in order to meet the user's desires. This is a list of less abstract engineering objectives which together ensure that each user requirement will be satisfied. Table 2 below lists the functional

requirements, and shows the requirement relationship tree relating each functional requirement to a parent user requirement.

Tree	ID	Details
U1	F1	Utilise a remote sensed data source which offers superior spatial and temporal resolution.
	F2	Develop neural network models which can predict sea ice concentration given suitable remotely sensed data.
	F3	Allow the user to search for the most recent data available for their location and serve relevant sea ice concentration estimates for the region.
U2	F4	Facilitate access to relevant estimates from existing sea ice concentration models.
	F5	Display estimates from existing models alongside new estimates for visual comparison in both Arctic and Antarctic regions.
U3	F6	Identify an algorithmic way to quantify the potential inaccuracy of any existing model estimates used.
	F7	Incorporate knowledge of these inaccuracies into the development of new models such that these errors are not transferred into the new estimates.
U4	F8	Facilitate access to a suitable dataset of geotagged in situ observations showing ice concentration conditions.
	F9	Provide a visual comparison of in situ observations with estimates from the existing model used for the same date and location.

*Table 2: Functional requirements table, showing how each user requirement has been analysed in terms of distinct functions. This allows the high-level user requirements to be defined according to a number of lower-level functional elements which must all be met in order to satisfy each user requirement.*

These functions can to some extent be treated independently since each one caters to a specific aspect of the end user requirements. However, since a lot of importance is placed on the ability to validate results and compare estimates with in situ observations, each aspect must be designed to function within a framework that is compatible with the rest of the validation process so that each set of results can be easily compared. This also necessitates some functions specifically dedicated to user interaction, despite not being intended as a full interactive data service, but rather as a first step towards such a service. Guidelines for selecting data sources for training and validation are also defined. Selecting appropriate data sources is particularly important in this case because the results of this investigation should be easily transferrable to other research activities being conducted by the client.

## Specifications

The specifications serve to fully define all important aspects of the investigation and should ensure that all functional requirements and user requirements can be met by

adhering to the list of specifications. These choices are critical because they define much of the investigative path to be followed. Each specification should be sufficient to completely define a particular aspect of the solution to be developed. Table 3 below contains a list of specifications and shows the requirement relationship tree relating each specification to a parent function and user requirement.

Tree	ID	Details
U1	F1	S1 SENTINEL-1 Ground Range Detected Extra Wide Swath Synthetic Aperture Radar images are acquired frequently in polar regions and offer significantly higher resolutions compared to the 10km resolution of existing sea ice concentration products.
	F2	S2 Neural network structure must produce an output of the same dimension as the input and should achieve a weighted mean absolute error of less than 10% on unseen test data.
		S3 Develop a straightforward sequential Convolutional Neural Network and adapt U-Net and DenseNet architectures for concentration mapping, not segmentation or classification.
	F3	S4 Provide a search interface which accepts a date range and location, in order to search for matching SAR images.
		S5 Save trained neural network models so that estimates can be generated and served on request for input data matching the user's search criteria.
U2	F4	S6 Download and use the OSI-401-b sea ice concentration product for each day, using the FTP server provided by Copernicus Marine Environment Monitoring Service.
	F5	S7 Generate a concentration patch from the full chart, corresponding to the SAR image of interest. The concentration patch should be at the same resolution as the input SAR image.
		S8 Use the user's location to determine if the Northern or Southern hemisphere product should be used.
U3	F6	S9 Use OSI-401-b <i>total uncertainty</i> as a measure of possible error in each estimate provided by the existing product.
		S10 Generate an uncertainty patch label corresponding to the SAR image of interest, in the same way as for the concentration patch, at the same resolution.
	F7	S11 When training new neural network models, pass both the concentration patch and uncertainty patch as the label data for each SAR image.
		S12 Develop a custom objective function which calculates the error of each estimate based in the level of certainty present in the label generated from the existing product.

	F8	<b>S13</b>	The Pangaea dataset of Southern Marginal Ice Zone photographs provides 233 geotagged images of sea ice concentration conditions obtained in 2019 [9].
U4	F9	<b>S14</b>	Allow a user to select any of the images from the Pangaea dataset, then locate the corresponding OSI-401-b product and display a reduced patch centred on the observation's location for visual comparison.

*Table 3: Specifications table, showing the design specifications derived from the functional requirements above. Each of the specifications listed is specific enough to be directly implemented, or used in the implementation of a larger process. Specifications are defined to meet the problem's low-level detail-specific requirements.*

Data source specifications are particularly important for this investigation. Factors to be considered include format, spatio-temporal resolution, historical availability, update frequency and timeliness. In this regard, SENTINEL-1 SAR images were selected as the choice of input image data. The advantages of SAR images were discussed in the literature review, and the SENTINEL-1 mission in particular offers an extensive database and is designed specifically for research use like this. The choice to use the OSI-401-b SIC product as a source of label information was informed through discussion with the client. This global SIC product is a popular choice for characterising ice coverage over large areas, as was required for this investigation. This product also includes a sample-wise uncertainty value, which is based on a known algorithmic uncertainty, and a stochastic uncertainty introduced by the filtering process applied to the data before publication. This uncertainty field can be incorporated into the training process as a way to prevent the models learning features which are known to be inaccurate or at least uncertain. The in situ observation dataset selected for model validation was created by the client and was the most recent set of observations available at the time of writing. The choice of neural networks, label generation structure, and training procedure was informed by the discussion on deep learning presented in the literature review. The comparison of U-Net and DenseNet also presents the opportunity for insight into the applicability of encoder-decoder versus fully convolutional network structures once modified for the concentration mapping task. Lastly, the purpose and function of each user interface is defined so that each can be tailored to suit a specific need, and present information to the user in the clearest possible way in order to meet the user requirements.

## Acceptance Tests

The user requirements analysis section is intended to guide research and development efforts throughout the investigation. Acceptance tests then are the set of checks which will be used to determine if the proposed solution is sufficient to meet the user's requirements. Acceptance tests are mostly focussed on assessing whether the specifications listed in the previous section have been met, since these are easily quantified, and are not open to subjective assessment. Table 4 below lists the acceptance test procedures which were used to assess the success of the proposed solution in adhering to the specifications, and also in meeting the end user's

requirements. The relationship tree relates each test to the specifications, functions, and user requirements it is intended to assess.

Tree	ID		Details
U1	F1	S1	<b>T1</b> Ensure that the resampled SAR image spatial resolution is better than the 10km sample spacing of OSI-401-b.
			<b>T2</b> Verify that the temporal resolution of SAR acquisitions in polar regions is better than the delayed once daily charts published by existing models.
	F2	S2	<b>T3</b> Ensure that each neural network model's output dimension matches the resampled SAR image dimension.
			<b>T4</b> Track each model's weighted MAE during development and ensure a prediction loss on test data less than 10%.
	F3	S4	<b>T5</b> Ensure that the search interface is able to retrieve SAR images for the date range and location specified.
		S5	<b>T6</b> Ensure that estimates can be generated for arbitrary SAR images using a model which has been saved previously.
U2	F5	S7	<b>T7</b> Verify that concentration labels are generated at the same resolution as the resampled SAR images.
			<b>T8</b> Verify that the pixel coordinate interpolation is accurate enough to reliably locate features in the full OSI-401-b concentration product.
			<b>T9</b> Ensure that the correct hemisphere's concentration product is downloaded based on the specified location.
U3	F6	S10	<b>T10</b> Verify that the uncertainty patches meet the same requirements as the concentration patches, and that coastline features are in correspondence between the two.
	F7	S12	<b>T11</b> Verify that the uncertainty weighted loss function penalises estimation errors in uncertain regions less than in more certain regions within a given patch label.
U4	F9	S14	<b>T12</b> Ensure that the correct location and data are used for each image in the Pangaea database when searching for corresponding SAR acquisitions.

*Table 4: Acceptance test procedures table, showing the tests which will be used to determine if the user requirements have been met successfully. Each test is focused on assessing a particular specification, or aspect of a specification, but together the complete set of tests will indirectly assess the efficacy of the proposed solution in terms of the higher-level functional and user requirements.*

Each acceptance test in Table 4 above is designed to be straightforward to assess, and to provide a pass/fail result (i.e. the requirement is either met successfully, or it is not). The majority of tests (T1, T5, T6, T7, T9, T11, T12) were met by design during the development of all data processing procedures, but are still defined here for accountability. Tests T2, T8 and T10 were assessed with dedicated validation experiments, shown and discussed in the Data Pre-Processing section. Tests T3 and T4 were met as a matter of process during neural network development. A summary of acceptance test procedure results is presented in Table 8 in the Results section.

## 4. Computational Setup

---

Every stage of the processing pipeline, machine learning development, and model evaluation described in the following sections was conducted online using Google Colaboratory running Python 3. This presents a number of useful advantages, such as free access to GPU hardware accelerators for machine learning development. But is not without its challenges, such as limited runtime and issues around storage virtualisation. Even so, the speed increase offered by the use of GPUs compared to CPUs is so significant that Colaboratory was an obvious choice in the absence of any other available hardware accelerators. The specific Python modules used at each stage will be mentioned as they are used, but for now it is worth mentioning that all image processing was done using *OpenCV*, *Matplotlib* was used to display all results, and *Keras* was used for all machine learning development, using the *TensorFlow* backend.

One of the complications which arise when trying to work with machine learning using an online development environment such as Colaboratory is the need to access large datasets for training each model. While premium services like Google Cloud or AWS EC2 allow the user to define custom runtime instances, which could include all the data needed for development, free services like Colaboratory always spin up as a fresh instance, and any previous work will have been lost. To make the most of this free service, a Google Drive folder was mounted as a remote storage device to each new instance. All files were then saved into this remote drive so that work could continue from one instance to the next. This method of data management has some significant speed limitations when dealing with larger datasets, which had to be accommodated in the processing pipeline. More discussion on this issue will follow below.

The development process in this investigation can be broken into three broad sections: data pre-processing, neural network development, and serving targeted predictions. Each of these sections of development was carried out in its own Google Colaboratory notebook, all of which can be found in the project's GitHub repository, which also includes the batched datasets used for training and evaluating each model, and the trained models themselves. The following three sections will discuss each of these phases of development in detail.

*GitHub Repository Link:*

[https://github.com/stefandominicu/FYP\\_ML\\_SIC](https://github.com/stefandominicu/FYP_ML_SIC)

## 5. Data Pre-Processing

---

This section will discuss the details of how training data was acquired, processed, labelled, and set up for the machine learning development process. It is dedicated to explaining the data processing pipeline which was developed in order to handle the vast number of SAR images used in each dataset, and the process of transforming the data into a form which could be readily used in machine learning development. At critical stages in the discussion which follows, validation results are presented which demonstrate the efficacy of the algorithms and scripts used at each stage. Finally, an overview of the datasets used in this investigation is presented, with some justification of the various parameter choices. All processing scripts can be found in the Data Acquisition & Processing notebook, available on GitHub.

### Creating a Labelled Dataset

In supervised machine learning, much of the challenge often lies in finding or creating suitable datasets on which to train the machine learning models. The following subsections discuss the methods used for creating the datasets used in this investigation and address many of the unique challenges faced in this process.

#### Define Parameters

The first step when creating a new image database was to define the key parameters: region of interest, start and end dates, and a descriptive name. This process is made more interactive in the Google Colaboratory environment by using *Forms* to accept user input in a more intuitive way compared to defining variables directly in the Python cell. *Forms* also allow for some simple data validation to be incorporated in fields such as date entries or drop-down selections, which offers some level of protection in an otherwise unrestricted development environment.

#### Download SENTINEL-1 Images

The SENTINEL-1 mission consists of a two-satellite constellation, in identical polar orbits, separated by 180 degrees in the orbital plane. The mission is managed by Copernicus, which is part of the European Space Agency. Each platform uses a synthetic aperture radar (SAR) instrument to scan the Earth's surface in a variety of modes. SENTINEL-1 SAR data is made free to use for research, and can be accessed in a variety of ways, which makes it well-suited for this task. Products are provided at a number of processing levels, ranging from raw measurement data, to high-level analysis of the satellite data covering various domains. For the purpose of this investigation, the product type was restricted to ground range detected, extra wide swath, dual polarisation images. Ground range detected (GRD) means that the measurement data has been projected onto the WGS84 Earth ellipsoid, and can therefore be treated simply as an image, without much concern for the complications that come with some other types of radar measurements. Extra wide swath (EW) is the acquisition mode recommended for coastal monitoring and sea ice detection and is comprised of five sub-swaths which are combined to form an approximately square aspect ratio. Lastly, dual polarisation (specifically HH+HV) refers to the SAR

instrument transmit and receive mode. In this configuration, the satellite transmits a single horizontally polarised radar signal, and receives both horizontally and vertically polarised signals through two receivers in parallel.

Each new acquisition is published within hours of being sensed, and can be viewed and downloaded through the Copernicus Open Access Hub, or via an API hub which is also maintained by the Copernicus team [10]. However, each product can range in size from 400MB to over 3GB, and typically consists of around 10,000 x 10,000 pixels. This is prohibitive when it comes to creating a dataset large enough to train a machine learning model, and the individual image size is also much too large for typical neural network architectures. Luckily each product is accompanied by a Quicklook file, which is an averaged and decimated PNG image which is used when previewing the product via the Open Access Hub [10]. For the dual polarisation products used in this investigation, the Quicklook image is structured as an RGB image, where the red channel contains the straight polarisation data (HH), the green channel contains the cross-polarisation data (HV), and the blue channel contains the average of the two different polarisation measurements. These files can be downloaded independently, without needing to download and store the much larger full measurement set.

To automate the process of downloading the SAR images in the Google Colaboratory environment, two Python packages were used. *Sentinelsat* was used to handle search queries and returns the metadata of all products matching the desired acquisition type, within the dates and regions of interest. *Requests* was then used to issue the HTTPS calls needed to download each product's Quicklook image and save the file to a Google Drive folder. This original image was then resampled using *OpenCV* to ensure that all input images conformed to the same shape, and the resampled image was then saved along with the original image.

A noteworthy limitation of using Google Drive for dataset storage was noticed at this point. Each of the downloaded images were saved with a unique filename (the unique identifier, or *uuid*, provided by Copernicus), and so there was no reason to order the files into directories beyond that – one directory for the original images and one for the resampled images. However, this quickly became an issue with datasets containing more than a few hundred images. When trying to load specific images for processing or training, the Google Drive interface would often timeout, likely due to the limitations of storage virtualisation. After further investigation, it was found that this is a known issue with Google Drive, which does not perform as expected with many files in a single directory. The solution then, is to subdivide the dataset into a directory tree, reducing the number of files in each directory to a manageable number. This was done according to acquisition date, using the following path convention:

*image\_folder / year / month / day / year\_month\_day\_uuid.png*

The date information was prepended to the filename so that the file could be located within the directory tree given only the filename string, without needed to query the Copernicus database to retrieve the corresponding date each time. Organising the images in this way allowed Google Drive to remain stable for arbitrarily large datasets,

and also reduced the access time per image, which helped to speed up the label generation procedure discussed later. However, single file access was still a significant bottleneck during neural network development, and this was addressed with the batching process discussed later in detail.

## Download OSI SAF Concentration Charts

The Ocean and Sea Ice Satellite Application Facility (OSI SAF) is run by the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) and provides a number of analysis products related to sea ice monitoring. These products are compiled using various satellite inputs and are each geared to particular applications. For this investigation, a global sea ice concentration product (OSI-401-b) was used. Of interest in this application is the fact that the product contains a layer of concentration values on a 10km grid spacing for both the Northern and Southern hemispheres, as well as a layer for uncertainty values based in the quality of the input data used to generate each concentration pixel value. This particular product is generated using measurements from a passive microwave (PMW) radiometer instrument on Defence Meteorological Satellite Program (DMSP) platforms.

This product is published daily and is distributed via a few different FTP server mirrors. The Copernicus Marine Environment Management Service (CMEMS) maintains one such server, which was used as the source for all OSI-401-b files. This was again automated in Python, to run within Google Colaboratory and save the product files to a Google drive folder. The Python package *ftplib* was used to retrieve the files for each day included in the desired range. The files were then saved in a directory tree ordered by year and month, for the same reasons as were discussed in the previous section regarding SENTINEL-1 data.

## Generate Label Patches

At this point both the SAR images and the concentration estimates have been downloaded. The remainder of this section is dedicated to manipulating data into a form which can be easily used for machine learning development. The objective is to generate a concentration label for each SAR image of the same resolution, so that neural network models can be trained to estimate the concentration map for an arbitrary SAR input image.

The central problem in doing this is identifying the corresponding concentration value for each pixel in a given SAR image. The *SentinelSat* package was used to query the product metadata using the Copernicus *OData* standard. This was done for each SAR image being labelled. Included in the product *OData* is a ‘*footprint*’ attribute, which lists the latitude and longitude of the four corners defining the image footprint on a WGS84 ellipsoid. The order in which these points are listed is determined in the satellite frame of reference and is shown in the Figure 2.

