An evaluation of deep learning semantic segmentation for land cover classification of oblique ground-based photography

by

Spencer Rose

B.Sc., University of Alberta, 1999
B.A., University of Victoria, 2003
M.A., University of Western Ontario, 2005

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

An evaluation of deep learning semantic segmentation for land cover classification of oblique ground-based photography

by

Spencer Rose

B.Sc., University of Alberta, 1999
B.A., University of Victoria, 2003
M.A., University of Western Ontario, 2005

Supervisory Committee

Dr. Yvonne Coady, Supervisor

Department of Computer Science

Dr. Eric Higgs, Departmental Member

School of Environmental Studies

Abstract

This thesis presents a case study on the application of deep learning methods for the dense prediction of land cover types in oblique ground-based photography. While deep learning approaches are widely used in land cover classification of remote-sensing data (i.e., aerial and satellite orthoimagery) for change detection analysis, dense classification of oblique landscape imagery used in repeat photography remains undeveloped. A performance evaluation was carried out to test two state-of the-art architectures, U-net[1] and Deeplabv3+[2], as well as a fully-connected conditional random fields model[3] used to boost segmentation accuracy. The evaluation focuses on the use of a novel threshold-based data augmentation technique, and three multi-loss functions selected to mitigate class imbalance and input noise. The dataset used for this study was sampled from the Mountain Legacy Project (MLP) collection, comprised of high-resolution historic (grayscale) survey photographs of Canada's Western mountains captured from the 1880s through the 1950s[4] and their corresponding modern (colour) repeat images. Land cover segmentations manually created by MLP researchers were used as ground truth labels. Experimental results showed top overall F1 scores of 0.841 for historic models, and 0.909 for repeat models. Data augmentation showed modest improvements to overall accuracy (+3.0% historic / +1.0% repeat), but much larger gains for under-represented classes.

Contents

Supervisory Committee ii

Abstract iii

List of Figures vi

List of Tables viii

Terminology xi

Acknowledgements xii

1 Introduction 1
- 1.1 Motivation and Application . . . 1
- 1.2 Research Objectives . . . 2
- 1.3 Research Contributions . . . 3
- 1.4 Thesis Structure . . . 3

2 Background 5
- 2.1 Overview . . . 5
- 2.2 Land Cover Classification and Change Detection . . . 5
- 2.3 The Mountain Legacy Project (MLP) . . . 8
- 2.4 Summary . . . 15

3 Related Work 17
- 3.1 Overview . . . 17
- 3.2 Taxonomy . . . 17
- 3.3 Traditional Image Segmentation Methods . . . 19
- 3.4 Deep Learning Methods . . . 25
- 3.5 Data Augmentation . . . 35
- 3.6 Loss Functions . . . 37
- 3.7 Summary of Current Study in Context . . . 39

4 Methodology 41
- 4.1 Overview . . . 41

4.2 Objectives . . . 41
4.3 Dataset . . . 42
4.4 Image Preprocessing . . . 44
4.5 Multi-class Semantic Segmentation . . . 49
4.6 Conditional random fields . . . 57
4.7 Evaluation Criteria and Metrics . . . 61
4.8 Summary . . . 67

5 Implementation **68**
5.1 Overview . . . 68
5.2 Experimental Setup . . . 68
5.3 U-net implementation . . . 73
5.4 Deeplabv3+ implementation . . . 77
5.5 Conditional Random Fields . . . 81
5.6 Summary . . . 82

6 Experimental Results and Analysis **83**
6.1 Overview . . . 83
6.2 Key Findings . . . 84
6.3 Model Training/Validation . . . 84
6.4 Experiment 1: Data Augmentation . . . 87
6.5 Experiment 2: Loss Functions . . . 90
6.6 Experiment 3: Conditional Random Fields . . . 92
6.7 Evaluation of Model Sensitivity . . . 93
6.8 Evaluation of Model Latency . . . 97
6.9 Limitations . . . 98

7 Conclusion and Future Work **99**
7.1 Conclusion . . . 99
7.2 Future Work and Challenges . . . 101

A Mountain Legacy Project Image Datasets **103**

B Experimental Results and Discussion **110**
B.1 Overview . . . 110
B.2 Model Training/Validation Losses . . . 113
B.3 Experiment 1: Data Augmentation . . . 116
B.4 Experiment 2: Loss Functions . . . 121
B.5 Experiment 3: Conditional Random Fields . . . 123
B.6 Evaluation of Model Sensitivity . . . 124
B.7 Evaluation of Tile Reconstruction . . . 127

C Example Segmentation Maps **128**

Bibliography **143**

List of Figures

2.1 Example historic image from DST.B showing LCC.A [5] and LCC.B [6] categorization schemes summarized in Table 2.1. LCC.B is slightly different from LCC.A in that B-MW is split into separate broadleaf forest (B) and mixedwood forest (MW) types, and similarly H-S is split into upland herbaceous (H) and upland shrub (S) types. These types apply to both repeat and historic images. Note that images from the test dataset, DST.C use LCC.B for categorization. (Image Ref: DST.B.H.2.4 in Appendix A) . 10

2.2 Pixel probabilities for land cover categories of combined historic capture images (scheme LCC.A) showing extracted compared with augmentation and merged databases. Dashed line indicates ideal class balance. See Table 5.2 for details. 16

2.3 Pixel probabilities for land cover categories of combined repeat capture images (scheme LCC.A) showing extracted compared with augmented data. Dashed line indicates ideal class balance. See Table 5.3 for details. . 16

3.1 Example segmentations obtained by Jean, et al.[5] using SVM-based piecewise segmentation. White represents forest, gray represents non-forest, and black represents uncategorized pixels. The MCC value (defined in section 4.7.2) is given for each image. 23

4.1 Image preprocessing and model training pipeline. 42

4.2 Model application and CRF post-processing pipeline. 43

4.3 Sample extraction (left) and augmented (right) tiles showing segmentation mask overlays. Randomized affine deformations to the image data were calibrated to ensure natural landscape features were not overly distorted. 47

5.1 U-net architecture [1] 75

5.2 Deeplabv3+ network architecture with ResNet101 backbone. 79

6.1 Comparison of output segmentations for images DST.A.H.2.1 (top), DST.B.H.2.4 (middle) on DLAB.H (historic) models, and image DST.B.R.2.6 (bottom) on DLAB.R (repeat) models. 92

6.2 Comparison of output segmentations for resampled versions of images DST.B.R.2.6 (top) and DST.B.H.2.3 (bottom) on model DLAB.H.2.3. From left, prediction masks shown are at full-sized (1.0×), scaled to 0.5×, and scaled to 0.25×. 94

6.3 (Top) Extractions from images DST.A.H.2.2 and DST.B.H.2.1 showing markings and/or scratches from glass plate negatives; (Middle) Ground-truth mask extraction; (Bottom) Segmentation output from model DLAB.H.8.3. 96

