Segmentation of Windthrow in High Resolution Capella SAR Images

Abie Marshall

Master of Science
Data Science
School of Informatics
University of Edinburgh
2022

Abstract

Windthrow occurs when strong winds uproot trees and is a common disturbance to European forests. Windthrow needs to be assessed quickly to mitigate the socio-economic impact however current solutions are lacking. Optical satellites require cloudless skies and airborne assessments rely on suitable flying conditions; both are unlikely after a major storm. Synthetic Aperture Radar (SAR) satellites alleviate this owing to their ability to see through clouds, day and night. However, current SAR windthrow detection algorithms rely on temporal change detection and are not timely. Capella currently offer high resolution SAR imagery of any scene on Earth with an average delivery time of 6 hours. This research is the first to apply Convolutional Neural Networks to Capella imagery, the first to automatically segment and identify windthrow in single SAR images and also presents a novel method for removing the "no data" values from satellite images which could be applied to other forms of satellite imagery. It was demonstrated that a U-Net like architecture is able to segment windthrow with competitive results to the temporal methodologies, which combined with Capella's quick data delivery offers a solution to rapidly mapping where windthrow has occured. This would allow forest managers to promptly identify and recover windthrown timber as well as quickly clearly blocked transportation links and mapping areas of potential fire risk. It was observed that performance was significantly better in images where the range dimension was perpendicular to the timber fell direction, a phenomena not mentioned in current literature.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Abie Marshall*)

Acknowledgements

I would like to thank the following people for helping with this research project. My partner *Sarah* for being incredibly supportive throughout my Masters, which includes agreeing to move to Edinburgh with me in the first place, being there through the challenging times and clambering over windthrown trees with me. *James Garforth* for offering invaluable guidance throughout this project and listening to my ramblings about forests and synthetic aperture radar. *Iain Woodhouse* and *Steve Hancock* for letting me take their course (Active Remote Sensing: Radar and Lidar) which will likely have more influence on my career than any other university course I have completed. I would like to further thank Iain for introducing me to the topic of this dissertation, getting me access to the Capella data and agreeing to co-supervise the project.

Table of Contents

Chapter 1

Introduction

Windthrow, demonstrated by Figure 1.1, occurs when strong winds uproot or snap trees and is the most prominent cause of disturbances to European forests, responsible for around 51% of all recorded damage. Over 3 million people are employed in Europe's forest sector which is estimated to be worth €120 billion and serves a wide range of societal services ranging from wood production, recreation and carbon sequestration. Windthrow therefore has far reaching socio-economic consequences and the ultimate ramification is largely dependent on how quickly the extent can be mapped. Salvage operations need to be established in order to avoid the secondary disaster of not being able to make use of the windthrown timber [1], the total extent determines whether standard clear felling operations need to be ceased in order to manage supply and demand and not negatively affect timber prices. Transportation links are often blocked and need to be identified and cleared as well as any potential damage to power lines [2]. Cataloging where windthrow occured is of interest to forest managers and insurers as future windthrow is likely to occur where it has previously occured [3]. Addtionally, cleaning up windthrow promptly is important because of complex environmental issues related to carbon sequestration [4], fire risk [5] and bark beetle outbreaks [6]. Compounding these factors is that extreme storm events are becoming more frequent across Europe [7].

Despite the importance, current approaches for mapping windthrow are lacking. Assessments are typically acquired via field surveys or interpreting optical imagery from airborne/satellite platforms. Field surveys are slow, expensive and only provide limited coverage. Flying airborne platforms is expensive and requires suitable weather conditions whilst passive optical satellites require cloudless skies; conditions often not present after a storm. These approaches are further limited in that imagery can only

Figure 1.1: Image of windthrow observed at Monynut obtained during fieldwork undertaken on 19th June 2022. Capella Imagery of this site was obtained in December 2021.

be obtained in daylight which frequently results in an indefinitely long period until an assessment can be made [8, 9]. Synthetic Aperture Radar (SAR) satellites alleviate these issues owing to their ability to capture images in all weather conditions, day and night. However, access to SAR imagery has historically been prohibitively expensive or required lengthy research proposals to be approved and the revisit times (time to re-image an area of Earth) have been on the orders of days [10]. ESA's Sentinel-1 mission is unique in this regard, providing openly accessible imagery since 2014 [11], unfortunately the resolution of Sentinel-1 imagery is not sufficient to unambiguously detect windthrow in a single images. Current approaches utilising Sentinel-1 instead rely on temporal change detection and are at best able to map windthrow in a time frame on the order of weeks; by which time clear optical imagery may be available [9].

Capella Space [12] and ICEye [13] are changing the paradigm of SAR products. Both companies operate a constellation of SAR micro-satellites and offer sub-meter resolution SAR imagery with hourly revisit times as a commercial enterprise. For the first time high resolution SAR imagery of any site on Earth is available essentially on demand for anyone willing to pay for it. In the wake of Storm Arwen in which an estimated 20% of Scottish annual timber yield was windthrown in a single evening [14], teams at Earth Blox [15] and the University of Edinburgh acquired 5 Capella images of sites affected by windthrow. Windthrow is easily identifiable as a distinct texture in these images where the forest appears combed over, however due to the large spatial size of satellite imagery manually identifying areas is still an arduous task. This work hypothesised that this distinct texture would instead be a strong enough feature to train a Convolutional Neural Network (CNN) to perform pixel-level predictions (*semantically segment*) in order to classify windthrow pixels, which combined with

Capella's quick data delivery offers a potential solution for rapid mapping of windthrow. CNNs continue to exhibit state-of-the-art segmentation performance within fields such as medical imaging [16] and are gaining growing interest in application to optical satellite imagery [17]. Despite this, CNN application to SAR imagery has lagged behind; likely a consequence of the sparsity of SAR datasets suitable for training segmentation models.

This research presents one of the few attempts to apply CNNs to SAR imagery and is the only example of applying CNNs to Capella imagery in open publication. This work contributes a labelled sub-meter resolution Capella SAR dataset and due to the non-existence of literature working with Capella products, also presented is a processing pipeline for manipulating the products into a format suitable for the training of CNNs. The processing pipeline includes a novel approach for dealing with the border of "no data" values which could be generalised to all forms of satellite data. This is the only known attempt to segment windthrow in individual SAR images and the inadequacy of classical machine learning approaches (Gaussian Naive Bayes; Logistic Regression) for segmentation of Capella imagery is highlighted, despite their popularity in the optical domain [18, 19]. Subsequently, it is demonstrated that a U-Net [20] inspired CNN architecture is able to mask Capella images in less than a minute and achieve equivalent performance to the sole example of utilising CNNs for segmenting windthrow in high resolution optical imagery [21]. These results are also competitive with established SAR temporal change detection approaches [9, 22], therefore proving that the proposed utilisation of Capella products presents the opportunity to map windthrow in a time frame potentially less than a day instead of weeks.

1.1 Dissertation Structure

This dissertation is organized as follows: Chapter 2 introduces Synthetic Aperture Radar, Convolutional Neural Networks, a review of previous approaches to segmenting windthrow along with general considerations when working with satellite data and how these are relevant to SAR. Chapter 3 describes the Capella images used in this research, the labelling procedure to create training data and the processing steps undertaken to manipulate the data into format suitable for training convolutional neural networks. Chapter 4 describes the evaluation metrics and methodology conducted for segmenting windthrow. Chapter 5 presents and discusses the experimental results. Chapter 6 concludes the dissertation and suggests future work.

Chapter 2

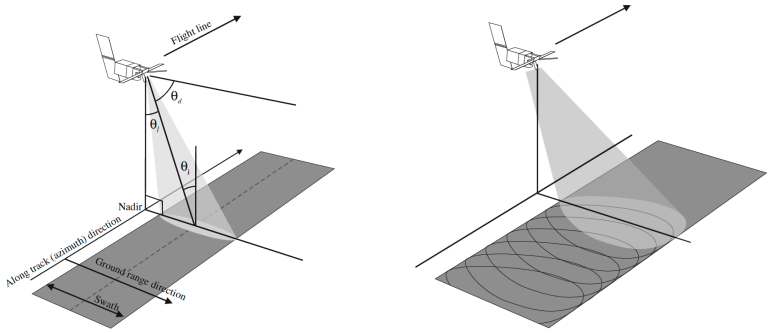
Background and Related Work

2.1 SAR Primer

Many researchers are familiar with *passive* optical remote sensors such as Landsat and Sentinel-2 which measure reflected radiation in the visible, near-infrared, and short-wave infrared portions of the electromagnetic spectrum. This is likely a consequence of the open data policy that these instruments have pioneered [23], but also because the interpretation of the imagery is not dissimilar to interpreting a normal photograph [24]. Synthetic Aperture Radar (SAR) is an alternative and less familiar *active* remote sensing technique in which pulses of microwave radiation are actively emitted.

Before describing SAR it is useful to consider Real Aperture Radar (RAR), the side looking nature of which is demonstrated by Figure 2.1a. The sensor emits a pulse of electromagnetic microwave radiation, it bounces off the surface (known as *backscatter*) and returns an *echo* to the sensor. Knowing the altitude of the sensor, together with the speed of light, this configuration allows the distances along the ground from which the echo returned to be accurately calculated by measuring the time delay. By further measuring the intensity of the returned echo (which is largely determined by the surface characteristics of the point on the ground from which the echo bounced) detailed 2D images of the ground can be constructed as the sensor moves in the direction along the flight path. RAR actively generates its own microwave illumination and perhaps the most useful property of this is that microwaves do not interact with the particles present in common atmospheric obstructions such as clouds, smoke, smog and sandstorms. This allows RAR systems to capture day and night, in all weather conditions.

Radar images have two dimensions known as range and azimuth in which the resolutions can and often do differ because they are governed by different sensor



(a) Real Aperture Radar (b) Synthetic Aperture Radar

Figure 2.1: (a) Geometry of RAR and associated terminology. As the instrument moves in the direction along the flight path detailed images of the ground can be built up [25]. (b) Geometry of SAR. The beam is designed to be wide in the azimuth direction to make adjacent footprints overlap the same points on the ground. These overlapping footprints can be combined to synthesise the characteristics of a larger effective antenna [25].

characteristics. Azimuth resolution is often the limiting factor in RAR as it is determined by the size of the instruments antenna and often requires impractically large antennas to achieve useful resolution. SAR alleviates this issue by synthesising a large effective antenna, the geometry of which is illustrated by Figure 2.1b.

In SAR the beam is designed to be wide in the azimuth dimension which allows the beam to overlap as the sensor moves along the flight path. The effective length of the synthesised antenna (and resulting azimuth resolution) is governed by how long the target dwells in the beam. SAR *spotlight mode* utilises this by steering the beam as the instrument flies past to cover the same point on the ground, see Figure 2.2. This results in much longer dwell times and finer azimuth resolution at the cost of image size.

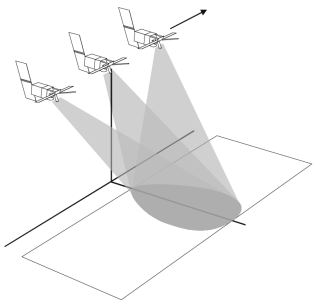


Figure 2.2: Spotlight mode. The beam is steered as the satellite flies past enabling longer dwell times and finer azimuth resolution of the scene. [25].

The side looking nature of SAR causes objects to geometrically distort. *Foreshort-*

ening causes tall objects, such as a mountains, to appear steeper on one side with a thin bright edge at the peak. Extreme foreshortening is called *layover* when the radar waves hit the top of an object before the base, hence the echos return from top return first and the object appears folded over on itself. *Shadowing* is similar to optical shadows, an object blocks the path of the radar signal and causes dark areas in the image where there is no returned signal [25]. Radar images contain artifacts which look like salt and pepper noise, known as *speckle*. Speckle is often described as noise however this is not correct as it is a repeatable phenomena resulting from interference between coherent echoes. Two images taken of the same scene with identical imaging geometry will produce the same speckle pattern. One way to reduce speckle at the cost of spatial resolution is by *multi-looking* where the formed synthetic antenna is split into multiple sub-antennas and each is used to generate an image of the scene. A multi-looked image with reduced speckle can be generated by taking an *incoherent* average of these images.

Radar images appear similar to gray scale images however the intensity of a pixel in a radar image is not indicative of color and instead represents the intensity of backscattered radiation, known as "radar brightness" or "Beta Nought", β^0. Beta Nought is influenced by properties of the radar system such as incidence angle, the surface properties of the scene such as roughness and dielectric constant [25] and is further influenced by the local topography. "Sigma Nought", σ^0, is a calibration which aims to make the image radiometry comparable between images regardless of incidence angle and often incorporates topographical information to provide more consistent results. Sigma nought calibration aims to make the variation in brightness between pixels across different images being solely down to properties of the ground surface instead of instrument or geometric specifics [25].