# Pixel Coordinate Interpolation Scheme

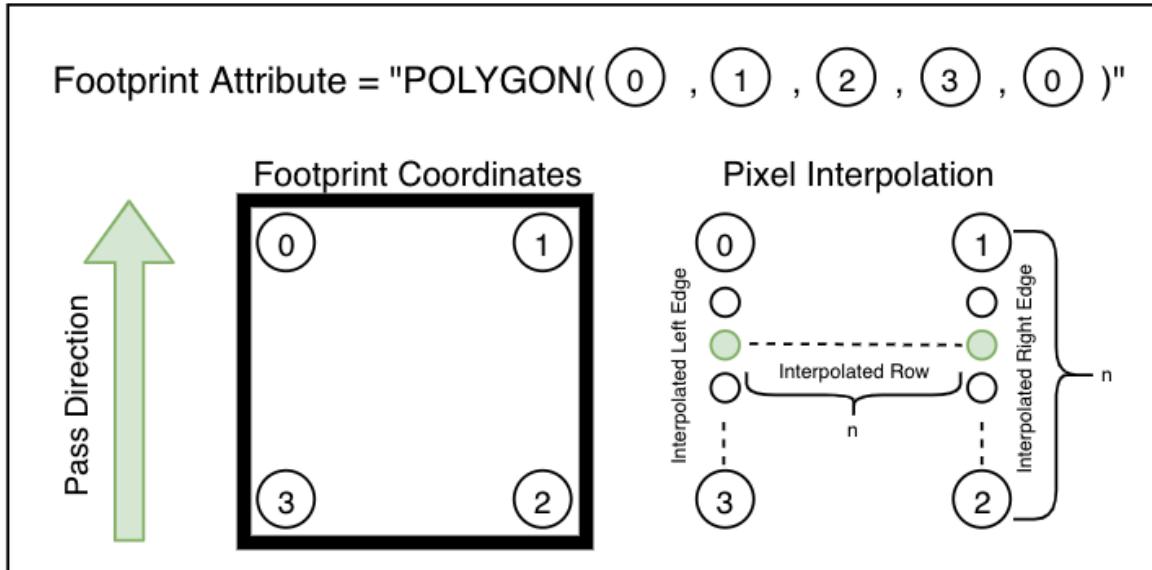
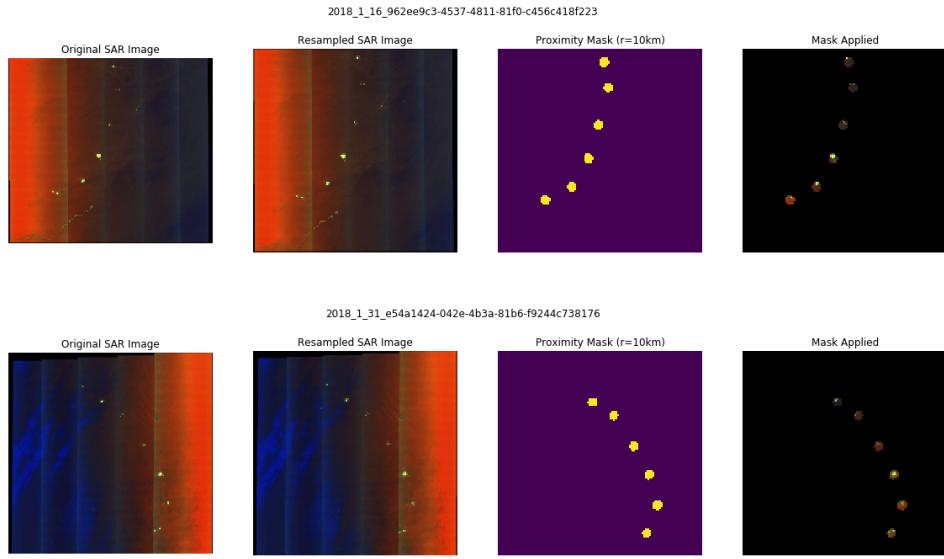


Figure 2: Pixel coordinate interpolation scheme. This diagram illustrates how the OData ‘footprint’ attribute is structured, and how it was used to generate a grid space interpolation of WGS84 coordinates, allowing the real-world location of each pixel in a SAR image to be determined and used for further processing.

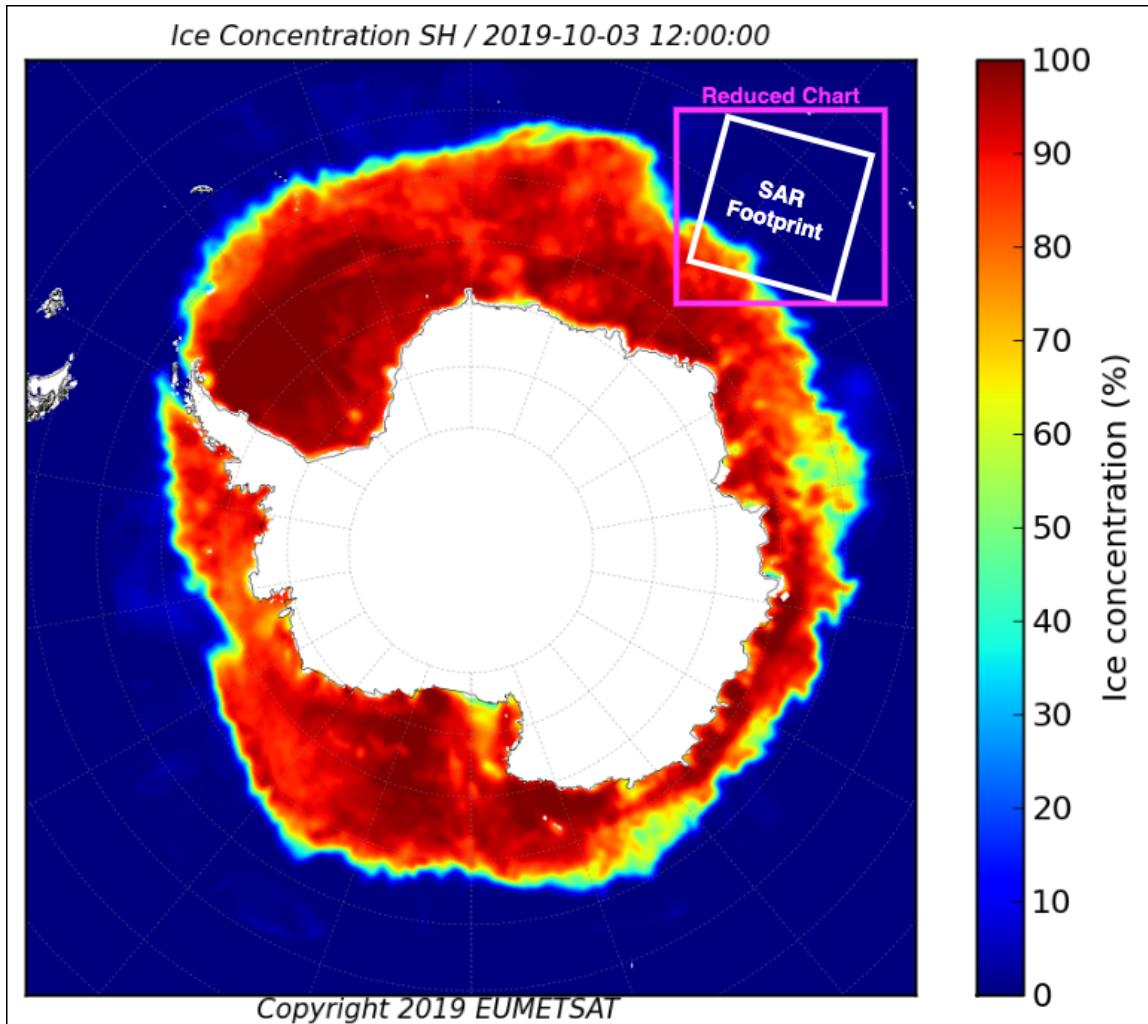
Once the coordinates of all four corners are known, finding the coordinates of any pixel ( $x, y$ ) in grid space can be achieved by interpolation. The details of this interpolation scheme are depicted in the Figure 2 above, and can be visualised as a ladder: first the coordinates of each pixel on the left and right edges are interpolated between corners as shown above(the ladder uprights), after which each row (or rung on the ladder) can be iteratively interpolated between pairs of coordinates on the left and right edges. Note that this method assumes negligible effects due to the curvature of the Earth, which is reasonable given the small footprint size. To verify that the interpolation scheme would give reliable results for each pixel, and to directly address acceptance test T8, an experiment was devised to visualise key locations within the image and determine whether or not the interpolated coordinates corresponded to the real-world coordinates for features within each image. For this test, a chain of small islands in the Southern Ocean were chosen as a test region. These particular islands were chosen because they offer a number of small, easily visible features on a scale that is similar to the size of the SAR images over that region. The coordinates of each island were found using Google Earth and used to generate a mask for each SAR image, masking all pixels which were not within 10km of any one of the islands. The *GeoPy* Python package was used to calculate the distance between each of the islands and the interpolated coordinates of each pixel in the image. A selection of the results of this test are shown in Figure 3 below.



*Figure 3: South Sandwich Islands pixel coordinate interpolation validation test results for two SAR images acquired in January 2018. The original SAR image is shown first, followed by the resampled image, which forces a square aspect ratio. The proximity mask is shown third, where each pixel is masked if its interpolated coordinates are not within a radius of 10km from the known coordinates of any of the islands in the chain. This mask is then applied to the resampled image, and shown fourth. The results show each island is clearly visible within the corresponding mask bubble, and therefore the coordinate interpolation algorithm is deemed to be accurate and reliable enough to use for the remainder of the patch label generation process.*

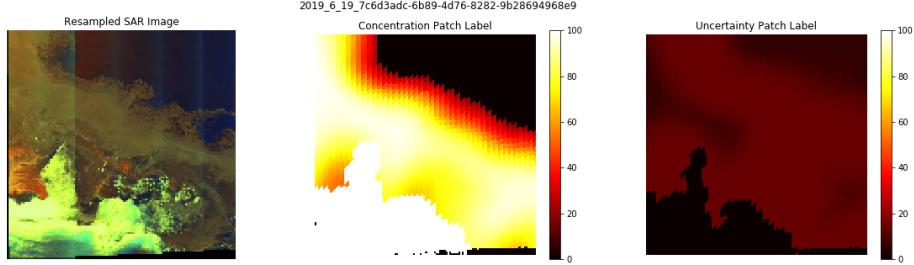
The results of this test clearly show that each of the islands appears within the corresponding mask bubble, no matter the image orientation, which serves to validate the accuracy of the interpolation scheme, showing that the WGS84 coordinates of each pixel in grid space can be reliably and accurately interpolated using only the known footprint coordinates.

Now all that remains to do to generate the concentration patch label is to iterate through each image pixel and use its interpolated coordinates to find the closest sample in the corresponding OSI SAF concentration chart. In practice, this was done in two stages in the interest of computational efficiency. First a reduced subset of the concentration chart was generated, ensuring that the SAR footprint was contained within the reduced chart. This is illustrated in Figure 4 below. Then for each pixel in the SAR image, the distance to each sample in the reduced concentration chart was calculated, and the value of the nearest sample taken as the value for that pixel in the concentration patch. Generating a reduced concentration chart in this way is significantly more efficient, because it means that in the following step there are far fewer concentration samples to be compared per pixel.



*Figure 4: OSI-401-b concentration chart illustration, with an arbitrary SAR image footprint overlaid to show the significance of first reducing the chart size to focus only on the portion of the concentration chart which is near to the SAR image. This significantly reduces the computational cost of finding the nearest concentration sample for each pixel in the SAR image.*

For the sake of explanation, this section has only mentioned generating a concentration patch. In fact, an uncertainty patch was also generated at the same time, using exactly the same method, but taking values from the total uncertainty layer included in the OSI-401-b product. Both of these patch labels were saved as PNG files following the same name and directory conventions as with the SAR images. Figure 5 below shows an example of a resampled SAR image, and the corresponding concentration and uncertainty patches, each generated as described above. The pixel spacing ( $rg \times az$ ) in this particular image is  $(3.26\text{km} \times 3.43\text{km})$ , which is an improvement compared to the  $10\text{km}$  sample spacing found in the OSI-401-b concentration chart. Since all SAR images used in this investigation were approximately the same size, both in terms of resolution and footprint dimensions, this means that even after decimating the Quicklook images, the pixel spacing is still superior to the sample spacing used in OSI-401-b. This pixel spacing is also dependant on the arbitrary resampling size used when processing each SAR image, and so could be adjusted if a further reduction in pixel spacing was required for a particular application.



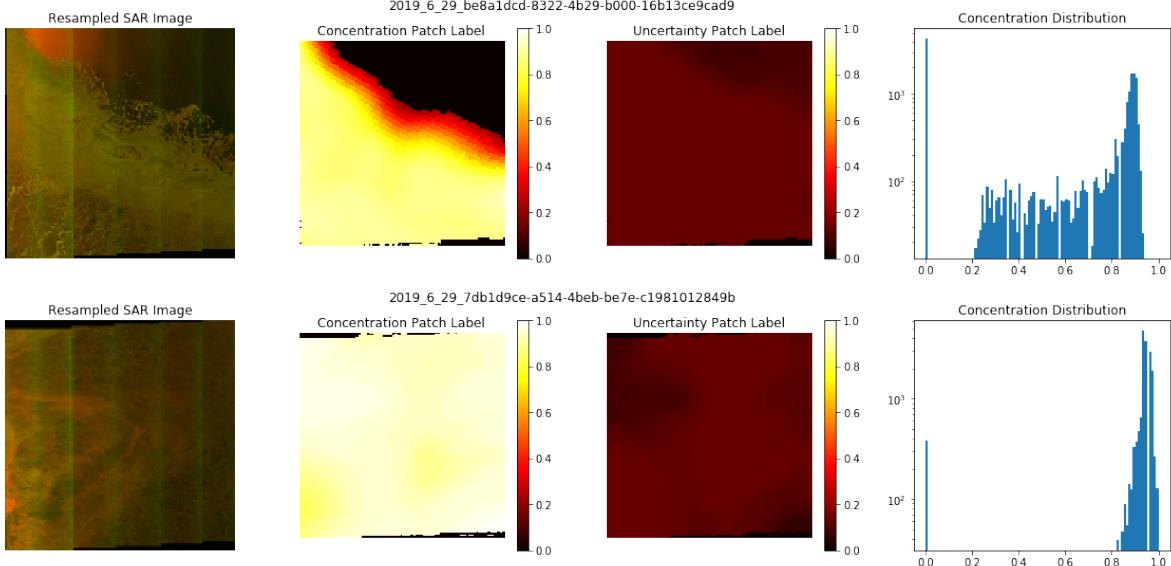
*Figure 5: Patch label validation, showing an example image-label set. The resampled SAR image is shown on the left, the concentration patch label in the middle, and the uncertainty patch label on the right. By visual inspection, it is clear that the features present in the SAR image are also reflected in both patch labels. This validates the label generation procedure, and also shows that the generated labels are at a higher resolution than the chart provided by the OSI-401-b product. The pixel spacing in this example is 3.26km x 3.43km.*

At this stage it is clear that patch labels can be generated for each SAR image in the dataset. Critically, these labels can be generated at the same resolution as the input images, which allows the machine learning models that follow to estimate ice concentration charts at a higher resolution than the OSI SAF data with which they were trained. This demonstrates that acceptance test T1 was passed successfully.

### Selective Filtering

Previous work using SAR images for machine learning have usually selected a handful of images with which to work, and in doing so it is ensured that the data being learned is in some way representative of the property of interest – in this case, sea ice concentration. However, in this experimental setup, all SAR images from the region of interest, and within the start and end dates selected were downloaded. This meant that many of the images contained entirely 0% concentration (open water), or 100% concentration (consolidated ice or land). These images are of no interest, since they do not contain any variation in ice concentration and will skew the training process in favour of single-value concentration patch estimations, which is undesirable. A small percentage of SAR images were also found to not be roughly square in aspect ratio, which means that the process of resampling the image to a fixed square aspect ratio significantly distorts the features contained in the image.

The solution proposed in this investigation is to filter the dataset prior to training and retain all images which are roughly square in aspect ratio and contain interesting variations in ice concentration such as at the ice edge where the concentration values in a single image span most of the 0% - 100% range. To gain some insight into the distribution of concentration values in each image, a histogram was plotted for a subset of the labelled images, which is shown below in Figure 6. Pixel-wise variance in the concentration patch label was used to quantify the spread of concentration values within each image, and a threshold was used to filter the dataset based on the concentration variance – images with a suitably high variance were marked as acceptable, while the others were excluded from all further use.



*Figure 6: Pixel-wise concentration distribution.* In this figure, the resampled SAR image, concentration patch label and uncertainty patch label are shown for two representative samples in the dataset. The histogram on the right shows the distribution of concentration values present in the concentration patch label for each sample. The first sample (top) is clearly more “interesting” in terms of concentration features, since it includes a section of ice edge, while the second sample (bottom) contains almost no variation, with most of the concentration values being greater than 80%. A variance threshold was used to filter all samples, ensuring only “interesting” were included.

The threshold value was tuned iteratively by viewing the accepted images generated with each threshold value and assessing whether the desired outcome had been achieved. If images which were visually deemed to be “uninteresting” were still being included, the variance threshold was increased, and the process repeated. This selective filtering process reduced the size of the dataset significantly but ensured that the data were representative of the concentration variations which are the focus of this investigation. Datasets spanning the Arctic Circle typically retained 15-20% of the total number of images from that region, and Antarctic datasets retained 30-35%. This difference is likely due to different acquisition schedules over the two poles, which is determined by the Copernicus team. Nevertheless, filtering the datasets in this way ensures that all images in the dataset contain “interesting” concentration information, representative of the features to be learned by neural networks later.

## Batching & Data Augmentation

When training neural networks on large datasets, and particularly image data, it is often preferable to use a batch training method, since it is far less resource demanding. *Keras* provides a number of helpful tools in this regard, and the *ImageDataGenerator* class is particularly powerful when it comes to feeding batches of image data into a neural network. The *ImageDataGenerator* class only facilitates processing of a single image at a time, so to process all three images (SAR, concentration, and uncertainty), three *ImageDataGenerator* objects were created, with identical parameters to ensure that the three images remained in correspondence. Data augmentation, which was comprised of random flips and rotations, was optionally applied to the dataset being used, so that its effect on the training process could be evaluated for each model. i.e. all tests were conducted both with and without data augmentation to compare the

efficacy of the augmentation process. A 5x5 median filter was also applied to all concentration and uncertainty patches, regardless of whether or not data augmentation was being used. This filter helped to smooth out the rather grainy patch labels, while preserving edges, as shown in Figure 7 below. The granularity in the unfiltered patches is due in part to the poorer 10km resolution offered by the OSI SAF concentration chart used but may also be affected by the single nearest neighbour approach used in generating the label patches. Nevertheless, the median filter applied had the desired effect of smoothing the patch without blurring.

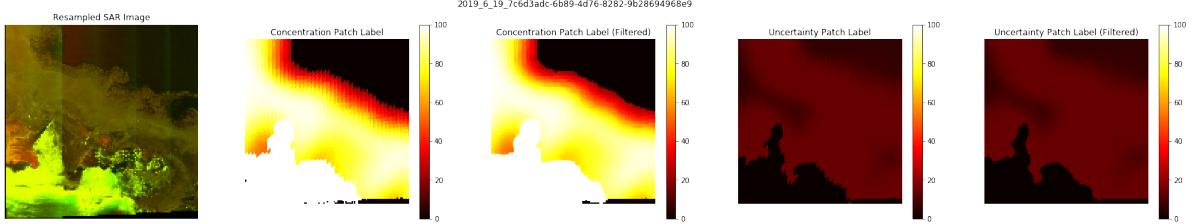


Figure 7: Median filter (5x5) applied to concentration and uncertainty labels. Both unfiltered and filtered patch labels are shown, so that the effect can be visually inspected. The median filter does well at smoothing the patch labels without blurring the image, and also doesn't introduce any concentration values which were not already present in the vicinity of each pixel to begin with.