C.1 DST.A.H.2.2 Image shows poor background visibility due to cloudiness. Scratches and marks visible across the middle of the photo. Bottom-right corner foreground pixels not classified. Moderate representation of minor classes [WT, S-I, RA]. DST.A.H.2.3 Image shows some visibility occlusion due to cloudiness and rain. Photo is grainy. Moderate representation of minor classes. [WT, S-I, H-S]. 129
C.2 DST.A.H.2.4: Some visibility occlusion due to cloudiness and rain. Emulsion edges are washed out (not classified). Photo is grainy. [S-I]. DST.B.H.2.1: Photo catalog markings visible on corners and along edges. Bottom corner regions contain foreground pixels not classified. [WL, H-S]. Output segmentations shown from DLAB.H.2.5. 130
C.3 DST.B.H.2.3: Photo image is distorted by plate inconsistencies. Emulsion edges are washed out with a wide gradient. Photo is also grainy. Significant border around the image is not classified. [WL, WT, H-S, S-I]. DST.B.H.2.5: Some visibility occlusion due to cloudiness and rain. Emulsion defects at edges. Photo is grainy. Large bottom-centre foreground region is not classified. [WL, WT, H-S, S-I] 131
C.4 DST.B.H.2.7: Relatively clear visibility. Some scratches and marks visible across the top left of the photo. [WT, WL, S-I, H-S, RA]. DST.B.H.2.2: Photo catalog markings visible on corners and along edges. Cloud shadows add significant photometric variation to forest cover. [WL, H-S, RA]. 132
C.5 DST.A.H.2.1. Image shows some visibility occlusion in distant background. Cloud shadows add photometric variation to forest cover. Bottom-right corner foreground pixels not classified. [S-I, RA]. Output segmentations shown from DLAB.H.2.5. DST.B.H.2.4: Cloud shadows add significant photometric variation to forest cover. Bottom-right corner foreground pixels not classified. [WL, WT, H-S, S-I]. DST.B.H.2.6: Scratches and marks visible across the middle of the photo. Some visibility occlusion due to cloudiness and rain, or due to emulsion defects. Middle-left region is not classified. [WL, WT, H-S, S-I]. 133
C.6 Example historic model segmentation results form images DST.C.H.1.1, DST.C.H.1.2 134
C.7 Example historic model segmentation results form images DST.C.H.1.7, DST.C.H.1.8 135
C.8 Example historic model segmentation results for images DST.C.H.1.5, DST.C.H.1.6 136
C.9 Example results using conditional random fields (CRF) for post-processing images. DST.A.H.2.2 shows improved classification of distant mountains and correction of NC classified regions; DST.B.H.2.7 shows significant correction of foreground minor class RA (regenerating area) and WT (wetlands) regions. 137
C.10 Example repeat model segmentation results for images DST.A.R.2.2, DST.A.R.2.3 138
C.11 Example repeat model segmentation results for images DST.A.R.2.4, DST.B.R.2.5 139
C.12 Example repeat model segmentation results for images DST.B.R.2.3, DST.B.R.2.5 140
C.13 Example repeat model segmentation results for images DST.A.R.2.1, DST.B.R.2.4, DST.B.R.2.6 141
C.14 Example repeat model segmentation results for images DST.B.R.2.2, DST.B.R.2.7 142

List of Tables

2.1 Land cover types used in LCC.A[5] and LCC.B[6]/LCC.C[7] classification schemes based on the Alberta Vegetation Inventory (Version 2.1.1., 2005) and Mcdermid et al. (2009)[8]. Note that types B/MW and H/S from LCC.B/C have been merged in LCC.A as B-MW and H-S, respectively, to improve class imbalance and simplify classification. 12

2.2 Summary of the MLP raw image datasets (DST.A[5], DST.B[6], DST.C[6]) 14

5.1 MLP training and testing image databases. Databases are created from tile extractions of training images in datasets DST.A.1 and DST.B.1. . . . 70

5.2 Class probability distributions for the historic capture databases (extracted versus combined DST.A, DST.B, merged with grayscaled repeat captures, augmented). Weights calculated using Equation 4.17. Total number of H-Mrg samples: 24,396. LCC.A categorization. 70

5.3 Class probability distributions for the repeat capture databases (extracted versus augmented database for combined DST.A, DST.B). Weights calculated using Equation 4.17. Total number of R-Aug samples: 12,038. LCC.A categorization. 71

5.4 U-net Architecture - Detailed specification. 76

5.5 Deeplabv3+ Architecture - Detailed specification. 80

6.1 Model training and validation losses (L_{CE}, L_{DSC}, L_F) for historic captures (logarithmic scale) and validation accuracy. Each model corresponds to different gradient weightings of loss functions, as summarized in Table 6.7. See also Sections 4.5 and 4.7.1. 85

6.2 Model training and validation losses (L_{CE}, L_{DSC}, L_F) for repeat captures (logarithmic scale) and validation accuracy. Each model corresponds to different gradient weightings of loss functions, as summarized in Table 6.8. See also Sections 4.5 and 4.7.1. 86

6.3 Experiment H.1: Results summary for data augmentation (historic captures). 87

6.4 Confusion matrices for models DLAB.H.1.1 (top left), DLAB.H.1.2 (top right), and DLAB.H.1.3/2.1 (bottom right) trained on H-Ext, H-Aug and H-Mrg databases, respectively (see Table 5.1). Class accuracy results for DLAB.H.1.3/2.1 are also shown. "Accuracy" is the overall pixel accuracy of predictions with respect to the ground-truth. $cAvg$ values equal the unweighted mean per class, and $wAvg$ equal the support-weighted mean per class. Support total: 449,533,967 pixels. For detailed analysis, see Appendix B.3.1). 88

6.5 Summary of accuracy metrics for extracted (H-Ext), augmented (H-Aug) and merged (H-Mrg) training databases. See Section 4.7.2 for metric definitions. 89
6.6 Confusion matrices for models DLAB.R.1.1 and DLAB.R.1.2 trained on R-Ext and R-Aug databases, respectively (see Table 5.1). Support total: 281,240,583 pixels. For detailed analysis, see Appendix B.3.2). 89
6.7 Experiment 2.1 Results Summary: Loss functions (historic captures). The asterisk ($*$) indicates inclusion of the loss in the gradient computation, and the w superscript indicates class weights were applied to the CE loss. Multi-loss computations used an equal weighting of the loss value (i.e. $L_{total} = \alpha L_{CE} + \beta L_{DSC} + \gamma L_F$, where $\alpha, \beta, \gamma \in [0, 1]$, see Section 4.5.4.4). Model trained on H-Mrg database for up to 20 epochs. Support is 449,533,967 pixels. 91
6.8 Experiment 2.2 Results Summary: Loss functions (repeat captures). The asterisk ($*$) indicates inclusion of the loss in the gradient computation, and the w superscript indicates class weights were applied to the CE loss. Multi-loss computations used an equal weighting of the loss value (i.e. $L_{total} = \alpha L_{CE} + \beta L_{DSC} + \gamma L_F$, where $\alpha, \beta, \gamma \in [0, 1]$ - see Section 4.5.4.4. Model trained on R-Aug database for up to 20 epochs. Support is 449,533,967 pixels. 91
6.9 Experiment 3: Conditional Random Fields Filter (historic models) 93
6.10 Summary of training and inference times for historic and repeat models. U-Net network totaled approximately 28M trainable parameters; Deeplabv3+ network totaled approximately 59M. Database sizes are listed in Table 5.1 97

A.1 Image and segmentation mask training dataset DSC.A.1. Segmentation masks use land cover classes (see LCC.A classes in Table 2.1) employed in a Landsat-based map of the same area (see Jean et al.[5]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca. 104
A.2 Image and segmentation mask test dataset DSC.A.2 (Testing). Segmentation masks use land cover classes (see LCC.A classes in 2.1) employed in a Landsat-based map of the same area (see Jean et al.[5]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca. 106
A.3 Image and segmentation mask training dataset DSC.B.1. Segmentation masks use land cover classes ((see LCC.B classes in 2.1) employed in a Landsat-based map of the same area (see Fortin et al.[6]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca 107
A.4 Image and segmentation mask test dataset DSC.B.2. Segmentation masks use land cover classes ((see LCC.B classes in 2.1) employed in a Landsat-based map of the same area (see Fortin et al.[6]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca 108