Electromagnetic waves have a property known as *polarisation*, which is defined to be the direction in which the electric field oscillates as the wave propagates through space. Waves are described in terms of perpendicular polarization planes known as *horizontal* and *vertical* which are defined by convention with reference to the Earth's surface. SAR systems can operate in 4 modes: HH, VV, HV, VH. The first letter denotes the transmit polarisation and the second denotes the receive polarisation. The polarisation mode influences measured pixel value as different modes are sensitive to different types of backscatter as demonstrated by Figure 2.3. HH indicates the presence of double bounces, caused by reflective surfaces at right angles to each other redirecting the incoming energy back to the sensor. To a lesser extent strong HH indicates rough surface scattering but this tends to be more sensitive to VV. Strong HV/VH comes from

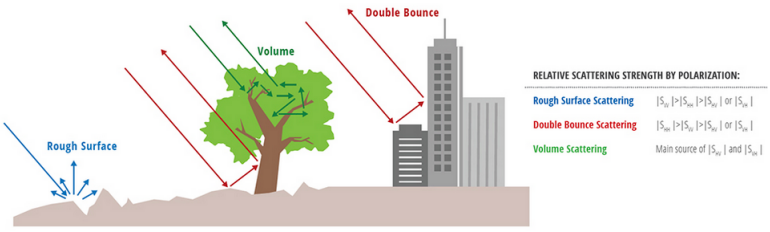


Figure 2.3: Sensitivity of types of backscatter to the polarisation mode. [24].

volume scattering, where the transmitted signal bounces several times within a medium before being redirected; resulting in a change of polarization. Multi-channel SAR images are formed by stacking the results from different polarisation modes together. This leads to the definitions of SinglePol (HH or VV channels only), DualPol (HH and VV) and QuadPol (HH, VV, HV and VV) imagery. SAR systems can operate at several wavelengths (X;C;L;P band) that interact uniquely with different media, therefore SAR images acquired of the same scene at different wavelengths may appear drastically different. As an example, Figure 2.4 demonstrates how different wavelengths penetrate into forest canopies; shorter wavelengths mostly interact (and therefore only image) the top of the canopy whilst longer wavelengths penetrate further and can image textures beneath the canopy.

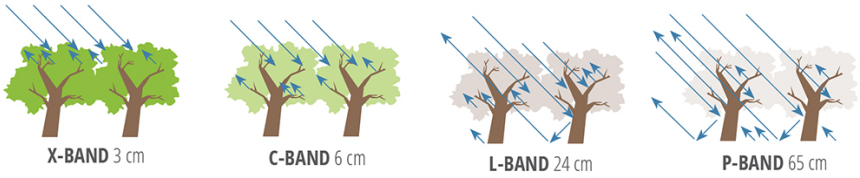


Figure 2.4: Wavelengths interactions with tree canopies. The listed length corresponds to the wavelength of the associated radar band [24].

2.2 The Capella Constellation

Capella operates a constellation of micro-satellites capable of producing SinglePol images in either HH/VV with revisit times averaging less than 4 hours. Users request imagery through a self serve console where they specify an imaging mode and tasking tier (time to get data) from 1;3;7 days, tasking requests are however usually accepted within minutes and on average data is delivered less than 6 hours after collection [10]. Images can be be acquired in spotlight (SP), sliding spotlight (SS) and stripmap (SM) and currently cost around $5000 an image, however exact pricing strategies are still being established. Stripmap mode has the largest scene size ($5km \times 20km$) but the

coarsest azimuth resolution. In spotlight mode, Capella's agile design allows the antenna beam to dwell on a scene for tens of seconds which allows sub-meter resolution imagery to be acquired at the cost of scene size ($5km \times 5km$). Sliding spotlight is a compromise between these two modes in terms of resolution and scene size ($5km \times 10km$) [26]. Capella's tasking additionally enables customers to specify many undiscussed (beyond the scope of this report) SAR acquisition parameters which influence the resulting image, allowing tailored imagery to be collected for specific tasks. Images are available in several data formats including Single Look Complex (SLC), Geocoded Ellipsoid Corrected (GEC) and Geocoded Terrain Corrected (GEO). SLC images achieve the highest spatial resolution however, for many applications SLC images are not useful because of the high level of speckle. Capella's GEC/GEO formats are multi-looked images with reduced speckle and only differ in how topographical information is incorporated into the calibration [26]. Speckle reduction between between SLC and GEC/GEO products is illustrated by Figure 2.5.

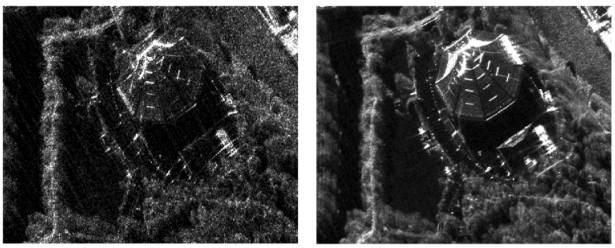
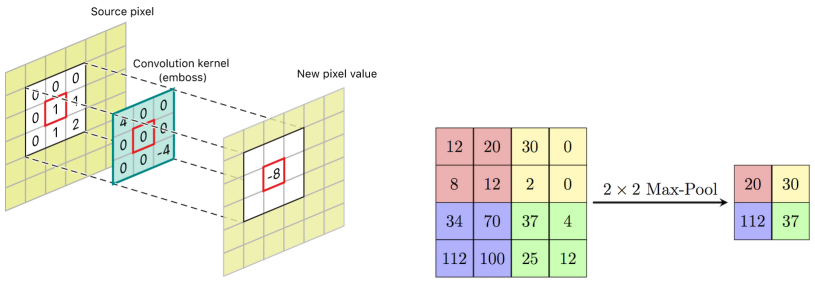


Figure 2.5: (left) SLC Capella image and (right) Capella multi-looked image of the Nippon Budokan, Tokyo, Japan. Increase in clarity can be seen in the multi-looked image and texture differences between grassy areas can now be distinguished [12].

2.3 Convolutional Neural Networks

In image processing a *kernel* is a matrix which can be applied to an image via a *convolution* to output a new image. *Kernel* is often used interchangeably with *filter* and in the deep learning community it is common to see the term *feature detector* with the resulting image referred to as a *feature map*. The convolutional operation is illustrated by Figure 2.6a; the kernel moves across the source image and at each position a new pixel value is calculated by summing the centre pixel with its local neighbours, weighted by the kernel values. Historically, kernel weights have been manually defined to extract features from images such as edge detection [27], for image processing tasks like blurring [28] and even within the SAR community for speckle reduction [29]. The resulting size of the convolved image can be controlled by specifying the



(a) Convolution Operation (b) Max Pooling Operation

Figure 2.6: (a) Example of a convolutional operation between a source image and kernel to result in a new pixel value within the output image [31]. (b) Example of a 2×2 max pooling operation with stride 2 [32].

stride and *padding* and one such use for this is to increase the spatial size of an image with up-sampling convolutions. Several up-sampling variants exist such as *transposed convolutions* (*deconvolutions*) and *bilinear convolutions*, both are described by [30].

The concept behind *Convolutional Neural Networks* (CNNs) is that the weights of the kernels that should be applied to extract features from the images are learnt instead of being manually specified. In contrast to other machine learning techniques, CNNs therefore account for spatial information instead of individual pixel values in isolation. CNNs initially showed potential for image classification when the AlexNet CNN won the ImageNet challenge in 2012 [33] and continue to exhibit state-of-the-art image classification performance and have since extended to image segmentation [34, 16].

CNNs architectures are typically built by stacking sequential convolutional layers combined with *non-linear* activation functions [35], *max pooling* [36], *batch normalisation* [37], *skip connections* [38] and *dropout* [39]. Introducing *non-linearity* is required otherwise the network effectively collapses into a single convolutional layer [35]. *Max pooling*, illustrated by Figure 2.6b, selects stronger invariant features, which leads to better generalisation and faster convergence [36]. The entirety of the training data for CNNs often does not fit into memory which leads to them being trained on *batches*. *Batch normalization* calculates the mean and standard deviation for the current batch at various points in the network and standardises the data such that the batch has 0 mean and unit variance. Batch normalisation enables faster and more stable training of deep neural networks, exactly why is still an open debate [40]. Skip connections pass information from one layer of the network to another without it passing through the intermediate layers; they have been shown to be essential in segmentation tasks at recovering spatial information lost during max pool downsampling [41]. Models are

described to have *overfitted* the training data when they do not generalize well to unseen data [42, 43]. Techniques to mitigate overfitting are often referred to as *regularization* and one such technique for CNNs is Dropout. Dropout randomly zeroes some of the outputs of the previous layer with some probability. This effectively forces the data to take different paths through the network and is a proven regularization technique [39].

2.3.1 U-Net architecture

U-Net, illustrated by Figure 2.7, was originally developed in 2015 for segmenting single channel medical images [20] and it remains a popular architecture for medical image segmentation [16] but has also shown potential in other fields, for example cloud masking in the remote sensing community [44]. Despite there being other segmentation architectures such as FCN [45] and SegNet [46], U-Net was chosen for the task of segmenting SinglePol Capella imagery because of the capacity of U-Net to learn from limited data [20, 16] and also because of its successes with medical images, which are often single channel intensity based images much like SAR.

The network consists of a contracting path with repeated applications of double 3x3 valid padded convolutions followed by ReLUs (DoubleConv+ReLU) before downsampling the spatial size by half with a 2x2 max pooling operation. The initial number of feature channels is set to 64 and this doubles after each downsampling. Following the contracting path is an expansive path with repeated application of 2x2 transposed convolution up-sampling that doubles the spatial size whilst halving the number of feature channels. The resulting feature map is concatenated with the centre crop of the feature map from the corresponding layer of the contracting path via a skip connection. Cropping is necessary due to spatial size reductions caused by repeated application of valid padded convolutions. The concatenated feature map passes through a Double-

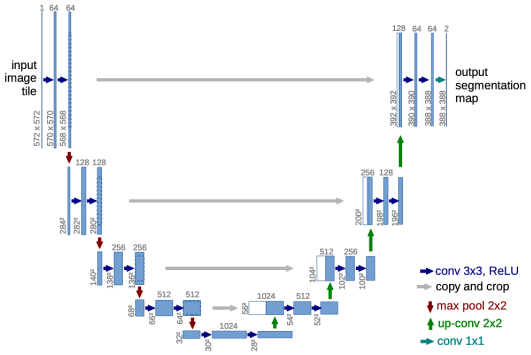


Figure 2.7: U-Net architecture as it was originally proposed [20].

Conv+ReLU and the output is passed to the next up-sampling layer. The final layer is passed through a 1x1 convolution that maps output to the desired number of segmentation channels. U-Net originally used 2 segmentation channels and probability masks of the inputs were generated using a softmax function [47]. In U-Nets source paper the images were tiled into 512x512 patches and the network outputted 388x388 predictions. To enable seamless segmentation of the original images an overlapping tiling strategy was incorporated that used *mirroring* to deal with the missing data brought about from the down-sampled output [20].

2.3.2 Optimization & Loss functions

Neural networks are trained by minimising a differentiable loss function with respect to the networks parameters. This loss function is used to calculate the gradient for each of the networks parameters via backpropagation [48] and then a gradient descent algorithm (*optimizer*) is employed to change the parameter values by a small step (*learning rate*) in the direction negative to their gradient such as to optimise the loss function. Many optimizers exist each with their own strengths and weaknesses [49], however the choice of loss function is arguably more important as this plays an essential role in determining the networks performance - optimising a poor choice of loss function will yield poor results regardless of the choice of optimizer [50]. For segmentation it is not possible to declare a universal loss function that will perform well for all tasks. Which loss function will perform best for a given task depends on properties of the dataset and is usually determined experimentally. Some general guidance is that focal loss functions work best for highly in-balanced datasets [51], whereas binary cross-entropy loss works well for balanced datasets [52]. Dice coefficient based loss functions can work well regardless of class imbalance [50, 53].

2.4 SAR Semantic Segmentation

Noted in the project proposal for this work [54]: publications utilising Capella products are sparse. Therefore, the literature was surveyed for how other SAR satellites had been used for mapping windthrow. To summarize this previous review, the potential of a single high resolution HH polarised X-Band images for detecting windthrow has been acknowledged. However, no attempts have been made to automatically segment windthrow in single images. Attempts instead rely on temporal change detection and

the most established method utilises Sentinel-1 (C-band) and is at best able to segment windthrow in ≈ 4 weeks and averages a *dice coefficient* [55] (Section 4.1) of ≈ 0.7 [54].

The vast majority of the literature regarding semantic segmentation in the remote sensing domain deals with passive optical imagery [17]. A large contributing factor to this is the near non existence of SAR datasets suitable for training segmentation models; the MSAW (Multi-Sensor All Weather Mapping) dataset collected over Rotterdam consists of airbourne quad-polarity X-band SAR imagery designed to be similar to Capella data and accompanying building masks is the only known example in the same sub-meter resolution category [56]. Despite the lack of literature, transferable considerations can be inferred from segmentation in the optical domain. Classical machine learning techniques achieve comparable performance to CNNs in multispectral optical remote sensing and this can be attributed to the large channel size (Landsat-8, 11 channels; Sentinel-2, 12 channels) increasing the likelihood that the desired segmentation class is separable in at least one of the channels [57], without the need to account for spatial information [18, 19]. Conversely, SAR images can at most have 4 channels (QuadPol) however, more common is 2 channels (DualPol) and sub-meter resolution imagery tends to be 1 channel (SinglePol) [58, 26]. The effect of the number of channels for forest segmentation in SAR imagery was investigated by [59]. Pixels were classified based on their channel values and significantly lower performance was observed when considering a single channel compared to when other channels are introduced. The poor performance for single channel segmentation is to be expected as many factors can cause pixels to have similar values in SAR imagery, see Section 2.1. As the Capella images are single channel, it is reasonable to expect classical machine learning approaches will exhibit poor segmentation performance.

