remote sensing

MDPI

Article

Deep Learning on Airborne Radar Echograms for Tracing Snow Accumulation Layers of the Greenland Ice Sheet

Debvrat Varshney [1], Maryam Rahnemoonfar [1,*], Masoud Yari [1], John Paden [2], Oluwanisola Ibikunle [2] and Jilu Li [2]

[1] Computer Vision and Remote Sensing Laboratory, University of Maryland Baltimore County, Baltimore, MD 21250, USA; dvarshney@umbc.edu (D.V.); yari@umbc.edu (M.Y.)

[2] Center for Remote Sensing of Ice Sheets (CReSIS), University of Kansas, Lawrence, KS 66045, USA; paden@ku.edu (J.P.); ibikunle.oluwanisola@ku.edu (O.I.); jiluli@ku.edu (J.L.)

[*] Correspondence: maryam@umbc.edu

Abstract: Climate change is extensively affecting ice sheets resulting in accelerating mass loss in recent decades. Assessment of this reduction and its causes is required to project future ice mass loss. Annual snow accumulation is an important component of the surface mass balance of ice sheets. While in situ snow accumulation measurements are temporally and spatially limited due to their high cost, airborne radar sounders can achieve ice sheet wide coverage by capturing and tracking annual snow layers in the radar images or echograms. In this paper, we use deep learning to uniquely identify the position of each annual snow layer in the Snow Radar echograms taken across different regions over the Greenland ice sheet. We train with more than 15,000 images generated from radar echograms and estimate the thickness of each snow layer within a mean absolute error of 0.54 to 7.28 pixels, depending on dataset. A highly precise snow layer thickness can help improve weather models and, thus, support glaciological studies. Such a well-trained deep learning model can be used with ever-growing datasets to aid in the accurate assessment of snow accumulation on the dynamically changing ice sheets.

Keywords: ice layer thickness; semantic segmentation; fully convolutional networks; Snow Radar

Citation: Varshney, D.; Rahnemoonfar, M.; Yari, M.; Paden, J.; Ibikunle, O.; Li, J. Deep Learning on Airborne Radar Echograms for Tracing Snow Accumulation Layers of the Greenland Ice Sheet. *Remote Sens.* **2021**, *13*, 2707. https://doi.org/10.3390/rs13142707

Academic Editors: Gareth Rees and Neil Arnold

Received: 9 June 2021
Accepted: 4 July 2021
Published: 9 July 2021

1. Introduction

Climate warming is rapidly reducing glaciers and ice-sheets across the world [1,2]. Research in climate science suggests that sea levels might rise by 80–90 cm as a result of increased melting of ice by the end of this century [2,3]. The climate affects the inputs and outputs to the ice sheet mass balance: snowfall adds mass to the ice sheet and melting, sublimation, and calving remove mass from the ice sheet. When an ice sheet is in equilibrium, the snowfall input is balanced with the outputs. Since climate change may lead to changes in snow accumulation, monitoring current and past snowfall is critical to projecting future snow accumulation, which, in turn, is needed to project mass balance.

Measurements obtained by airborne radar sensors, such as the Snow Radar [4], are one of the preferred methods of monitoring the effects of ongoing climate change on Greenland precipitation and run-off due to their ability to measure accumulation rate patterns at a large-scale [5]. The sensor produces a 2D profile of the snow accumulated over multiple years, where the horizontal axis represents the along-track direction, and the vertical axis represents the layer depth. These profiles along with their annual changes can be used to monitor the impact of climate change. The Snow Radar, due to its fine vertical resolution resulting from using an ultra-wide bandwidth, specifically helps in capturing internal snow layers. However, tracing each of these layers manually for all the flight campaigns is laborious and time consuming. The data collected by this sensor is further processed using signal processing techniques to produce grayscale echograms (Figure 1) which show profiles of internal ice layers at various depths. The bright regions in these images, with whiter pixels, are due to surface layers giving high reflectance; which generally correspond to the greatest snow density variations in the ice. As the ice layer depth increases, the radar return signal decreases in power [6], giving darker regions in these images. Further, each internal ice layer can be traced to the accumulated snow during the winter season of a particular year. Hence, by analyzing the depth of each of these internal ice layers (the bright and dark regions combined), glaciologists can assess the annual snow accumulation rate and use these values for improving weather and climate studies [5].

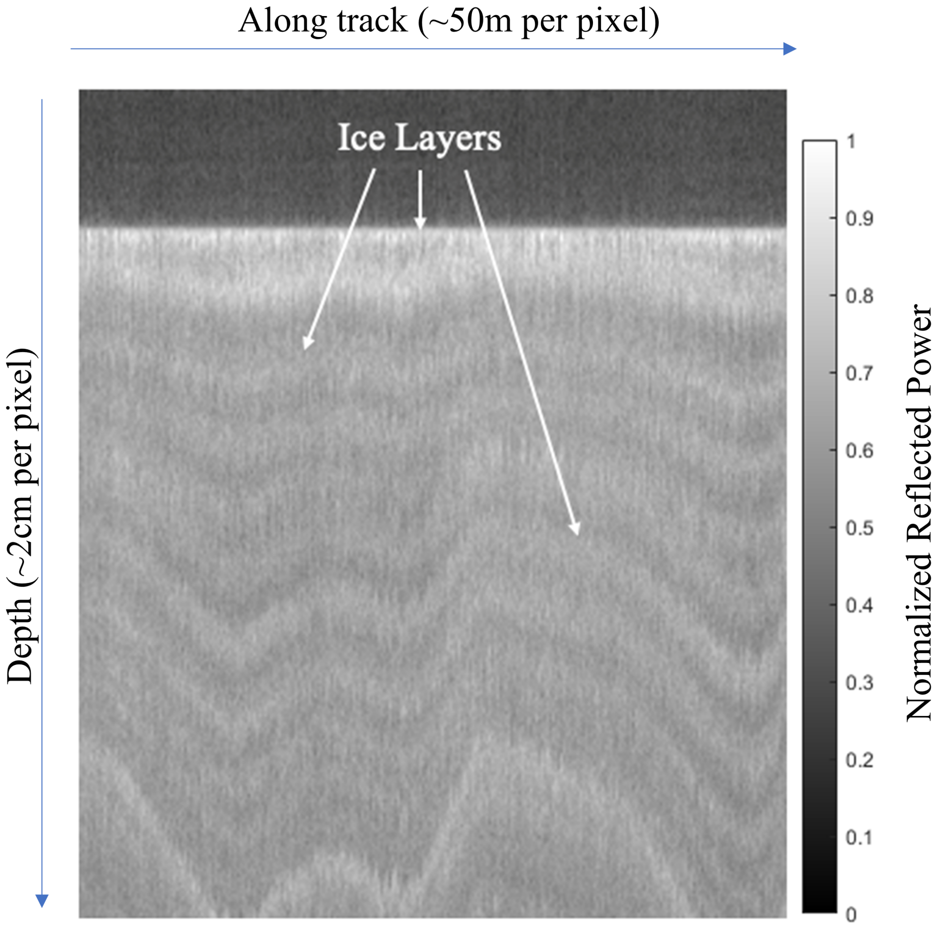


Figure 1. A Snow Radar echogram showing multiple internal layers of ice.

Developing algorithms to accurately detect and track the air-snow boundary (also called surface), and particularly the internal ice layers of the ice sheet, has become an important research topic. Detecting the internal layers is more challenging since they are compact and, hence, visually indistinguishable from each other. In this domain, MacGregor et al. [7], Koenig et al. [5], Rahnemoonfar et al. [3], Rahnemoonfar et al. [8], and Yari et al. [9,10] developed algorithms which could detect all the internal layers from a given echogram. MacGregor et al. [7] and Koenig et al. [5] developed semi-automatic techniques, which required manual correction to the algorithm's output, whereas Yari et al. [9,10] and Rahnemoonfar et al. [3] developed fully automatic techniques, which did not require manual corrections. Although the former techniques were semi-automatic, they could detect each internal layer uniquely with a separate label-value. The latter techniques, on the other hand, although fully automatic, labeled all the detected internal layers with the same value. In our work, we aim to detect all the internal layers, each with their own respective label-values (i.e., a 'multi-class' output), through a fully automatic technique.

Automated image processing has recently been empowered with the help of deep learning [11]. Fully convolutional networks (FCN), which are modified versions of convolutional neural networks (CNN), extract complex image features and help in image understanding. They can semantically segment an image and create multi-class maps, where each pixel can be categorized as belonging to a unique object present in the image. This has made them useful for various applications of computer vision, such as medical image understanding [12,13], remote sensing [14], and autonomous driving [15].

In this paper, we create multi-class output of Snow Radar images using FCNs. By multi-class output, we mean that we detect each ice layer in the image, and assign a separate, unique label to it. This helps us in identifying the individual extent of each internal ice layer and estimating its thickness. Using this methodology, we further assess the performance of our networks over two multi-temporal datasets, captured during different campaigns and flight paths over Greenland between the years 2009 and 2017. We first train our networks on a dataset from 2009 to 2012, and test it on a dataset from 2013 to 2017. To estimate the effect of denoising on the data, we also train and test our networks on the denoised dataset from 2009 to 2017.

Our paper is divided into the following sections: Section 2 highlights automated and semi-automated techniques to detect various kinds of ice layers in radar images. The section also describes recent state-of-the-art FCNs for semantic segmentation. Section 3 gives an insight into the Snow Radar dataset which we use to detect internal ice layers. Section 4 describes our methodology, which covers image cropping and denoising for pre-processing, detailed description of the FCN architectures for ice layer tracking, and post-processing steps to visualize the FCN outputs. Section 5 gives the hyperparameter details and also describes the evaluation metrics we use to analyze network performance. Section 6 gives the results, and Section 7 concludes the paper. The code used in this study is available at: http://github.com/BinaLab/ice-layer-tracking (accessed on 1 June 2021).

2. Related Work

Past work on radar image processing for ice layer tracking has focused on two-layer tracking [16–21] or multi-layer tracking [3,9,10]. These works detected ice layers in a binary format, that is, they detected either the presence or absence of an ice layer at each pixel. In this section, we describe these past techniques and also describe the FCNs that we use to detect internal ice layers in a multi-class format.

2.1. Ice Layer Tracking Techniques

2.1.1. Ice Surface and Bottom Detection

Automated techniques, such as Ferro and Bruzzone [17], Carrer and Bruzzone [16], Rahnemoonfar et al. [19,20], and Xu et al. [21], have been very useful for tracking the ice surface and bottom. Carrer and Bruzzone [16] developed a hidden markov model to process planetary echograms, whereas Ferro and Bruzzone [17] coupled Steger and Weiner filtering with denoising methods to detect linear features from radar data acquired over icy regions. Rahnemoonfar et al. [19] used a level set approach to evaluate airborne radar imagery, and Rahnemoonfar et al. [20] used anisotropic diffusion followed by a contour detection model to identify ice and bedrock layers. These methods were primarily developed to detect the surface and bottom layers of the echograms.