A.5 Image and segmentation mask test dataset DSC.C. Segmentation masks use land cover classes ((see LCC.C classes in 2.1) employed in a Landsat-based map of the same area. All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca 109

B.1 Summary of experiments and model descriptors. 111
B.2 Measured gaps between training and validation losses (L_{CE}, L_{DSC}, L_F) for historic and repeat models. See also Sections 4.5 and 4.7.1. 115
B.3 Experiment H.1: Results summary for data augmentation (historic captures). 117
B.4 Accuracy metrics and confusion matrix for model DLAB.H.1.1 trained on the extracted database (H-Ext, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 118
B.5 Accuracy metrics and confusion matrix for model DLAB.H.1.2 trained on the augmented database (H-Aug, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 118
B.6 Class accuracy and confusion matrix for model DLAB.H.1.3/2.1 trained on the merged database (H-Mrg, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 119
B.7 Summary of accuracy metrics for extracted (H-Ext), augmented (H-Aug) and merged (H-Mrg) training databases. See Section 4.7.2 for metric definitions. 120
B.8 Class Accuracy and confusion matrix for model DLAB.R.1.1 trained on extracted database (R-Ext, see Table 5.1). Support total: 281,240,583 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 120
B.9 Class Accuracy and confusion matrix for model DLAB.R.1.2 trained on augmented database (R-Aug, see Table 5.1). Support total: 281,240,583 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 121
B.10 Experiment 3: Conditional Random Fields Filter (historic captures) . . . 123
B.11 Class accuracy and confusion matrix for model DLB-H.2.3 + CRF trained on the merged database (H-Mrg, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. 124

Terminology

Dominant class	Semantic class where the aggregate class pixel count is greater than the ideal balanced pixel probability.
Image segmentation	The partitioning of a digital image into non-overlapping regions.
Land cover	Observed vegetation, geomorphic or anthropogenic cover on the Earth's surface.
Minor class	Semantic class where the aggregate class pixel count is lower than the ideal balanced pixel probability.
Object detection	Classification of different regions in an image as countable instances of object classes, often using bounding boxes to mark each object.
Oblique photographs	Land-based photographs with an oblique angle of incidence.
Repeat photography	The technique of detecting landscape change using multiple photos from the same location and viewpoint captured at different points in time.
Scene parsing	Semantic segmentation of an image where a significant number of pixels are classified as stuff rather than object instances.
Segmentation map	Pixel-level classification mask for a given image generated through a segmentation method.
Semantic segmentation	The assignment of a semantic label to each pixel in a digital image.

Acknowledgements

I could not have written this thesis without a great deal of patience, support and assistance from those around me.

I would like to first thank Dr. Yvonne Coady, who first convinced me to take up CS graduate studies, and whose knowledge, fearless optimism and wise guidance helped me see this work through to the end. I feel truly grateful we crossed paths.

I would next like to thank Dr. Eric Higgs, the founder of the Mountain Legacy Project, for his expertise and steady encouragement throughout the process. It was Eric who launched my thesis collaboration with MLP, and introduced me to my research partner, James Tricker. James has provided keen feedback at every stage of this work and was generous with his knowledge – I always look forward to our regular chats.

I would like to thank Stewart Arneil, at the UVic Humanities Computing and Media Centre, for first connecting me with the Mountain Legacy Project, and for his many years of guidance and honest opinion.

I would like to acknowledge my other Mountain Legacy Project colleagues. In particular, I'd like to thank Prof Emerita Mary Sanseverino, who has provided me with some of the most important background and context for this study, along with very helpful feedback and support.

I would also like to thank my CS professors at UVic, and in particular, Dr. Kwang Moo Yi for his excellent introduction to the subject of deep learning for computer vision, and Dr. Sean Chester for his high standard of instruction and for teaching me how to better program with modern architectures.

I am also grateful to Prof Emeritus Kellogg Booth, whom I have had the great pleasure of working with over the years, and whose support has made my studies possible.

Finally, I would like to express my deepest love and gratitude to my long suffering partner Tanya and daughter Holly for their incredible support, patience and understanding during this journey.

Dedicated to Tanya and Holly

Chapter 1

Introduction

1.1 Motivation and Application

The documenting of historical landscape changes is fundamental to our scientific interpretation of ecological dynamics, and informs practices of ecosystem intervention and restoration [9]. Through qualitative and quantitative analysis of observed shifts in the Earth's (bio)physical land cover, we are better able to gauge the extent or trajectory of landscape scale ecosystem change and human impact. Changes in landscape patterns are furthermore central to the study of wide-ranging problems, including climatic variability[10], agricultural development[11], flooding and flood management[12], deforestation and forest succession[13], vegetation phenology[14], ecosystem biodiversity[6], forest fires[15], and insect infestations[16], to name a few[17].

Such diverse research employs a broad scope of methods to identify, describe, and quantify variation in vegetation cover, land use, and geomorphic processes. A key methodological approach is to evaluate the spatial extent of variation across multitemporal images [18], which can be used to support inferences about the nature of the change and its driving factors[19]. Conventional methods use remote sensing techniques to extract land cover data from satellite and aerial orthoimagery[20] (images orthogonal to the Earth's surface) – e.g., vegetation cover or geomorphological features – that can be quantified using segmentation and classification at both landscape and class levels. High-resolution aerial and satellite image time series (SITS)[21] data have proliferated over the past four decades[22], not only expanding our imagery of the Earth's surface[23], but also becoming widely available for new research and analysis of natural landscapes and geomorphic processes [24] [25][26] [27].

Repeat photography – the practice of taking multiple photos from the same location and viewpoint, but at different points in time [28] – offers an alternative, but under-utilized approach to detecting and analyzing landscape change. Unlike remote-sensing techniques, repeat photography looks at variation in time-series oblique images captured using (typically) ordinary cameras at ground level. Such landscape photography predates technologies developed for capturing aerial and satellite imagery by several decades, offering far greater depth to its historical record of ecological change [18][19]. Photographs are also abundantly available, less costly and, for some topological studies, provide a higher spatial resolution of sloped terrain[6].

At the same time, recent developments in deep learning semantic segmentation, such as the fully convolutional neural network (FCN)[29] and its variants, have advanced techniques for detecting landscape change. Semantic segmentation, a "dense prediction" task that assigns a semantic label (e.g. land cover category) to each pixel in a digital image, can be used to map land cover types at image resolution. Several deep learning methods have been evaluated for use of remotely sensed imagery [30] [25] [31] [32] [33] [34] [35] [36]. Similar methods for conventional photographic imagery at the landscape scale, however, remain largely unproven[34]. The adoption of deep learning methods applied to repeat photography for land cover analysis and change detection at scale require demonstrable successes and accessible tools.

1.2 Research Objectives

1. To determine the effectiveness of deep learning methods for land cover classification of oblique ground-based imagery through a performance evaluation of two state-of-the-art segmentation networks. The dataset used in this study consists of historic and modern images from the extensive Mountain Legacy Project (MLP) collection[37], a publicly accessible and spatially extensive database for studying the ecosystem dynamics of mountain landscapes. Based at the University of Victoria, the MLP research group investigates landscape, ecological and cultural change in the mountains of western Canada. This thesis builds on previous work by MLP researchers[18][5][38][6] to advance new methods of analysis for mountain research.

2. To develop and evaluate a multi-faceted approach to address specific dataset challenges – limited ground truth data; class imbalance (dominant and under-represented semantic classes); high-variance input data (strong variation in the appearance of training images) – that impact the performance of segmentation neural networks.