Architectures similar to U-Net have recently been applied to SAR images and offers a potential segmentation solution. A TernausNet model was used to segment buildings in sub-meter resolution SAR imagery and achieved a dice coefficient of 0.21 [56, 60]. The most comparable study to this research was the segmentation of roads in sub-meter resolution SinglePol TerraSAR-X HH images using U-Net [61]. Despite the thin nature roads making them challenging to segment and the fact that road pixels represented the minority of the pixels in the images (around 5% overall) they achieved a respectable dice coefficient of 0.63. Although not SAR imagery, it is worth noting the only example of CNN segmentation of windthrow in optical satellite imagery; achieving a dice coefficient also of 0.63 [21].

The large spatial size of satellite imagery combined with memory constraints mean

that the images need to be tiled down into smaller patches when training CNNs. Seamless segmentation masks are created by stitching together the predictions for the individual tiles. The chosen patch size has significant influence on the models performance. It is beneficial to use large patches to capture spatial patterns, however as patch size increases the size of the training batches decrease. This influences batch normalization layers resulting in a balance between large patches to account for spatial patterns and large batches which improve regularization and training speed [62]. It was found that several tiling and stitching mechanisms have been utilised in the optical remote sensing literature. Sometimes the overlapping/mirroring strategy originally proposed with U-Net [20, 62] was mimicked, however it was most common to use non-overlapping patches of various sizes with the most frequent size being 256×256 pixel patches [44]. Regardless of the tiling strategy used, there is often little justification and little experimental evidence of the impact of the particular approach [63].

A feature common to optical and SAR satellite imagery is that the image is angled and surrounded by a border of "no data" values, see Figure 3.1. This is a consequence of the projections that are applied to the images to geocode them into a coordinate system such that the images are in the correct location on the Earths surface [62]. Several different methods for dealing with the "no data" values have been observed. One approach is to take the average pixel values of the data pixels for each image and in-paint the "no data" values with these averages [62]. Another approach incorporates the "no-data" values as their own class and performs multi-class segmentation predicting the "no data" values simultaneously with areas of interest [64]. The satellite images are sometimes cropped such that the resulting crop only contained actual data pixels [21]. However, most common in the literature is to not detail how the "no data" values were handled so it is assumed that they were not [44, 65].

Once Sigma Nought Calibrated, SAR pixels have a range on a logarithmic scale roughly between $-70 \rightarrow 20$ and are visually similar to grey scale images. *Contrast stretching* (*"Normalization"*) is a pre-processing technique for grey scale images that stretches the range of intensity values that the image contains to span a desired range of values, usually the full range of pixel values that image type concerned allows. This increases contrast between an image's relative highs and lows and enhances subtle differences [66]. Contrast stretching has been shown to improve performance of CNNs trained on grey scale images in general [67]. Further, contrast stretching is a useful mechanism for transforming pixel value ranges of single channel imagery such that they lie between 0 - 255 which is standard for many CNN architectures.

Chapter 3

Data

Capella delivers data for all product modes and product types in a 3-file bundle known as the Capella TIFF+JSON format. The Capella TIFF+JSON bundle includes one GeoTIFF image along with two JSON metadata files (STAC; Extended). The TIFF file contains an angled raster image surrounded by a black border of "no data" values, see Figure 3.1. The STAC file contains standard geospatial metadata [68]. The extended json file contains Capella specific metadata.

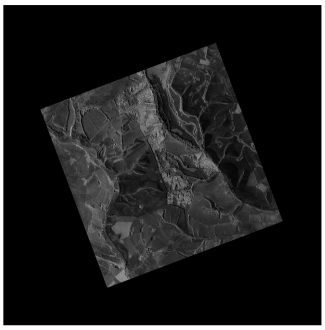


Figure 3.1: Capella GEC spotlight image of the Monynut forest area acquired on 17/12/21, see Table 3.1 for more details.

Capella's multi-looked GEC product type was the chosen format for this analysis because GEC images provide unmatched image clarity and spatial resolution with low amounts of speckle. GEC images are therefore ideally suited for visual literal image interpretation as is the case with detecting windthrow [69]. The 5 images used for this research are of various scenes across Scotland and were acquired in the wake of Storm Arwen [70]. The images can be downloaded through the Capella console [71] with appropriate permissions; the properties of these images are detailed in Table 3.1.

Site	Date	Mode	# of Looks	Polarisation	Range Resolution	Azimuth Resolution	Pixel Size
Monynut	17/12/21	SP	9	HH	0.73 m	0.64 m	24152 x 24342
Monynut	11/01/22	SP	9	VV	0.73 m	0.64 m	23918 x 24148
Glenisla	17/12/21	SP	9	HH	0.73 m	0.64 m	21519 x 21263
Glenisla	15/12/21	SS	4	HH	0.88 m	1.07 m	14343 x 21906
Fetterangus	15/12/21	SS	4	HH	0.82 m	1.01 m	15790 x 21915

Table 3.1: Properties of the 5 Capella GEC images used for this research.

3.1 Labelling

To enable the training of machine/deep learning models binary masks the same spatial size as the original images were generated where a pixel value of 1 corresponded to "windthrow" and a pixel value of 0 corresponded to "not windthrow". ESA's SNAP toolbox [72] was chosen for this task due to its confirmed compatibility with the Capella TIFF+JSON format [73]. Each Capella image was first imported into SNAP and then converted to BEAM-DIMAP format [74] to enable the creation and persistence of vector containers for drawing polygons around windthrow as well as the generation of the new band (channel) required for the binary mask [75]. It was observed that SNAP applies an incorrect Sigma Nought calibration when loading Capella images, this did not affect this research as SNAP was only used for generating windthrow masks however the issue was raised with the SNAP developers [76].

Some areas of windthrow in the Capella images are obvious however other areas were more ambiguous as forest stands can appear different in SAR images due to microwave interaction. This can be attributed to variations in structural properties between stands such as size, orientation and spatial patterns of trees themselves, but also of their branches and leaves [77, 78, 79]. Optical imagery is a useful aid in correctly interpreting SAR images as it allows one to compare what is actually happening on the ground to how different surface textures present themselves in SAR imagery. To help resolve the ambiguity and ensure generated masks were as correct as possible the geographical extent of each Capella TIFF file was imported into Google Earth Pro [80] and historical high resolution MAXAR [81] images of each scene from 2021 before Storm Arwen occurred were exported to aid in the labelling.

Figure 3.2 demonstrates a patch of obvious windthrow; the forest appears combed over and individual tree trunks are visible, compared with the MAXAR imagery where the forest is standing. Figure 3.3 demonstrates the different appearances of various types of forest stands, dense stands appear sunken compared to fielded areas around

them whereas less dense stands look somewhat combed in appearance and could be confused with windthrow without the optical aid. A less obvious patch of windthrow is demonstrated by Figure 3.4; the SAR image reveals that there appears to be a discontinuity in the forest which could be could be caused by structural variations in the forest stand, however the optical image reveals this is not the case and therefore this must be windthrow.

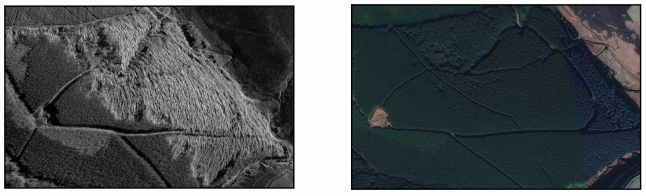


Figure 3.2: (left) Obvious windthrow in the Monynut HH image (right) MAXAR image of the same scene.

Figure 3.3: (left) Various ground surfaces and differing forest stands in the Monynut HH image (right) MAXAR image of the same scene.

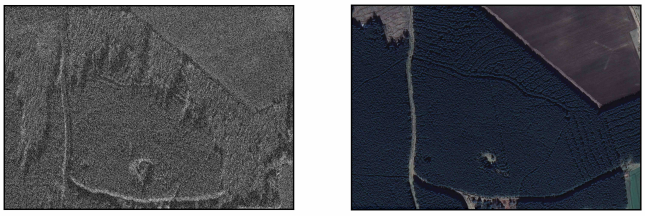


Figure 3.4: (left) Less obvious windthrow in the Fetterangus image (right) MAXAR image of the same scene. The Fetterangus image had less clarity and higher levels of speckle that other images.

Visually, windthrow was much more identifiable in the HH images which is consistent with the literature [82]. Therefore, only the HH images were utilised for this research. During this analysis it was noted that windthrow appears brighter overall,

with the individual trunks visible only when the range dimension (incoming energy direction) is perpendicular to the direction of the felled timber. This is the case for Monynut SP; Glenisla SS, roughly the case for Glenisla SP however for Fetterangus SS the range dimension was parallel to the windthrow. This is a phenomena not mentioned in the literature and the exact reason for it is not known. It is speculated that when the trunks are perpendicular to the incoming energy they are blocked less by leaves and present more opportunities for double bounces to occur. Conversely, if the trunks are orientated parallel to the incoming energy they may be masked by layover and shadow from standing timber and also covered more by their own canopies given the shallow penetrating capabilities of X-band, see Figure 2.4. These combined make windthrow more ambiguous under literal interpretation in this configuration (J Brown [Capella Space] & I Woodhouse , 2022, personal communication, 28 June).

Polygons were manually drawn around identifiable areas of windthrow which were collated into a mask and a virtual binary band the same spatial size as the original input was created using SNAP's "band maths" [83] tool where pixels contained within this mask were set to 1 and all other pixels were set to 0. The binary band for each image was exported as a GeoTIFF to serve as labels for all experiments.

3.2 Processing

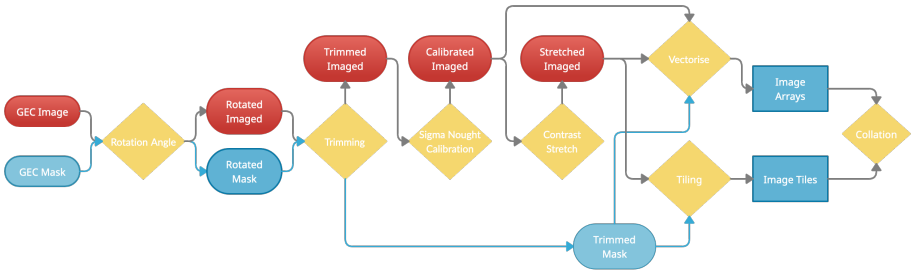


Figure 3.5: The processing procedure that each individual image and associated binary mask underwent. The end result were arrays that could be used for classical machine learning algorithms and image tiles that could be used for training convolutional networks. Processing steps are shown are diamonds, inputs and intermediate outputs are shown as oblongs, outputs are shown as rectangles.

The goal of the processing procedure was to calibrate the images and tile them into smaller patches together with their masks to enable the training of CNNs. Addtionally, the images and masks were "vectorised" (flattened) into 1D arrays to enable the training

of classical machine learning baselines. The processing code was written in Python 3 [84] and made use of rasterio [85], GDAL [86], numpy [87], scipy [88] and OpenCV [89]. Due to memory constraints the code was developed locally as a PyPi package against test GeoTIFFs [90] before being installed into a high RAM GPU Google Colab environment [91] where it could be run against the real Capella GeoTIFFs. The full procedure is illustrated by Figure 3.5 and each step is elaborated more in the following sections.

3.2.1 Rotation Angle & Trimming

As demonstrated by Figure 3.1, the Capella TIFF comes with the image angled surrounded by a large black border of "no data values" which come about from various geographic processing steps [62]. Such processing steps are useful for geographic related tasks however the resulting "no data" values are a hindrance when applying deep learning. Existing approaches for dealing with these "no data" values outlined in Section 2.4 are in-painting, defining a "no data" class, cropping and doing nothing. Each of these approaches has disadvantages. In-painting passes large regions of non applicable imagery to a model which is arguably a waste of computational resource. Similarly, passing the no data values as their own class for the network to learn is inefficient and somewhat redundant considering that the value that represents no data will be present in the images meta data and a no data mask can readily be generated. Cropping the image down to only contain data pixels has the advantage that the trained models only see relevant pixels but has the obvious disadvantage that not all of the available data is being utilised. Doing nothing is similar to in-painting in that large amounts of useless data is passed to the network.

To improve upon these existing methodologies alternative techniques were explored the with the aim to maximise use of data pixels whilst minimising the use of "no data" pixels. Initially, it was investigated whether information about the applied projections was present in the Capella metadata such that reversing them could be attempted in an effort to remove the "no data" values. However, the only related data describes the affine transformation matrix that transforms the image from coordinate space to its projected georeferenced coordinate space, known as a geotransform [92]. This maintains orientation and therefore does not reduce the "no data" values.