However, as a result of using Google Drive for image storage, file access speed became a bottleneck in the training process. i.e. the GPU was not able to work continuously at full capacity, because images could not be read from file quickly enough. However small this delay, it occurred on every batch, through every epoch, which significantly slowed the overall training process. This was overcome by pre-processing the batches of images data and saving each batch to file. First the list of samples in the dataset was split for training and testing, and for each of these segments a batch set was generated. The *NumPy* Python package's *savez* method was used to save a dictionary containing a batch of SAR images, concentration patches, and uncertainty patches all to a single file. With a batch size of 16, the number of file access calls required when loading a new batch for training is reduced by a factor of 48, while also containing all calls to a single directory instead of jumping to a new directory for each of the 48 files which make up a batch (recall the directory tree structure used to store each image and patch label).

With each batch of fully pre-processed data now saved to file, all that is required is a way to recall these batches as needed and separate the input SAR image from the label patches. Again, *Keras* has a straightforward way to do this – the *Sequence* class. This class is designed to integrate well with all of the built-in *Keras* training and evaluation methods and allows users to define exactly how their data is handled and presented to the neural network. For this investigation, the *Sequence* class was extended to provide methods for accessing individual elements (either SAR, or one of the patch labels) which was used extensively when displaying results. When feeding data into a neural network, the class was coded to provide the data in the tuple format expected by the *Keras* training and evaluation methods, and could also be configured to only supply an input SAR image (and no label data) to allow estimations to be made with no knowledge of the correct concentration patch. The patch labels could then be retrieved

externally for visual comparison with the model estimations, as shown later in the results section.

## Summary of Pre-Processing Pipeline

The graphic in Figure 8 was prepared in order to summarise the last few sections, which have discussed in great detail exactly how all datasets were created and processed. The intention here is to give a slightly more abstracted view of the pre-processing pipeline, so that the overall flow of data can be understood, without getting bogged down in the details of each step.

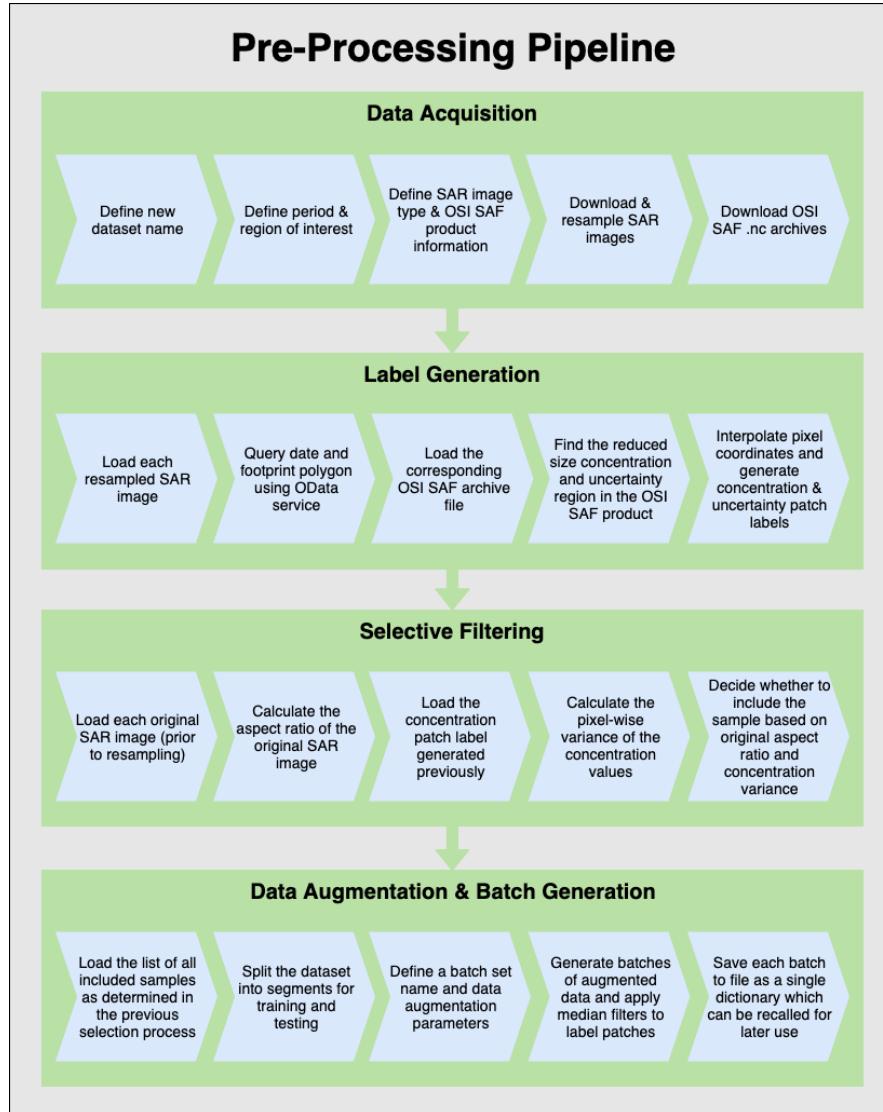


Figure 8: Summary of the data pre-processing pipeline developed for this investigation. This graphic is a summary of the detailed discussion over the previous sections on dataset management. The pipeline was divided into four major stages, and this graphic illustrates how each stage links together, as well as how data moves through each stage in the pipeline.

Note that Figure 8 does not include the extended *Sequence* class which was used to feed the saved batches to the neural network for training. No additional pre-processing takes place after the batches are saved, and the *Sequence* objects are simply used to recall each batch and present it to the neural network in the appropriate form.

## Summary of Datasets Used

After the pre-processing pipeline had been designed, built, and validated as described in the previous sections, four batch sets were generated to be used for all machine learning development which followed. Since a large part of the problem being investigated is the lack of transferability observed in existing numerical models from Arctic to Antarctic environments, datasets were generated from both regions, allowing for some comparisons to be made later on. From each of these datasets, two batch sets were generated, one using data augmentation, and one not. This will allow for some evaluation of the efficacy of data augmentation in this situation. Table 5 shows the parameters of each batch set. Batch set names follow a naming convention of “North” if the dataset was taken from the Arctic, and “South” if the dataset was taken from the Antarctic. If data augmentation was used in the batch generation process, then “\_A” is appended to the batch set name. A similar naming convention was used for the neural network models discussed later.

Batch Set	Start Date End Date	ROI	Data Augmentation	Total Images	Selected Images
North	2019-01-01 2019-07-01	Arctic Circle	No	21962	3797
North_A	2019-01-01 2019-07-01	Arctic Circle	Yes	21962	3797
South	2019-01-01 2019-07-01	Antarctic Circle	No	6060	2081
South_A	2019-01-01 2019-07-01	Antarctic Circle	Yes	6060	2081

Table 5: Summary of batch sets used in this investigation. Each batch set has a unique name, and is created from a parent dataset as described in the previous section. The table also shows the total number of images acquired in the parent dataset, and the number of images selected by the concentration variance threshold filter.

Both datasets were limited to the same time period, January 2019 to June 2019, and their regions of interest (ROI) were either the Arctic Circle or the Antarctic Circle. Table 5 includes the total number of pictures downloaded for the time period over the defined region, as well as the number of images which were selected in the concentration variance threshold filtering process.

It is worth noting that over three times as many SAR images were available within the Arctic Circle compared to the Antarctic Circle, despite the image type, start and end date, and all other parameters being the same. This is likely explained by the way in which the Copernicus team define the acquisition schedules so obtain as many useful images as possible. Recall that the GRD EW SAR products used in this investigation are intended for use in coastal monitoring applications, so there is no reason for the satellites to take measurements over large land masses. The Arctic Circle contains many small land masses, and so there is proportionally more coastline compared to the Antarctic Circle where the continental land mass presents no interest for the acquisition mode being used. Therefore, many more acquisitions will be made over the whole Arctic region, while in the Antarctic, acquisitions are only made over the coastline and MIZ.

## 6. Neural Networks

This section on neural network development is dedicated to explaining the details of how each model was implemented and iteratively developed. The training and evaluation processes are discussed afterwards, with comments on some difficulties that were encountered along the way. All code used in this section can be found in the Neural Network Implementation notebook, available on GitHub.

### Neural Network Architectures

Three different neural network architectures were used in this investigation, with the intention of comparing the performance and complexity of each, in order to determine which was best suited to the application of predicting sea ice concentration from resampled SAR images. In each case, the model's hyperparameters were iteratively tuned until the mean absolute error on unseen test data was below the 10% tolerance required to successfully pass acceptance test T4. The three model types are discussed below, followed by a summary of the final parameters used for each architecture. All neural network models were developed using *Keras*.

#### Convolutional Neural Network

Before investigating any exotic model architectures, a relatively straightforward sequential Fully Convolutional Neural Network (FCNN) was implemented to act as a baseline with which to compare the predictions of more complex models implemented later. Each layer was comprised of a convolutional layer with  $n \times (3 \times 3)$  filter kernels and ReLU activation functions, followed by a dropout layer with an adjustable dropout percentage. A simple two-layer version of this architecture is shown in Figure 9 below.

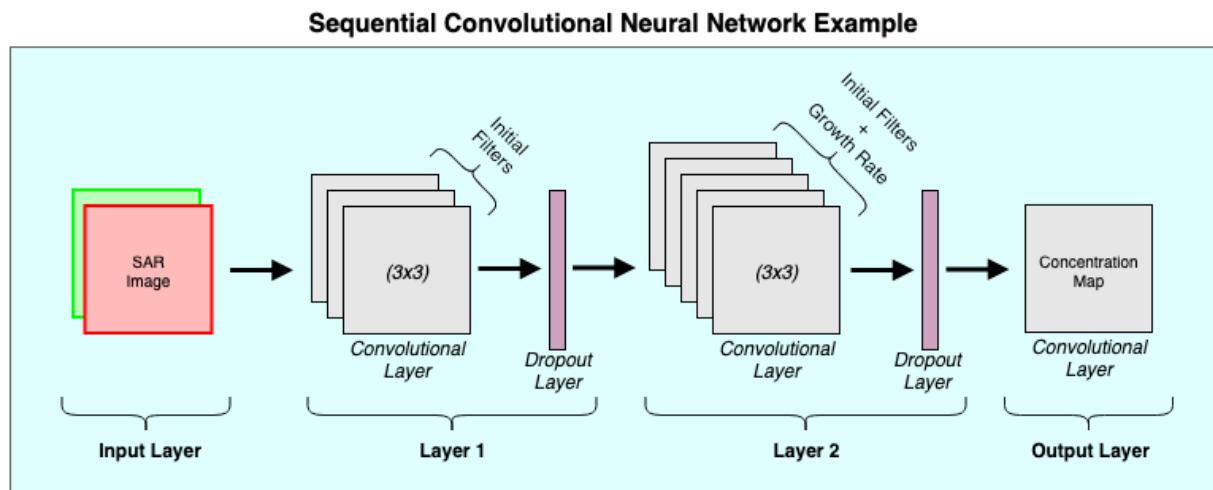


Figure 9: An illustration of the sequential FCNN architecture. This example contains two hidden layers, and illustrates how the input SAR image data is propagated through the network, and how the number of convolutional filters is increased at each layer.

The convolution operators were padded so as to maintain the same image resolution throughout the network, which meant that the output layer was simply a single convolutional filter with a Sigmoid activation so that all pixel values fall in the range (0,1) representing the estimated concentration percentage. Each model was defined by

the following hyperparameters: number of layers, initial filters, growth rate, and dropout rate. The roles of the number of layers and dropout rate hyperparameters are fairly self-explanatory. Initial filters defines the number of filter kernels in the first convolutional layer, and growth rate define how much the number of filters will be increased by after each layer. The final configuration used in this investigation is listed in Table 6 below.

## U-Net Adapted

The U-Net architecture was introduced in 2015 and has been successful in many image segmentation tasks [8]. Since segmentation in this context is little more than pixel-wise classification, this architecture can be easily modified for concentration mapping, which is essentially pixel-wise regression. By simply modifying the final output layer, all of the advantages of this network structure are retained, and very little design work is required. Figure 10 below shows the U-Net architecture as described in the original paper.

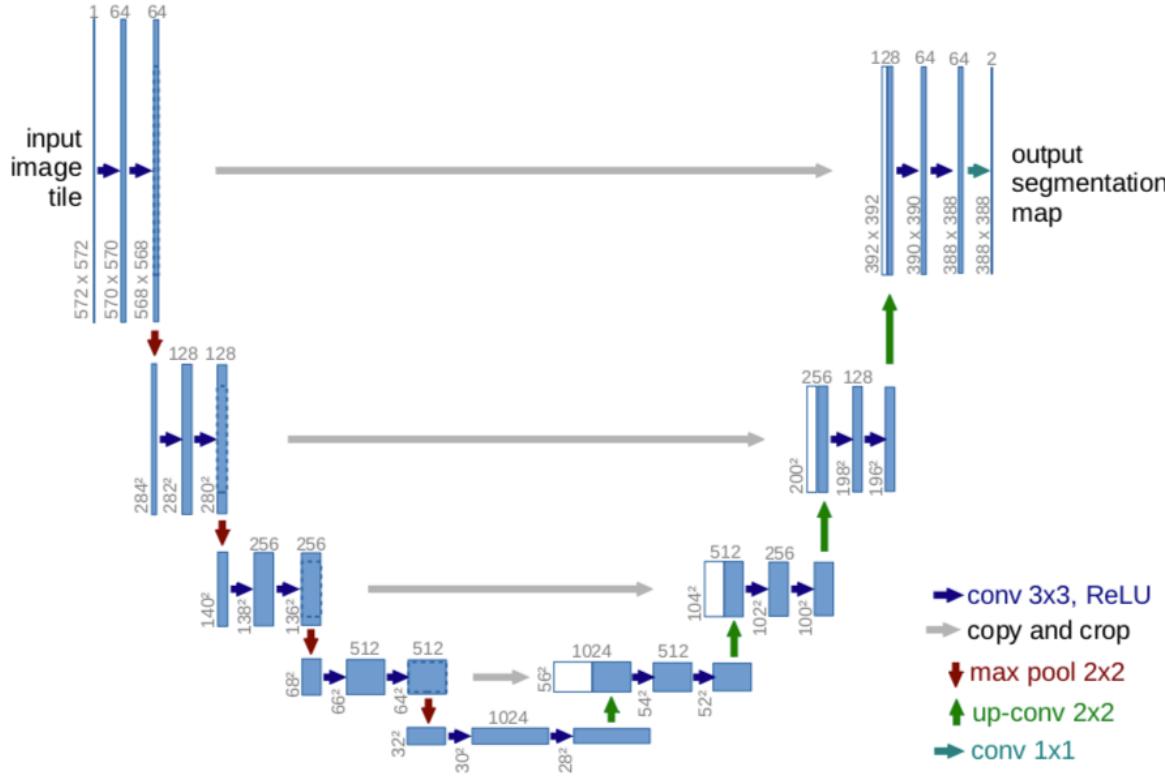


Figure 10: A diagram illustrating the original U-Net architecture. With each additional layer, the number of filters is increased, but feature maps are also passed across to layers closer to the output, which allows for better gradient propagation and feature reuse [8].

A Keras implementation of the U-Net architecture found on GitHub was used for the bulk of the model definition [11]. Some simple changes were made to the convolutional padding parameters so that the output dimension was the same as the input dimension, and the segmentation output layer was replaced with a single filter convolutional layer with a Sigmoid activation function, identical to the output configuration used in the previous section's sequential CNN. The U-Net models were defined by the same set of hyperparameters as the sequential CNN above, however the number of layers parameter

refers to the depth of the network in this case, not its length. Initial filters, growth rate and dropout rate behave in the same way as above. The final configuration used in this investigation is listed in Table 6 below.

### DenseNet Adapted

Building on the success of skip connections as demonstrated in U-Net, another unique architecture was introduced where convolutional layers were densely connected to one another, called DenseNet [7]. The network was originally intended for use on classification problems, and Figure 11 below describes how the dense blocks (containing densely connected convolutional layers) are arranged sequentially with some transition operators in between each block. Although this architecture is intended for classification tasks, it can be easily converted for concentration mapping by removing max pooling layers (which reduce the feature map dimension), padding all convolutional layers to maintain the original image dimension, and replacing the SoftMax output layer with a single filter convolutional layer and Sigmoid activation function, as was used in the previous two architectures.

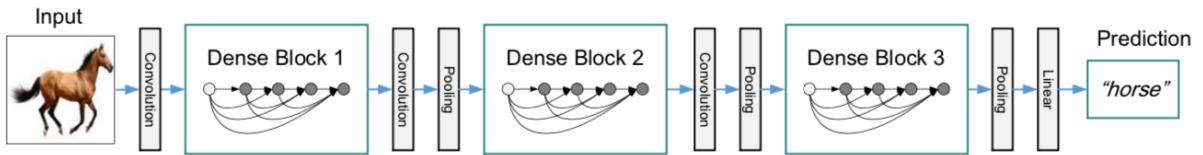


Figure 11: A diagram illustrating the original DenseNet architecture. Each dense block consists of densely connected convolutional layers. Although it was originally designed for classification, the structure was easily modified to predict the concentration maps needed in this investigation [7].

Although the original paper's work is available in an open source format, the original implementation used *Caffe*. Instead, an alternative *Keras* implementation of this architecture was used as a template for this investigation, found on GitHub [12]. Besides the changes to convolution padding and the removal of max pooling as described above, the model's weight decay hyperparameter was also set to zero. The purpose of this hyperparameter is to penalise excessively large weights in any of the model layers, in order to prevent overfitting to particular training examples, and overreliance on particular feature sets. However, this interfered with the training process by diluting the training loss value. So instead of minimising the objective function by estimating more accurate concentration maps, the model training process simply minimised the layer weights, leading to very poor visual performance despite the training loss following the expected exponential decay trajectory. This was solved by removing the weight decay penalty, and relying only on the loss due to the estimated concentration map itself. The final configuration used in this investigation is listed in Table 6 below.

### Summary of Models Used

For each of the model architectures described above, a number of variants were trained and evaluated. Each structure was iteratively tuned over a number of generations, until the estimation mean absolute error on unseen data was below the acceptable accuracy threshold defined in the Requirements Analysis section. The final configuration of each model is listed in Table 6 below.

Base Architecture	Structure	Description
CNN	Layers: 10 Initial Filters: 32 Growth Rate: 32 Dropout Rate: 0.2 Parameters: 3,043,937	A straightforward sequential CNN, where each layer is comprised of a convolutional layer followed by a dropout layer.
U-Net	Layers: 4 Initial Filters: 128 Growth Rate: x2 Dropout Rate: 0.5 Parameters: 124,108,545	Based on the successful image segmentation network, this model used skip connections to feed low-level features forward to layers near the output.
DenseNet	Dense Blocks: 4 Dense Layers: 8 Growth Rate: 8 Dropout Rate: 0.2 Parameters: 186,459	Based on the novel classification model architecture, this model was modified to retain the same image dimension throughout the network.