3. To evaluate the effectiveness of using a fully-connected conditional random fields (CRF) model[39] [3] in post-processing to boost performance.

1.3 Research Contributions

This work makes the following research contributions:

1. A performance evaluation of two state-of-the-art deep convolutional neural networks (DCNNs), U-net[1] and Deeplabv3+[2], for the classification of land cover types for oblique ground-based photography. This is the first study of deep learning land cover classification methods to use images from the Mountain Legacy Project collection[37]. Performance is evaluated based on a comparison with available manually-classified ground-truth segmentation maps created by MLP researchers[5][12][6].

2. A novel approach to data augmentation, specifically developed to mitigate severe class imbalance in semantic segmentation.

3. A performance evaluation of the use of conditional random fields as a post-processing step to improve segmentation maps predicted by the models.

1.4 Thesis Structure

The following is a brief outline of the thesis. In Chapter 2, key context for this thesis is introduced by way of a general discussion of the research problem of land cover classification and a rationale for the study. In Chapter 3, a review of recent research literature related to the application of deep learning to semantic segmentation and complex scene parsing is presented, as well as techniques for data augmentation and different loss functions both used to address MLP dataset challenges of limited ground truth data, class imbalance and noisy data. A short review of the use of conditional random fields (CRFs) to boost performance is also included. In Chapter 4, the methodological approach of this study is presented, including formal definitions of the problem of multi-class semantic segmentation using convolutional neural networks and conditional random fields. The framework used in evaluating experimental results is also presented. In Chapter 5, the proposed implementation and optimization of the U-net and Deeplabv+ architectures, and CRF model are described. The experimental results are then presented in Chapter 6, which include an evaluation and analysis. Chapter 7 concludes with a short summary

and discussion of future research directions in deep learning for land cover classification and segmentation.

Chapter 2

Background

2.1 Overview

In this chapter, the key contextual themes of the thesis are introduced by way of a general discussion of the research problem, a justification for the study, and a summary of its aims and challenges. Section 2.2 introduces the problem of land-cover classification for oblique landscape repeat photography. While land cover classification and analysis predominantly use satellite and aerial imagery extracted from remote sensing data, repeat photography for oblique ground imagery offers an abundant but under-utilized resource to map landscape composition, quantify change, and understand landscape legacies and ecological history[18][6][40]. Section 2.3 presents an overview of the Mountain Legacy Project (MLP) collection, a vast archive of historic and modern repeat landscape photography captured in the mountainous regions of Western Canada. Images sampled from the MLP collection, along with corresponding manually-created ground truth segmentation maps, form the dataset for this study. Some of the important characteristics and challenges of the dataset for land cover classification are also discussed.

2.2 Land Cover Classification and Change Detection

2.2.1 Approaches Based on Remote Sensing Data

Growth in remote sensing data used in the analysis of landscape change over the past forty years has fueled demand for fully-automated pixel-scale classification and segmentation methods to maximize speed, throughput and performance in the analysis of natural landscapes and geomorphic processes [26][34]. Deep convolutional neural networks (DCNNs), in particular, have recently enabled ecologists to leverage spatial context to

identify objects and segment landscapes in high-resolution remotely sensed imagery[26], leading to the development of many new high performance techniques for the automatic mapping of land cover types in orthoimagery [30] [25] [31] [32] [33] [41] [34] [35] [36]. The field of remote sensing scene classification remains an active and challenging task that will likely continue to see performance improvements through use of deep learning.

2.2.2 Approaches Based on Repeat Photography

While methods for land cover quantitative research are predominantly applied to aerial and satellite remote sensing data, repeat photography using ordinary oblique ground-level imagery presents a valuable and less-explored class of alternative techniques. In the natural and environmental sciences, repeat photography is a long-established approach to documenting and detecting landscape change[18][13][28].

Unlike the near-uniform scale of orthoimagery, the oblique vantage point of ground-level photography creates a continuous variation in scale that can complicate the mapping of landscape features in a photograph to absolute spatial coordinates[18][13]. Such spatial referencing of images is often needed to quantify changes[19]. Remote sensing methods also rely on multispectral and hyperspectral data to classify different species of vegetation[23]), whereas landscape photography is limited to the visible spectrum, which reduces the discriminating information [42]. Existing automatic segmentation and classification approaches for aerial and satellite imagery therefore do not work for oblique photographs.

However, numerous studies have explored a range of approaches to accurately estimate relative land cover composition in repeat ground photographs. Geographic information system (GIS) analytical software applied to oblique photo-pairs has been used to categorize and quantify vegetation change[18] [13]. Software such as the WSL Monoplotting Tool[43] and the Image Analysis Toolkit (IAT)[38] have been used to georeference or orthorectify repeat photographs (i.e., map the image spatial data to a known geographic coordinate system corrected for topographic variation) allowing for the extraction and analysis of ecological information at uniform scale [19] [44]. Recent advances in "virtual" photos[7] (VP) – virtual 3D representations of terrain in a given ground photograph – have allowed for the generatation of georeferenced viewsheds based on the photo's land cover segmentation mask. Other methods have also been developed for land cover estimation that segment and classify oblique images without coordinate or spatial transformations [12] [6]. Fortin, et al.[6], for example, showed oblique photographs can provide

reasonable models of landscape composition based on a comparison of land cover proportions quantified by oblique photographs of the Canadian Rocky Mountains, with those quantified by satellite images.

2.2.3 Use of Oblique Ground-level Photography in Spatial Analysis

Repeat photography has a long history as a source of qualitative information on long-term ecological processes, that has played a complementary role to remote sensing change detection technique [28]. However, the spatial data extracted from oblique ground-based images presents a number of specific advantages to aerial and satellite orthoimagery. The historical record for remote sensing data is relatively short compared with ordinary photography. Aerial imagery largely developed in the 1930s and 1940s[18] and remote sensing satellite imagery first became available in 1972 with Landsat[45], whereas ground-based photography predates both by several decades. The MLP collection of over 120,000 historic photographs from the Geological Survey of Canada, for example, spans the 1880s through the 1950s[4]. Oblique landscape photography therefore offers far greater depth to the record of ecological change than that available using remote sensing imagery databases. Oblique photographs can also present a higher degree of detail in land cover maps, such as in the representation of steep and narrow landscape features [6], whereas with remote sensing orthomagery, which forces a projection of angled surfaces, textural detail is reduced [46]. Details of sloped terrain, such as mountain rock, ice and snow, in particular, are therefore better captured in oblique photographs[38], while narrow landscape features, such as wetland valleys, present with higher resolution[4].

Given the predominance of remote sensing data in change detection analysis, it is not surprising that comparatively few land cover classification methods have been developed for oblique ground-level photography. Jean, et al. (2015) [5] proposed a machine learning method to classify landscape images from the MLP collection by land-cover class. The work generated coarse resolution binary segmentation maps for forest versus non-forest classification, which can be used with post-classification change detection methods, such as aggregate pixel count comparisons. Another patch-based approach using convolutional neural networks was proposed by Bayr and Puschmann (2019) [47] in the analysis of natural landscape repeat photography. This method produced similarly coarse resolution segmentations of multiple land-cover classes. It is considered one of the first studies to investigate the classification of multiple vegetation types (i.e. woody, herbaceous and grassy vegetation) in landscape photographs. The proposed approach was able to detect clear in trends in woody vegetation change, despite the low resolution and noisy segmentation maps generated. Buscombe and Ritchie (2018) [34] studied the application of CRFs and DCNNs for the semantic segmentation of landscape-level

and high-vantage imagery. Their proposed hybrid semantic segmentation method combines DCNN image classification of small regions in the imagery, with the fine-grained localization of fully-connected conditional random fields (CRFs)[3] for pixel-level classification [34]. They also implemented a CRF-based method to generate DCNN training and testing data using minimal manual supervision.