Therefore, a new technique was developed for this project inspired by the approach described by [93] for document skew detection and correction. A no data mask was

generated for each image from which the coordinates of a minimum area rectangle that bounded all the "data" pixels was determined. The skew angle of this bounding rectangle was calculated and the affine transformation matrix required to remove the skew angle from the bounded rectangle was determined using OpenCVs implementation of the algorithm described by [94]. This transformation was applied to the image with the end result being an image which was no longer skewed, albeit still surrounded by a border of "no data" values. The image was therefore trimmed by removing any columns or rows that solely consisted of "no data" values. To ensure the mask associated with the image was still correct, the same transformation was applied to the mask and the rows and columns that were removed from the image were also removed from the mask. The procedure is illustrated by Figure 3.6 and Table 3.2 details image information after this process.

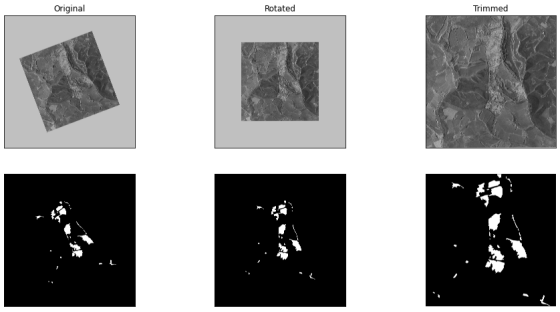


Figure 3.6: The rotation and trimming processing steps demonstrated for one of the Capella images and its accompanying mask.

Site	Rotation Angle	Windthrow Percentage	Processed Pixel Size (Range x Azimuth)
Monynut HH SP	$-20.56°$	5.8%	14523 x 14257
Glenisla HH SP	$-5.57°$	4.0%	14189 x 14480
Glenisla HH SS	$9.23°$	3.9%	8303 x 17451
Fetterangus HH SS	$-15.26°$	1.3%	8259 x 16991

Table 3.2: Capella imagery after rotation, trimming and calibration. Rotation angle is the angle of rotation required to remove the tilt from the image. Windthrow percentage is the percentage of pixels that represent windthrow in the rotated and trimmed image.

3.2.2 Calibration & Contrast Stretching

After rotation and trimming the images were manually Sigma Nought calibrated in order to minimise the differences in image radiometry. For Capella images Sigma Nought (in

decibels) is calculated from the *Digital number* (pixel value) for GEC products using Equation 3.1

$$\sigma^0_{db} = 20 \cdot \log_{10}(SC \cdot DN), \tag{3.1}$$

Where *SC* is the scale factor (found in the extended image metadata) which accounts for internal calibration of the radar subsystem and *DN* is the Digital Number [95]. Once calibrated, the images were further contrast stretched such that the pixel range was between 0 - 255 according to equation 3.2,

$$P_{out} = a + (P_{in} - c) \cdot \left(\frac{b-a}{d-c}\right) \tag{3.2}$$

where P_{out} is the resulting pixel value, P_{in} is the input pixel value, a and b are the desired lower and upper limit respectively $(0-255)$, and c and d are the lowest and highest values currently present in the image respectively. Outlying pixels with either a very high or very low value can severely affect the value of c or d which leads to unrepresentative scaling. Therefore, a more robust approach is to take a histogram of the image and select c and d at percentiles in the histogram, say 5th and 95th. Any pixels with value below the 5th percentile value would be set to the lower limit and any pixels above the 95th percentile value would be set to the upper limit.

The distribution of SAR images tends to be very asymmetric with the majority of pixels representing low intensity back-scattering followed by a long tail of bright high intensity pixels. The multi-looking process already filters a large amount of outliers via speckle reduction and given that windthrow tends to present itself as brighter than its surroundings an asymmetric stretch using the 5th and 99th histogram percentiles was implemented. The intent of this was to enhance separation amongst the bright pixels by collapsing low intensity pixels to the same value. This adds further noise reduction to the low intensity pixels and increases the range of values that the bright pixels can be stretched into.

3.2.3 Vectorisation

To convert the images into a format suitable for the training of classical machine learning algorithms, each image and its associated masks were flattened into 1D arrays of equal length. These arrays were shuffled and split 80%;10%;10% into smaller training, validation and test arrays. The resulting data splits for each image were collated into

Site	Training Pixels	Validation Pixels	Testing Pixels
Monynut	165,643,529	20,705,441	20,705,441
Glenisla (SP)	164,365,376	20,545,672	20,545,672
Glenisla (SL)	115,916,523	14,489,565	14,489,565
Fetterangus	112,262,937	14,032,866	14,032,866
Total	558,188,365	69,773,543	69,773,543

Table 3.3: Properties of the training, validation and testing vectorised splits. These numbers apply to the Sigma Nought calibrated arrays, contrast stretched arrays and arrays formed from the masks labels.

larger arrays now containing data for each image. Both the calibrated images and the further contrast stretched images were vectorised in identical manners in order to investigate whether the contrast stretching was beneficial for the task of segmenting windthrow. Information regarding the vectorised data splits is given by Table 3.3.

3.2.4 Tiling

Several patch sizes and techniques exist in the literature and are often utilised with little justification [44, 63]. The choice was taken to utilise a non-overlapping tile strategy, partly because such tiling strategies are simple to implement and have demonstrated good performance for other remote sensing segmentation tasks [44, 63], but also remove the need for mirroring required by the overlapping tile strategy originally proposed with U-Net [20]. This choice was influenced by the geometrical distortions that are present in SAR imagery due to its side looking nature. Phenomena such as shadow and layover are dependent on the direction of the incoming energy, therefore mirroring around any tile which contains geometric distortions would result in nonsensical patterns which will never actually be present in SAR imagery. This is less of a consideration for other tasks like U-Nets original application of segmentation of biological cells, which look similar when the image is mirrored.

The contrast stretched images were shown to be beneficial for classical machine learning, see Section 5.1 and therefore only the contrast stretched images were tiled. For the tiling itself, a methodology was designed to enable a systematic comparison of the effect of different tile sizes, the end result is illustrated by Figure 3.7. The approach required specifying a maximum and minimum patch size with the constraint that the minimum patch size could be reached by repeatably dividing the maximum patch size by 2. Each individual image had columns removed from the right hand side

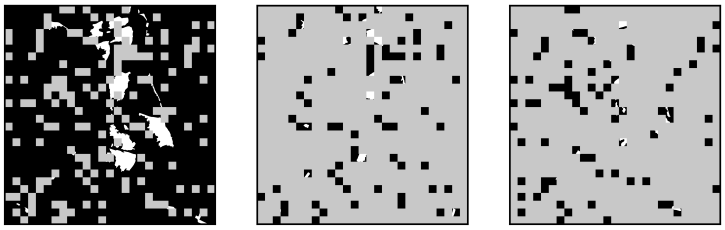


Figure 3.7: Tiling strategy illustrated when applied for 512×512 pixel patches to the mask of the Monynut HH image. From left to right: training split, validation split, testing split. The tiles are in their correct spatial location and grey areas indicate tiles are not present for this split. This tiling strategy means these splits are the same for smaller patch sizes which would be visually identical even though the patch would be smaller.

and rows removed from the bottom until its rows and columns were directly divisible by the maximum patch size - the same columns and rows were removed from the associated mask. The image and mask were tiled into patches of the maximum patch size and patches were named in matrix index notation [96] such that they could be stitched back together to form the original image/mask. The tiles were shuffled and split 80%;10%;10% into training, validation and test sets and the individual splits were collated into top level splits that represented tiles from all images. The maximum patch size was divided by 2 and these top level tiles were each further tiled into 4 smaller patches and so on until the minimum patch size had been achieved. This approach ensured that the same pixels were used for the training, validation and test sets regardless of patch size which enabled investigations into the effect of patch size without any bias; something currently lacking in the remote sensing literature [63]. A maximum patch size of 512 and a minimum patch size of 128 were specified and Table 3.4 details properties of the different tile splits for the individual images.

	Tile Sizes								
	512×512			256×256			128×128		
Site	Train	Val	Test	Train	Val	Test	Train	Val	Test
Monynut	606	75	75	2424	300	300	9696	1200	1200
Glenisla (SP)	606	75	75	2424	300	300	9696	1200	1200
Glenisla (SL)	436	54	54	1744	216	216	6976	864	864
Fetterangus	424	52	52	1696	208	208	6784	832	832
Total	2072	256	256	8288	1024	1024	33152	4096	4096

Table 3.4: The resulting number of tiles for each image and data split when processed with the described tiling strategy. Contrast stretched images and associated masks were subject to the same tile processing.

Chapter 4

Methodology

The experimental methodology can be divided into two phases. (1) The training of classical baselines which is mostly intended to demonstrate their inadequacy for segmenting single channel SAR images. (2) A deep learning approach using a U-Net inspired architecture. Given that there is almost no literature around applying deep learning to SAR data an ablation study [97] with the U-Net model was performed to measure the impact of different patch sizes, loss functions [50], up-sampling techniques [30], augmentations [98] and dropout [39]. The code was written in Python3 and training was performed in high RAM Google Colab environments with Nvidia P100 GPUs [99].

4.1 Evaluation Metrics

As demonstrated by Table 3.2 windthrow represents a minority of the total pixels in the data, about 4% overall. Therefore accuracy would be a poor evaluation metric for segmentation performance; 96% accuracy could be achieved by masking the entirety of the images as "not windthrow". Models were instead evaluated based on dice coefficient; a metric derived from recall and precision. Recall is a metric that provides insight into the performance at capturing all *true positives* (windthrow pixels which were classified as windthrow) whilst disregarding *false positives* (non windthrow pixels which were classified as windthrow). Precision is a metric that provides insight into the amount of classified windthrow pixels that were actually windthrow. The calculation of recall and precision is illustrated by Figure 4.1. Often these two metrics conflict with each other and to find the optimum balance dice coefficient [55] (also known as F1 score) was utilised as it is the harmonic mean between precision and recall, see Equation 4.1.

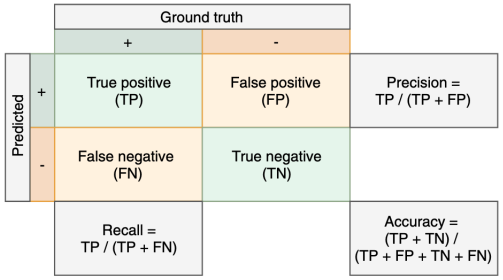


Figure 4.1: Confusion matrix between predictions and ground truth for binary labelled data. Illustrated is how recall, precision and accuracy are derived from this matrix. Figure credit [62].

$$Dice = \frac{2}{Recall^{-1} + Precision^{-1}} \tag{4.1}$$

Dice coefficient ranges between 0 & 1 and is high when predictions match the labels well and do not extend outside of them which makes it a robust metric for evaluating segmentation models. A dice score of 0 means no agreement between labels and predictions, 1 means perfect agreement and everything in-between can be interpreted relative to these extremes.

4.2 Classical Baselines

Classical baselines were trained using scikit-learn [100] with Gaussian Naive Bayes and Logistic regression being the chosen classifiers. Gaussian Naive Bayes classifiers work under the assumption that considered classes are independent, they are advantageous in that they are simple to train and can handle large amounts of data [101]. Logistic regression belongs to the class of generalized linear models and is popular due to its probabilistic interpretation; class label predictions for a given input range between 0 & 1 which can be thresholded to give binary predictions [102]. These models were chosen partly because they are popular within the remote sensing community [19, 18, 103] but mainly because their scikit-learn implementations supported incremental learning [104]. The latter is important due to memory constraints as a consequence of the large arrays generated from the vectorisation process requires that algorithms be trained on batches instead of the whole dataset at once. Each model was trained twice: once on the sigma nought calibrated pixels; once on the further contrast stretched pixels. Image and mask training pixels were randomly shuffled in the same manner and split into batches of

1×10^5 pixels. This size was arbitrarily chosen and resulted in roughly 5582 training batches. The models were trained according to their defaults in scikit-learn [105, 106] and were evaluated on the validation and test sets (batched in the same manner). The Gaussian Naive Bayes assigned each evaluated pixel a binary label whereas the Logistic regression model assigns each pixel a probability of being windthrow, which was thresholded at 0.5 to yield binary labels. These binary labels allowed the dice coefficient between the predictions and ground truth labels to be calculated. Extensive experiments were not conducted with the classical baselines as the literature strongly indicated that these methodologies would perform poorly for SinglePol SAR images [21, 59].

4.3 Deep learning model

The architecture used for the deep learning aspect of this work is illustrated by Figure 4.2 and the implementation is similar to the original U-Net architecture. The 3×3 convolutions were modified to use same padding instead of valid padding to enable seamless segmentation of full images for the chosen non-overlapping tiling strategy. Batch normalisation was introduced between the 3×3 convolutions and ReLUs which significantly reduces the training time compared to the original U-Net. Combining batch normalization and dropout can cause issues and therefore following the advice of [107], dropout was introduced only after the final Batch norm + ReLU. The final layer was modified to be a sigmoid, which in combination with the same padded convolutions meant that the output was a single channel probability mask the same spatial size as the input in which each pixel was assigned a probability of being windthrow. This probability mask could be further thresholded to produce a binary windthrow mask for the input patch which allows the dice score to be calculated. The architecture was implemented in PyTorch [108] and built upon the popular U-Net repository provided by [109]. The network was modified such that the initial feature channel size could be specified and the type of convolutional up-sampling could be chosen between deconvolution and bilinear [30].