Table 6: Summary of neural network models used in this investigation. The details and important differences are shown in the table, and more details on each model can be found in the preceding sections.

## Training & Evaluation

Model training and evaluation was conducted with a number of combinations of model architecture and batch set selection, but all models were trained for 50 epochs in total. The available batch sets are shown in Table 5. To differentiate the various different trained model configurations, the following naming convention was used:

$$\text{ModelName\_TrainSet\_TestSet\_Augmentation}$$

For example, *CNN\_N\_N* denotes a model based on the sequential CNN architecture, which was trained and tested using the Northern hemisphere dataset, with no data augmentation used. *DenseNet\_NS\_S\_A* denotes a model based on the DenseNet architecture, trained first on the augmented Northern hemisphere dataset for a fraction of the total number of epochs, and then on the augmented Southern hemisphere dataset for the remaining epochs, before being tested on the Southern hemisphere dataset. In these cases, the Northern/Southern epoch ratio was 32/18 which is in proportion with the number of images available in each dataset. The full list of model training configurations used in this investigation are shown in Table 7 below, which summarises the results achieved in each configuration.

## Objective Function

One of the primary objectives of this investigation was to train a model on data from an existing numerical model, without inheriting all the inaccuracies of that model. The proposed solution presented in this section was to incorporate the OSI-401-b uncertainty field into the objective function calculation so that the model would not be heavily penalised for estimation errors if the concentration label had a high degree of uncertainty. The intention was for the neural network model to learn from features where the existing numerical models are certain. With sufficiently large datasets, the

model would be exposed to enough features with a high level of certainty for the overall training process to be successful.

The section on patch label generation above described how both concentration and uncertainty patches were generated for each SAR image in the dataset. During training, the customised *Sequence* object was used to feed both of these normalised patches to the *Keras fit\_generator* method as the desired output. Each model was compiled to use a custom loss function which first separated the label into its two layers so that the concentration and uncertainty patches could be used independently. The loss function then computed the pixel-wise prediction error and scaled it by the pixel-wise “certainty”, which was defined as “one minus uncertainty” for each pixel. Figure 12 shows the uncertainty weighted mean absolute error loss function Python definition.

```

3 def uncertainty_weighted_MAE(y_true, y_pred):
4     # y_true contains the concentration label, and the uncertainty as two two channels
5     y_target      = y_true[:, :, :, :, :1]
6     uncertainty  = y_true[:, :, :, :, 1:]
7     # Calculate the weighted MAE
8     loss = K.abs(y_pred - y_target)
9     loss = loss * (K.ones_like(loss) - uncertainty) # Scale the error by the 'certainty' of the label
10    return K.mean(loss, axis=-1)
```

Figure 12: Python definition for the uncertainty weighted mean absolute error function used as the objective function for all models in this investigation. The customised Sequence class passes both concentration and uncertainty patches through as the target output. The loss function separates the two layers so that each can be used independently in calculating the loss for each sample’s estimation.

The reasons for defining a custom loss function like this are so that multiple pieces of information can be utilised in determining the loss for each sample, and so that the loss can be constructed as desired for any application. In this case, the estimation error is scaled by the pixel-wise certainty, which prevents the model from learning uncertain details from the existing numerical models as readily as it would without any kind of uncertainty compensation.

### Swath Stitching Artifacts

Although the SENTINEL-1 Extra Wide Swath (EW) SAR products are the most suitable for this application, as recommended by Copernicus, their composition presented a challenge during training. Each EW SAR image is comprised of five sub-swaths, which have been stitched together to form the full image. The vertical lines seen in the resampled SAR images in Figure 6 for example, are the result of this stitching process. This artefact is quite prominent in some images, and more subtle in others, and had a tendency to interfere with the training process. Gaussian, mean, and median filters of various kernel sizes were used in an attempt to remove the stitching lines before training, but none of these were effective. The solution which gave the best results was simply to reduce the learning rate, which prevented each model from learning any strong associations with these artefacts.

## 7. Targeted Predictions

---

The focus of this section is to explain how the search and comparison interfaces were developed, which serve to combine the work done in the previous two sections into a straightforward interface. This allows each model's estimations to be easily compared with each other and provides some location context by plotting the SAR image footprint on a world map. The code used in this section can be found in the Search Interface notebook, available on GitHub.

### Pass Direction Error

One of the main objectives for this interface was to offer some context around the SAR acquisition, to make it easier to interpret the concentration estimations. To do this, it was decided that the SAR images should be rotated into the Earth's reference frame (i.e. North pointing up), and then the footprint should be overlaid on a map so that features such as coastlines can be visually located in the SAR images. Figure 4 gives some idea of how the SAR footprints appear rotated in the Earth's reference frame. However, after implementing this rotation and footprint overlay process, some images still appeared to be incorrectly rotated. Extensive troubleshooting revealed that the cause was an inconsistent mislabelling of some SAR products' pass direction attribute, which caused the image to be pre-processed incorrectly by Copernicus. This error is explained in detail in the remainder of this section.

### Copernicus Pre-Processing of SAR Images

Both SENTINEL-1 satellites are in near-polar orbits. The satellite tracker map in Figure 13 shows how the satellite's path sweeps up and down over the Earth's surface, while moving East to West (in the Earth's reference frame). The convention is to label the South to North portion of each orbit as ASCENDING, and the North to South portion as DESCENDING.



Figure 13: Satellite tracker website used to show the path of SENTINEL-1A over the surface of the Earth. Each portion of the orbit is labelled as either ASCENDING or DESCENDING, as indicated. [13]

SAR measurements are obtained when the sensor scans rows of pixels from left to right (in the satellite's frame of reference), and fills rows in order from the bottom up, as the satellite moves over the surface. This is illustrated in Figure 14 as the Acquisition Order and resembles a bottom-left origin structure. When the full image is 'displayed', the rows are reordered using the conventional top-left origin (again, in the satellite frame of reference). This is shown as the Display Order in Figure 14 below. This means that each image will appear upside down in the satellite frame of reference.

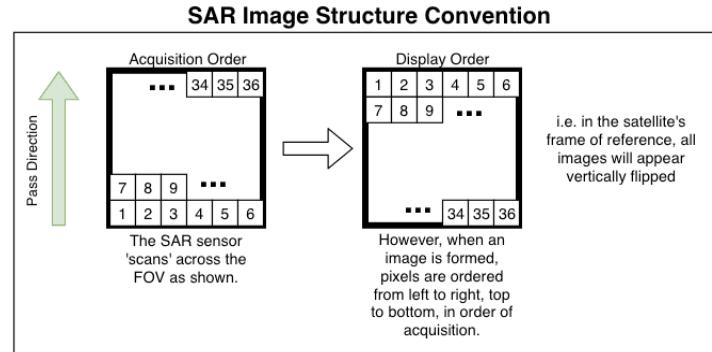


Figure 14: SENTINEL-1 pixel-wise image structure convention, showing how the pixels are ordered in both acquisition and display, thereby explaining the vertical flip which must be accounted for in pre-processing.

Because the satellite platform is North-facing on ASCENDING passes and South-facing on DESCENDING passes, the image orientation needs to be corrected after acquisition. When working with full measurement products, this correction can be applied using the SNAP toolbox provided by Copernicus for use with SAR data. However, Copernicus applies this correction automatically for the Quicklook images so that the Open Access Hub preview appears correctly orientated when viewed in a browser [10]. Figure 15 is a summary of this pre-processing corrective step applied by Copernicus before supplying the Quicklook image.

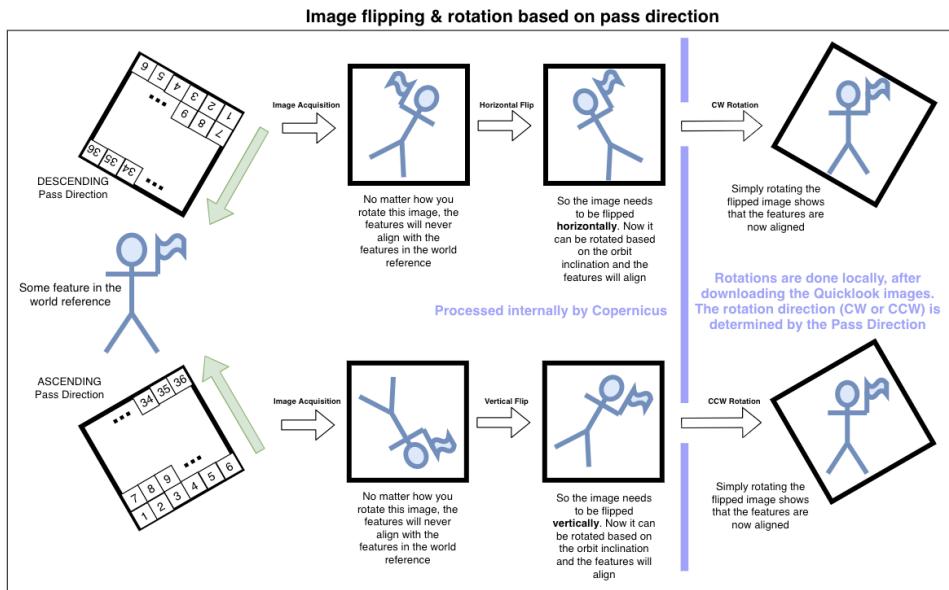


Figure 15: A diagram explaining the pre-processing steps applied to SENTINEL-1 Quicklook images, based on pass direction. Only the final output is accessible for download, once all pre-processing steps are complete. The locally applied rotation is also shown, to illustrate how each pass direction needs to be rotated in a different direction in order for the features in the image to align correctly in the Earth's reference frame.

The final rotation step is specific to this application, and is not important for understanding this issue. The point is that the Quicklook images are processed differently depending on the pass direction attribute, and that only the final result is accessible for download.

## Evidence of Mislabelling

The Open Access Hub [10] was used to troubleshoot this issue, using a few broad searches based on pass direction. Figure 16 shows a search result for products labelled as DESCENDING. Looking at the map (top right) it seemed clear from the layout of all the footprints that these products are in fact DESCENDING. Further validation was found when viewing the selected product's preview, which shows a clear correspondence between the land features included in the product footprint (bottom left) and Quicklook image (bottom right), albeit with a slight rotation error since the Quicklook has not been aligned with the Earth's reference frame.

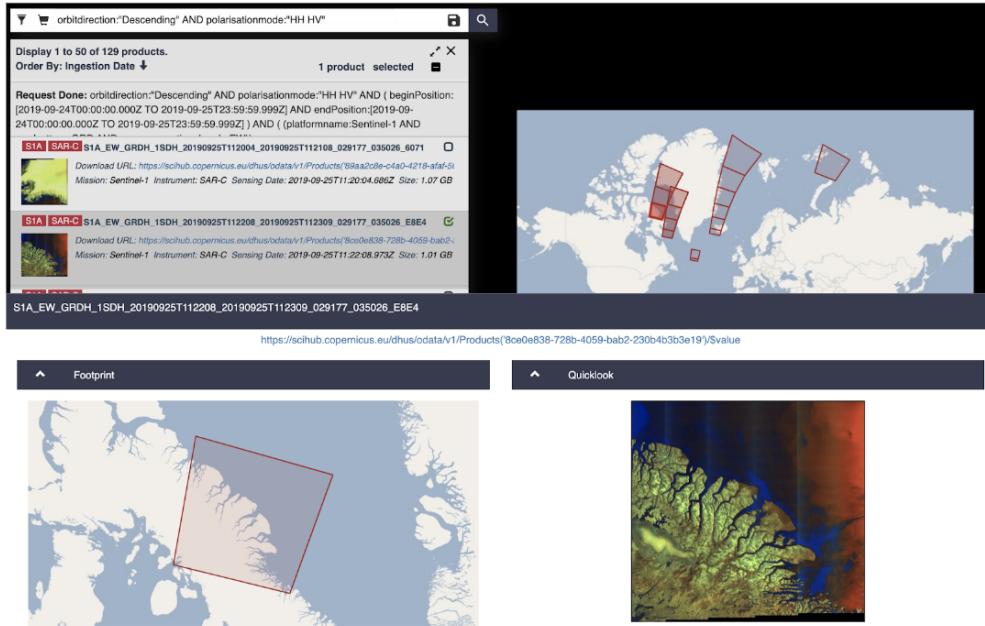


Figure 16: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only DESCENDING products were included. A preview of one product is included (bottom), showing that the footprint and Quicklook are in agreement [10].

In contrast, performing the same search, but filtering for ASCENDING products instead gives the results shown in Figure 17. Judging by the collection of footprints shown on the map (top right) these products were actually obtained on a DESCENDING pass. Viewing a particular product preview revealed that the land features included in the product footprint are no longer in correspondence with the same features in the Quicklook image. It appears as if the Quicklook image had been rotated by 180 degrees.

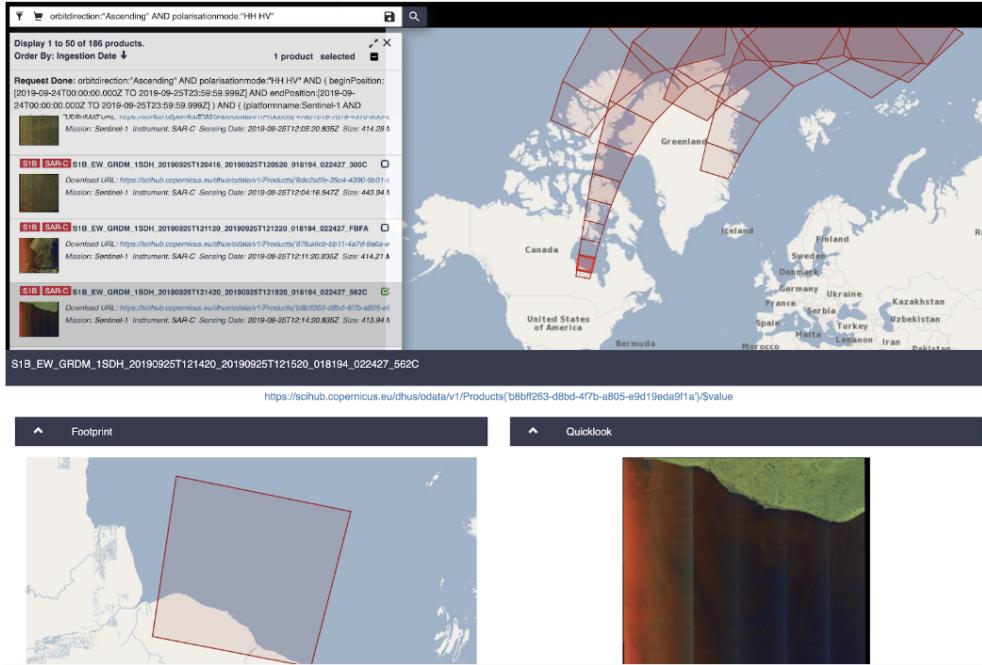


Figure 17: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only ASCENDING products were included. However the trajectories inferred from the collection of footprints suggests that these are actually DESCENDING products. A preview of one product is included (bottom), showing that the footprint and Quicklook are not in agreement, but in fact the Quicklook is rotated by 180 degrees [10].

By then increasing the number of results shown per page, a larger collection of footprints was plotted on the map, shown in Figure 18. These products are all labelled as ASCENDING, but it seems that many of them were actually acquired on DESCENDING passes (judging by the inferred trajectories). In Figure 18, a number of products were selected (shown in brighter red) which appeared to be correctly labelled, and their product previews confirmed that the Quicklook images were correctly orientated. However, all other products shown in the search results exhibit the same 180 degree rotated Quicklook image as discussed above.



Figure 18: Screenshot of an Open Access Hub search showing SENTINEL-1 GRD EW HH+HV products. In this search, only ASCENDING products were included, and the number of results per page increased to show more footprints on the map. It appears that the selected products are correctly labelled as ASCENDING, while the rest are actually DESCENDING products [10].

With the pass direction attribute identified as a potential source of error, the pre-processing steps illustrated in Figure 15 were applied to hypothetical mislabelled images, to see if this error explained the observations. Figure 19 shows the outcome when applying these processes with an incorrect pass direction attribute - the Quicklook image should be flipped horizontally, but it is instead flipped vertically due to the incorrect pass direction label, which has the net effect of rotating the Quicklook image by 180 degrees from its correct orientation.

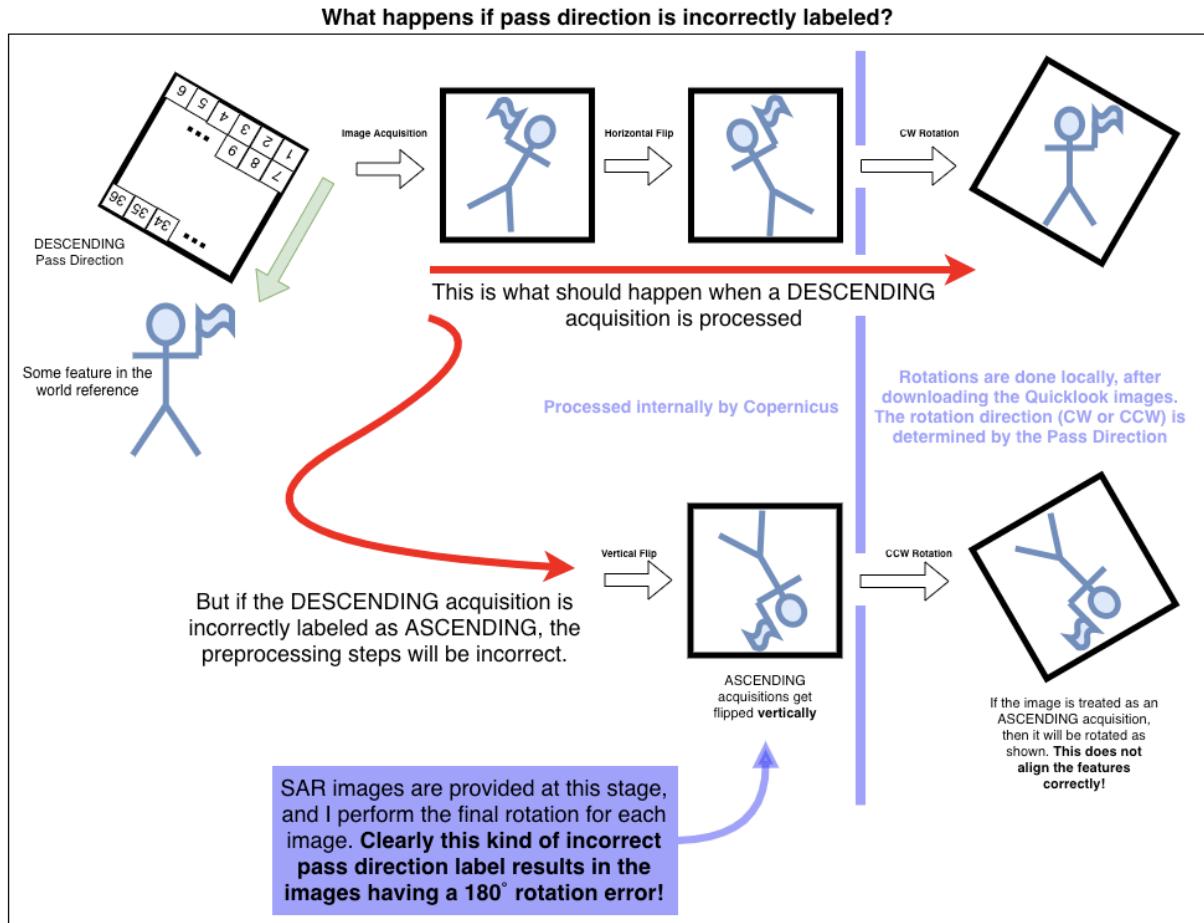


Figure 19: A diagram explaining what happens during pre-processing if a product's pass direction is mislabelled, based on the pre-processing steps discussed above. If this were to happen, it is clear that the Quicklook image would appear rotated by 180 degrees.