Recent work in fully convolutional neural networks (FCN)[29] [48] [49] [1] [50] [51] [2], has advanced the state-of-the-art in semantic segmentation that has opened up new opportunities for end-to-end pixelwise landscape classification. FCN-based networks[42] have been adapted for the computer vision task of scene parsing: the dense prediction of visual scenes that feature predominantly amorphous or "stuff-like" object classes, such as barren rock, water, mixed vegetation, or sky, and which are inherent to oblique landscape photographs. Harbaš, et al. (2018)[42], for example, developed roadside scene parsing for the navigation of autonomous vehicles that was among the first neural networks developed to extract features strictly from the visible spectrum (i.e. ordinary camera images) to detect different forms of vegetation. A full discussion of recent work in deep learning semantic segmentation is presented in Chapter 3.

2.3 The Mountain Legacy Project (MLP)

Since it was created in 1998, the Mountain Legacy Project (MLP)[37] at the University of Victoria has supported numerous research initiatives exploring the use of repeat photography to study ecosystem, landscape, and anthropogenic changes. MLP hosts the largest systematic collection of mountain photographs, with over 120,000 historic photos captured by the Geological Survey of Canada, the Dominion Topographic Survey, and other surveys, from the late 19th to mid-20th centuries to create topographic maps of the Canadian Cordillera [52] [53] [54], and other locations. These original photographs were preserved on large format glass plates, mainly at Library and Archives Canada and the British Columbia Archives. Over the years, MLP field researchers have digitized these historical survey images, as well as captured approximately 9,000 modern repeats, which are used to investigate landscape-level change. The MLP collection has been a rich resource for cross-disciplinary research studies [18][4] [38] [12] [6] [7], and a proving ground for new software tools and algorithms used in the classification and analysis of landscape images [55] [56] [5] [19] [38] [7].

2.3.1 MLP Image Dataset

This study used MLP collection images sampled from both its historic survey and modern repeat photographs. The original unprocessed historic (grayscale) images consist of very high-resolution digital scans of glass plate photographic negatives originally held in special facilities at LAC's Gatineau Preservation Facility (Gatineau, Québec) [12]. The fixed lens box cameras used by the surveyors to expose images on glass plate negatives (approximately 102 mm by 152 mm) resulted in photos of both high physical resolution and longevity [5]. Glass plates were scanned with high-quality scanners at 2,400 dots per inch (dpi) and stored as 16-bit TIFF grayscale images with lossless compression. The historic images used in this study were previously downsampled in size to be within the range of 19 to 39 megapixels. Historic images were also cropped and inverted for analysis. Each repeat and historic image pair also has to be manually aligned using specialized software tools [38]. The modern repeat images in the dataset were photographed by MLP field researchers using high-quality digital cameras at the historic survey sites located in British Columbia and Alberta. Modern repeat images use three-channel RGB format and match their paired historic image in size. The modern photographs were comparatively clear and sharp, due to the high-quality camera equipment [12].

All of the photographs used in this study had corresponding ground-truth segmentation maps (or masks) of land cover classes at the original image resolution, created using manual techniques. These masks are used both as training and validation targets for the DCNN model, as well as ground truth to evaluate test output segmentations. Manual segmentation typically involves the use of field experts who classify pixel classes in a photograph using a graphics tablet for high precision [4]. A typical method is to draw polygonal contours around specific landscape elements identified visually as belonging to a specific class [47]. The manual labor required to create segmentation maps of sufficient accuracy and detail for land cover change detection and analysis is painstaking, costly and time-consuming[7][5].

Segmentation masks were created for two previous MLP studies, with a combined count of 120 high-resolution image pairs with masks (i.e., a very small fraction of the full MLP collection). The first segmentation dataset (labelled **DST.A)**, previously used by MLP researcher Frederic Jean, et al. in an earlier image segmentation study [5], consisted of 60 historic/repeat image pairs and corresponding mask segmentations. Land cover regions were manually classified using a custom software tool called "Image Labeler," developed at the University of Victoria [12]. The second dataset (labelled **DST.B**) of 46 image pairs and masks was initially developed by MLP researchers Fortin, et al. (2015) for biodiversity change at locations in the Willmore Wilderness Park[4], and for a study on landscape-level composition estimates of oblique photographs[6]. DST.B images were

selected for manual segment classification based on criteria of clarity and sharpness, with no exposure or focus issues, and minimal foreground, to maximize usable pixels [6]. Images were also selected that capture a view from valley to peak, to give the full range of possible habitat classes, and which are geographically dispersed to give a wide-range of land cover variation [6]. A third dataset, DST.C[7], was also combined with some of the images from DST.A and DST.B to test the proposed network architectures. Details about the combined datasets are summarized in Table 2.2 and in Appendix A. All images and segmentation masks are publicly available under a Creative Commons License, and can be freely downloaded from an open access repository [5] [6].

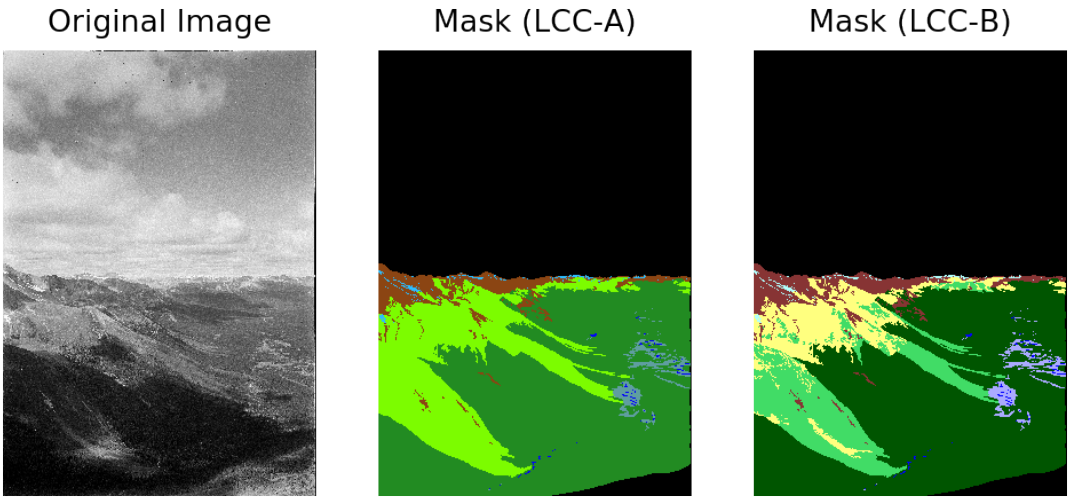


FIGURE 2.1: Example historic image from DST.B showing LCC.A [5] and LCC.B [6] categorization schemes summarized in Table 2.1. LCC.B is slightly different from LCC.A in that B-MW is split into separate broadleaf forest (B) and mixedwood forest (MW) types, and similarly H-S is split into upland herbaceous (H) and upland shrub (S) types. These types apply to both repeat and historic images. Note that images from the test dataset, DST.C use LCC.B for categorization. (Image Ref: DST.B.H.2.4 in Appendix A)

Segmentation masks used in DST.A and DST.B follow two different land cover classification (LCC) schemes labelled LCC.A and LCC.B, respectively, and summarized with type descriptions in Table 2.1. Both schemes use comprehensive well-defined categories[5][6] based on the Alberta Vegetation Inventory (Version 2.1.1., 2005) and Mcdermid et al. (2009)[8]. DST.A uses an eight-category system for classifying vegetation and non-vegetation land cover types, plus a "not-classified" (NC) category used to omit image regions from the downstream analysis. Vegetation types include coniferous forest (C), regenerating areas (RA), and two merged types: broadleaf/mixedwood forest (B-MW), and upland herbaceous/shrub (H-S). Non-vegetation types include barren rock (S-G-R – i.e. sand, gravel, rock), water (WT), wetland (WL), and permanent snow/ice (S-I). LCC.B is slightly different from LCC.A in that B-MW is split into separate broadleaf

forest (B) and mixedwood forest (MW) types, and similarly H-S is split into upland herbaceous (H) and upland shrub (S) types. These types apply to both repeat and historic images. Note that images from the test dataset, DST.C use LCC.B for categorization.