The input tiles were normalized to lie between 0 & 1 which was simply achieved by dividing the patches by 255 as only the contrast stretched patches were used. Normalization helps networks learn faster [111, 37] and other initial normalisation techniques could have been investigated such as standardising the input tiles. However, the introduction of batch normalization layers make this initial step somewhat superfluous and thus the simplest technique was chosen. For each epoch the training tiles and mask tiles

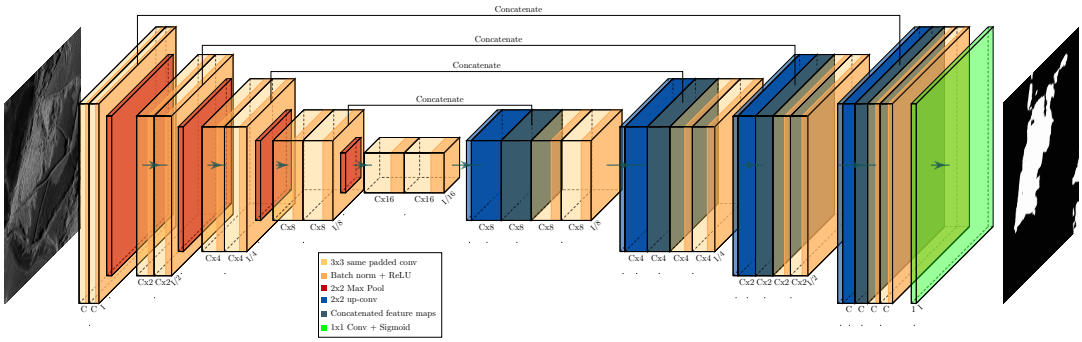


Figure 4.2: The deep learning architecture that was used for this work, heavily inspired from U-Net [20]. C is the initial number of feature channels in the first layer of the architecture and I is initial the spatial size of the input image (or patch). The number of feature channels and spatial size for subsequent layers are given in relation to the initial values. The input is a patch from one of the Capella images and the output is a pixel wise segmentation map for that patch. Diagram created using [110].

were identically shuffled before being split into smaller training batches and associated mask batches. Shuffling the data ensures that the batches that are presented to the network for each epoch are unique. Each batch was passed through the network and predictions were generated for each enclosed tile and from these predictions the loss for the batch was calculated with reference to mask tiles. The parameter gradients were derived from this loss and an optimisation step was taken. To monitor the networks ability to generalise to unseen data and to avoid overfitting the training set, after each epoch the network was evaluated on the validation set using dice coefficient as the metric. To further explore the networks ability to generalise, on completion of training the epoch which performed best on the validation set was evaluated on the test set, again using dice coefficient as the metric.

The optimizer used for training was ADAM which differs from optimizers like Stochastic Gradient Descent (SGD) [112] in that it maintains an adaptive learning rate (up to an initial upper limit) for each network parameter which updates as the network trains, instead of assigning each parameter the same learning rate. ADAM was chosen because it has proven faster convergence than optimizers like SGD [113]. There is debate as to whether ADAM is more prone to converging towards solutions with poorer generalisation when compared to other optimizers like SGD [114], however given the time constraints of the project, this potential pitfall was deemed acceptable in favour of faster training. The default PyTorch implementation was utilised which uses the optimizers hyperparameters recommended in the original paper [113, 115].

Parameter	Description
Loss Function	The loss function to optimise during training.
Tile Size	The tile size to use during training.
Batch Size	The batch size to use as part of the training procedure, largely determined by hardware memory constraints.
Epochs	The number of epochs (passes of the entire dataset) to complete as part of the training procedure. All conducted experiments were trained for 50 epochs.
Initial Learning Rate	Defines the initial learning rate for the ADAM optimiser. For ADAM, this specifies the upper limit of the step that can be taken when modifying the parameter values. For all experiments an initial learning rate of 0.001 was chosen, as recommended in the original paper [113].
Threshold	Defines a value between 0 & 1 that pixel values in the probability mask must be greater than for them to be declared as windthrow.
Initial Channel Size	Defines the number of channels in the initial feature map in the first layer of the U-Net model. All subsequent layers channel sizes are derived from this initial channel size, see Figure 4.2.
Bilinear	Boolean parameter for whether of not the up-sampling convolutions use bilinear up-sampling (True) or transposed convolution (deconvolution) up-sampling (False), see [30] for how they differ.
Transformations	PyTorch transformations and augmentations that were applied to the training data. A full list of available augmentations is given by [116].
Dropout probability	The probability to zero elements of the input tensor during training. The zeroing of elements is chosen according to a Bernoulli distribution [117].

Table 4.1: The parameters (and a description of them) that could be tuned during the model training procedure.

4.4 Deep Learning Experiments

All modifiable training parameters are detailed in Table 4.1 and the following sections discuss the combinations which formed the basis for the conducted experiments. All experiments were trained for 50 epochs, used an initial learning rate of 1×10^{-3}, L2 regularization [118] of 1×10^{-8} and model weights were initialized by the LeCun uniform initializer [119]. Each model took around 4 hours to train and around 30-45s to create a full prediction mask for one of the Capella images.

4.4.1 Patch & Batch Size

The trade off between batch and patch size has been noted in the remote sensing literature but is rarely elaborated. The effect was investigated by establishing the largest possible hardware limited batch size for the 512 patches, which was determined to be 4, and from this the largest batch sizes for the 256 and 128 patches were derived to be 16 and 64 respectively. Specifying the batch sizes in this manner ensured that the same number of gradient steps would be taken for each epoch for each size of patch, ensuring that any differences would be down to the interaction between batch and patch size. For each patch/batch pair the model was trained against dice loss with an initial channel size of 64 without augmentation or dropout. Dice loss was chosen as an educated guess for these initial experiments as it has been shown to work well for datasets with mild class imbalance and is capable of handling different spatial patterns like those observed in the windthrow masks [50]. The effect of different up-sampling

techniques was concurrently investigated, with the described experiment repeated for both deconvolution and bi-linear up-sampling. Finally, the thresholding level is a tunable hyper-parameter and whether this influenced generalisation performance was investigated by evaluating the dice coefficient for the validation set with binary masks produced at 0.3;0.5;0.7 thresholds. The effect of smaller batches for the smaller patch sizes was not investigated as the benefits of larger batches is well established [40, 62].

4.4.2 Loss function

The combination of patch; batch; up-sampling; threshold that generalised best to the validation and test set was taken forward to investigate the performance of different loss functions. The choice of loss function plays an essential role in determining the model segmentation performance and one cannot say conclusively which loss function will work better on a particular dataset. Therefore the performance of the following loss functions was investigated: binary cross-entropy (BCE) loss, focal loss, dice loss, Tverksy loss and focal Tverksy loss. These loss functions were chosen as they represent a spread of functions that have been shown to perform uniquely, dependent on the distribution of the given dataset [50]. Except for BCE loss, these functions are not included with PyTorch and therefore implementations of them provided to the following Kaggle competition were integrated into the training procedure [120]. What follows is a succinct description of each of these loss functions; the hyper-parameter combinations which were trained and results are discussed in Section 5.

BCE loss is defined as a measure of the difference between two probability distributions for a given random variable and focuses on maximising the probability that the correct pixels are predicted. BCE loss is widely used for segmentation tasks and performs best for distributions with balance amongst the classes [52]. For convenience, defining the predicted class probabilities for binary classes by Equation 4.2, allows the BCE loss to written as Equation 4.3

$$p_t = \begin{cases} p, & y = 1 \\ 1 - p, & y = 0 \end{cases} \tag{4.2}$$

$$L_{BCE}(y, p) = -\log(p_t) \tag{4.3}$$

where y is the true class label and p is the predicted probability. Focal loss is a variant of BCE loss that down-weights the contribution of easy examples to allow the

model to focus of learning harder examples. Focal loss was designed to extend BCE loss to perform well for datasets with highly imbalanced classes [51]. By again making use of Equation 4.2, Focal loss can be defined by 4.4

$$L_{Focal}(y,p) = -\alpha(1-p_t)^{\gamma}\log(p_t) \tag{4.4}$$

where α and γ are hyperparameters and setting $\alpha = 1; \gamma = 0$ recovers the standard BCE loss.

Dice Loss modifies the dice coefficient metric into a loss function and effectively calculates the dice coefficient by working with predicted probabilities instead of thresholded binary labels. This loss function is region based instead of distribution based like BCE/Focal loss and focuses on maximising the intersection between predictions and corresponding masks. Not being distribution based allows dice loss to perform well regardless of class imbalance [53]. Dice loss is defined by Equation 4.5

$$L_{Dice}(y,p) = 1 - \frac{2yp+1}{y+p+1} \tag{4.5}$$

where y is the true class label, p is the predicted probability and a 1 is introduced in the numerator and denominator to avoid division by zero in the case of $y = p = 0$.

Tversky loss is similar to dice loss but is weighted by α and β constants that control how false positives and false negatives are penalised in the loss function respectively as their values increase [121]. Tversky loss is defined by Equation 4.6.

$$L_{Tversky}(y,p) = 1 - \frac{1+yp}{1+yp+\alpha(1-y)p+\beta(1-p)y} \tag{4.6}$$

For the case of $\alpha = \beta = 0.5$ Tversky Loss reduces to Dice loss. The β parameter in particular has application in situations where models can be misleadingly performant by making highly conservative predictions [121]. Focal Tverksy loss is a variant of Tverksy loss which similar to focal loss, down-weights the contribution of easy examples and is defined by Equation 4.7

$$L_{FocalTverksy}(y,p) = (L_{Tversky}(y,p))^{\gamma} \tag{4.7}$$

4.4.3 Ablation Studies

The loss function which generalised best to the validation and test sets from the previous experiments was utilised for further ablation studies. From Figure 4.2 it can be seen that the initial feature map channel size greatly influences the complexity of the model. Lightweight versions of U-Net with smaller initial feature map channel sizes have found success in other remote sensing segmentation tasks [122, 62] and for certain medical image segmentation tasks [123]. Therefore, it was investigated whether smaller initial feature map channel sizes could achieve comparable performance to that of the standard 64 that U-Net is usually trained with. Less complex models are less likely to overfit the training data but likely have a lower performance ceiling [43]. Perhaps more relevant to this task is that less complex models offer faster training and prediction times, which is useful when the goal is to rapidly segment windthrow boundaries.

Augmentation encompasses a suite of techniques that increase the size of the training dataset which can enable more performant models to be trained. Many augmentations techniques exist and investigating them all was beyond the time frames of this project [98]. Therefore, only brightness/contrast augmentations and horizontal/vertical flips were investigated. Brightness & contrast jittering was chosen under the hypothesis that it may be able represent the differences in shades between SAR images. The jittering was implemented by specifying α between $[0, 1]$ such that a factor (ϕ) for each batch was uniformly drawn between $\phi = U[1-\alpha, 1+\alpha]$, brightness and contrast modifications were then made in random order for each image in the batch relative to its pixel values and according to ϕ. Horizontal/Vertical flips were chosen as forests in SAR images appear different dependent on the direction of incoming radiation and it was believed flips may help a model to learn these differences. A flip probability was defined and each patch was given the opportunity to flip horizontally and then vertically with this probability. Finally, whether dropout would be beneficial for the most performant combinations was investigated. Models were retrained with varying levels of dropout, which was only implemented after the final DoubleConv+ReLU layer.

Chapter 5

Results & Discussion

5.1 Classical Baselines

The results of the Gaussian Naive Bayes (GNB) and Logistic Regression (LR) classifiers are shown in Table 5.1. The LR classifiers achieved comparable results to what was achieved with LR for classifying windthrow in optical images, however the GNB classifiers performed significantly better [21]. Contrast stretching was shown to be beneficial and this influenced the decision to use contrast stretched patches for the deep learning learning model. Performance between the validation and test sets remained comparable and this is likely because no evaluation was performed on the validation during training. The model was evaluated on the validation and test set after training and therefore had no opportunity to acquire an affinity towards the validation set. A more rigorous approach that monitored validation performance after each training batch may have improved the LR performance. The GNB model performed surprisingly well, achieving comparable dice coefficient to that for building segmentation in SAR imagery with a deep learning approach [56]. This perhaps indicates that the independence assumption taken by GNB classifiers is somewhat sensible for windthrow pixels. Regardless, the resulting dice scores are poor.

Model	Image Type	Val Dice Score	Test Dice Score
GNB	σ_0 Cal	0.157	0.159
GNB	Contrast Stretched	0.201	0.198
LR	σ_0 Cal	0.066	0.067
LR	Contrast Stretched	0.082	0.082

Table 5.1: Results for the classical baselines. GNB (Gaussian Naive Bayes). LR (Logistic Regression)

Model	Patch Size	Batch Size	Up-Sampling technique	Best Val Epoch	Best Val Dice Score	Test Set Dice Score	Test Val Diff
512Deconv	512	4	Denconvolution	27	0.495	0.420	-0.075
512Bilinear	512	4	Bilinear	9	0.525	0.427	-0.098
256Deconv	256	16	Denconvolution	33	0.629	0.626	**-0.003**
256Bilinear	256	16	Bilinear	42	0.649	**0.632**	-0.017
128Deconv	128	64	Denconvolution	50	0.690	0.558	-0.132
128Bilinear	128	64	Bilinear	36	**0.713**	0.582	-0.131

Table 5.2: Results of the patch size, batch size and up-sampling technique experiments. Validation and test dice scores are calculated from binary masks thresholded at 0.5.