This discussion shows that the proposed error in pass direction labelling does in fact cause the Quicklook images to appear rotated by 180 degrees from their correct orientation, just as was observed when viewing products on Open Access Hub [10]. This obviously leads to incorrect results after being further rotated to align with the Earth's reference frame, a process which is also dependant on the pass direction attribute.

It is worth noting that this discrepancy was only observed in the Northern Hemisphere related to SENTINEL-1 GRD EW HH+HV products, where DESCENDING products have been labelled as ASCENDING. Without more insight into the underlying cause for this discrepancy, it is not reasonable to comment on whether this error is likely to occur in other products.

## Workaround Solution

While troubleshooting this issue, it was noticed that the product *OData* ‘*footprint*’ attribute (a set of points defining the footprint polygon) follows a predictable structure: the order in which the points are listed is always top-left, top-right, bottom-right, bottom-left (in the satellite’s frame of reference). This is depicted in Figure 20 below.

## OData 'footprint' Attribute Convention

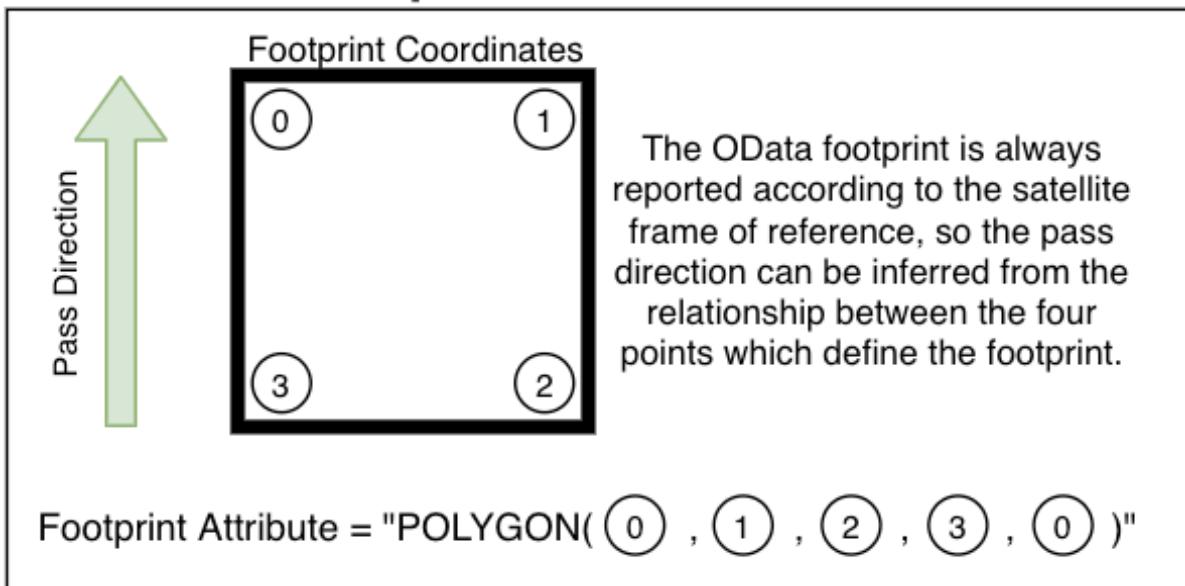


Figure 20: A diagram explaining the convention governing the order in which the footprint points are listed in the *OData* ‘*footprint*’ attribute. This can be leveraged to determine the satellite’s actual orientation at acquisition time, without relying on the ‘pass direction’ attribute.

This insight allows the actual pass direction to be determined without relying on the attribute reported by either *OData* or *OpenSearch*, which are both unreliable. One way to do this is to compare the latitudes of points 0 and 3. If the latitude of point 0 is greater than that of point 3, then the satellite was North-facing at acquisition time, and the product is therefore ASCENDING. If not, then the product is DESCENDING. This derived pass direction was then compared to the reported *OData* attribute. If the two agree, then the pass direction attribute is correct, and the Quicklook image has been correctly pre-processed before being downloaded. If not, then the product was incorrectly labelled, and therefore the Quicklook image was incorrectly pre-processed before being downloaded. This erroneous processing could then be easily corrected by rotating the image by 180 degrees (which corresponds to ‘undoing’ the incorrect flip, and then performing the correct flip). Once corrected, further processing can continue as usual.

The discussion contained in this section was submitted to the Copernicus team for review, and has been acknowledged as a known issue. It has been labelled as a low priority issue since most research activity uses the full SAR measurement, not the Quicklook image, which is not processed in the same way. This enquiry has been noted as the first time this issue has been reported externally, and the Copernicus team has stated that they intend to implement a fix in the next processor revision.

## Navigational Support

While existing sea ice concentration chart products, such as OSI-401-b, tend to focus on achieving global coverage in a single chart, the aim of this investigation was to produce a more focused concentration estimation for specific regions on demand. This kind of product is much more useful when providing close navigational support for shipping and research activities. A boilerplate search interface was developed in Google Colaboratory using *Forms* to facilitate interaction, which allows a user to search for SAR images covering a particular location and date range. A screenshot of the *Form* interface and an arbitrary result set is shown in Figure 21 below.

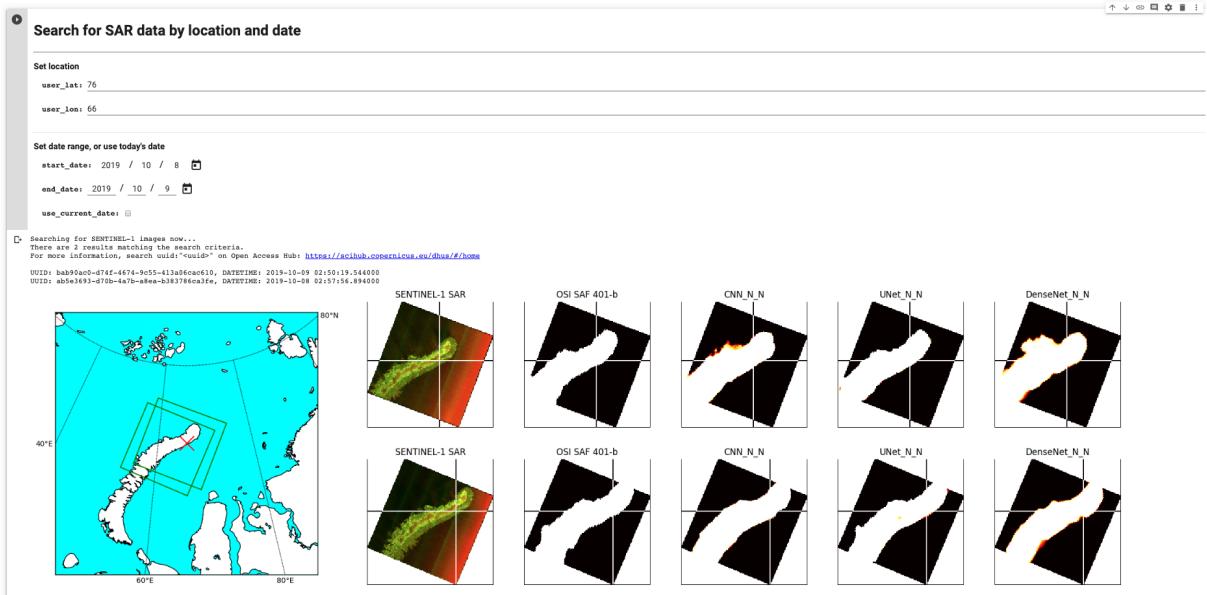


Figure 21: This interface was developed to allow a user to search for the most recent SAR data available for their location, or for historical data if needed, and then to view the estimates produced by a number of models side-by-side. The map on the left offers location context, and includes the footprint of each SAR measurement.

The scripts used to query and download SENTINEL-1 SAR images from the Copernicus database, those used to download the corresponding OSI-401-b concentration charts, and those used to generate patch labels for each SAR image were functionally identical to the processes used in the data pre-processing pipeline. Each was described earlier in detail, and so will not be repeated in this section. Noteworthy additions made in this section are the image footprint display map, seen on the left in Figure 21 above, the fact that all patches have been rotated to align with the Earth’s reference frame, making it easier to correlate features across multiple image patches, and the addition of a location cross hair which shows the location of interest in each patch.

The map plot shown in Figure 21 above was dynamically generated at runtime and is centred on the specified location of interest. The *Basemap* Python package was used to generate the desired map projection, which is designed as a *Matplotlib* toolkit, so it integrates well with existing plotting facilities in Python. Among other things, this package allows the stereographic map projection parameters to be customised, and also provides a convenient way to convert WGS84 coordinates into grid space coordinates on the map projection plot, which allows for overlays to be plotted directly on the same

axes. This is evident in the central red “X” marking the location of interest, and the green footprint polygons illustrating the coverage of each SAR image.

The process of rotating the patches to align with the Earth’s reference frame is actually quite simple, although the pass direction error discussed in the previous section complicated matters significantly. However, once a suitable workaround was implemented, the rotation process could be reliably completed. First the *Basemap* package was used to convert the four corners of each SAR image’s footprint from WGS84 latitude and longitude into grid space coordinates within the map projection plot. Then using trigonometry and the relative positions of each point, the angles  $\theta_1$  and  $\theta_2$  were calculated, as defined in Figure 22 below. The average of these two angles was then taken to be the degree of rotation required for a particular SAR image and the associated patches. The rotation was performed using *OpenCV*, specifically the *getRotationMatrix* and *warpAffine* functions. However, before applying the rotation, the rotation matrix was modified to avoid the corners of the image being truncated in the rotation process [14].

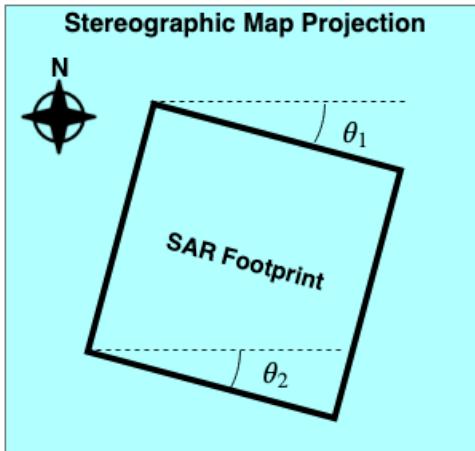


Figure 22: Illustration of an arbitrary SAR image footprint projected onto a stereographic map. The angles which the top and bottom edges make with the horizontal are defined as  $\theta_1$  and  $\theta_2$  respectively, as shown in the diagram. These angles were calculated using trigonometry and the known coordinates of each corner of the footprint polygon. The two angles were averaged to determine the required degree of rotation.

In addition to the location of interest being shown in the map projection to the left-hand side of the search interface results, the location was also plotted on each patch using cross hairs to avoid cluttering the region of interest with a large marker of some sort. To do this, the pixel coordinate interpolation scheme was used to generate a distance mask prior to rotating the patches, where the value of each pixel represented the distance of that pixel from the specified location. This distance mask was then rotated along with the image patches, and then the minimum value in the rotated distance mask was used to define the crosshair location in the rotated patches. This way the cross hairs themselves are not rotated and appear aligned in the Earth’s reference frame.

This interface served a number of purposes. Firstly, it was a very useful tool when developing the data pre-processing pipeline, as it allowed for specific edge cases to be visually inspected and validated. As an example of this, consider a SAR image whose

footprint straddles the International Date Line. If the footprint coordinates are naïvely interpreted, as they most often could be anywhere else on the Earth, both the pixel coordinate interpolation scheme, which is critical for patch label generation, and the process used to plot the image footprint on the map shown above cease to function as expected, due to the longitude discontinuity found at the International Date Line. Secondly, it provides a very direct way to visually compare the concentration estimations generated by each model. This is particularly useful when an interesting image, or one with some challenging anomalies, is found. Comparing the model estimations in this way gives some insight into how each model performs on specific features. And finally, this interface offers a proof of concept for the intended workflow in the context of serving targeted predictions for close navigational support.

## Evaluation of Existing Models

A secondary objective for this investigation was to use much of the infrastructure developed in terms of acquiring and processing SAR images, filtering data by location and date, and extracting the relevant subset of OSI-401-b concentration data for each SAR image in order to compare estimates made by the existing numerical sea ice concentration models with a set of recent in situ observations. A 2019 research expedition to the MIZ in the Southern Ocean collected a large set of images of the sea ice conditions, each tagged with its location and date [9]. This dataset is ideal for understanding the validity of the existing models in the Southern Ocean where extensive in situ observation data is lacking. Figure 23 shows a screenshot of the interface used to compare OSI-401-b predictions with in situ observations of sea ice conditions.

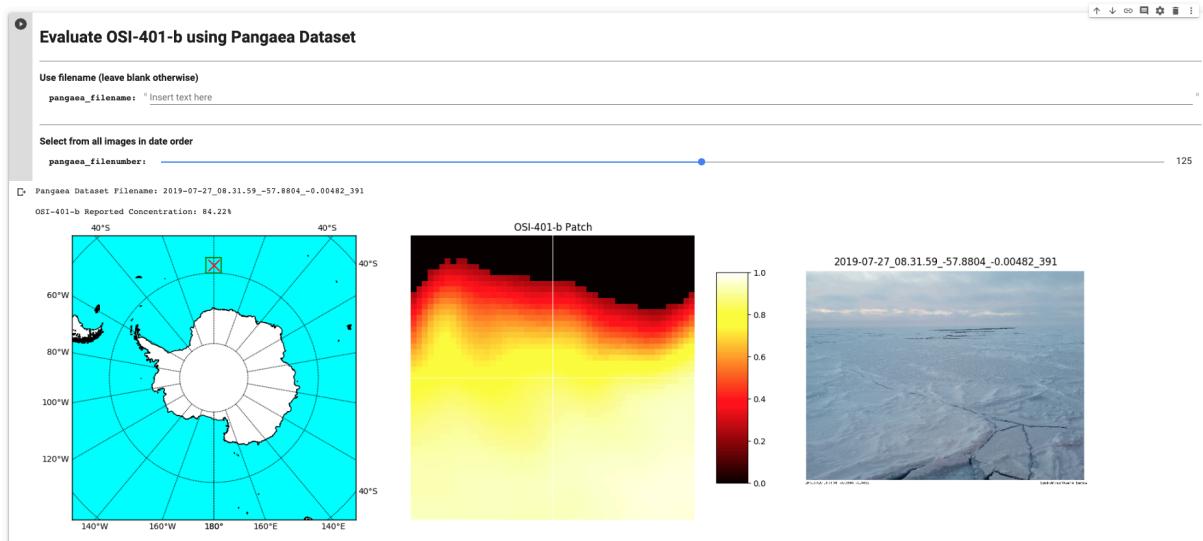


Figure 23: This comparison interface shows images taken in the MIZ and compares them with the OSI-401-b estimates for the same day and location, with the exact location indicated with crosshairs.

The interface shown in Figure 23 was derived from the search interface used in the previous section but was modified to only show results based on the contents of the observation dataset. This way each observation datapoint could be compared, without having to manually enter the date and location of each image in the dataset.

## 8. Results

---

This section will present the consolidated results from each of the areas of focus discussed over the previous chapters. First the model training and validation outcomes are presented, followed by a set of concentration estimates which can be visually inspected. Some interesting in situ observation comparisons are then shown, and lastly the outcomes of each acceptance test is justified.

### Training & Evaluation

Model performance during and after the training procedure is presented in two ways in this section. Models are compared primarily based on their final performance metrics, but the loss trajectories throughout the training process are also used to gain some insight into each model's behaviour. Both the weighted MAE objective function and a pure MAE metric were tracked during all training procedures.

### Performance Metrics

Table 7 shows all combinations of model architecture and training strategy used in this investigation, explains the choice of training data for each strategy, and presents the final performance metrics achieved by each model at the end of 50 epochs.

Model Name	Train Data	Test Data	Weighted MAE (Train, Test)	MAE (Train, Test)
CNN_N_N	North	North	(0.0640, 0.0697)	(0.0727, 0.0793)
CNN_N_N_A	North_A	North	(0.0674, 0.0792)	(0.0763, 0.0897)
CNN_S_S	South	South	(0.0579, 0.0660)	(0.0652, 0.0748)
CNN_S_S_A	South_A	South	(0.0624, 0.0755)	(0.0703, 0.0855)
CNN_NS_S	North + South	South	(0.0561, 0.0631)	(0.0631, 0.0715)
CNN_NS_S_A	North_A + South_A	South	(0.0597, 0.0742)	(0.0675, 0.0842)
UNet_N_N	North	North	(0.0145, 0.0386)	(0.0166, 0.0437)
UNet_N_N_A	North_A	North	(0.0174, 0.0468)	(0.0200, 0.0530)
UNet_S_S	South	South	(0.0150, 0.0324)	(0.0170, 0.0369)
UNet_S_S_A	South_A	South	(0.0218, 0.0693)	(0.0248, 0.0786)
UNet_NS_S	North + South	South	(0.0165, 0.0325)	(0.0187, 0.0370)
UNet_NS_S_A	North_A + South_A	South	(0.0198, 0.0477)	(0.0225, 0.0541)
DenseNet_N_N	North	North	(0.0603, 0.0797)	(0.0683, 0.0904)
DenseNet_N_N_A	North_A	North	(0.0645, 0.124)	(0.0730, 0.138)
DenseNet_S_S	South	South	(0.0585, 0.120)	(0.0658, 0.130)
DenseNet_S_S_A	South_A	South	(0.0632, 0.130)	(0.0709, 0.141)
DenseNet_NS_S	North + South	South	(0.0581, 0.0813)	(0.0654, 0.0897)
DenseNet_NS_S_A	North_A + South_A	South	(0.0633, 0.0830)	(0.0711, 0.0919)

Table 7: Summary of results obtained from each training strategy for each model type under investigation.