Land cover types used for classification

LCC.A	LCC.B/C	Category
		Not categorized (NC)
		Broadleaf forest (B) Greater than 75% broadleaf trees.
		Mixedwood forest (MW) 26 - 74% broadleaf trees, the rest largely composed of coniferous trees and/or shrubs.
		Coniferous forest (C) Greater than 75% coniferous trees.
		Upland Herbaceous (H) Less than 25% shrub cover, less than 6% tree cover
		Upland Shrub (S) Greater than 25% shrub cover, less than 6% tree cover.
		Regenerating Area (RA) Fire boundaries clearly identified or clear sign of recent timber harvesting.
		Barren rock (S-G-R) Soil, sand, gravel, or rock.
		Wetland (WL) "Wet" or 'aquatic moisture regime.
		Water (WT) 6% or greater flowing or standing water.
		Snow/Ice (S-I) Permanent ice and snow.

TABLE 2.1: Land cover types used in LCC.A[5] and LCC.B[6]/LCC.C[7] classification schemes based on the Alberta Vegetation Inventory (Version 2.1.1., 2005) and Mcdermid et al. (2009)[8]. Note that types B/MW and H/S from LCC.B/C have been merged in LCC.A as B-MW and H-S, respectively, to improve class imbalance and simplify classification.

2.3.2 Segmentation Ground-truth Errors

Manual segmentation introduces ground-truth errors that needed to be incorporated into the evaluation of the proposed automated approach (see section 4.7.3). MLP historic photography, in particular, presented difficult challenges for both manual and automated segmentation [18] [5] [12] [6] – some of which are common to oblique, ground-based photographs, such as perspective distortion, and noise, while others are specific to the dataset, for example, the lack of colour information. The following lists the primary segmentation errors, based on the analysis presented in Rhemtulla et al. (2002) [18], Taggart-Hodge, et al. (2016) [12], and Fortin et al. (2019)[6].

1. **Foreground/Background Representation**: Perspective distortion in ground-based photos can greatly enlarge the image foreground, leading to an over-representation of foreground pixels compared to those in the background. Pixels in the foreground of an oblique image represent a much smaller area than pixels in the background, and therefore foreground pixels were typically omitted (i.e., not classified) during manual segmentation [18] [6]. Furthermore, objects in the foreground can obscure other objects of interest in mid- and background (for example, tree growth, or a close range rock shelf), and are therefore also omitted. In the case of DST.B, Fortin et al. (2019) selected images for their study that did not contain large foreground areas.

2. **Category Granularity**: Granularity describes the coarseness (broadness) or fineness (narrowness) of the scope of a land cover class. Categorization schemes LCC.A and LCC.B use a granularity selected to align with existing systems that enable the comparative measurement of trends in vegetation change [12]. However, this chosen granularity can result in less accurate classification of land cover categories at a species level. This has proven particularly the case for the single-channel historic images [12]. The use of comparatively broad categories (see Table 2.1) was favoured over finer categories that require more effort for the expert classifier.

3. **Photometric/Spectral**: Historic photos lack colour information and therefore provide limited spectral data for segmentation and classification [12]. In Taggart-Hodge, et al. (2016), researchers converted the modern images to grayscale, so pairs of single channel images can be compared. Test results on four separate MLP image pairs showed significant classification errors for some land cover textures, for example, broadleaf/mixedwood forest (B-MW) and grassland herbaceous/shrub (H-S) [12].

Dataset	Capture	Images	Categorization
DST.A A.1	Historic (Grayscale)	60	LCC.A
	Repeat (Colour)	60	LCC.A
DST.B A.3	Historic (Grayscale)	46	LCC.B
	Repeat (Colour)	46	LCC.B
DST.C A.5	Historic (Grayscale)	8	LCC.B
	Repeat (Colour)	2	LCC.B

TABLE 2.2: Summary of the MLP raw image datasets (DST.A[5], DST.B[6], DST.C[6])

4. **Class Boundaries** Transitions between class regions (interpreted segments) in an image can be visually difficult to define, which results in an ambiguous interpretation of pixel classes for some regions.

5. **Tracing Error**: Multiple researchers are typically involved in the image segmentation process, each of whom may contribute slightly different interpretations of pixels classes [12]. Classes that pose particular challenges include regenerating areas (RA), broadleaf/mixedwood (B-MW) and herbaceous/shrub (H-S). Historical images have also been consistently more challenging to segment than repeat images using manual methods, due to flaking, scratches and other imperfections on the glass plate negatives that obscure the image. These deviations were also mitigated during the segmentation process through clear protocols, supervised training, and comparisons of results [12]. The use of high precision drawing tablets also helped to reduce tracing error.

2.3.3 Dataset Challenges

Preliminary diagnostic tests on the MLP collection images revealed two key dataset challenges for land cover classification and analysis: (1) severe class imbalance; and (2) high variance in the appearance of visual features. The pixel class distributions of the combined extraction dataset (DST.A and DST.B) helped to identify minor (underrepresented) and dominant (over-represented) semantic classes. These class profiles are summarized in Figures 2.2 and 2.3, as well as Tables 5.2 and 5.3. The between-class imbalance shown in the charts highlights three deficiencies of the dataset that typically prove problematic for supervised learning: (1) Undersampling of critical minor classes; (2) Oversampling of the non-categorized class; and (3) Limited labeled data. For the historic capture images, severe undersampling of minor classes B-MW, WL, WT, S-I ($< 1\%$) was found, as well as oversampling of the noncategorized (NC) class ($> 50\%$), which accounts for more than half of the pixels in the training samples. repeat capture images showed a very similar profile with the same minor and dominant classes.

The second problematic characteristic of the dataset is the wide variation in the visual appearance of landscape features. This variation is due to photographic conditions (e.g., illumination and weather conditions), occlusions (e.g., due to fog or cloud cover), and image noise or quality. Noise can include photographic artefacts, digitization artefacts, scratches, localized emulsion defects, "salt and pepper" artefacts, and other occasional but visible damages to the glass plate negatives for the historic photos [5]. Noise errors appear far less problematic for the repeat image datasets. The lack of colour information in the historic photographs also limits the complexity of features for classification[5][12], as mentioned above. Historic and repeat images were furthermore created using different camera technologies, resulting in different pixel intensities and textural variation [5].