5.2 U-Net Model

Results from investigating the concurrent effects of patch size, batch size, up-sampling technique are delineated in Table 5.2. Models were trained with dice loss and the choice of patch/batch size combo greatly influenced the final performance and generalisation capability; larger 512 patches perform worst on the validation set whilst 128 patches perform best with 256 patches performance in between. 512 and 128 patches experience a significant drop in performance when generalised to the test set, observed to a lesser extent with the 256 patches. All models outperformed the classical baselines.

This interaction can be further explored by inspecting the training curve shown by Figure 5.1. The greatest reduction in loss is achieved in the initial epochs, after which the loss steadily decreases for each model at a similar rate. The benefits to training speed for larger batches when utilising batch normalization is well known so it is suspected that even though the chosen patch / batch size combinations present the same number of pixels to the model, smaller patch sizes have the opportunity to present a wider spread of pixels. When batch normalisation standardises the data the calculated means and standard deviation are therefore more representative of the underlying distribution of the pixels; leading to faster training. It is suspected the poor performance from the 512 tiles is a consequence of the model learning slowly with these tiles, whilst the 128 tiles faster learning is potentially leading to overfitting evidenced by the large drop in performance on the test set. Training with 256 tiles appears to be a good balance between training speed and mitigating overfitting.

Table 5.2 further shows that for each batch/patch combination bilinear up-sampling achieved higher validation/test dice score than deconvolutional up-sampling. Similar results have been observed when segmenting trees in high resolution airbourne optical imagery [124]. Deconvolutional up-sampling has been shown to introduce checkerboard

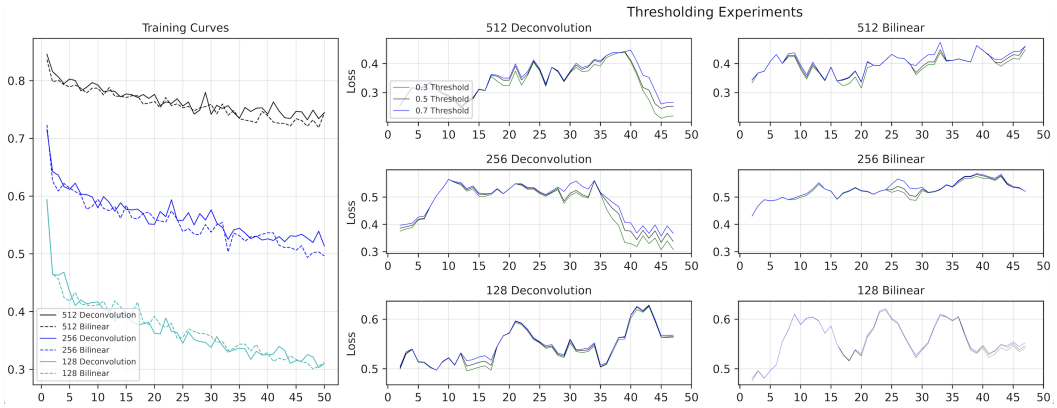


Figure 5.1: Training curves for each batch/patch size combinations (left). Also shown are plots which show how the dice coefficient for the validation set varies when calculated with binary masks produced at different probability thresholds (middle & right). For each graph y-axis is loss and x-axis is epochs.

artifacts [125] which are not present with bilinear up-sampling; it is hypothesised that during U-Nets expansive path these checkerboard artifacts could be misinterpreted as strong features by the model and be distracting to the segmentation task. Figure 5.1 also illustrates the effect on the validation dice score by thresholding at 0.3;0.5;0.7. The validation dice score for each combination when plotted against the training epoch was quite erratic so to better understand trends the data is presented as a 5-point moving average. The validation dice score remained quite comparable regardless of threshold level and the epoch with the highest validation dice score was the same for each model regardless of threshold. This indicates that U-Net is capable of producing probability masks with confident predictions which results in the chosen threshold level being unimportant as the probability mask already closely resembles generated binary masks. Based on these results a patch size of 256, batch size of 16, bilinear up-sampling and a threshold level of 0.5 was chosen for subsequent experiments.

The loss function hyper-parameter combinations that were explored are detailed in Table 5.3, where the baseline is the most performant model from the previous experiments. An inspection of these results reveals that none of the experimented loss functions exceeded the baselines performance on the validation set and only the Tverksy0802 model was able to slightly exceed the baseline when generalised to the test set. Figure 5.2 illustrates the validation dice score against epochs for each loss function which is again presented as a 5 point moving average to aid in interpretation. For visual clarity the models are grouped as focal loss experiments and dice derived loss functions. The focal loss experiments demonstrate that BCE loss has the least erratic curve and

Model	Loss Function	α	β	γ	Best Val Epoch	Best Val Dice Score	Test Set Dice Score	Test Val Diff
Baseline	Dice	-	-	-	42	**0.649**	0.632	-0.017
BCE	BCE	-	-	-	37	0.641	0.614	-0.027
Custom	BCE + 0.1 Dice	-	-	-	41	0.636	0.595	-0.041
Focal	Focal	0.8	-	2	23	0.581	0.523	-0.058
Tverksy0604	Tverksy	0.6	0.4	-	37	0.628	0.615	-0.013
Tverksy0703	Tverksy	0.7	0.3	-	33	0.620	0.626	0.006
Tverksy0802	Tverksy	0.8	0.2	-	41	0.618	**0.634**	**0.016**
Tverksy0208	Tverksy	0.2	0.8	-	44	0.561	0.492	-0.069
Tverksy0307	Tverksy	0.3	0.7	-	5	0.564	0.518	-0.046
Tverksy0406	Tverksy	0.4	0.6	-	50	0.605	0.594	0.011
FocalTverksy1.5	Focal Tverksy	0.5	0.5	1.5	30	0.617	0.617	0.000
FocalTverksy2	Focal Tverksy	0.5	0.5	2	22	0.582	0.538	-0.044
FocalTverksy3	Focal Tverksy	0.5	0.5	3	10	0.604	0.501	-0.103

Table 5.3: Results of loss functions experiments, see Section 4.4.2. Each model was trained for 50 epochs with channel size 64, patch size 256, batch size 16 and thresholded at 0.5.

remains quite constant in performance past 20 epochs. The baseline dice model is more erratic, averaging lower performance overall but with spikes of performance at certain epochs above that achieved by BCE. Dice loss demonstrates an increasing trend up to around 40 epochs before falling off which is indicative that overfitting has begun to occur. To investigate the benefit of simultaneously maximising pixel probabilities (BCE Loss) and intersection between mask and prediction (Dice Loss) a model was trained with a custom loss of $BCE + 0.1Dice$. The dice loss was weighted by 0.1 to bring its observed loss range during training of $0.72 \rightarrow 0.49$ to be comparable of that observed for the BCE loss $0.14 \rightarrow 0.03$ such that they would be weighted somewhat equal during training. This custom loss curve sat somewhere between BCE and Loss and even averaged the best perform between epochs 35-40, however this did not translate to better test set generalisation where it performed worse than BCE and Dice. The effect of incorporating focal loss aspects into BCE and Dice loss was investigated for various γ values was shown to not be beneficial to performance.

Figure 5.2 additionally illustrates the effects of varying α and β within the Tverksy loss function. Higher α weights precision more important than recall, conversely higher β weights recall higher than precision. It can be seen that higher α values perform comparably to standard Dice $(\alpha = \beta = 0.5)$ whereas higher β values perform significantly worse. This suggests that there are some examples of windthrow that the architecture is struggling to learn how to segment, even when it is rewarded for not being precise with its predictions. This combined with the performance drop observed when

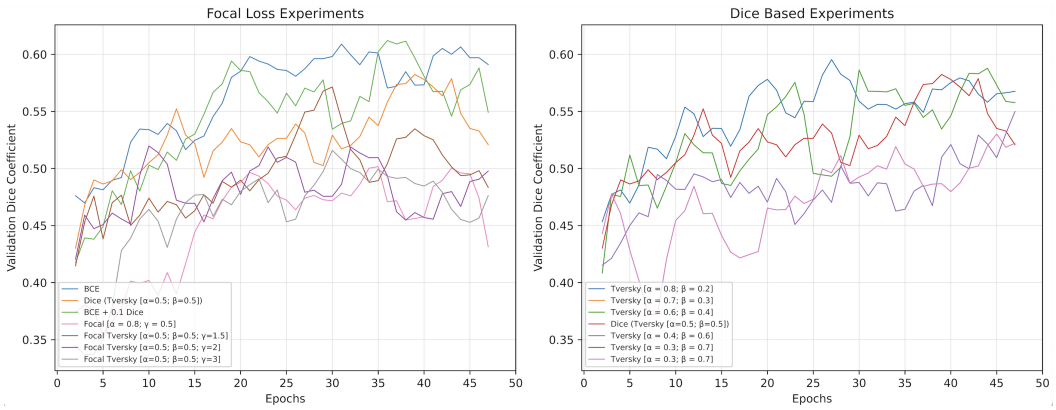


Figure 5.2: Validation Dice score for focal loss experiments (left) and dice score derived loss functions (right). Data is presented as a 5 point moving average.

focal aspects are introduced into the loss functions indicates that the class imbalance between “windthrow” and “non windthrow” pixels is not the limiting performance factor. Instead performance seemed to be limited by the architecture not being able to cope when asked to focus on harder examples. To investigate this complete masks for each image were generated using the baseline dice model and the dice score for each complete image mask was calculated, the results of which are illustrated by Figure 5.3.

Figure 5.3 reveals that the models segmentation performance is related to the geometry that the SAR image was acquired with, this correlated with how easy windthrow was to manually identify during labelling. The highest performance is achieved in the two images where the direction of the felled timber is perpendicular to the range dimension (Monynut SP; Glenisla SS). This performance drops off when the fell direction is somewhat perpendicular to the range dimension (Glenisla SP) and the model is incapable of segmenting windthrow when the fell direction is parallel to the range dimension (Fetterangus SS). Interestingly, the higher spatial resolution from SP over SS mode is not a deciding factor in final performance as the Glenisla SS image covers the area of the Glenisla SP image and despite having lower spatial resolution it demonstrates fewer omissions than the Glenisla SP image of the same area. Failure to segment any windthrow in the Fetterangus SP is the suspected reason for the model architecture struggling when asked to focus on hard examples. This perhaps shouldn’t come as a surprise the since the areas of windthrow in this image are visually distinct compared to the other 3 images and could only be identified during the labelling stage with the aid of optical imagery. Further exacerbating the challenge of segmenting this image is that it contained the lowest proportion of windthrow leading to few examples for the architecture to learn from. Small amounts of training data often causes overfitting and

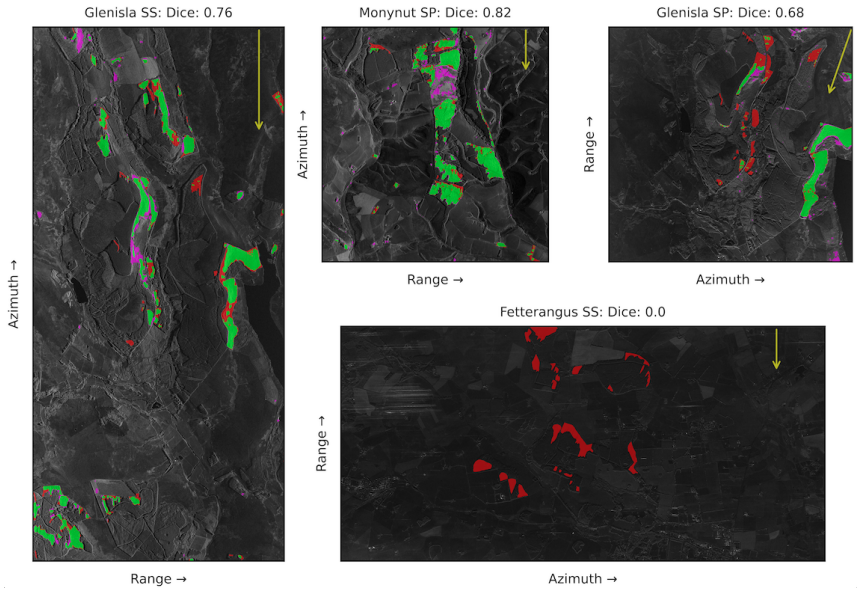


Figure 5.3: Full predictions masks and associated dice score for each image. Green represents areas of agreement between labels and predictions (intersect), Red represents areas which should have been predicted as windthrow but were not (omissions), pink represents areas which were predicted as windthrow but should not have been (commissions). The yellow arrow is the approximate direction in which the timber fell. The Dice score for the complete image mask is presented for each image.

it may be the case that the architecture has overfitted the other 3 images. Despite this challenge, the combined test set dice score of 0.63 when utilising Dice Loss did coincidentally match that achieved by the previous study utilising U-Net for road segmentation in SAR imagery [61]; which also matched the dice score achieved for segmentation of windthrow in optical imagery [21]. The test set results are competitive with the most established SAR temporal change detection approach utilising Sentinel-1 which averages a dice score of ≈ 0.7 [9]. However, omitting test tiles from the Fetterangus image yields a reduced test set dice score of 0.71.