2.1.2. Internal Ice Layer Detection

Tracking all internal layers of an ice sheet is challenging since the layers are stacked very close to each other. There were several works in this field, such as MacGregor et al. [7], Koenig et al. [5], d. P. Onana et al. [22], and Carrer and Bruzzone [16], which used semi-automated techniques to detect internal layers. These algorithms were not fully automatic and required human correction. Hence, they were not scalable to larger datasets. Recent efforts, such as Yari et al. [9,10], Rahnemoonfar et al. [3,23], and Oluwanisola et al. [24], developed fully automatic techniques which applied deep learning to track and identify internal ice layers. Yari et al. [9,10] and Rahnemoonfar et al. [3] used multi-scale networks to detect ice layers in a binary format, i.e., they detected the presence and absence of an ice layer at each pixel, irrespective of which year that layer belonged to. Oluwanisola et al. [24] further used an iterative neural network to detect each ice layer separately, but their method was tested only on synthetic, simulated images and failed to detect layers on real radar images. To track and annually date every ice layer from a real radar image, all ice layers need to be distinguished from each other and detected uniquely.

In our work, we track the compact, closely spaced internal ice layers and distinguish them from each other through a fully automatic approach. We are able to detect each ice layer uniquely and calculate its thickness, which can be used for glaciological studies. We take the advantage of deep learning for its recent success and scalability to large datasets. In particular, we use FCNs for their ability to detect complex image features and assign a unique label to each pixel of a given input image.

2.2. Fully Convolutional Networks

FCNs have become a popular computer vision technique to create segments of an image and give a contextual label for every pixel. Where CNNs are used to classify the entire image with a label, FCNs are used to classify each pixel of an image with a label. Due to this, FCNs have been extensively used for semantic segmentation of images for a variety of problems. From medical image analysis [12,13] to autonomous driving [15] to satellite imagery [14], they have shown a high amount of accuracy in detecting complex image features. The varied use cases and optimal performance of these networks has turned this technique into a foundational concept in computer vision [25]. Architectures, such as UNet [13], introduced encoder-decoder modules with skip connections, whereas PSPNet [26] introduced spatial pyramid pooling to analyze a scene from various scales. DeepLabv3+ [25] introduced dilated convolutions in the spatial pyramid pooling modules to refine segmentation maps. Such techniques have improved feature detection from images, and we will be describing them in further detail in Section 4.

3. Dataset

3.1. Characteristics

The Center for Remote Sensing of Ice Sheets (CReSIS) has publicly available Snow Radar data captured during the NASA Operation Ice Bridge Mission. It is composed of two datasets [27,28] that were captured at different locations of the Greenland Ice Sheet. Koenig et al. [5] published the first set [27], where the data was captured across the entire Greenland from the years 2009 to 2012. Montgomery et al. [29] published the second set [28], which focused on the Southeast Greenland region. This dataset was a temporal extension of the first as it was captured between 2009 and 2017. From hereon, we will refer to these datasets as ‘Entire Greenland Dataset (EG)’ and ‘Southeast Greenland Dataset (SG)’ datasets, respectively. The images that they contain are described in Table 1.

Table 1. Datasets and their description.

Dataset	Region	Timespan (Years)	Number of Images
EG	Entire Greenland	2009 to 2012	21,445
SG	Southeast Greenland	2009 to 2017	2037
EG-2012	Entire Greenland	2012	2621

SG contains 2037 images, whereas EG contains 21,445 images. Out of these, the data collected in the year 2012 for EG contains the most varied images, captured over both wet and dry zones, with more than ten internal layers visible in the dry zones. Hence, we consider this set of 2621 images from EG’s 2012 data as a special set for training our neural network classifiers. The datasets are tabulated in Table 1 for the reader. The Greenland regions which they have captured are shown in Figure 2.

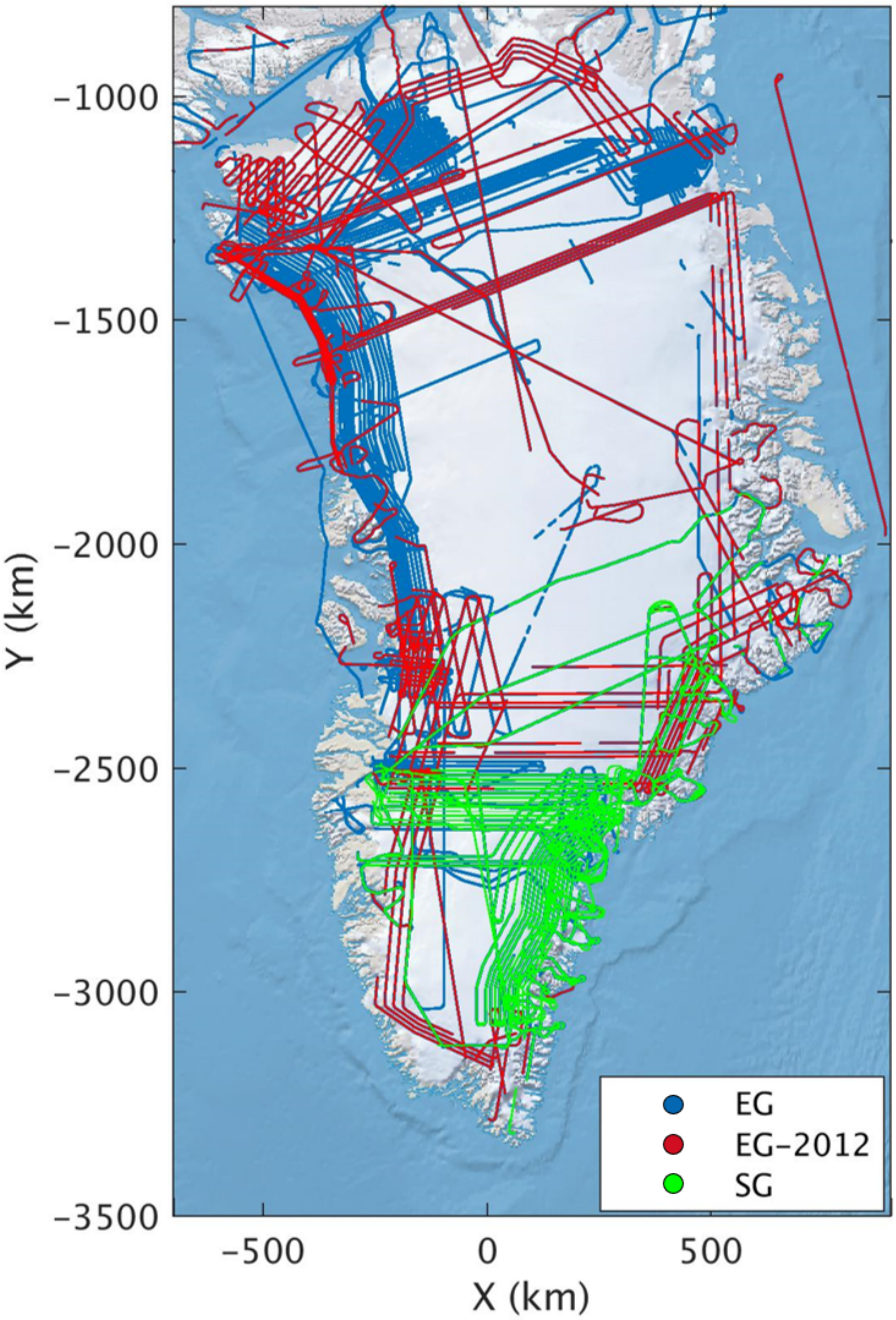


Figure 2. Map showing the flight paths where the airborne Snow Radar was flown to capture the images.

Further, the annotations for both EG and SG are available as outputs of Koenig et al. [5]. In these annotations, each ice-layer from a particular year (i.e., present at a unique depth in the ice sheet) is marked as a unique label value and is considered as a separate class.

3.2. Challenges

Radar images can be very noisy, and the internal layers in a typical Snow Radar are visually indistinguishable from each other. They fade out at most regions and do not have sharp boundaries to determine their extent. There is not a lot of variation in texture and contrast between different ice layers, which are typically needed for a CNN to extract features from images. There can also be surface tracking errors, as well as constructive and destructive along-track stacking. Most of these issues can be seen in a typical radar image, as shown in Figure 3a.

These issues make it very hard for glaciologists to manually trace and track the ice layers. While manually tracing the ice layers, an annotator is bound to make mistakes and create incorrect labels, such as those shown in Figure 3b. In this figure, labels for deeper layers are not present. Out of the labels which are present, they do not extend across the entire image, and are incomplete in the middle. These issues of lack of distinct features in the radar image, as well as lack of proper training labels, make it difficult for a CNN or FCN to learn information from them. To address these issues, we introduce a methodology described in Section 4 to crop out and denoise the input radar image, as well as increase the training labels available for each layer.

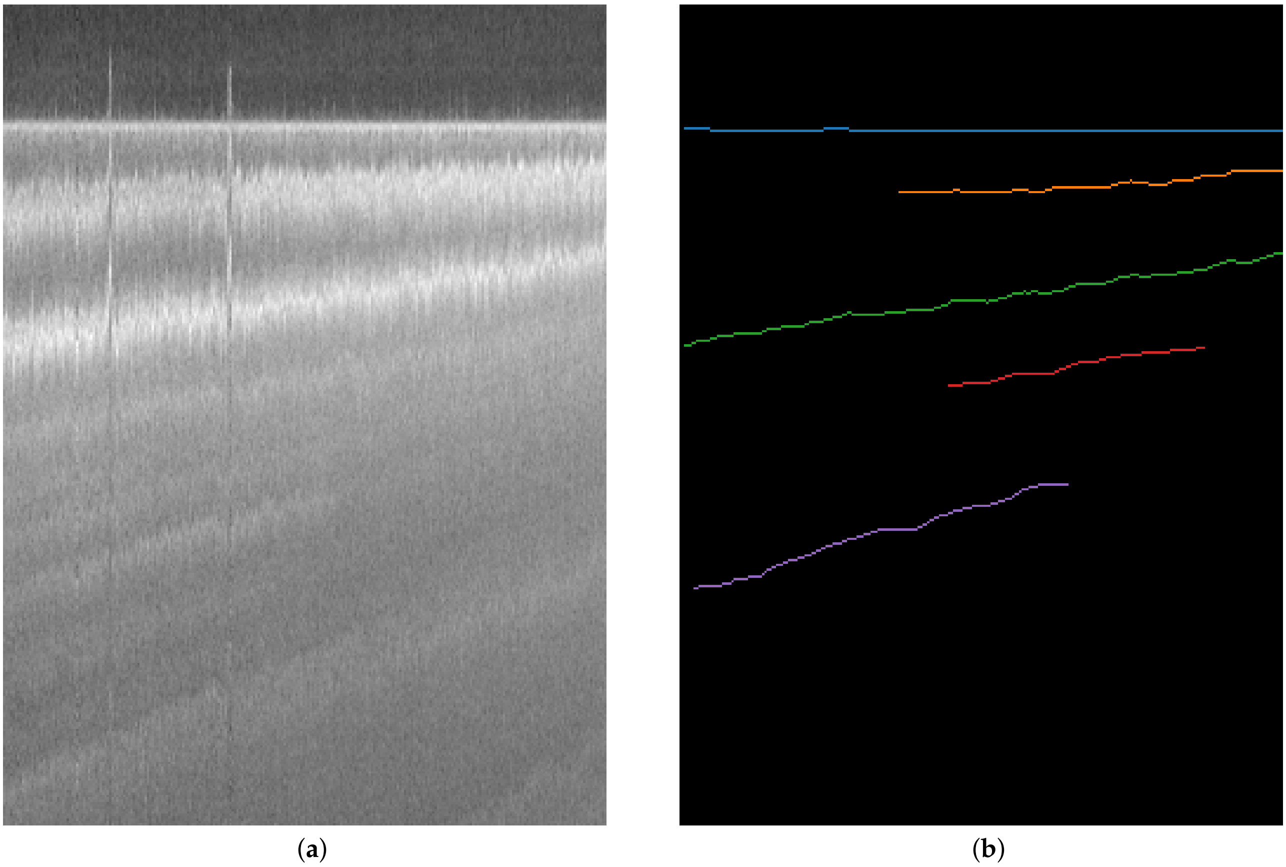


Figure 3. Noisy radar image (**a**) and its labels (**b**). Both are generated through Koenig et al. [5]. The deep layers in the radar image cannot be seen properly, and the labels available for them are incomplete.