2.4 Summary

While deep learning approaches for the mapping of land cover types have been the overriding choice for remote-sensing image analysis[23], similar approaches for oblique repeat photography, a comparatively untapped source of landscape data, have been given far less focus. New and robust deep learning segmentation architectures developed for such applications as complex scene parsing, robotic vision, and autonomous driving, however, offer new possibilities for the analysis of oblique repeat photography, an abundant but under-utilized resource. In this chapter, some of the trends in the use of deep learning for land cover image classification were introduced. As well, the dataset for this study – the Mountain Legacy Project (MLP) collection, a vast archive of historic and modern repeat landscape photography captured in the mountainous regions of Western Canada – was described along with key dataset challenges for this study: limited annotated data, class imbalance, and input variation.

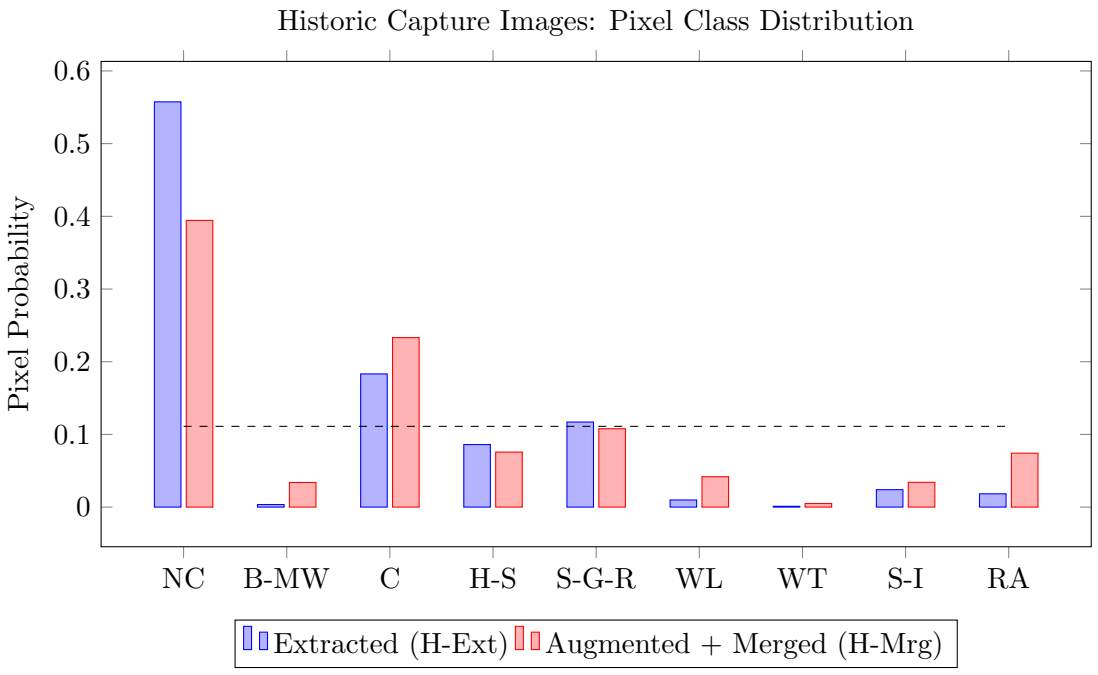


FIGURE 2.2: Pixel probabilities for land cover categories of combined historic capture images (scheme LCC.A) showing extracted compared with augmentation and merged databases. Dashed line indicates ideal class balance. See Table 5.2 for details.

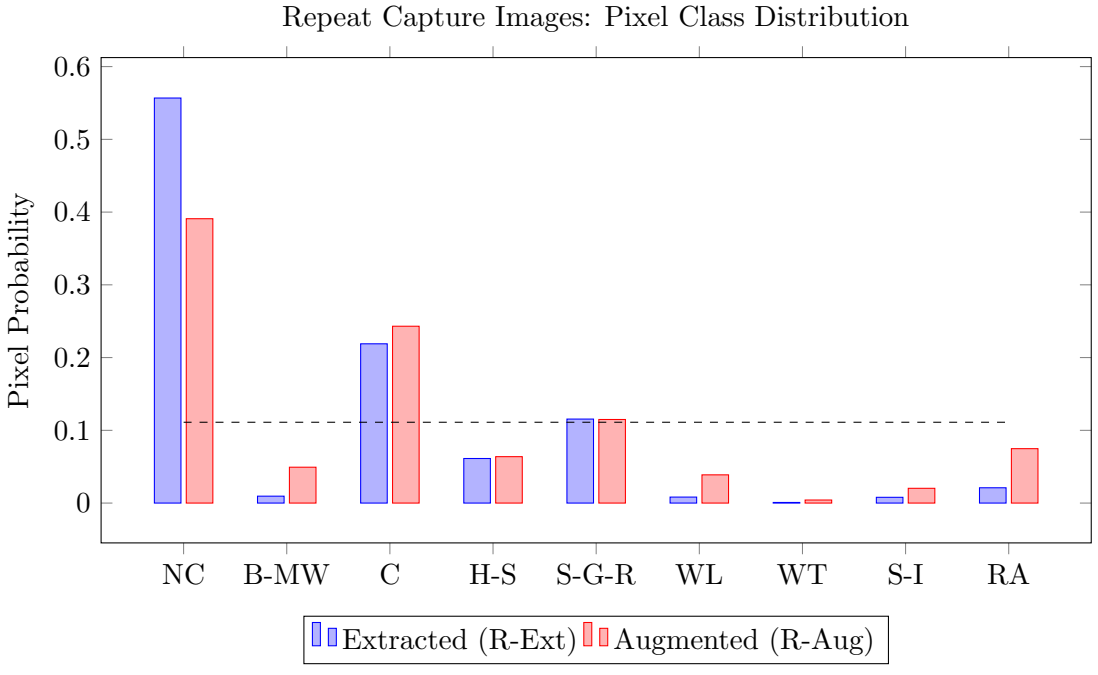


FIGURE 2.3: Pixel probabilities for land cover categories of combined repeat capture images (scheme LCC.A) showing extracted compared with augmented data. Dashed line indicates ideal class balance. See Table 5.3 for details.

Chapter 3

Related Work

3.1 Overview

In this chapter, we present the main literature review covering the theory and methods applied to semantic segmentation of complex visual scenes. This review is intended to properly situate the objectives of this study within the vast research literature on the subject. We begin with some preliminaries in section 3.2.1 by identifying and describing the history and nature of the general problem of semantic segmentation. Section 3.3 surveys some of the traditional image segmentations algorithms and techniques, many of which predate deep learning methods. In section 3.4, we survey the evolution of deep convolutional neural networks (DCNNs) from image classification to semantic segmentation, and discuss some of the recent segmentation architectures. In sections 3.5 and 3.6 we review various methods of data augmentation and DCNN loss functions previously used in segmentation problems, respectively. Section 3.7 concludes by restating the objectives of this thesis in the context of the preceding research.

3.2 Taxonomy

3.2.1 Semantic Segmentation

Semantic segmentation combines the process of image segmentation and object recognition (classification) with the goal of parsing a digital image into objects or regions that are meaningful and analyzable. Whereas basic image segmentation partitions an image into non-overlapping regions or segments that are distinguishable to human perception [57], semantic segmentation is a far more difficult task, as it further assigns each "pixel" – the atomic element of a digital image – a semantic class or label. (Here we understand

digital images to be comprised of two-dimensional arrays of elemental pixels, where each pixel has an intensity value, normally recorded as an 8-bit digital number that ranges from 0 to 255, and location address in the image. For photographic images of natural scenes, the intensity value represents the measured illumination in a given wavelength band reflected from an object.) Semantic segmentation is among the most challenging and intensely studied problems in computer vision. Its diverse applications span robotic vision [58] [59], biomedical imaging [1] [60] [50] [61], autonomous driving [62] [63] [64] [42] [65], and, related to this thesis, ecological change detection and land cover classification [66] [67] [5] [34] [47].