Multiple sets of training data indicate that the model was partially trained on the first set, and then trained to completion using the second set.

Two things worth noting are: in all cases, the weighted MAE scores are less than the pure MAE for each strategy, which confirms that the objective function is indeed penalising the estimation error less in the presence of uncertainty; and with the exception of three DenseNet versions, all models were able to achieve a weighted MAE on unseen test data less than 10% as required for acceptance test T4.

## Loss Trajectories

Observing the progression of loss metrics throughout the training process can offer some insight into model behaviour and may inform decisions when tuning hyperparameters. The full sets of loss trajectories are included in Appendix A, but Figure 24 below shows a particularly interesting case.

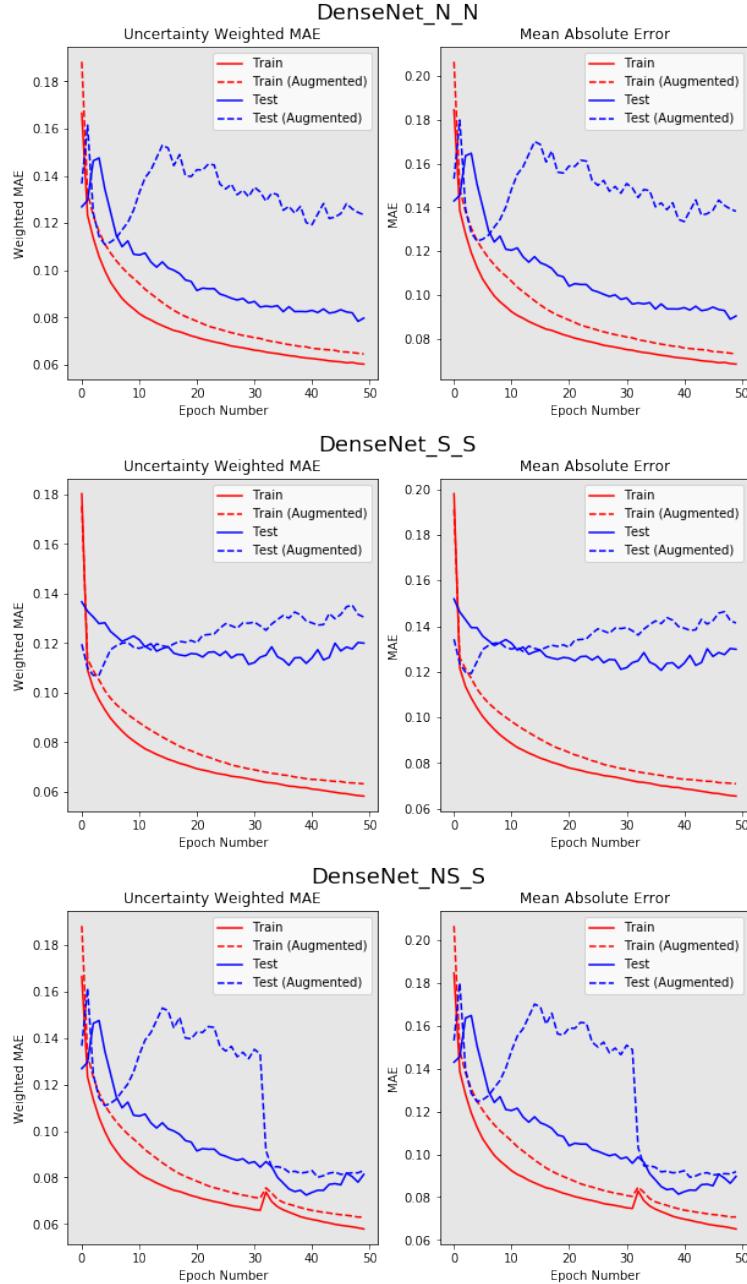


Figure 24: Loss trajectories for DenseNet, covering all training strategies. The divergence of test results from the training curves is noteworthy, and warrants further discussion.

There are a number of interesting features in the DenseNet loss trajectories, all regarding the divergence of test results from the decreasing training error curve, which was far more pronounced in the DenseNet training examples than with any other model architecture or training scheme. These features will be analysed in the discussion following this section.

## Visual Inspection

Another important way to evaluate each model's performance is by visually inspecting the results. While performance metrics give a quantitative understanding of the model's performance across an entire dataset, visual inspection of just a few samples can give a more intuitive understanding of where each model performs well, and where it struggles. For the purpose of reporting on each model, a single "interesting" sample was chosen from each of the datasets used so that any observed differences could be solely attributed to architecture and training strategy. The following samples were used:

*Northern Hemisphere*  
2019\_3\_28\_1211d9d2-39f4-4571-b38f-916d4a8a7b3e

*Southern Hemisphere*  
2019\_1\_8\_37fd589e-ebc9-45aa-b8b4-1daca4741f8d

Following the convention used in Table 7 for training strategies, all models intended for testing on Northern hemisphere were evaluated using the first sample, and those intended for testing on Southern hemisphere data were evaluated using the second sample. The full set of evaluation results is shown in Appendix B, and two examples are shown in Figure 25 below. Their significance will be discussed in the following section.

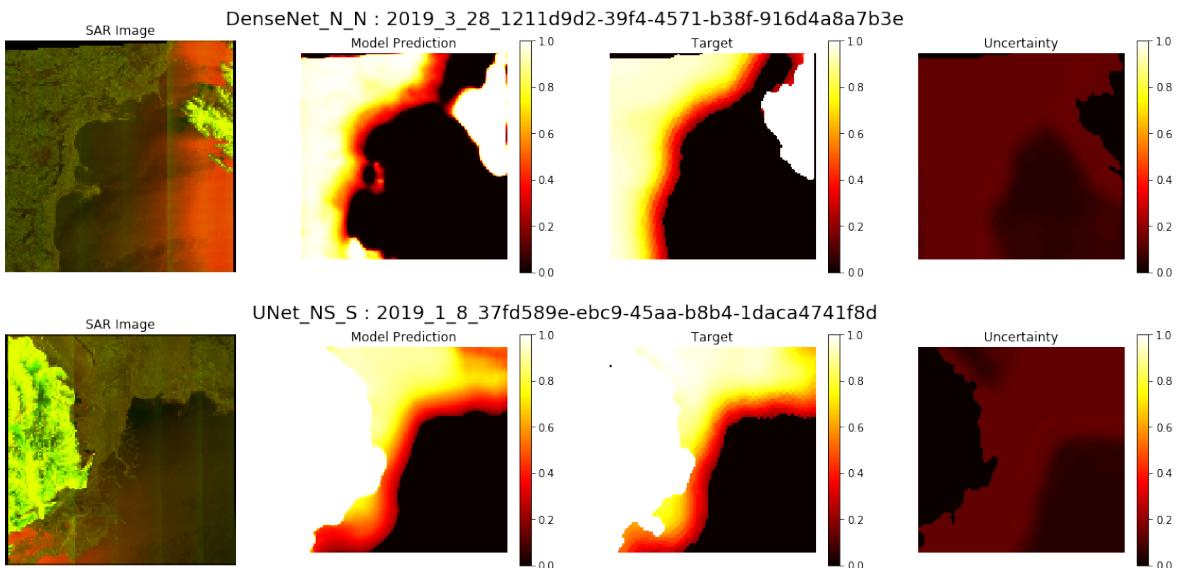
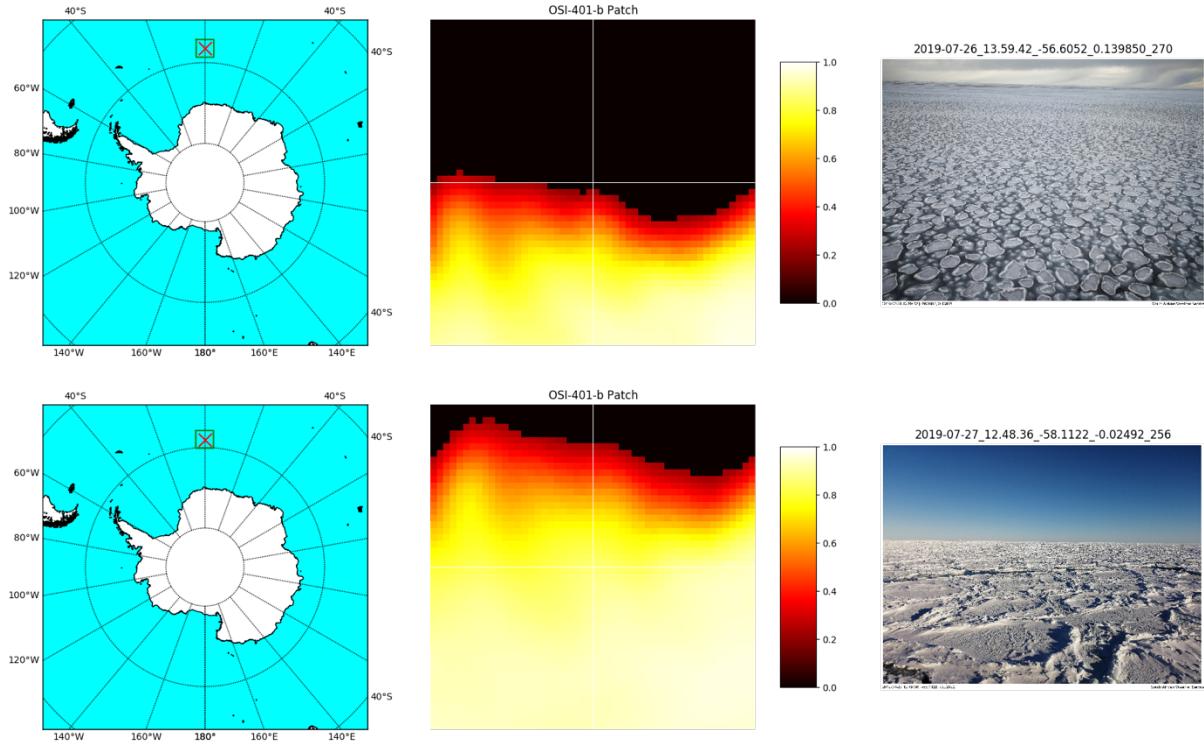


Figure 25: Two examples of how each trained model was visually evaluated using a pre-selected sample. The top row shows a DenseNet\_N\_N estimate, and the bottom row shows a UNet\_NS\_S estimate. The significance of these particular concentration estimates will be discussed further below.

## Validity of Existing Models

The previous section introduced a comparison tool developed to evaluate the reliability of the PMW-derived model used in this investigation. Figure 26 below shows two more items selected from the set of in situ observations which illustrate the unreliability of these models in the Antarctic. An additional set of observation comparisons is included in Appendix C, but only the two shown here will be included in the discussion which follows.



*Figure 26: In situ observations from the MIZ compared with PMW-derived concentration estimates for the same day and location. Each chart was generated using the evaluation tool developed for this investigation, which was presented in the previous section.*

## Acceptance Tests

Table 8 below lists the acceptance test procedure IDs defined in the Requirements Analysis, and provides evidence justifying the outcome of each test. Refer to Table 4 for the definitions of each acceptance test procedure, and details regarding the specifications each test was meant to assess.

ID	Outcome
T1	<b>Pass.</b> Pixel spacing was shown to be in the order of 3km after decimation (see Figure 5)
T2	<b>Pass.</b> Using the Search Interface often delivers two SAR results acquired within the last 24 hours, while the most recent OSI-401-b product may be older than 24 hours.
T3	<b>Pass.</b> By design, each architecture produced an output of the same dimension as the input.
T4	<b>Fail.</b> Three DenseNet versions exceeded the 10% weighted mean absolute error threshold.
T5	<b>Pass.</b> Demonstrated in the Navigational Support section when searching for SAR data based on date and location.
T6	<b>Pass.</b> Demonstrated in the Navigational Support section when displaying concentration estimates for previously unseen SAR data.
T7	<b>Pass.</b> Demonstrated in Figure 5 (patch label validation).
T8	<b>Pass.</b> Pixel coordinate interpolation validation test (see Figure 3).
T9	<b>Pass.</b> The Search Interface demonstrates that the correct hemisphere's concentration products can be downloaded on demand.
T10	<b>Pass.</b> Demonstrated in Figure 5 (patch label validation).
T11	<b>Pass.</b> Achieved by design (see Figure 12) and validated in performance metrics (see Table 7).
T12	<b>Pass.</b> Demonstrated in comparison tool (see Figure 23) and in Appendix C.

*Table 8: Acceptance test procedure outcome table, showing the outcome of each test, and justifying each result in reference to work done or results shown in the course of this investigation.*

## 9. Discussion

---

This section presents some discussion around the important results shown in the previous section. The intention is primarily to understand the results in the context of the state of affairs described in the literature review, and in terms of the client's expectations defined in the user requirements section.

### Spatio-Temporal Resolution

The need for improved spatio-temporal resolution in sea ice models was discussed in the literature review at the beginning of this document, and one of the primary motivators for choosing to focus on SAR images was the far superior spatial resolution offered by these platforms compared to PMW instruments. This investigation showed that even after the original SAR measurement has been decimated to a much smaller image size, the pixel spacing is still an improvement over PMW data. Since the neural network architectures were designed to replicate the input dimension in their estimates, all of the concentration estimates presented here offer a resolution better than the original data with which they were trained. As evidence of this, consider the first estimate shown in Figure 25 generated using the DenseNet architecture. The model prediction tile captures the small feature near the centre of the SAR image fairly well, which is not reflected at all in the OSI-401-b data.

Improvements in temporal resolution are more difficult to demonstrate in this context, but perhaps the clearest example can be found in the Search Interface introduced in the Navigational Support section. Quite often, when searching for SAR data over the last day (i.e. *use\_current\_date checked*) in high latitudes, two results were returned. However, the most recent OSI-401-b product available belonged to the previous day, since each daily average is only published the following morning. This means that the most recent data available could be over 24 hours old, and that the estimates generated from much more recent SAR data present a clear advantage, particularly in the more dynamic Southern Ocean MIZ where sea ice conditions tend to evolve more quickly.

### Learning vs Numerical Models

The reasons for pursuing a learning approach to generating sea ice concentration estimates were based on the hypothesis that a sufficiently well-designed learning structure would be able to overcome the inaccuracies inherent in the PMW-derived models used for training, even if these inaccuracies are not fully-understood. This is similar to how the learned models were able to learn higher-resolution features than were present in any one sample of the training data. This is a non-trivial claim, but the results have shown positive signs indicating that the underlying theory has promise. As an example, consider the second estimate shown in Figure 25, and particularly the lower left hand corner of each tile. The 'target', which is a patch label from OSI-401-b, clearly shows a small region of high concentration, which does not seem to correlate to any feature in the SAR image to the left. Incidentally, the U-Net generated estimate does not include this feature, and so it would seem that the learned model has been

able to differentiate in some way between features which are accurately represented in the training labels, and others which may be misleading.

### Arbitrary Loss Definition

Perhaps the most useful result of this investigation in the context of trying to create reliable new models using less reliable existing models is the implementation of a certainty-weighted objective function. In the limited scope of this investigation, the weighting relied solely on the uncertainty attribute published with the OSI-401-b concentration product. But this theory can be applied to any arbitrary definition of ‘certainty’ in future research. For example, if one were to generate a ‘bias map’ using a series of in situ observations for a given region which quantifies any discrepancies between estimates and reality, this bias map could be incorporated into the objective function in a similar way, and may allow the learning process to correct for errors in the training data.

### Data Augmentation

Data augmentation is a commonly used tool in machine learning. It is particularly useful when dataset size is limited, or to avoid overfitting and in turn help the model generalise better to unseen test data. However, the loss trajectory results presented in the previous section and in Appendix A suggest the opposite. In all cases where the use of augmented data was compared with an otherwise identical neural network and training strategy, the test results were worse when the model was trained with augmented data. i.e. the use of augmented data decreased model generalisation ability. One possible explanation for this effect is that there may be so little difference between training data and test data that ‘overfitting’ in training actually yields better test results. However, it should be noted that this is merely conjecture, and any further investigation into this result is out of scope.

### Multi-Stage Training

It is not uncommon to schedule or dynamically adjust hyperparameters at key points during a model training process, and this is often referred to as multi-stage training. However, in this investigation this staging approach was used to change the entire dataset. This has a similar effect to transfer learning, although transfer learning tends to be applied in a cross-domain fashion, whereas in this case both datasets were of sea ice concentration. The rationale for this hybrid training strategy was to allow the model to learn most of its feature associations from Northern hemisphere data, which is known to be more accurate and reliable, before completing the training process with Southern hemisphere data in order to help the model generalise to the new environment. The most interesting observation as a result of this is shown in Figure 24, specifically the DenseNet\_NS\_S loss trajectory. While training on Northern hemisphere data for the first 32 epochs the train and test scores diverge, which is likely due to overfitting and the fact that the DenseNet model used had far fewer parameters compared to the other two model types (see Table 6). However, almost as soon as the source of training data was switched to the Southern hemisphere after 32 epochs, the test scores rapidly converged with the training scores. As can be seen in Figure 24, this allowed the

DenseNet\_NS\_S strategy to achieve a 4% reduction in weighted MAE on test data compared to the DenseNet\_S\_S, which both used the same Southern hemisphere test data (see Table 7). This presents another interesting result demonstrating the effects of mixed-data training strategies like this, which could certainly be investigated further in future research.

## Reliability of Existing Models

As a secondary objective, this investigation aimed to offer some insight into the reliability of the PMW-based OSI-401-b sea ice concentration product used for generating all label information. The comparison tool introduced in Figure 23 allowed for a number of recent in situ observations to be retrospectively compared to the concentration estimates for the same day and location. The results shown in Figure 26 and in Appendix C show that there are significant errors in the concentration estimates within the Southern Ocean MIZ. Both observations shown in Figure 26 have been judged by experts to represent 100% ice concentration by surface area, and yet OSI-401-b estimates show 0% and approximately 80% respectively. While these two samples are among the more drastic examples of this error, they illustrate beyond doubt that the naïve application in the Southern hemisphere of estimation models designed and calibrated for the Northern hemisphere will not yield reliable results.