4. Methodology

Manual annotation of echograms showing internal ice layers is tedious, time consuming, and prone to error, as the images are noisy and the layers are visually indistinguishable. The manual annotation has lead to incomplete labels for most layers, and, in other cases, there are no labels available for a layer. Hence, this creates an issue for training FCNs which are highly data-hungry algorithms. We resolve these issues by a set of pre-processing techniques which incorporate image cropping and image denoising. Out of the two pre-processing methods, image cropping is the most important as it helps us prepare complete training labels for FCNs. Using FCNs will help in learning the spatio-contextual information from the radar images, which would help detect the common features of every ice layer. The outputs from the pre-processing techniques are fed into state-of-the-art FCNs, which is then followed by post-processing steps.

4.1. Pre-Processing

4.1.1. Image Cropping

The sparsity in the available annotations, listed out in Section 3.2, becomes a challenge in using them with FCNs. Hence, we introduce a set of steps which crop out the complete, useful labels from the labeled image and the same portion from its corresponding radar image. The labels are then additionally populated so as to increase the training labels for each layer for the purpose of semantic segmentation.

For image cropping, the idea is to find a bounding box, which contains complete layers having a consecutive set of labels. Once we find the coordinates of this bounding box, we can simply crop out that portion from the labeled image, and the corresponding radar image, for further processing. All our bounding boxes are across the entire width of the image, and only their vertical coordinates may vary. For example, a bounding box with coordinates $(x_1, y_1), (x_2, y_1), (x_1, y_2)$, and (x_2, y_2), we set x_1 as 0 and x_2 as the width of the image w. In addition, note that one image can have multiple bounding boxes to result in multiple crops of an original image.

To find out the consecutive set of complete labels for our bounding boxes, we first discard all the incomplete layers, i.e., turn their label values to 0, the background class. For example, the second, fourth, and fifth layers in Figure 3b are incomplete, which we discard from our processing. Then, beginning from the top of the image, we look for sets of layers whose label values are consecutive, such as $2, 3$ or $4, 5, 6$, etc. In such a manner, we find no consecutive sets in Figure 3b. For other images, whenever we find a consecutive set of layers, we also calculate the minimum row index (l) of the layer with the lowest label value (i.e., the first, or the 'top' layer in the consecutive set) and the maximum row index (h) of the layer with the highest label value (i.e., the last, or the 'bottom' layer in the consecutive set). For an efficient processing of images, we add a margin of five to both the indices and set them (index + margin) as the vertical coordinates, i.e., y_1 and y_2 of the bounding box for cropping. The width of the bounding box is the width w of the entire label image, i.e., x_1 is 0, and x_2 is w. The bounding box coordinates, thus, found on the labeled image are also used to crop out the matching portion from the corresponding radar image.

Coordinates for cropping are explained in Equation (1). For a minimum row index l and a maximum row index h (i.e., $l < h$), in a consecutive set, the bounding box can have the coordinate values, as shown in Equation (1). Here, w is the width of the image, and $(x_1, y_1), (x_2, y_2)$ form the diagonally opposite corners of the bounding box. Once these are calculated, we use the same coordinates to extract matching regions from radar image.

$$(x_1, y_1), (x_2, y_2) = (0, l-5), (w, h+5). \tag{1}$$

Take Figure 4, for example. The 'ground truth' here, and subsequently throughout the paper, refers to labels generated through Koenig et al. [5] and does not imply in-situ observations. Figure 4 has labels numbered 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 18, 19, and 20, where each label signifies a unique ice layer. In this set, layers labeled 1, 4, 15, 16, and 17 are missing. Moreover, layer labeled 12 is incomplete and does not annotate the entire ice layer across the image. As such incomplete and missing layers would not help in semantic segmentation, we completely remove the layer labeled as 12, i.e., we turn it into the background class. This means that we covert all pixels with value 12 to have a value 0 (the background class). So, if you remove layer 12, the consecutive sets formed from the original set of labels for this image would be: {2,3}, {5, 6, 7, 8, 9, 10, 11}, {13, 14}, and {18, 19, 20}. The top and bottom layers of these sets, as explained the previous paragraph, would be layers {2, 5, 13, 18} and {3, 11, 14, 20}, respectively, to give us four cropped labeled images ('Cropped Layers' in Figure 4).

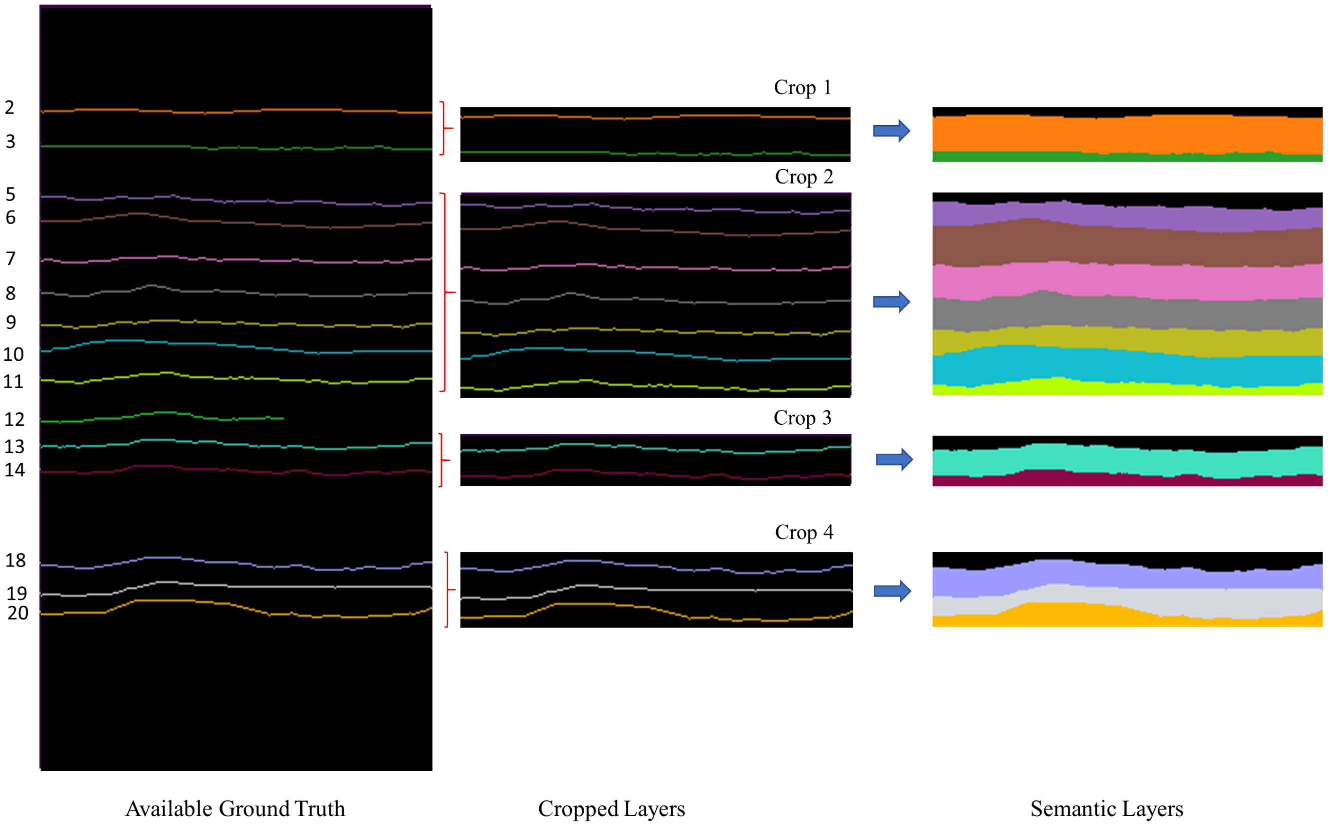


Figure 4. Processing the training labels: We crop out the 'Available Ground Truth', into multiple segments, as shown in 'Cropped Layers'. These cropped portions are then prepared for semantic segmentation, as shown in 'Semantic Layers'.

The cropped labels, each of height 1 pixel and width w pixels, only demarcate the surface of an ice layer. The ice layer is still deeper than a single pixel. To make our neural network learn the entire deep extent of each ice layer, we populate the intermediate region between two complete layers, with the upper label. The 'Semantic Layers' column in Figure 4 shows the outputs after re-populating. Once we have these semantic layer-labels prepared, we crop out the matching portion from the corresponding radar image, using the previously calculated bounding boxes. The cropped out radar images are shown in Figure 5.

4.1.2. Denoising

Zhang et al. [30] found that, by integrating ResNet's [31] residual blocks in UNet's [13] architecture, the noise from an input image can be removed. This architecture, called DRUNet, consists of four scales with skip connections. The architecture can handle various noise levels, σ, i.e., the standard deviation of an assumed additive white Gaussian noise (AWGN). We use noise level 15 for all our experiments.

4.2. FCN Architectures

Background of CNNs, fully connected and fully convolutional layers is well explained in Section 4 of Rahnemoonfar et al. [3]. In this section of our paper, we give some details about the architectures of UNet [13], PSPNet [26], and DeepLabv3+ [25], which we use to perform semantic segmentation on Snow Radar images.