It can be seen in Figure 5.3 that the Fetterangus image is less bright than the other 3 images which led to the hypothesis that Brightness/Contrast augmentations would aid the architecture in segmenting areas of windthrow in this image. Additionally, the effect of horizontal/vertical rotations were investigated to deduce whether this would mitigate the observed dependence on the imaging geometry. Dice loss was utilised for these experiments which were conducted simultaneously with varying the initial feature map channel size, the results of which are listed in Table 5.4. The baselines in this table refer to models trained without augmentations for a specific Initial Channel

Model	Loss	Channel size	Jitter	Flip %	Best Val Epoch	Best Val Dice Score	Test Set Dice Score	Test Val Diff	Fetterangus Dice Score
64Baseline	Dice	64	-	-	42	**0.649**	0.632	-0.017	0.0
64Dice0125J	Dice	64	0.125	-	45	0.644	**0.635**	-0.009	0.0
64Dice025J	Dice	64	0.250	-	43	0.612	0.590	-0.022	0.0
64Dice05J	Dice	64	0.500	-	50	0.580	0.519	-0.061	0.0
64Dice05F	Dice	64	-	0.5	31	0.600	0.602	0.002	0.0
32Baseline	Dice	32	-	-	42	0.607	0.627	0.020	0.0
32Dice0125J	Dice	32	0.125	-	40	0.606	0.603	-0.003	0.0
32Dice025J	Dice	32	0.250	-	20	0.595	0.596	0.001	0.0
32Dice05J	Dice	32	0.500	-	50	0.592	0.618	0.026	0.0
32Dice05F	Dice	41	-	0.5	31	0.557	0.595	**0.038**	0.0
16Baseline	Dice	16	-	-	38	0.612	0.630	0.018	0.0
16Dice0125J	Dice	16	0.125	-	43	0.598	0.584	-0.014	0.0
16Dice025J	Dice	16	0.250	-	49	0.600	0.577	-0.023	0.0
16Dice05J	Dice	16	0.500	-	49	0.601	0.561	-0.061	0.0
16Dice05F	Dice	16	-	0.5	42	0.591	0.598	0.007	0.0

Table 5.4: Results of jittering and horizontal/vertical flipping augmentation experiments. Each model was trained for 50 epochs with patch size 256, batch size 16, thresholded at 0.5 and used bilinear up-sampling.

Size (ICS). An interesting result from these experiments is that smaller ICSs 32; 16 achieve almost identical test set performance to that achieved with a model trained using U-Nets standard 64 channels. These models were quicker to train, taking 1 hour; 30 minutes respectively for 32; 16 ICS compared to the 4 hours it took to train with 64 ICS. Predicted masks for a full image were also quicker, taking approximately 12s; 6s for 32; 16 ICS respectively compared to the 45s it took to mask a full image with 64 ICS. Considering that the goal is to rapidly segment windthrow, it is encouraging that simpler architectures achieve comparable performance to more complex models.

Table 5.4 further reveals that contrast/brightness augmentations did not improve performance; no model that utilised augmentations was able to map any of the windthrow in the Fetterangus image and only one combination (64Dice0125J) slightly improved upon its respective baselines test set performance - all others reduced performance. Despite the previous hypothesis that contrast/brightness augmentation may be beneficial, it was instead concluded that they likely interfere with the Sigma Nought calibration which leads to them being detrimental to performance. Horizontal/vertical flips of the training tiles was found to be similarly unhelpful for each channel size, the suspected reason for this is the geometry inherent in SAR images. Certain features will only appear in certain positions of the images and therefore training with flipped images may encourage the model to learn nonsensical features that will never be present in SAR imagery; leading to poorer generalisation.

Model	Loss Function	Dropout Proportion	Best Val Epoch	Best Val Dice Score	Test Set Dice Score	Test Val Diff
DiceBaseline	Dice	-	42	0.649	0.632	-0.017
Dice025D	Dice	0.25	50	0.614	0.569	-0.045
Dice05D	Dice	0.5	33	0.612	**0.651**	**0.039**
BCEBaseline	BCE	-	37	0.641	0.614	-0.027
BCE025D	BCE	0.25	45	**0.665**	0.611	-0.054
BCE05D	BCE	0.5	25	0.641	0.614	-0.027
CustomBaseline	BCE + 0.1 Dice	-	41	0.636	0.595	-0.041
Custom025D	BCE + 0.1 Dice	0.25	43	0.627	0.594	-0.045
Custom05D	BCE + 0.1 Dice	0.5	37	0.624	0.604	-0.020

Table 5.5: Results of the dropout experiments. Each model was trained for 50 epochs with channel size 64, patch size 256, batch size 16, thresholded at 0.5 and used bilinear up-sampling.

The final experiments investigated the effect of different proportions of dropout, which tends to encourages an architecture to learn more sparse feature representations. It was believed that these sparse representations may improve the ability to segment the less obvious windthrow as they would regulate the model away from focusing on specific features which may be only present in the obvious examples of windthrow. Dropout was investigated with Dice Loss, BCE Loss and the previously described combination of them. Dice Loss was chosen as it had generalised best except from an extreme α weighted Tversky loss. This Tversky loss would not encourage improving the performance of the harder examples and was therefore not utilised. BCE loss was chosen as it was consistent in training but plateaued relatively early and dropout may extract further performance. The choice to investigate the combination of these loss functions was motivated mostly by curiosity. The results of these experiments are listed in Table 5.5, BCE Loss with 0.25 dropout was the only model that was able to exceed the original Dice Loss baseline on the validation set, however this did not extend to better test set generalisation. Dropout of 0.25 reduced the test set performance for each Loss function respective to their baselines, whereas 0.5 dropout matched or exceeded the test set performance respective to baseline for each loss function. Dice loss with 0.5 dropout was the model that generalised best to the test set across all experiments, achieving a dice coefficient for all test tiles of 0.65 and 0.73 when Fetterangus tiles are omitted, leading to the conclusion that relatively high dropout is beneficial for the task of windthrow segmentation.

Chapter 6

Conclusion

SARs ability to image day and night in all atmospheric conditions, combined with quick data delivery offered by Capella and ICEye's constellations enables sub-meter resolution satellite imagery to be reliably obtained almost on demand. The potential use cases for these products are vast and the commercial SAR market is expected to grow rapidly as more parties become aware of products on offer. This research demonstrates that Capella imagery combined with a U-Net inspired CNN offers a potential solution to the unsolved problem of rapid and reliable identification of windthrow; the current most reliable approach utilises Sentinel-1 and gives an overview in approximately 4 weeks. The presented methodology has the potential to reduce this to less than 1 day and the most performant model achieves a competitive dice score of 0.73 when generalised to unseen data from images with preferred *geometry* and *polarisation*.

U-Net was expected to achieve good results when applied to SAR imagery given its successes across other fields and this project confirmed this. An interesting observation is the importance of the acquisition properties of the SAR imagery. Consistent with the literature, it was found that windthrow is most readily identifiable in HH polarised images. It was further discovered that whether the fell direction of the timber was perpendicular to the range dimension of the image significantly influences the ease of manual windthrow identification and model segmentation performance, a phenomena previously unobserved. 3 of the 4 utilised HH images were acquired with this geometry and achieved competitive segmentation results whilst the remaining image had the range dimension parallel to the fell direction and the model was incapable of segmenting windthrow in this image. Another observation is that data augmentation techniques (brightness/contrast jittering; flipping) regarded to be useful for optical imagery were not as obviously beneficial for SAR. Such idiosyncrasies are important to establish to

enable researchers to best make use of these products, especially considering that they come at a cost.

Whether the convenience of mapping the extent of windthrow in less than a day outweighs the associated financial cost will ultimately be determined by the authorities that are impacted by the socio-economic consequences of windthrow. This work does however demonstrate that the methodology is viable, with the caveat that the flexibility offered by Capella when specifying acquisition parameters should be utilised. The polarisation mode should be HH and the potential fell direction should be taken into consideration when determining the orientation of the SAR image dimensions.

6.1 Future work

More sophisticated approaches for improving segmentation performance in the image with the non-preferential geometry could be explored. These include introducing the estimated fell direction as a parameter, experimenting with soft labels where the harder examples are labelled < 1, modifying U-Net to perform multi-class segmentation with multiple windthrow classes based geometry or training separate models based on image geometry. Whilst useful to explore, from a practical standpoint it makes more sense to task the constellation to acquire imagery in a geometry where windthrow is easily identifiable. An interesting research avenue would instead be to investigate the potential of stripmap images as these were not utilised for this research. Stripmap images have the largest spatial size and therefore provide the most coverage for the cost but have the coarsest resolution. Resolution was however not found to be a deciding factor in segmentation performance. Therefore if stripmap images yield competitive performance they offer the potential for rapid and widespread identification of windthrow. If resolution becomes a limiting factor, Capella are launching satellites with upgraded imaging capabilities in 2023 which may mitigate this [126].

Transfer learning (using the weights from a model trained on another dataset starting point) has been successful for classification and segmentation tasks in optical imagery where the dataset is limited. However, it has been observed that models trained on optical imagery do not transfer well to SAR [61]. An interesting experiment would be to train a model using the QuadPol images from the larger MSAW dataset and perform transfer learning for windthrow segmentation using this model. Capella images are SinglePol therefore it is recommended the QuadPol image be spliced such that only the HH channel is used for this experiment.

Bibliography

[1] Barry Gardiner, Andreas Ralf Thorsten Schuck, Mart-Jan Schelhaas, Christophe Orazio, Kristina Blennow, and Bruce Nicoll. *Living with storm damage to forests*, volume 3. European Forest Institute Joensuu, 2013.

[2] Johan ES Fransson, Andreas Pantze, Leif EB Eriksson, Maciej J Soja, and Maurizio Santoro. Mapping of wind-thrown forests using satellite sar images. In *2010 IEEE International Geoscience and Remote Sensing Symposium*, pages 1242–1245. IEEE, 2010.

[3] Áine Ní Dhubháin and Niall Farrelly. Understanding and managing windthrow. 2018.

[4] Esther Thürig, Taru Palosuo, Jürg Bucher, and Edgar Kaufmann. The impact of windthrow on carbon sequestration in switzerland: a model-based assessment. *Forest Ecology and Management*, 210(1-3):337–350, 2005.

[5] Hong S He, Bo Z Shang, Thomas R Crow, Eric J Gustafson, and Stephen R Shifley. Simulating forest fuel and fire risk dynamics across landscapes—landis fuel module design. *Ecological Modelling*, 180(1):135–151, 2004.

[6] Christophe Bouget and Peter Duelli. The effects of windthrow on forest insect communities: a literature review. *Biological Conservation*, 118(3):281–299, 2004.

[7] Rupert Seidl, Mart-Jan Schelhaas, Werner Rammer, and Pieter Johannes Verkerk. Increasing forest disturbances in europe and their impact on carbon storage. *Nature climate change*, 4(9):806–810, 2014.

[8] Alata Elatawneh, Adelheid Wallner, Ioannis Manakos, Thomas Schneider, and Thomas Knoke. Forest cover database updates using multi-seasonal rapideye

data—storm event assessment in the bavarian forest national park. *Forests*, 5(6):1284–1303, 2014.

[9] Marius Rüetschi, David Small, and Lars T Waser. Rapid detection of windthrows using sentinel-1 c-band sar data. *Remote Sensing*, 11(2):115, 2019.

[10] Craig Stringham, Gordon Farquharson, Davide Castelletti, Eric Quist, Lucas Riggi, Duncan Eddy, and Scott Soenen. The capella x-band sar constellation for rapid imaging. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 9248–9251. IEEE, 2019.

[11] Dirk Geudtner, Ramón Torres, Paul Snoeij, Malcolm Davidson, and Björn Rommen. Sentinel-1 system capabilities and applications. In *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 1457–1460. IEEE, 2014.

[12] Gordon Farquharson, Davide Castelletti, Craig Stringham, and Duncan Eddy. An update on the capella space radar constellation. In *EUSAR 2021; 13th European Conference on Synthetic Aperture Radar*, pages 1–4. VDE, 2021.

[13] Vladimir Ignatenko, Pekka Laurila, Andrea Radius, Leszek Lamentowski, Oleg Antropov, and Darren Muff. Iceye microsatellite sar constellation status update: Evaluation of first commercial imaging modes. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 3581–3584. IEEE, 2020.

[14] Confor. Storm arwen: update on impact on forests and woods. https://www.confor.org.uk/news/latest-news/storm-arwen-aftermath/, 2021. [Accessed August 17, 2022].

[15] Earth Blox. Effortless, no-code access to the power of earth observation. https://www.earthblox.io/, 2022. [Accessed August 17, 2022].

[16] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021.

[17] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing*, 152:166–177, 2019.