## User Requirements

The final discussion point is on the relevance of these results to the user requirements defined in Table 1. The Search Interface allows delivery of concentration estimates for any arbitrary location and does so with the improved spatio-temporal resolution made possible through the use of SAR images. These estimation models have been thoroughly compared through a wide range of test configurations and have always been shown alongside the PMW-based estimate for comparison. The OSI-401-b uncertainty attribute was used in the objective function to account for known inaccuracies in the existing model. A comparison tool was developed to allow in situ observations to be easily compared to estimates from existing models in order to comment on the reliability and accuracy of these models. And lastly, in terms of the acceptance test procedures defined in Table 4, all but one test has passed and justification is shown in Table 8. Failure to pass this single test was insignificant in the scope of the full investigation and has not impacted on the extent to which the user's expectations have been satisfied.

## 10. Conclusions

---

The problem of sea ice concentration estimation is clearly far from solved, although a number of interesting results have been presented over the course of this investigation. Satellite instrument data sources have been under-utilised in the past, but developments in deep learning, particularly in the field of image processing, have led to promising results in this context. The outcome of this investigation is a detailed implementation of an end-to-end workflow for estimating concentration maps from SAR images, based on three neural network models trained on data from existing passive microwave concentration estimation models. The neural network development process was a primary focus of this investigation, and many of the results led to insights on the effects of various data processing techniques used in the course of development. Care was taken throughout the development process to validate results and procedures, thus ensuring that the proposed workflow is scalable and robust. The estimation models developed can be improved in many ways but have already shown signs of being able to learn higher resolution features than are contained in any single training sample, and the use of a novel weighted objective function allows the training process to account for known uncertainty within the training data. A comparison tool was presented which allows existing concentration estimates to be compared with in situ observations and has offered more evidence asserting that the use of Arctic models in the Southern Ocean marginal ice zone will not yield reliable results.

For any future research conducted in this direction, a number of recommendations have already been made, such as defining a more effective ‘certainty’ metric to be used in the objective function, investigating the effects of data augmentation in this context which have appeared to behave counterintuitively in these results, or proposing more effective ways to utilise the benefits of multi-stage training shown here. Additionally, one should consider alternative satellite data sources which may have been overlooked, and also consider scaling up the neural network input dimension to make better use of the high resolution SAR data which is so freely available.

## References

---

- [1] A. Alberello, L. Bennetts, P. Heil, C. Eayrs, M. Vichi, K. MacHutchon, M. Onorato and A. Toffoli, “Drift Of Pancake Ice Floes In The Winter Antarctic Marginal Ice Zone During Polar Cyclones,” 2019.
- [2] R. Kwok, S. S. Pang and S. Kacimi, “Sea ice drift in the Southern Ocean: Regional patterns, variability, and trends,” *Elem Sci Anth*, vol. 5, no. 32, 2017.
- [3] C. L. V. Cooke and K. A. Scott, “Estimating Sea Ice Concentration From SAR: Training Convolutional Neural Networks With Passive Microwave Data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4735-4747, 2019.
- [4] S. Häkkinen and D. J. Cavalieri, “Sea ice drift and its relationship to altimetry-derived ocean currents in the Labrador Sea,” *Geophysical Research Letters*, vol. 32, no. 11, 2005.
- [5] M. Vichi, C. Eayrs, A. Alberello, A. Bekker, L. Bennetts, D. Holland, E. Jong, W. Joubert, K. MacHutchon, G. Messori, J. F. Mojica, M. Onorato, C. Saunders, S. Skatulla and A. Toffoli, “Effects of an Explosive Polar Cyclone Crossing the Antarctic Marginal Ice Zone,” *Geophysical Research Letters*, 2019.
- [6] D. Notz, “Challenges in simulating sea ice in Earth System Models,” *Wiley Interdisciplinary Reviews: Climate Change*, vol. 3, no. 6, pp. 509-526, 2012.
- [7] G. Huang, Z. Liu and L. van der Maaten, “Densely Connected Convolutional Networks,” 2016.
- [8] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” 2015.
- [9] M. de Vos, C.-L. Ramjukadh, C. G. Liesker, M. de Villiers and C. T. Lyttle, “Southern Ocean Marginal Ice Zone Photographs from a 2019 Winter Research Cruise (SA Agulhas II),” 2019. [Online]. Available: <https://doi.pangaea.de/10.1594/PANGAEA.905497>. [Accessed 5 10 2019].
- [10] E. Copernicus, “Copernicus Open Access Hub,” 2019. [Online]. Available: <https://scihub.copernicus.eu/dhus/#/home>.
- [11] K. Żak, “keras-unet,” 2019. [Online]. Available: <https://github.com/karolzak/keras-unet>.
- [12] seasonyc, “densenet,” 2018. [Online]. Available: <https://github.com/seasonyc/densenet>.

- [13] N2YO.com, “N2YO.com,” 2019. [Online]. Available:  
<https://www.n2yo.com/?s=39634>.
- [14] A. Rosebrock, “PyImageSearch,” 2017. [Online]. Available:  
<https://www.pyimagesearch.com/2017/01/02/rotate-images-correctly-with-opencv-and-python/>. [Accessed 9 2019].

## Appendix A – Neural Network Loss Trajectories

---

### CNN Training Progression

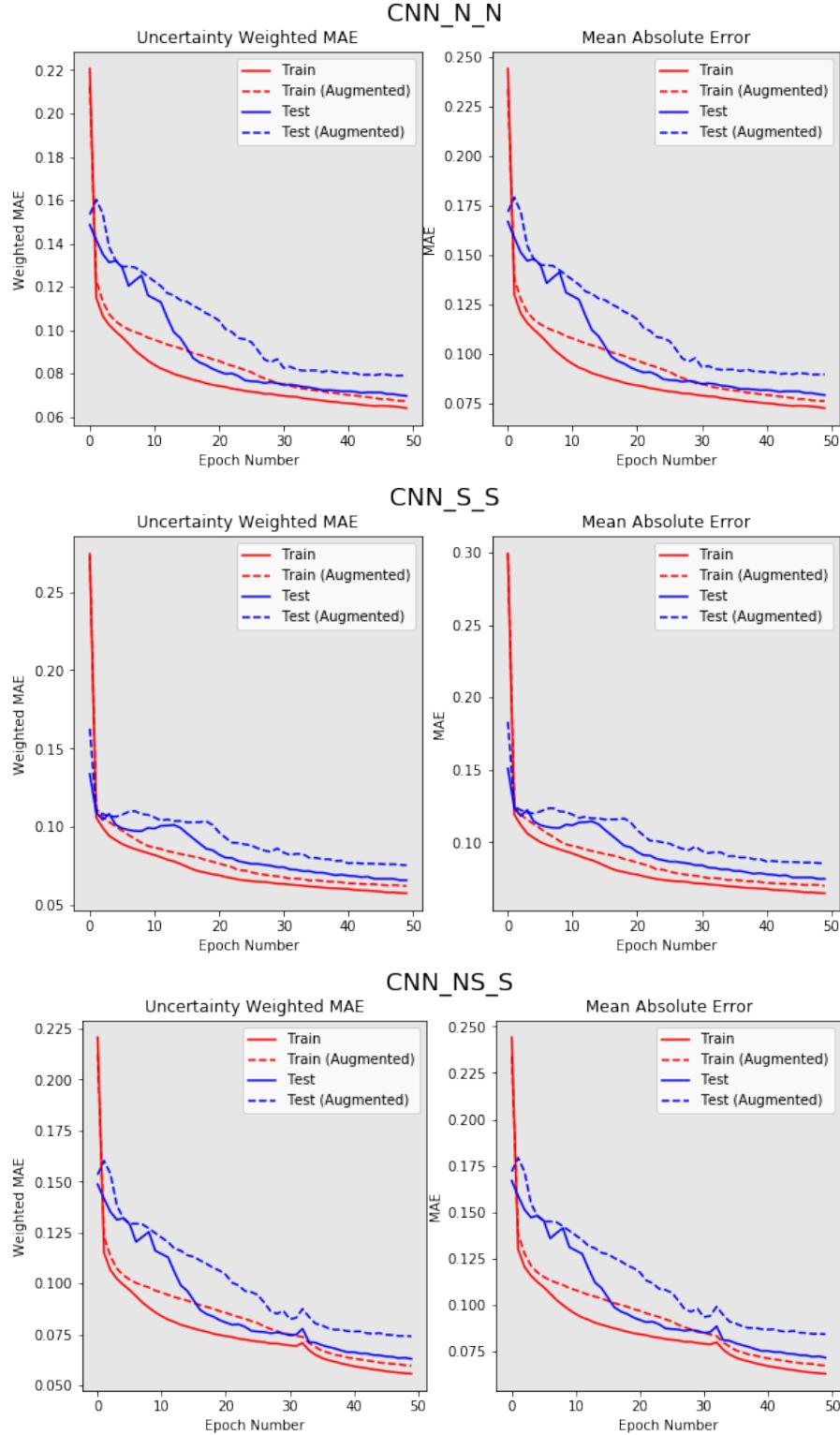


Figure 27: CNN loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets.

## U-Net Training Progression

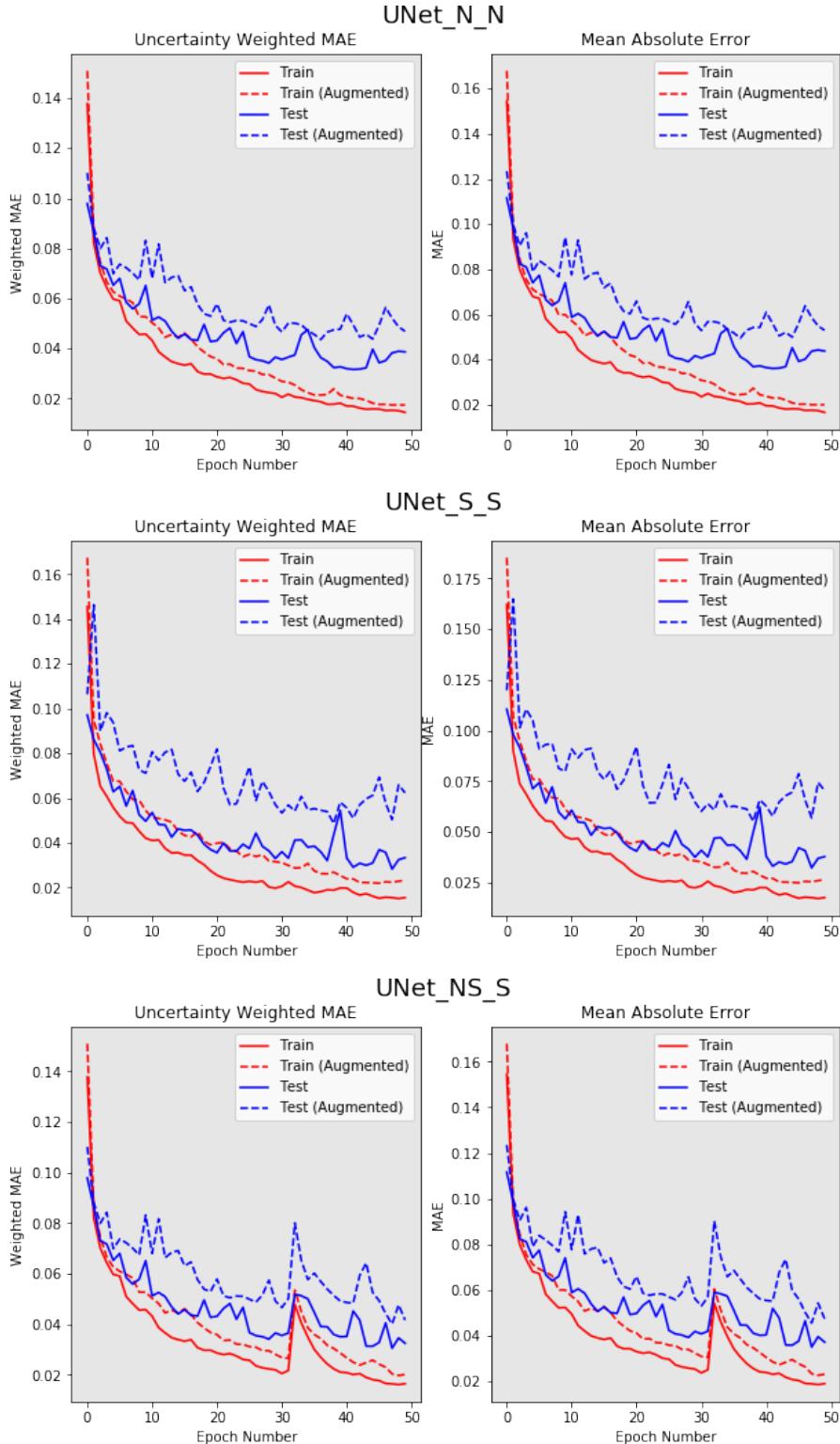


Figure 28: U-Net loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets.

## DenseNet Training Progression

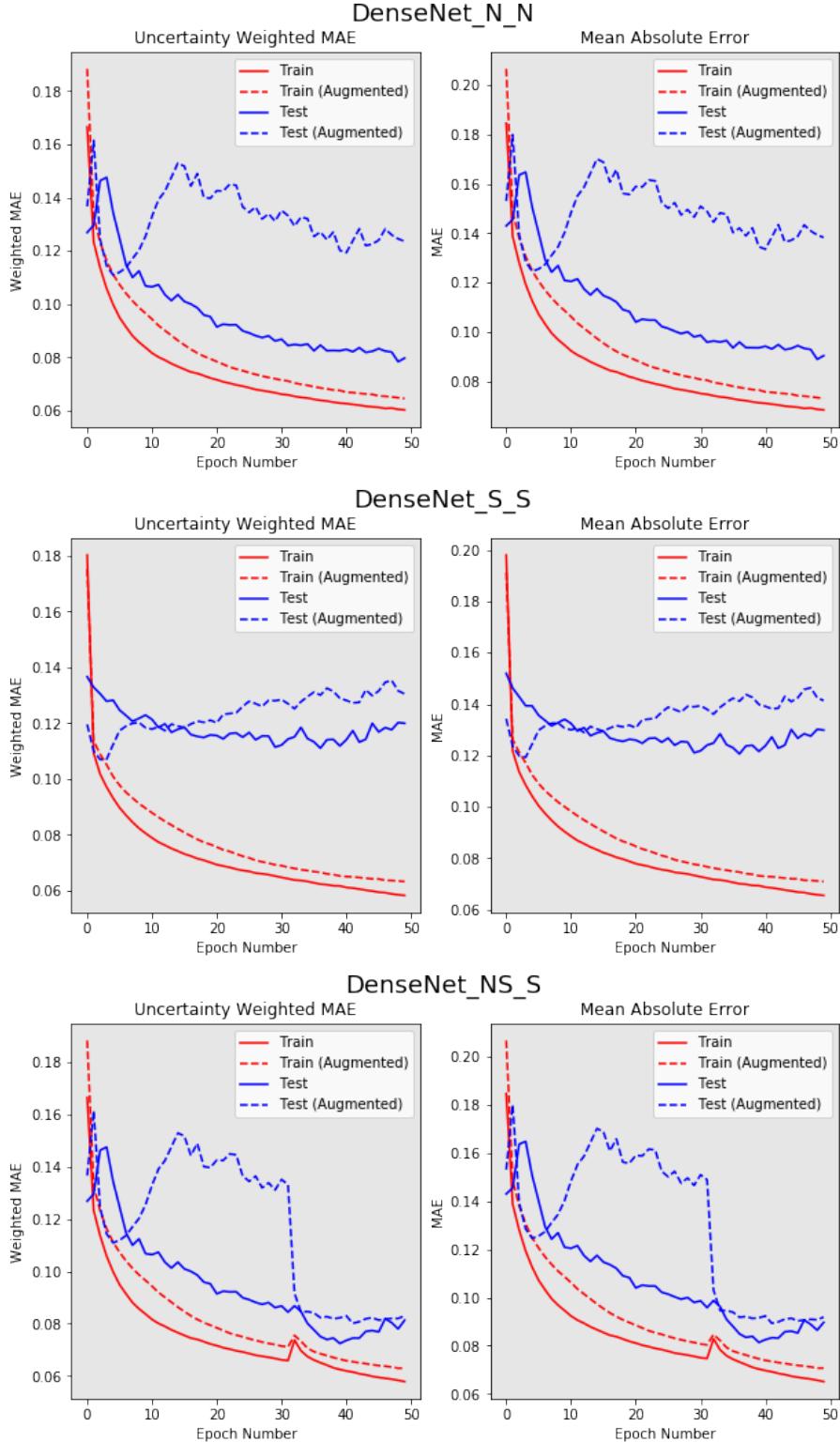


Figure 29: DenseNet loss trajectories plotted throughout the course of each training strategy. Both weighted MAE and pure MAE are tracked. The plot for each strategy includes the results from training on both the augmented and non-augmented datasets.