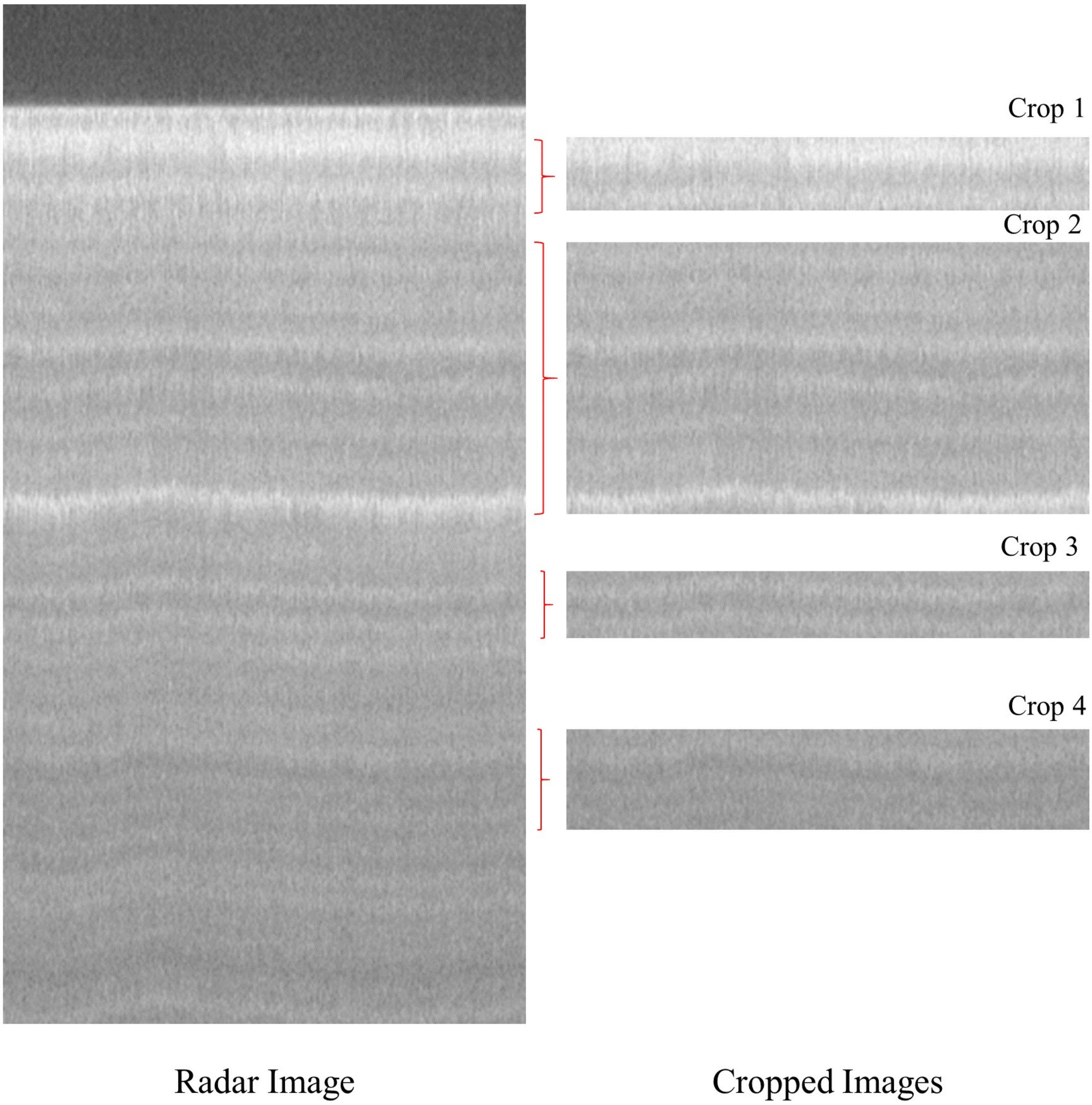


Figure 5. Corresponding radar image of Figure 4, where matching portions have been cropped out.

4.2.1. UNet

UNet [13] is made up of two symmetric modules: an encoder and a decoder. Both are stacked beside each other and form a U-shaped network architecture. The encoder downsamples the input image to form features maps in high resolution. These feature maps are then upsampled by the decoder to extract more global information from them. In between, there are skip connections between various features, and this methodology was found useful in detecting tissue deformations. UNet's success in processing biomedical images [32,33] inspired its application in remote sensing [34,35] and autonomous driving [36], among other domains. Its architecture is shown in Figure 6, where the green arrows depict forward connections, and purple arrows depict skip connections. 'I/x' denotes a downsampled, spatially reduced, input image 'I', while 64, 128, 256, and 512 denote the number of channels at each layer.

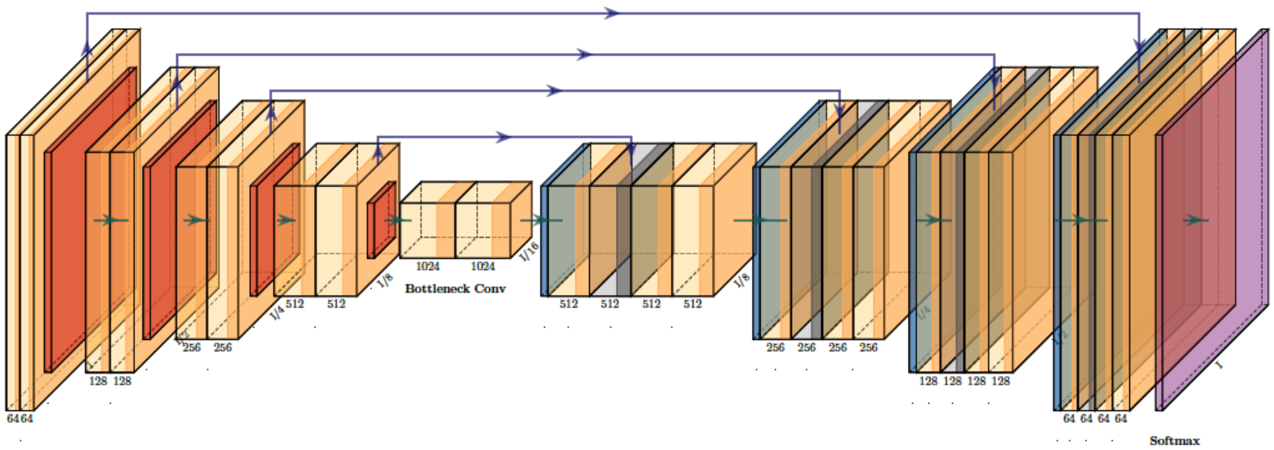


Figure 6. UNet [13] architecture. Green arrows depict forward connections, and purple arrows depict skip connections. 'I/x' denotes a downsampled, spatially reduced, input image 'I', while 64, 128, 256, and 512 denote the channel numbers at each layer.

4.2.2. PSPNet

Zhao et al. [26] noted that most FCNs lacked a global contextual understanding of pixels. To address this issue, they introduced a module which could act as a prior for the images. This was the pyramid pooling module which could extract global, as well as local, information from the images. There are four scales of the module in this architecture, shown in Figure 7. At the largest scale (the red block of the pyramid pooling module of Figure 7), the module learns a global perspective, while, for smaller scales, the module learns contextual representation which are local to that particular scale. We use PSPNet with a ResNet-50 [31] backbone, but, unlike ResNet, PSPNet has an additional loss function at the end of the fourth block of ResNet which deeply supervises [37] its architecture. The network learns through a weighted combination of two loss functions: the main loss and the additional loss.

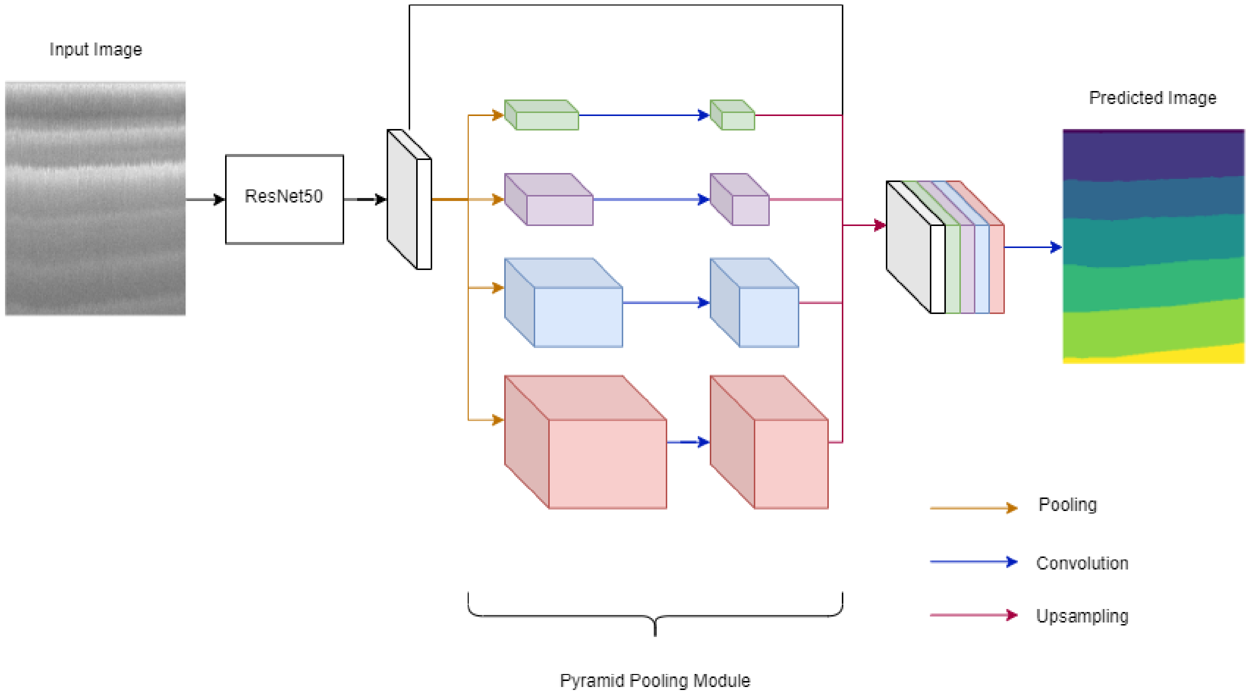


Figure 7. PSPNet [26] architecture.

4.2.3. DeepLabv3+

The DeepLabv3+ [25] architecture is shown in Figure 8. Here, we use ResNet-50 [31] as the backbone network. The backbone network is followed by an atrous spatial pyramid pooling module (ASPP) which contains multiple dilated convolutions at different rates (the red cells in Figure 8). The output from the ASPP is finally upsampled and concatenated with feature maps from the backbone network to decode the encoder outputs. There is a final upsampling at the end of the decoder which gives us the network outputs.