[18] Aaron E Maxwell, Timothy A Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, 2018.

[19] André Hollstein, Karl Segl, Luis Guanter, Maximilian Brell, and Marta Enesco. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in sentinel-2 msi images. *Remote Sensing*, 8(8):666, 2016.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Dmitry E Kislov and Kirill A Korznikov. Automatic windthrow detection using very-high-resolution satellite imagery and deep learning. *Remote Sensing*, 12(7):1145, 2020.

[22] Mihai A Tanase, Cristina Aponte, Stéphane Mermoz, Alexandre Bouvet, Thuy Le Toan, and Marco Heurich. Detection of windthrows and insect outbreaks by l-band sar: A case study in the bavarian forest national park. *Remote Sensing of Environment*, 209:700–711, 2018.

[23] Zhe Zhu, Michael A Wulder, David P Roy, Curtis E Woodcock, Matthew C Hansen, Volker C Radeloff, Sean P Healey, Crystal Schaaf, Patrick Hostert, Peter Strobl, et al. Benefits of the free and open landsat data policy. *Remote Sensing of Environment*, 224:382–385, 2019.

[24] Africa Ixmuca Flores-Anderson, Kelsey E Herndon, Rajesh Bahadur Thapa, and Emil Cherrington. The sar handbook: comprehensive methodologies for forest monitoring and biomass estimation. Technical report, 2019.

[25] Iain H Woodhouse. *Introduction to microwave remote sensing*. CRC press, 2017.

[26] Capella. Sar imagery products guide v3.4.2. https://support.capellaspace.com/hc/en-us/articles/4626115099796-SAR-Imagery-Products-Guide, 2022.

[27] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[28] Robert A Hummel, B Kimia, and Steven W Zucker. Deblurring gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38(1):66–80, 1987.

[29] Jyoti Jaybhay and Rajveer Shastri. A study of speckle noise reduction filters. *signal & image processing: An international Journal (SIPIJ)*, 6(3):71–80, 2015.

[30] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[31] Apple. Blurring an image. https://developer.apple.com/documentation/accelerate/blurring_an_image, 2022. [Accessed August 17, 2022].

[32] FirelordPhoenix. File:maxpoolsample2.png. https://computersciencewiki.org/index.php/File:MaxpoolSample2.png, 2018. [Accessed August 17, 2022].

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[34] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[35] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[36] Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE international conference on signal and image processing applications (ICSIPA)*, pages 342–347. IEEE, 2011.

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[40] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[41] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep learning and data labeling for medical applications*, pages 179–187. Springer, 2016.

[42] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[43] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[44] Abie Marshall. Informatics research review: Cloud masking in satellite multispectral images using fully convolutional networks. 2022.

[45] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[46] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[47] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[48] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[49] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[50] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[51] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[52] Ma Yi-de, Liu Qing, and Qian Zhi-Bai. Automated image segmentation using improved pcnn model based on cross-entropy. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pages 743–746. IEEE, 2004.

[53] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[54] Abie Marshall. Informatics project proposal: Segmentation of windthrow in high resolution capella sar images using fully convolutional networks. 2022.

[55] Thorvald A Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

[56] Jacob Shermeyer, Daniel Hogan, Jason Brown, Adam Van Etten, Nicholas Weir, Fabio Pacifici, Ronny Hansch, Alexei Bastidas, Scott Soenen, Todd Bacastow, et al. Spacenet 6: Multi-sensor all weather mapping dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 196–197, 2020.

[57] Shi Qiu, Zhe Zhu, and Binbin He. Fmask 4.0: Improved cloud and cloud shadow detection in landsats 4–8 and sentinel-2 imagery. *Remote Sensing of Environment*, 231:111205, 2019.

[58] Airbus. Terrasar-x product guide. https://www.intelligence-airbusds.com/files/pmedia/public/r459_9_20171004_tsxx-airbusds-ma-0009_tsx-productguide_i2.01.pdf, 2022. [Accessed August 17, 2022].

[59] Alessandro Lapini, Simone Pettinato, Emanuele Santi, Simonetta Paloscia, Giacomo Fontanelli, and Andrea Garzelli. Comparison of machine learning methods applied to sar images for forest classification in mediterranean areas. *Remote Sensing*, 12(3):369, 2020.

[60] V. Iglovikov and A. Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *ArXiv e-prints*, 2018.

[61] Corentin Henry, Seyed Majid Azimi, and Nina Merkle. Road segmentation in sar satellite images with deep fully convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(12):1867–1871, 2018.

[62] Jacob Høxbroe Jeppesen, Rune Hylsberg Jacobsen, Fadil Inceoglu, and Thomas Skjødeberg Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote sensing of environment*, 229:247–259, 2019.

[63] Bohao Huang, Daniel Reichman, Leslie M Collins, Kyle Bradbury, and Jordan M Malof. Tiling and stitching segmentation output for remote sensing: Basic challenges and recommendations. *arXiv preprint arXiv:1805.12219*, 2018.

[64] M Joseph Hughes and Robert Kennedy. High-quality cloud masking of landsat 8 imagery using convolutional neural networks. *Remote Sensing*, 11(21):2591, 2019.

[65] Alistair Francis, Panagiotis Sidiropoulos, and Jan-Peter Muller. Cloudfcn: Accurate and robust cloud detection for satellite imagery with deep learning. *Remote Sensing*, 11(19):2312, 2019.

[66] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[67] Vimala Mathew, K Ramesh, and B Prabha. Performance improvement of facial expression recognition deep neural network models using histogram equalization and contrast stretching. In *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–6. IEEE, 2021.

[68] Matthew Hanson. The open-source software ecosystem for leveraging public datasets in spatio-temporal asset catalogs (stac). In *AGU Fall Meeting Abstracts*, volume 2019, pages IN23B–07, 2019.

[69] Capella. Capella releases two new sar data products. https://www.capellaspace.com/capella-releases-two-new-sar-data-products/, 2021. [Accessed August 17, 2022].

[70] David Smart, Simon H Lee, and Edward Graham. Some aspects of the unusual climatology of storm arwen (2021). *Weather*, 77(1):31–33, 2022.

[71] Capella. Console & api. https://www.capellaspace.com/data/console-api/, 2022. [Accessed August 17, 2022].

[72] ESA. Snap download. https://step.esa.int/main/download/snap-download/, 2022. [Accessed August 17, 2022].

[73] Capella. Software compatibility with capella sar data. https://support.capellaspace.com/hc/en-us/articles/4401757536916-Software-Compatibility-with-Capella-SAR-Data, 2021. [Accessed August 17, 2022].

[74] Carsten Brockmann. Demonstration of the beam software–a tutorial for making best use of visat. In *Proceedings of the MERIS user workshop, ESA ESRIN, Frascati, Italy, published on CD-Rom, ESA SP-549*, 2003.

[75] Marco Zuhlke, Norman Fomferra, Carsten Brockmann, Marco Peters, Luis Veci, Julien Malik, and Peter Regner. Snap (sentinel application platform) and the esa sentinel 3 toolbox. In *Sentinel-3 for Science Workshop*, volume 734, page 21, 2015.

[76] SNAP Forum. Capella product format. https://forum.step.esa.int/t/capella-product-format/35216, 2022. [Accessed August 17, 2022].

[77] Walter E Westman and Jack F Paris. Detecting forest structure and biomass with c-band multipolarization radar: Physical model and field tests. *Remote sensing of environment*, 22(2):249–269, 1987.

[78] Myron Craig Dobson, Fawwaz T Ulaby, Thuy LeToan, Andre Beaudoin, Eric S Kasischke, and Norm Christensen. Dependence of radar backscatter on coniferous forest biomass. *IEEE Transactions on Geoscience and remote Sensing*, 30(2):412–415, 1992.

[79] Marc L Imhoff. A theoretical analysis of the effect of forest structure on synthetic aperture radar backscatter and the remote sensing of biomass. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):341–351, 1995.

[80] Google. Google earth pro on desktop. https://www.google.com/earth/about/versions/?gl=GB&hl=en#earth-pro, 2022. [Accessed August 17, 2022].

[81] MAXAR. Product overview. https://www.maxar.com/products, 2022. [Accessed August 17, 2022].

[82] Leif EB Eriksson, Johan ES Fransson, Maciej J Soja, and Maurizio Santoro. Backscatter signatures of wind-thrown forest in satellite sar images. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 6435–6438. IEEE, 2012.

[83] NASA. The band maths expression editor. https://seadas.gsfc.nasa.gov/help-8.2.0/desktop/ExpressionEditor.html, 2022. [Accessed August 17, 2022].

[84] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[85] Rasterio. Rasterio: access to geospatial raster data. https://rasterio.readthedocs.io/en/latest/, 2022.

[86] GDAL. Gdal. https://gdal.org/, 2022. [Accessed August 17, 2022].

[87] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[88] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[89] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[90] Markus Müller. fake-geo-images. https://github.com/up42/fake-geo-images, 2020. [Accessed August 17, 2022].

[91] Ekaba Bisong. *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress, 2019.

[92] GDAL. Geotransform tutorial. https://gdal.org/tutorials/geotransforms_tut.html, 2022. [Accessed August 17, 2022].

[93] Reza Safabakhsh and Shahram Khadivi. Document skew detection using minimum-area bounding rectangle. In *Proceedings International Conference on Information Technology: Coding and Computing (Cat. No. PR00540)*, pages 253–258. IEEE, 2000.

[94] Herbert Freeman and Ruth Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.

[95] Capella. Sar imagery products format specification. https://support.capellaspace.com/hc/en-us/articles/5607458273940-SAR-Imagery-Products-Format-Specification, 2021. [Accessed August 17, 2022].

[96] Wikipedia. Index notation: Two-dimensional arrays. https://en.wikipedia.org/wiki/Index_notation#Two-dimensional_arrays, 2022. [Accessed August 17, 2022].

[97] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *J. Mach. Learn. Res.*, 22(82):1–6, 2021.

[98] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[99] Matt Martineau, Patrick Atkinson, and Simon McIntosh-Smith. Benchmarking the nvidia v100 gpu and tensor cores. In *European Conference on Parallel Processing*, pages 444–455. Springer, 2018.

[100] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[101] Harry Zhang. The optimality of naive bayes. *Aa*, 1(2):3, 2004.

[102] Jan Salomon Cramer. The origins of logistic regression. 2002.

[103] Qi Cheng, Pramod K Varshney, and Manoj K Arora. Logistic regression for feature selection and soft classification of remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 3(4):491–494, 2006.

[104] scikit-learn developers. Strategies to scale computationally: bigger data, incremental learning. https://scikit-learn.org/0.15/modules/scaling_strategies.html#incremental-learning, 2022. [Accessed August 17, 2022].

[105] scikit-learn developers. sklearn.naivebayes.gaussiannb. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html, 2022. [Accessed August 17, 2022].

[106] scikit-learn developers. sklearn.linearmodel.sgdclassifier. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html, 2022. [Accessed August 17, 2022].

[107] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2682–2690, 2019.

[108] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[109] Milesi Alexandre. Pytorch implementation of the u-net for image semantic segmentation with high quality images. https://github.com/milesial/Pytorch-UNet, 2022. [Accessed August 17, 2022].

[110] Haris Iqbal. Latex code for making neural networks diagrams. https://github.com/HarisIqbal88/PlotNeuralNet, 2022. [Accessed August 17, 2022].

[111] Zhibin Liao and Gustavo Carneiro. On the importance of normalisation layers in deep learning with piecewise linear activation units. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[112] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[113] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[114] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[115] Torch Contributors. Adam. https://pytorch.org/docs/stable/generated/torch.optim.Adam.html, 2022. [Accessed August 17, 2022].

[116] Torch Contributors. Transforming and augmenting images. https://pytorch.org/vision/main/transforms.html, 2022. [Accessed August 17, 2022].

[117] Torch Contributors. Dropout. https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html, 2022. [Accessed August 17, 2022].

[118] Ekachai Phaisangittisagul. An analysis of the regularization between l2 and dropout in single hidden layer neural network. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 174–179. IEEE, 2016.

[119] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 31, 2018.

[120] Severstal. Severstal: Steel defect detection. can you detect and classify defects in steel? https://www.kaggle.com/code/bigironsphere/loss-function-library-keras-pytorch/data, 2022. [Accessed August 17, 2022].

[121] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pages 379–387. Springer, 2017.

[122] Dan López-Puigdollers, Gonzalo Mateo-García, and Luis Gómez-Chova. Benchmarking deep learning models for cloud detection in landsat-8 and sentinel-2 images. *Remote Sensing*, 13(5):992, 2021.

[123] Haoran Lu, Yifei She, Jun Tie, and Shengzhou Xu. Half-unet: A simplified u-net architecture for medical image segmentation. *Frontiers in Neuroinformatics*, 16, 2022.

[124] Kaili Cao and Xiaoli Zhang. An improved res-unet model for tree species classification using airborne high-resolution images. *Remote Sensing*, 12(7):1128, 2020.

[125] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[126] Capella. Capella space unveils next generation satellite with enhanced imagery capabilities and communication features. https://www.capellaspace.com/capella-space-unveils-next-generation-satellite-with-enhanced-imagery-capabilities-and-communication-features/, 2022. [Accessed August 17, 2022].