## Appendix B – Neural Network Concentration Estimates

### Using Arctic Training Data

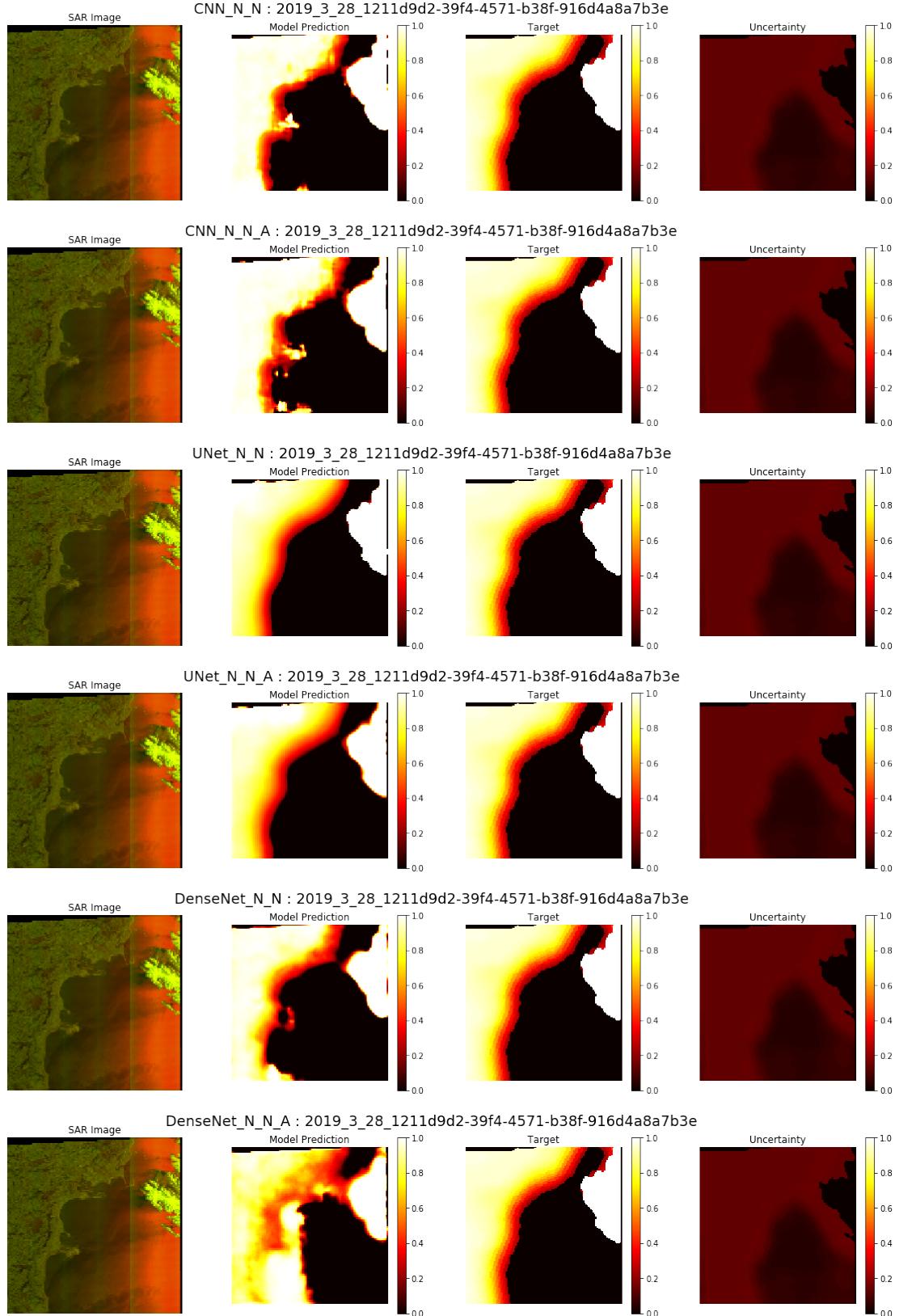


Figure 30: A set of concentration estimates generated using Northern hemisphere trained models.

## Using Antarctic Training Data

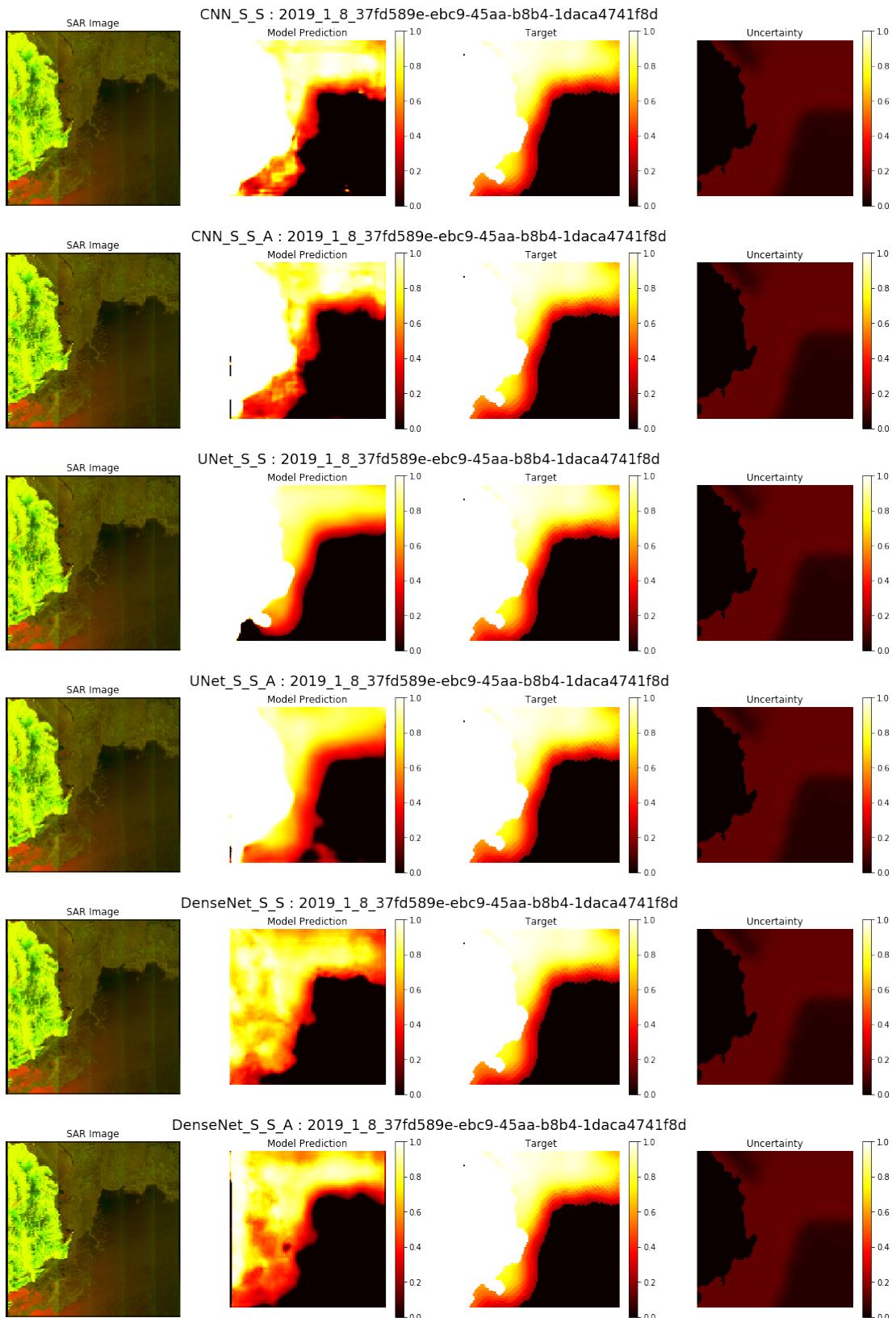


Figure 31: A set of concentration estimates generated using Southern hemisphere trained models.

## Using Hybrid Training Scheme

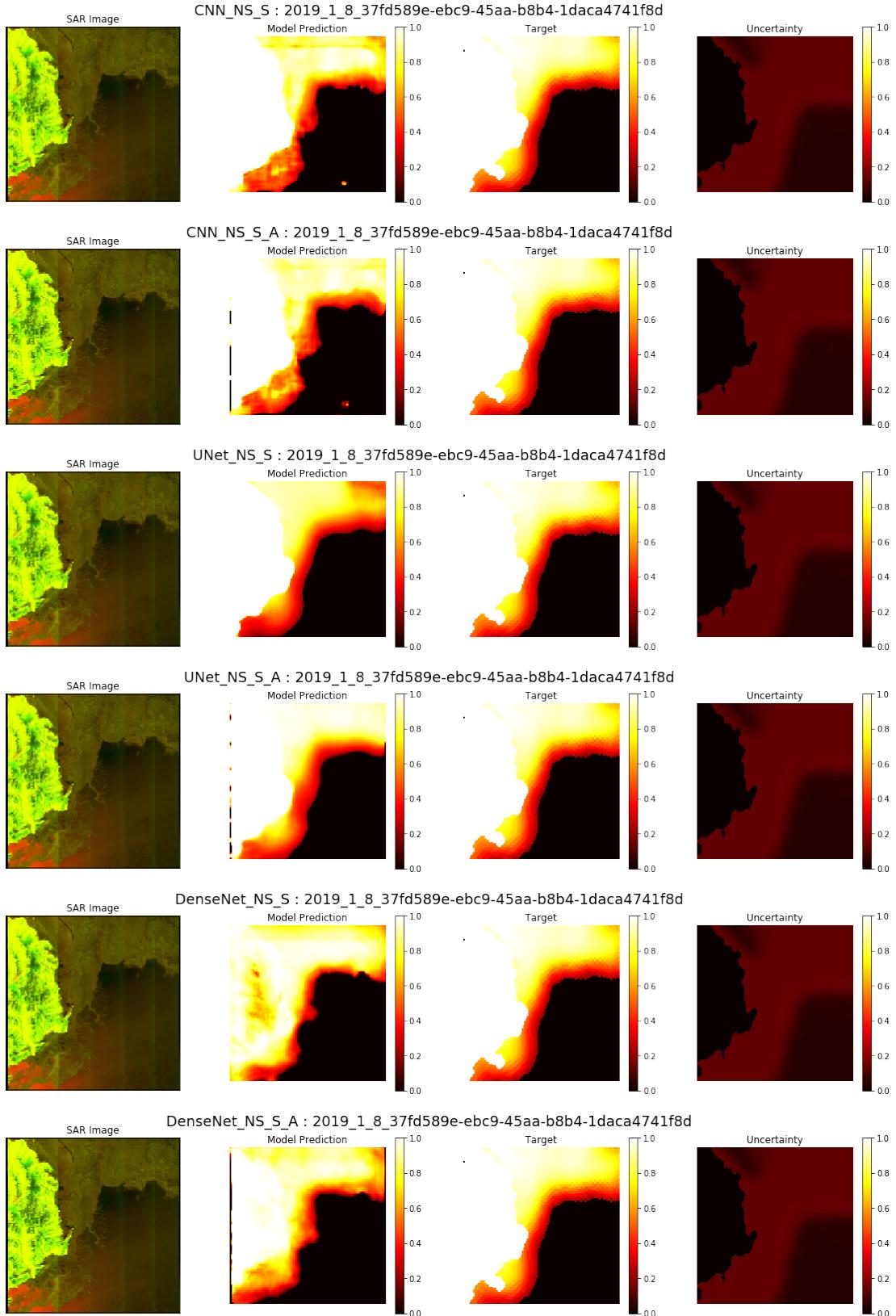
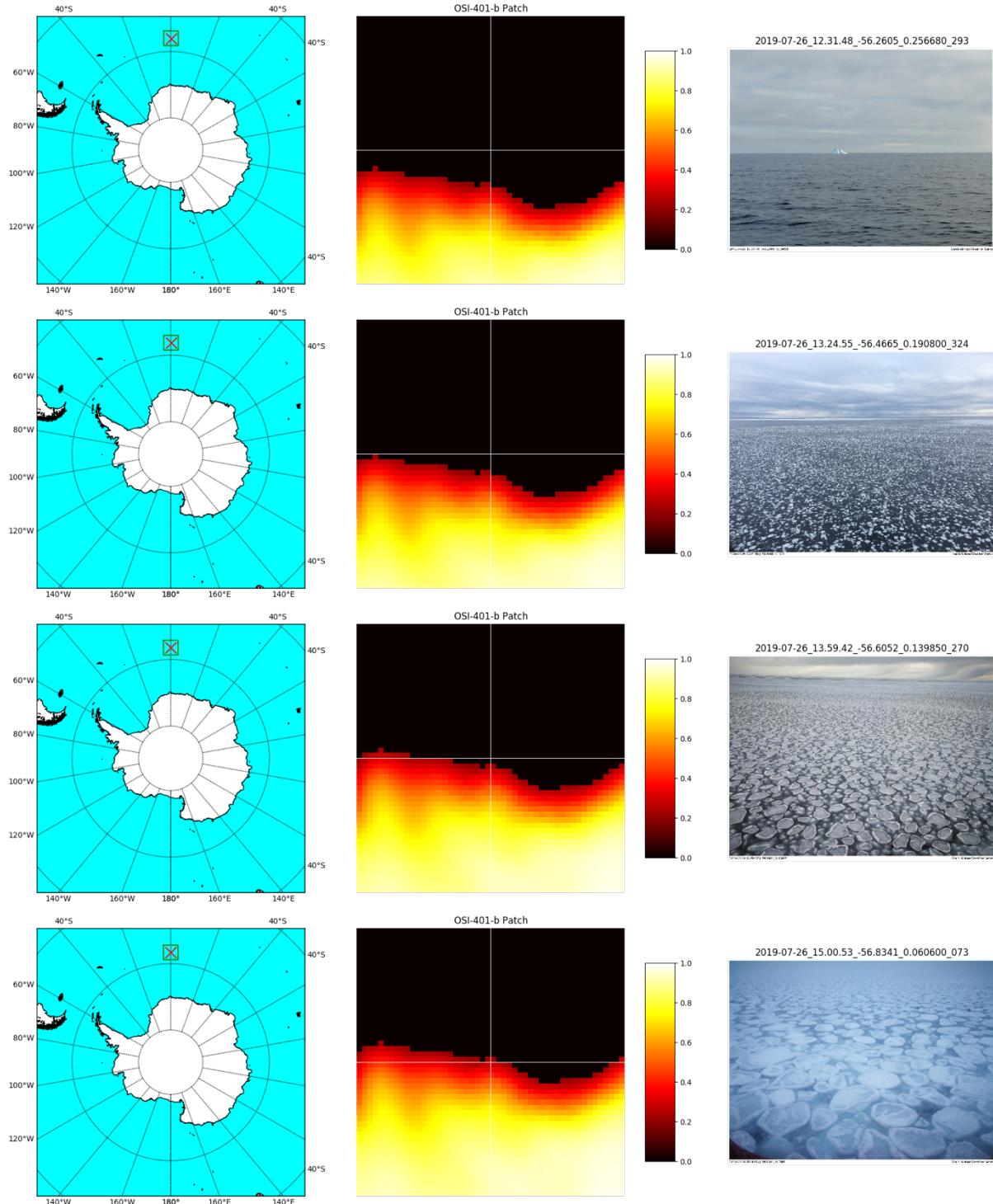
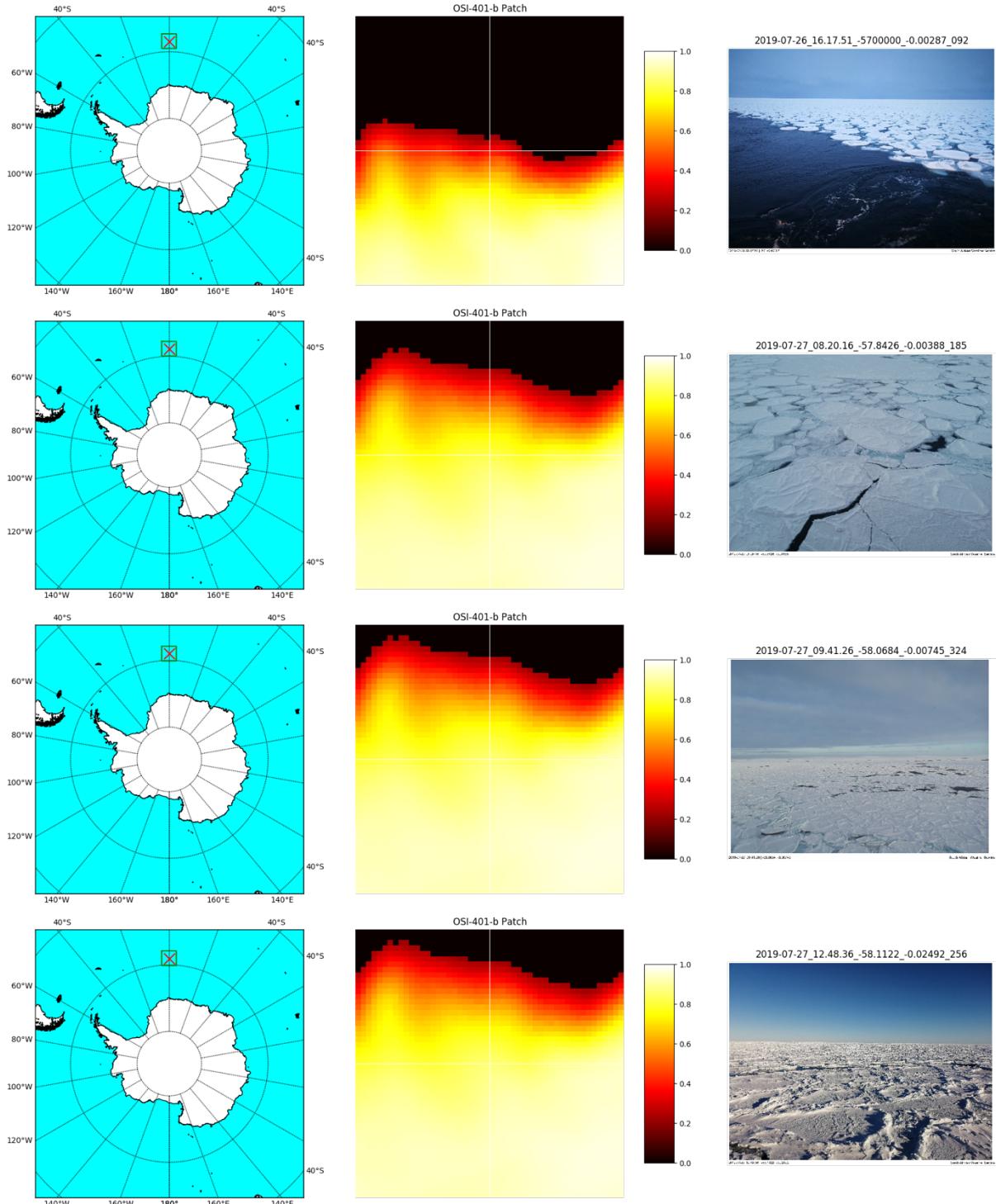


Figure 32: A set of concentration estimates generated using models trained partially on each hemisphere's data.

## Appendix C – In Situ Observation Comparisons





*Figure 33: A set of in situ observation comparisons generated using the comparison tool developed for use in this investigation. The PMW-derived concentration estimate shown is from the same day as each image, and the location is indicated on the concentration chart with crosshairs.*

# EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer ([Zulpha.Geyer@uct.ac.za](mailto:Zulpha.Geyer@uct.ac.za); Chem Eng Building, Ph 021 650 4791). Students must include a copy of the completed form with the final year project when it is submitted for examination.

<b>Name of Principal Researcher/Student:</b>	Stefan Dominicus		<b>Department:</b>	ELECTRICAL ENGINEERING	
If a Student:	YES	Degree:	BSc (Eng) Mechatronics	Supervisor:	Prof. Amit Mishra
<b>If a Research Contract indicate source of funding/sponsorship:</b>			NA		
<b>Research Project Title:</b>	Sea Ice Concentration Estimation Techniques Using Machine Learning.				

## Overview of ethics issues in your research project:

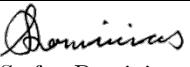
Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES	NO
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	NO
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

## I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	Date
Principal Researcher/Student:	 Stefan Dominicus	13 October 2019

This application is approved by:

Supervisor (if applicable):	 Prof. Amit Mishra	13 October 2019
HOD (or delegated nominee): Final authority for all assessments with NO to all questions and for all undergraduate research.	Janine Buxey	13 October 2019
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		