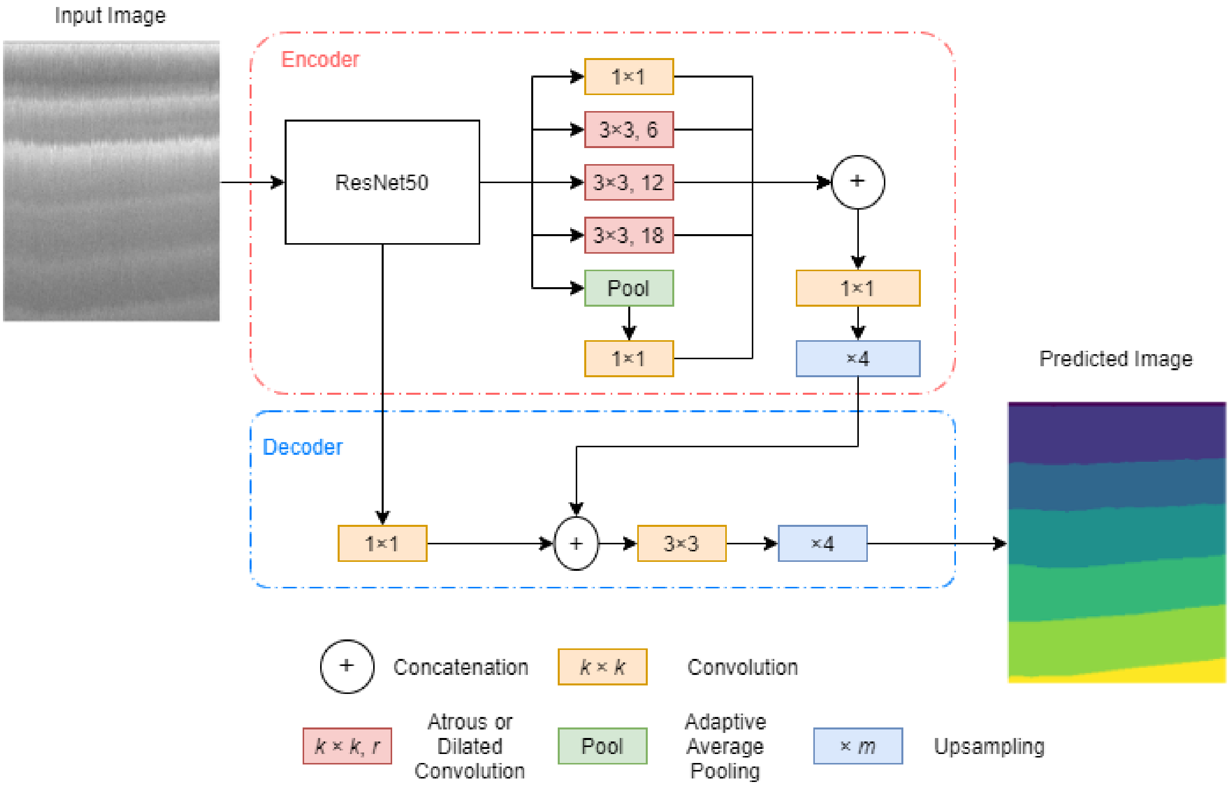


Figure 8. DeepLabv3+ [25] architecture with ResNet50 [31] backbone.

4.3. Tracing of Annual Snow Accumulation Layers

The networks described above give us pixel-wise outputs, similar to the ones shown in 'Semantic Layers' of Figure 4. These semantic layers need to be converted back to traced annual snow accumulation layers, similar to 'Cropped Layers' of Figure 4, that are useful for glaciologists. We do so by going through each pixel of every column and discarding all duplicate entries of a label. This means that, in every column, a label is preserved at the row where it first appears, and it is turned to 0 (background) wherever it appears thereafter. This way, we assign each unique label to just one single pixel in every column. Recall that the unique layer-labels at the input of the networks were annotated by glaciologists and represent snow accumulation layers, such as the bottom radar echogram, in Figure 3 of Koenig et al. [5]. When seen from all columns of the entire image and multiple consecutive images, the single pixels per column, for each label, become a traced annual snow accumulation layer that may extend hundreds to thousands of kilometers.

5. Experimental Setup

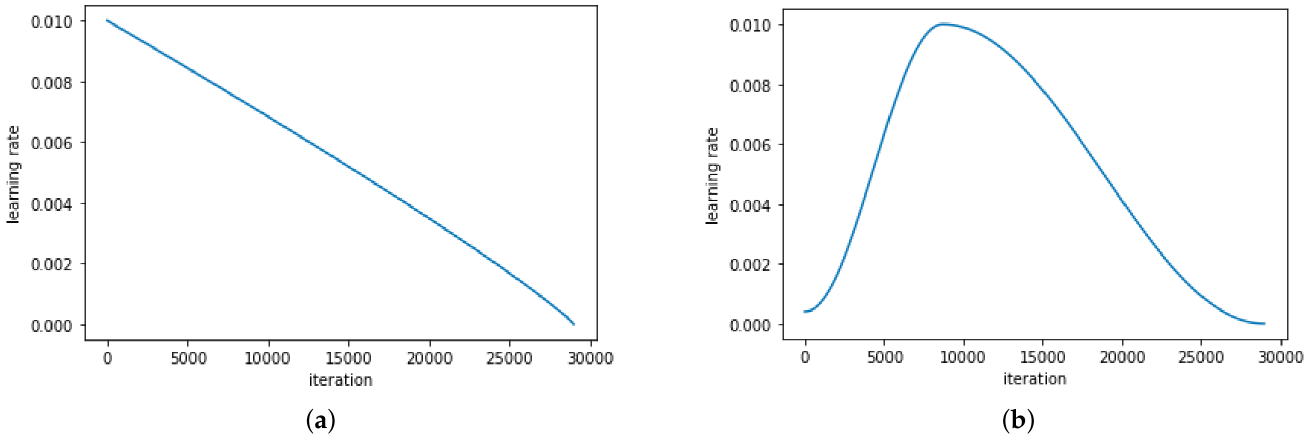
This section explains the hyperparameters, evaluation metrics for our experiments, and how we assess the detected ice layer thickness. We train the networks in multiple iterations, where a single 'batch' of images is used for every iteration. When all the images of a dataset have been used for training a network, this constitutes one 'epoch'. Below, we explain three categories of experiments we carried out on images cropped as per Section 4.1.

1. Semantic segmentation of EG-2012:- EG's 2012 data contains more than two thousand images having a variety of features, from those containing only the ice surface, to those having twenty to thirty internal layers. Hence, we use this data subset to find which of the three state-of-the-art FCNs will be useful for ice layer tracking. We keep 260 images for testing and the remaining 2361 for training. We train each network with two different learning rate strategies for 200 epochs each.
2. Training on past data and predicting on future data:- For this category of experiments, we train the most generalized network obtained over EG-2012 and train it on the entire EG dataset, i.e., over the years 2009 to 2012. We then predict the ice layers on 'future years', 2013–2017, from the SG dataset. Training on past data before 2012, and testing on data post-2012, would give us a sense of applicability of our current method for future datasets, in general. We train on past years of EG and not of SG because the former has a larger, and varied, number of images. Further, we experiment with denoising all the images before feeding them into the FCN in order to improve layer detection. In this category, we train the network for 70 epochs.
3. Training and testing on combined dataset:- Here, we use the network which gave the least mean absolute error (described below) over EG-2012 and train it on the entire Snow Radar data we have, i.e., EG and SG combined. We split the dataset into 80% for training and 20% for testing. As this is a large dataset, it would help in nourishing FCN. Here, again, we experiment with training on both regular and denoised images, for 70 epochs.

All denoising experiments with were carried out with noise level (σ) 15 [30].

5.1. Hyperparameters

We used ResNet-50 as the backbone network for DeepLabv3+ and PSPNet architectures. We started training with a learning rate of 0.01 and used 'Poly', as well as 'OneCycle', learning rate schedulers to change the learning rate during each iteration. Poly learning rate scheduler linearly reduces the learning rate from a starting value (0.01) to zero at the end of training (Figure 9a). The OneCycle learning rate scheduler anneals [38] the learning rate, as shown in Figure 9b. We used a momentum of 0.9 also annealed in a similar fashion, and a constant weight decay of 10^{-4}. We trained all networks (three FCNs, each with two learning rate schedulers) for 200 epochs with a binary cross entropy loss. The EG-2012 dataset, after cropping, resulted in 1158 images, which we split into batches of 8 images each. This resulted in 145 iterations per epoch. Figure 9 shows the learning rate schedulers for the total iterations across all 200 epochs.



(a) (b)

Figure 9. The two learning rate schedulers (LRS): (**a**) Poly and (**b**) OneCycle.

5.2. Evaluation Metrics

We calculate overall accuracy in Equation (2), mean Intersection Over Union (IoU) per label in Equation (3), and F-score in Equation (4) on the network outputs to assess the performance of our networks. In the three equations, k represents the number of labels. TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively:

$$\text{Accuracy} = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \tag{2}$$

$$\text{mean IoU} = \frac{1}{k}\sum_{i=1}^{k}\frac{\text{Predicted Output}_i \cap \text{Ground Truth}_i}{\text{Predicted Output}_i \cup \text{Ground Truth}_i}, \tag{3}$$

$$\text{F-score} = \frac{TP}{TP + \frac{1}{2}\frac{FP}{FN}}. \tag{4}$$

We calculate the mean thickness of each predicted layer and the semantic ground truth labels by counting the total occurrences of each label (k) and dividing it by the total number of columns in the image, i.e., its width w. This can help us give the mean absolute error (MAE) for all the layers across each image. Calculating MAE is shown in Equation (5), where ith layer, $i \in [1, k]$, p_i is the predicted mean thickness from the network outputs, and t_i is the mean thickness from the semantic ground truth layers.

$$\text{MAE} = \frac{1}{k}\sum_{i=1}^{k} \mid p_i - t_i \mid, \tag{5}$$

where p_i is the predicted mean thickness, and t_i is the true mean thickness of the ith layer.

6. Results and Discussion

Here, we discuss all the results obtained from our various experiments on different types of datasets. We calculated the accuracy, mean IoU per (layer) class, and F1 score based on Equations (2)–(4), respectively. Further, images obtained through Zhang et al. [30]'s denoising architecture gave us a slightly cleaner set of images shown in Figure 10.

6.1. Segmentation and Thickness Estimation of EG-2012

Tables 2 and 3 show the accuracy and mean IoU, respectively, on EG-2012 data for six experiments (three FCNs, each having two learning rate schedulers). In both these tables, the highest values obtained across all experiments are highlighted in bold. In these tables, 'LRS' stands for learning rate scheduler, as shown in Figure 9.

Poly LRS gave a higher performance with PSPNet and DeepLabv3+, both of which are multi-scale architectures. On the other hand, OneCycle gave a higher performance with UNet. This implies that annealing the learning rate hampers learning in case of multi-scale pooling architectures. In general, out of all the three FCNs, PSPNet performed the best, giving a high mean IoU and accuracy. The quantiative and qualitative performance of UNet were the worst, as can be seen in Tables 2 and 3 and Figures 11 and 12, respectively. UNet gave incomplete and 'shattered' lines, while completely missing out a lot of the layers.

UNet's poor performance can be attributed to its lack of a multi-scale architecture. DeepLabv3+ captures intricate details, as well as the global context of the image as a prior. This helps the architecture to learn the contrast variation for layer detection, as well sees the layers from a global perspective, in order to form their structure.

We also calculate the MAE of the predicted thicknesses and report them in Table 4. In general, all networks did a fair job in precisely predicting the layer thicknesses with an average MAE of 5.28 pixels across all datasets. MAE of DeepLabv3+ was the best, 2.36 pixels on the training set and 3.59 pixels on the test set. PSPNet came close, with MAE of 2.80 pixels on the training set and 3.63 pixels on the test set. UNet, gave the worst MAE of 6.17 pixels on the test set.

The tracking error in pixels can be translated to depth errors for known vertical pixel size which is determined by fast-time sampling frequency. For example, the vertical pixel size for EG-2012 is 2.07 cm. Thus, DeepLabv3+'s MAE of 2 to 4 pixels (our best result, depending on the learning rate and the dataset split) will translate to approximately 4 cm to 8 cm in snow with a dielectric constant of 1.53.

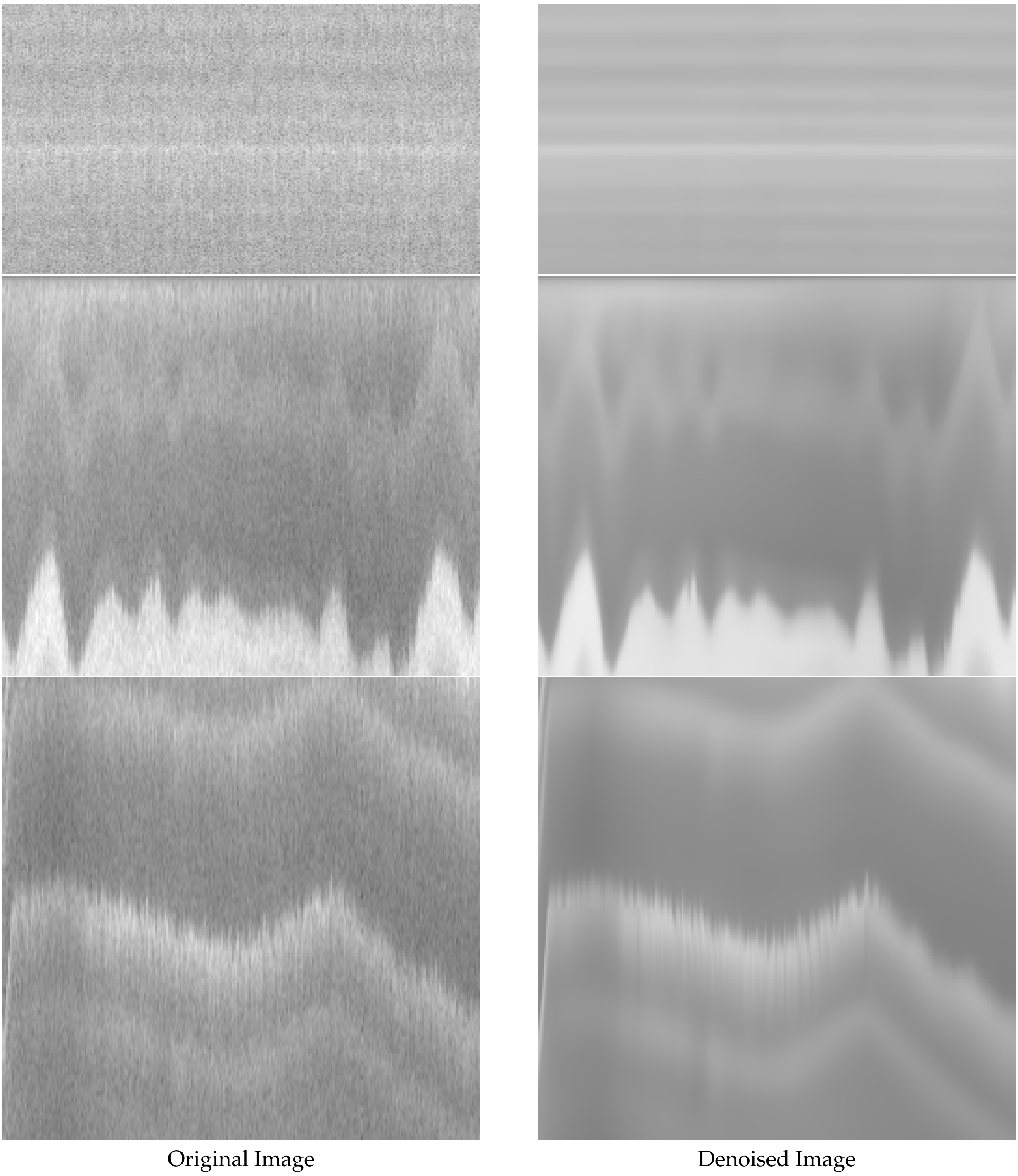


Figure 10. Some sample radar images on the left, and their denoised images on the right. Each radar image is from different years within EG or SG, and has a different resolution.

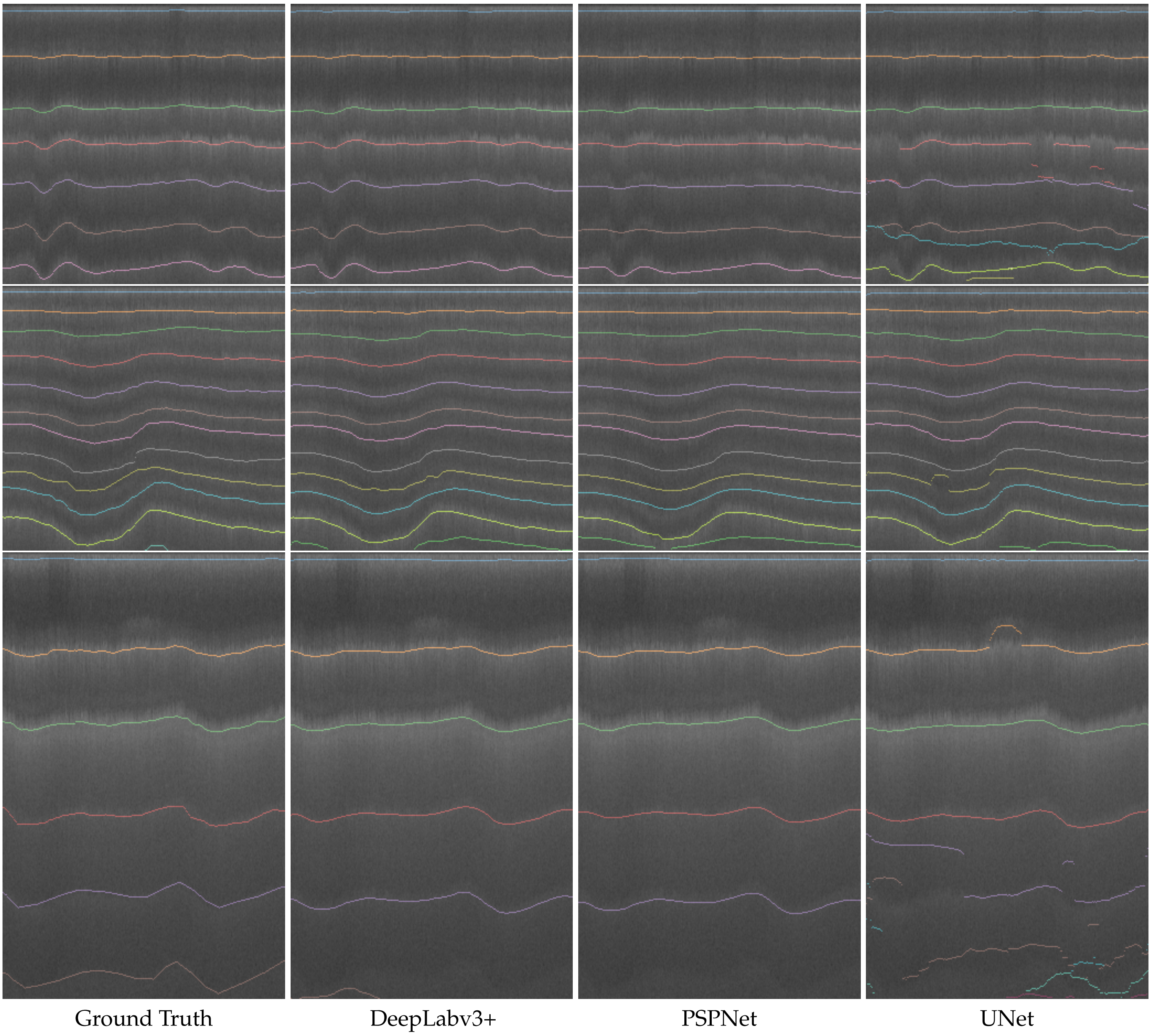


Figure 11. Radar images overlaid with Ground Truth layers (**first column**) and layers predicted from the three FCNs: DeepLabv3+ (**second column**), PSPNet (**third column**), and UNet (**fourth column**). The results shown are from each network's best output in terms of mean IoU (Table 3). More examples are in Figure 12.

Table 2. Network accuracy on the train and test sets. Numbers in bold represent the best results.

Network-LRS	Train	Test
UNet-Poly	0.755	0.714
UNet-OneCycle	0.856	0.792
PSPNet-Poly	**0.948**	**0.899**
PSPNet-OneCycle	0.938	0.867
DeepLabv3+-Poly	0.957	0.887
DeepLabv3+-OneCycle	0.935	0.886

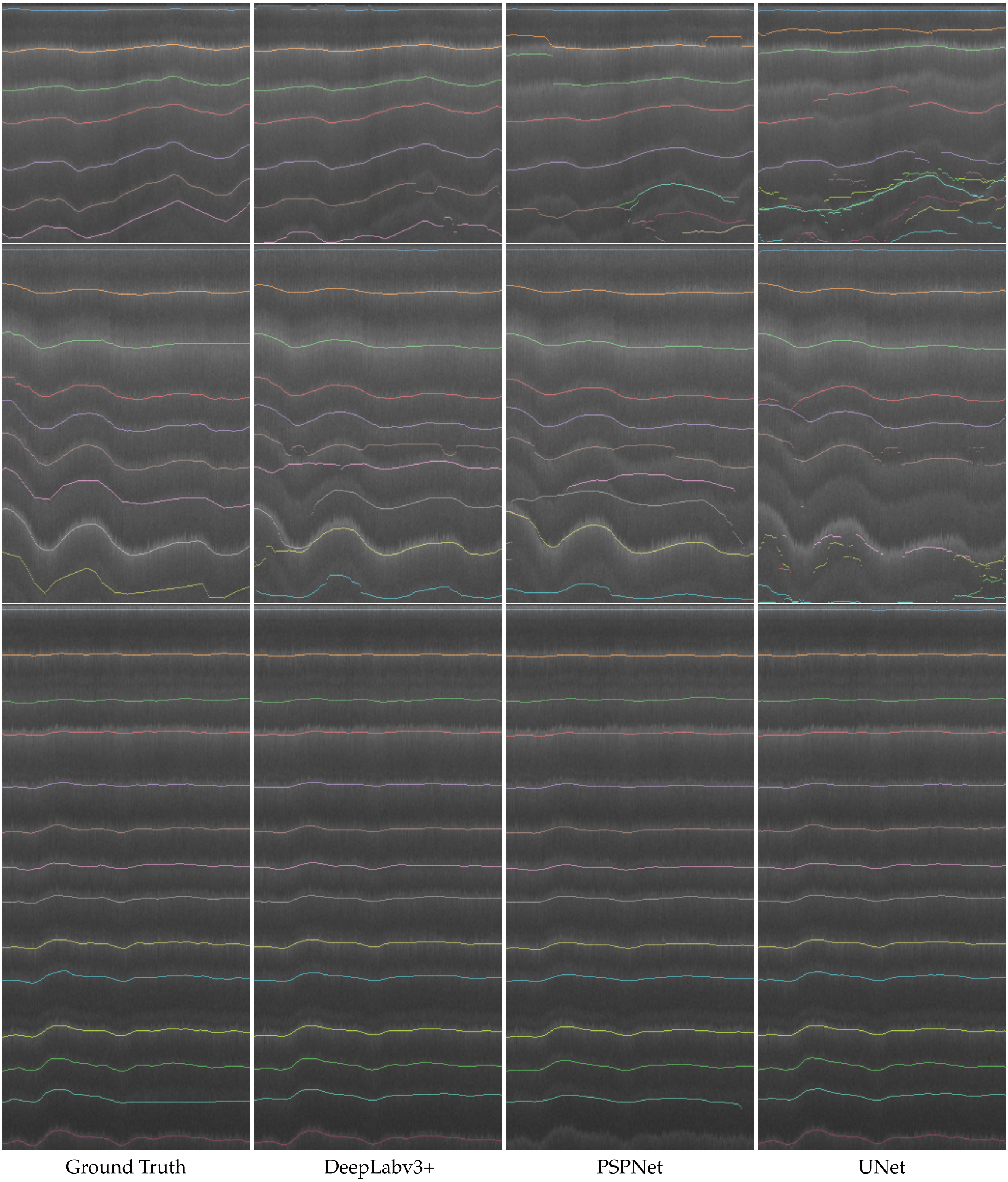


Figure 12. Radar images overlaid with Ground Truth layers (**first column**) and layers predicted from the three FCNs: DeepLabv3+ (**second column**), PSPNet (**third column**), and UNet (**fourth column**). The results shown are from each network's best output in terms of mean IoU (Table 3). More examples are in Figure 11.

Table 3. Mean Intersection over Union (IoU) obtained through the six experiments on the train and test sets. Numbers in bold represent the best results.

Network-LRS	Train	Test
UNet-Poly	0.387	0.343
UNet-OneCycle	0.549	0.438
PSPNet-Poly	**0.737**	**0.650**
PSPNet-OneCycle	0.728	0.589
DeepLabv3+-Poly	0.734	0.590
DeepLabv3+-OneCycle	0.676	0.595

Table 4. The mean absolute error (MAE) in pixels achieved by different architectures, obtained over all the layers calculated on our data splits: train and test. Numbers in bold represent the best results.

Network-LRS	Train	Test
UNet-Poly	7.95	8.75
UNet-OneCycle	5.22	6.17
PSPNet-Poly	2.80	3.63
PSPNet-OneCycle	4.03	5.62
DeepLabv3+-Poly	**2.36**	3.75
DeepLabv3+-OneCycle	3.08	**3.59**

6.2. Training on Past Data and Predicting on Future Data

Since, in the previous subsection, we saw that DeepLabv3+ with a Poly learning rate scheduler gave the lowest MAE, we used this network to learn image features from the EG dataset, and test it on 2013–2017 of SG dataset. This will allow us to see how, training on recent or past years, will help predict ice layer changes for a future unseen dataset. We also experimented with learning from denoised images, as well.

From Figure 13, we see that a network trained over a previous time-span (2009–2012) was successfully able to detect internal layers from a future unseen time-span (2013–2017) with a nearly 80% accuracy. Further, training on a denoised set of EG images helps in achieving a higher accuracy and F-score on both EG and SG's 2013–2017 datasets, as compared to training on non-denoised images. EG images are slightly noisy, so learning from their denoised versions helped in extracting layer features overall.

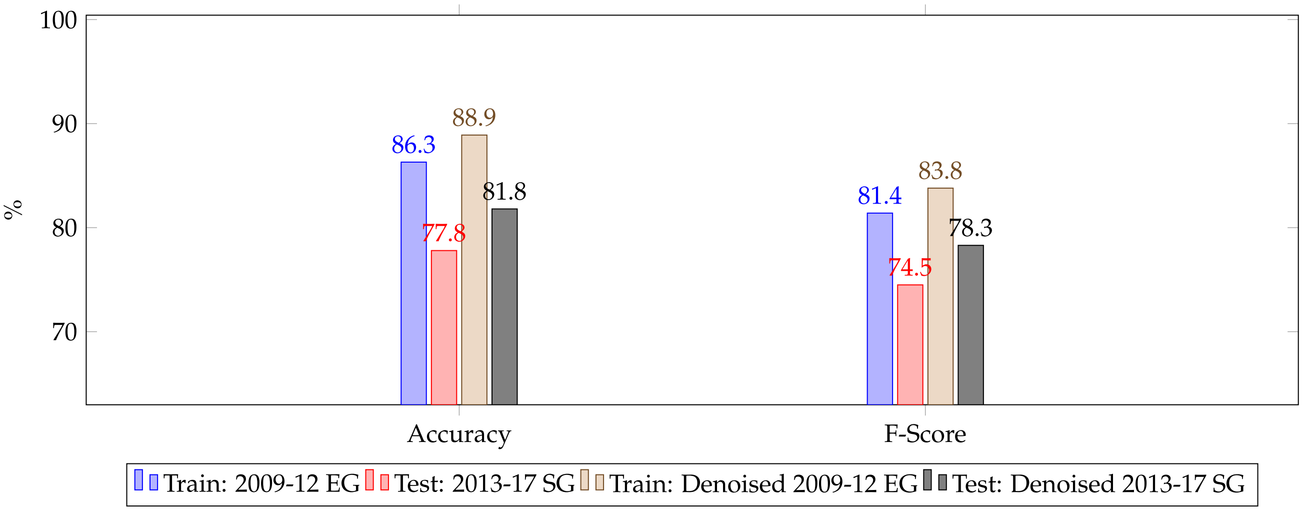


Figure 13. Accuracy and F-score obtained by training on 2009–2012 dataset and testing on 2013–2017 dataset.

6.3. Training and Testing on Combined Dataset

Figures 14 and 15 show the quantitative results of training a DeepLabv3+ with Poly learning rate on the entire dataset (EG+SG), as well as its denoised version. Figure 14 shows the accuracy and F-score over the training set, while Figure 15 shows the accuracy and F-score over the test set. On both, the test set and its denoised version, we see that the network was able to achieve nearly 80% accuracy in detecting internal layers.

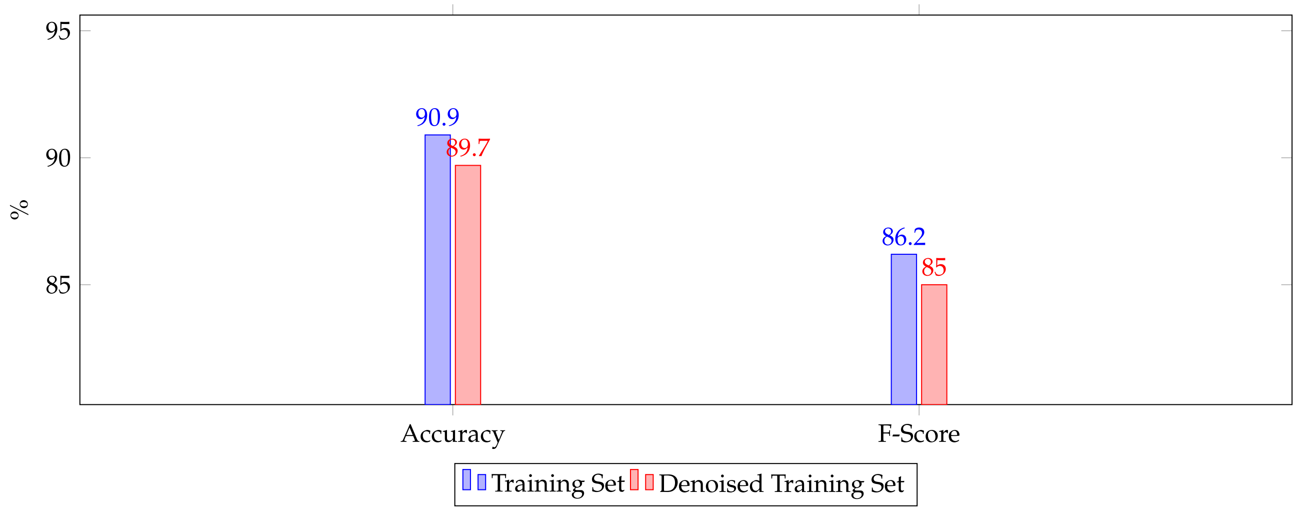


Figure 14. Accuracy and F-1 score obtained over the combined training set and the combined denoised training set.

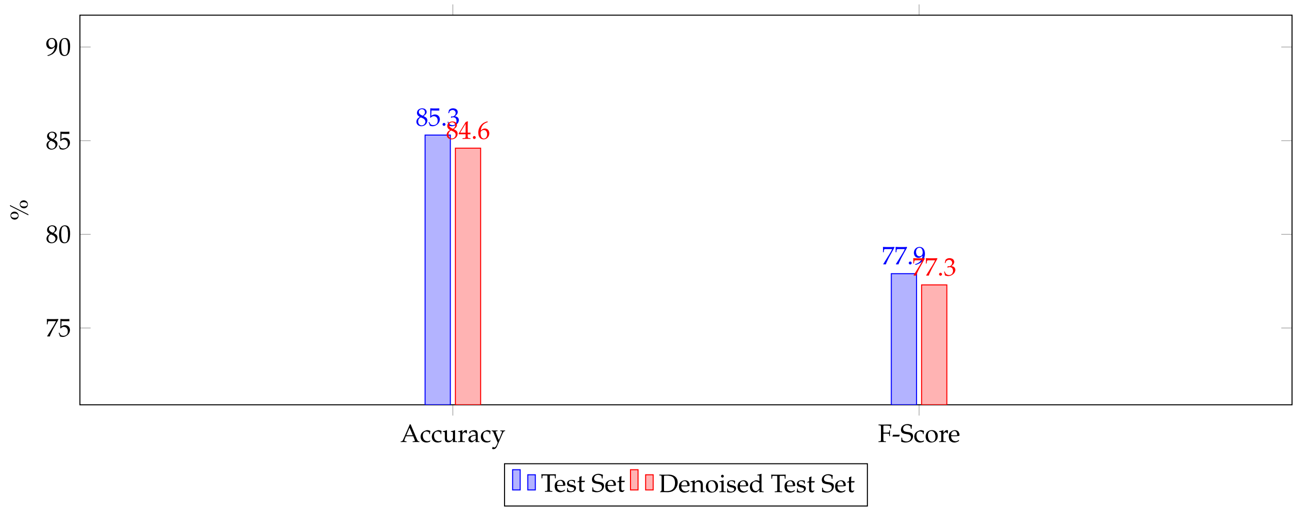


Figure 15. Accuracy and F-1 score obtained over the combined test set and the combined denoised test set.

We see that denoising does not help while training on the entire combined dataset. Both accuracy and F-score decrease on the denoised training set (Figure 14) and the denoised test set (Figure 15). We believe that this is because the amount and type of noise present in EG and SG images are different. Hence, training a network on varying images which have been denoised with the same noise level will not help the neural network learn a uniform set of features. Therefore, there is further scope to denoise the two datasets with separate noise levels which can support feature learning.

Having found that regular, non-denoised images work well while training the combined dataset, we show the accuracy obtained by this network over the entire (training + test) data from each year in Tables 5 and 6, respectively. In these tables, we also show the pixel size in the vertical direction, of each year's images, to make the calculated MAE more meaningful for ice layer tracking by glaciologists.

Table 5. Pixel size in the vertical (depth) direction, as well as the accuracy, F-score, and MAE, obtained over each year of EG.

Year	Vertical Pixel Size (cm)	Accuracy (%)	F-Score (%)	MAE (Pixels)
2009	1.42	87.3	82.2	1.27
2010	1.39	93.5	88.0	0.75
2011	3.19	92.1	86.5	0.54
2012	2.07	90.2	85.1	1.36

Table 6. Pixel size in the vertical (depth) direction, as well as the accuracy, F-score, and MAE, obtained over each year of SG.

Year	Vertical Pixel Size (cm)	Accuracy (%)	F-Score (%)	MAE (Pixels)
2009	1.39	94.8	88.2	5.97
2010	1.37	90.7	85.1	3.26
2011	1.54	87.0	82.1	2.19
2012	1.03	87.7	82.8	6.41
2013	2.11	88.5	83.4	0.98
2014	1.03	89.3	84.2	2.01
2015	1.05	90.2	85.3	7.25
2016	1.05	89.6	85.1	7.28
2017	0.83	89.8	85.3	5.79

7. Conclusions

This work introduces fully convolutional networks to semantically segmenting internal ice layers seen through Snow Radar images captured over the Greenland Ice Sheet. We also introduce a cropping methodology that can overcome any lack of missing or incomplete training labels. We train on over 15,000 images, ranging between the years 2009–2017, and captured over different regions of the ice sheet. Through our methodology, we were able to detect the position and thickness of each individual ice layer accumulated over the ice sheet in a specific year. We got a mean absolute of thickness estimation ranging between 0.54 to 7.28 pixels, depending on the year.

In this manner, a well trained model, which has learnt various aspects of the ice sheets from different time periods, can be used to predict the position, thickness, and, thus, the change of any ice layer captured at a later time period. Such data-driven models can further be improved by embedding physical knowledge in the networks, and, thus, support glaciological studies.

Author Contributions: M.R. designed the experiments, supervised the project, and acquired funding; D.V. carried out pre- and post-processing of the data, performed and analyzed the experiments, and wrote the manuscript; M.Y. gave valuable insights for training the neural networks and processing the images; J.P., O.I., and J.L. helped in understanding the dataset, its characteristics, the snow profile it captures, and the Greenland regions it represents. Everybody engaged in constructive, biweekly discussions to structure the project and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by NSF BIGDATA awards (IIS-1838230, IIS-1838024), IBM, and Amazon. We are grateful to Joseph MacGregor and Brooke Medley for their active feedbacks.

Data Availability Statement: Datasets used in this study are available at the following links respectively: EG—https://data.cresis.ku.edu/data/temp/internal_layers/NASA_OIB_test_files/image_files/greenland_picks_final_2009_2012_reformat/ (accessed on 1 June 2021). SG—https://data.cresis.ku.edu/data/temp/internal_layers/NASA_OIB_test_files/image_files/Corrected_SEGL_picks_lnm_2009_2017_reformat/ (accessed on 1 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rignot, E.; Mouginot, J.; Scheuchl, B.; van den Broeke, M.; van Wessem, M.J.; Morlighem, M. Four decades of Antarctic Ice Sheet mass balance from 1979–2017. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 1095–1103. [CrossRef] [PubMed]
2. IPCC. *IPCC AR5 WG1 Summary for Policymakers*; IPCC: Paris, France, 2014
3. Rahnemoonfar, M.; Yari, M.; Paden, J.; Koenig, L.; Oluwanisola, I. Deep multi-scale learning for automatic tracking of internal layers of ice in radar data. *J. Glaciol.* **2020**, *67*, 39–48. [CrossRef]
4. Gogineni, S.; Yan, J.B.; Gomez, D.; Rodriguez-Morales, F.; Paden, J.; Leuschen, C. Ultra-wideband radars for remote sensing of snow and ice. In Proceedings of the IEEE MTT-S International Microwave and RF Conference, New Delhi, India, 14–16 December 2013; pp. 1–4
5. Koenig, L.S.; Ivanoff, A.; Alexander, P.M.; MacGregor, J.A.; Fettweis, X.; Panzer, B.; Paden, J.D.; Forster, R.R.; Das, I.; McConnell, J.R.; et al. Annual Greenland accumulation rates (2009–2012) from airborne snow radar. *Cryosphere* **2016**, *10*, 1739–1752. [CrossRef]
6. Arnold, E.; Leuschen, C.; Rodriguez-Morales, F.; Li, J.; Paden, J.; Hale, R.; Keshmiri, S. CReSIS airborne radars and platforms for ice and snow sounding. *Ann. Glaciol.* **2019**, 1–10. [CrossRef]
7. MacGregor, J.A.; Fahnestock, M.A.; Catania, G.A.; Paden, J.D.; Prasad Gogineni, S.; Young, S.K.; Rybarski, S.C.; Mabrey, A.N.; Wagman, B.M.; Morlighem, M. Radiostratigraphy and age structure of the Greenland Ice Sheet. *J. Geophys. Res. Earth Surf.* **2015**, *120*, 212–241. [CrossRef] [PubMed]
8. Rahnemoonfar, M.; Johnson, J.; Paden, J. AI Radar Sensor: Creating Radar Depth Sounder Images Based on Generative Adversarial Network. *Sensors* **2019**, *19*, 5479. [CrossRef] [PubMed]
9. Yari, M.; Rahnemoonfar, M.; Paden, J.; Oluwanisola, I.; Koenig, L.; Montgomery, L. Smart Tracking of Internal Layers of Ice in Radar Data via Multi-Scale Learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5462–5468
10. Yari, M.; Rahnemoonfar, M.; Paden, J.; Koenig, L.; Oluwanisola, I. Multi-Scale and Temporal Transfer Learning for Automatic Tracking of Internal Ice Layers. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020.
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
12. Fu, H.; Cheng, J.; Xu, Y.; Wong, D.W.K.; Liu, J.; Cao, X. Joint Optic Disc and Cup Segmentation Based on Multi-Label Deep Network and Polar Transformation. *IEEE Trans. Med. Imaging* **2018**, *37*, 1597–1605. [CrossRef] [PubMed]
13. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
14. Varshney, D.; Gupta, P.K.; Persello, C.; Nikam, B.R. Deep Convolutional Networks for Cloud Detection Using Resourcesat-2 Data. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 9851–9854
15. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223
16. Carrer, L.; Bruzzone, L. Automatic Enhancement and Detection of Layering in Radar Sounder Data Based on a Local Scale Hidden Markov Model and the Viterbi Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 962–977. [CrossRef]
17. Ferro, A.; Bruzzone, L. Automatic Extraction and Analysis of Ice Layering in Radar Sounder Data. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 1622–1634. [CrossRef]
18. Ilisei, A.; Bruzzone, L. A System for the Automatic Classification of Ice Sheet Subsurface Targets in Radar Sounder Data. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3260–3277. [CrossRef]
19. Rahnemoonfar, M.; Fox, G.C.; Yari, M.; Paden, J. Automatic Ice Surface and Bottom Boundaries Estimation in Radar Imagery Based on Level-Set Approach. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5115–5122. [CrossRef]
20. Rahnemoonfar, M.; Habashi, A.A.; Paden, J.; Fox, G.C. Automatic Ice thickness estimation in radar imagery based on charged particles concept. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 3743–3746.
21. Xu, M.; Crandall, D.J.; Fox, G.C.; Paden, J.D. Automatic estimation of ice bottom surfaces from radar imagery. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 340–344.
22. de Paul Onana, V.; Koenig, L.S.; Ruth, J.; Studinger, M.; Harbeck, J.P. A Semiautomated Multilayer Picking Algorithm for Ice-Sheet Radar Echograms Applied to Ground-Based Near-Surface Data. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 51–69. [CrossRef]
23. Rahnemoonfar, M.; Yari, M.; Paden, J. Radar Sensor Simulation with Generative Adversarial Network. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020.
24. Oluwanisola, I.; Paden, J.; Rahnemoonfar, M.; Crandall, D.; Yari, M. Snow Radar Layer Tracking Using Iterative Neural Network Approach. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020.

25. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Computer Vision–ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 833–851.
26. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
27. CReSIS. NASA OIB Greenland 2009–2012. Available online: https://data.cresis.ku.edu/data/temp/internal_layers/NASA_OIB_test_files/image_files/greenland_picks_final_2009_2012_reformat/ (accessed on 1 June 2021).
28. CReSIS. NASA OIB Greenland 2009–2017. Available online: https://data.cresis.ku.edu/data/temp/internal_layers/NASA_OIB_test_files/image_files/Corrected_SEGL_picks_lnm_2009_2017_reformat/ (accessed on 1 June 2021)
29. Montgomery, L.; Koenig, L.; Lenaerts, J.T.M.; Kuipers Munneke, P. Accumulation rates (2009–2017) in Southeast Greenland derived from airborne snow radar and comparison with regional climate models. *Ann. Glaciol.* **2020**, *61*, 225–233. [CrossRef]
30. Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Van Gool, L.; Timofte, R. Plug-and-play image restoration with deep denoiser prior. *arXiv* **2020**, arXiv:2008.13751.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Guan, S.; Khan, A.A.; Sikdar, S.; Chitnis, P.V. Fully Dense UNet for 2-D Sparse Photoacoustic Tomography Artifact Removal. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 568–576. [CrossRef] [PubMed]
33. Weng, Y.; Zhou, T.; Li, Y.; Qiu, X. NAS-Unet: Neural Architecture Search for Medical Image Segmentation. *IEEE Access* **2019**, *7*, 44247–44257. [CrossRef]
34. Jiao, L.; Huo, L.; Hu, C.; Tang, P. Refined UNet: UNet-Based Refinement Network for Cloud and Shadow Precise Segmentation. *Remote Sens.* **2020**, *12*, 2001. [CrossRef]
35. Cui, B.; Chen, X.; Lu, Y. Semantic Segmentation of Remote Sensing Images Using Transfer Learning and Deep Convolutional Neural Network with Dense Connection. *IEEE Access* **2020**, *8*, 116744–116755. [CrossRef]
36. Baheti, B.; Innani, S.; Gajre, S.; Talbar, S. Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 14–19 June 2020.
37. Lee, C.Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-Supervised Nets. In *Proceedings of Machine Learning Research, Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 9–12 May 2015*; Lebanon, G., Vishwanathan, S.V.N., Eds.; PMLR: San Diego, CA, USA, 2015; Volume 38, pp. 562–570.
38. Smith, L.N.; Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*; Pham, T., Ed.; International Society for Optics and Photonics (SPIE): Baltimore, MD, USA, 2019; Volume 11006, pp. 369–386.