Unlike image classification, which interprets the image as an object, semantic segmentation performs "pixel-level classification" using "dense prediction," in that the classifier accurately interprets the input digital image at its highest granularity, therefore, performing both recognition and segmentation of two or more classes. Basic (non-semantic) image segmentation, in contrast, forms regions or structures without classification. This form of segmentation is therefore not well-defined, as many possible segmentations may be equally appropriate [68].

Semantic segmentation is also distinct from object detection, in that the latter distinguishes separate instances of a given object class, such as "tree" or "house" [69][68]. In object detection, the surrounding stuff or material, such as the "sky" or "forest" typically serves as context for the detection of more salient objects [69]. Semantic segmentation, in contrast, assigns a given class to every pixel; that is, rather than being individuated as instances, objects are clumped into common regions.

Stuff and objects form the two primary types of visual objects of interest in image classification and segmentation [69] [70]. Stuff, as defined in Forsyth, et al. [71], is the "homogeneous or repetitive pattern of fine-scale properties [that have] no specific or distinctive spatial extent or shape," whereas "an object has a specific size and shape". Hence, objects roughly correspond to well-formed entities in a visual scene. Alexe, et al. [72], developed a measure of the "objectness" of a pixel-region, and argued that objects have at least one of three following characteristics: (1) well-defined closed boundary in space; (2) distinct from its surroundings; (3) salient or unique. Stuff, in contrast, consist of pixel clusters that lack well-defined boundaries, or can be considered amorphous objects.

In natural imagery, the appearance of objects and stuff varies significantly according the environment in which they appear. Different imaging conditions result from variation in spatial distancing between the camera sensor, vantage point, and the target scene, as well as different object scales. Given these conditions, the minimum expectation for segmentation algorithms is to partition images into (a) relatively homogeneous (i.e.

exhibit statistical regularities or patterns) and (b) semantically significant groups of pixels, which may be further processed into meaningful objects[73]. With more complex segmentation methods (e.g. convolutional neural networks), we further account for spatial and semantic content of the entire image [26].

3.2.2 Scene Parsing (Natural Landscapes)

Scene parsing can be considered a variant of semantic segmentation that primarily interprets pixels of the amorphous "stuff" in images. Hence, what distinguishes scene parsing from other semantic segmentation methods is that a significant number of pixels are classified under stuff classes (e.g., barren rock, forest, or sky) rather than object classes (e.g., an individual person or tree). By predicting the label (class), location and shape of visual element, scene parsing provides a complete interpretation of every pixel in a scene. The goal of scene parsing methods is therefore well aligned with the classification of different land-cover categories in landscape photography, given the exclusively amorphous characteristics inherent to oblique landscape photographs that lack well-defined objects – i.e., landscape segmentation has no object classes. Though boundaries between classes are ideally smooth (not ragged) and spatially accurate [74], regions within a natural landscape scene do not necessarily have the well-defined, closed boundaries of individual objects, but interface in compound curves, and form class regions of wide-ranging sizes and shapes. In land cover studies of oblique photos, some classes allow definitions that are distinct, stable and constant such as those between barren rock and overhead sky, or between water and surrounding vegetation. However, boundaries become fuzzier between other classes, such as between broadleaf, mixed-wood, and coniferous forests [12].

Using the preceding terminology, we next consider different methods for image segmentation, semantic segmentation and, by extension, scene parsing.

3.3 Traditional Image Segmentation Methods

Image segmentation is the task of partitioning an image into salient regions of homogeneous characteristics (e.g. texture), and has been a core research area in computer vision for several decades. Segmentation methods are considered important functions for higher-level applications, including semantic segmentation and scene understanding and parsing [75]. Here we define traditional methods as those that do not use neural networks (DCNNs), but which make heavy use of domain knowledge, or which may use local features extracted from the image itself [68].

The following summarizes some of the common traditional methods found in the research literature, and grouped into the following approaches: (1) Local Feature-based Methods; (2) Texture-based Methods; (3) Machine Learning Methods; and (4) Conditional Random Fields. Given the large number and variety of techniques developed for image segmentation, this list is not intended to provide an exhaustive survey.

3.3.1 Local Feature-based Methods

Local features (or feature descriptors) describe an image in a way that reduces it to its most important information, and can be used to break down images into regions based on criteria such as similarity and homogeneity. The most common local features used for segmentation include (1) Pixel colour (e.g., three features for RGB, three features for HSV (hue, saturation, value) and one feature for the grayscale); (2) Histogram of oriented gradients (HOG)[76], which extracts the gradient and orientation of image edges; (3) Histogram of Local Binary Patterns (HLBP)[77], which thresholds the neighborhood of each pixel and encodes it as a binary number (also invariant to grayscale intensity and rotation); (4) Scale-invariant feature transform (SIFT)[78], which extracts image keypoints invariant to image scale and rotation; (5) Speeded-up robust features (SURF)[79], another scale- and rotation-invariant detector and descriptor; (6) Bag-of-visual-words (BOV) histograms [80], which count occurrences of certain patterns within a patch of the image; (7) Superpixels[81], which defines features in neighbouring pixels that share properties of contour, texture, brightness and continuity evaluated using information-theoretic measures.

Thresholding methods, for example, use an intensity or colour histogram to classify pixels for different regions of an image [82], and are quite simple and effective in differentiating foreground and background regions [83]. Edges are another useful feature for segmentation. Boundaries between two adjacent regions typically indicate a discontinuity of pixels (i.e., a difference in pixel colour, intensity, texture) that allow for edge detection, while a second processing step normally links edges into chains which correspond better with boundaries in an image. Edge-based segmentation represents a large group of methods – e.g., Elder, et al. (1996) [84] and Gevers and Smeulders (1997)[85], and Sun, et al. (2007)[86], and which include edge thresholding [87] and edge relaxation [88]. Graph-based methods use local features that admit segmentation by treating images as fully-connected graphs, where each node corresponds to an image pixel. Although several variants exist[89] [90], the basic method of segmenting the graph is to break links that cross between segments, such that edges are selected from the graph based on whether a given pixel's neighboring pixels are connected by undirected edges. Weights on each edge can be used to measure the dissimilarity between pixels. Local

features are also used in hybrid segmentation methods that can involve machine learning classifiers, neural networks, or other methods[83].

3.3.2 Texture-based Methods

Texture, which has been a topic of considerable research since the 1960s [91], has played a founding role in methods developed for image segmentation and classification [92]. In computer vision, texture analysis aims at representing image texture in a model that is invariant to different visual appearances of the same textural class, and which can characterize and discriminate image textures [93]. This typically involves extraction of intrinsic features from the original dimensional space that describe texture information. The defining characteristic of texture representations is to pool information extracted locally and uniformly from the image, by means of multiple local features that encode texture images for comparison and matching with other textures [94]. Textural segmentation methods are wide-ranging, and include using fractal dimensions with an unsupervised k-means clustering [95], analysis of local spectral histograms [96] [97] to discriminate region appearances and localize region boundaries; and component analysis of Gabor transforms[98] to extract textural features [99]. Texture-based methods are most suitable where traditional thresholding, or other local feature techniques cannot be effectively used [100].

3.3.3 Conditional Random Fields

Conditional Random Fields (CRF) are a class of discriminative probabilistic graphical models used for structured prediction [101]. Introduced by Lafferty, et al. (2001)[102], CRFs incorporate information about the neighboring context of a given input, such as an image pixel, to allow for the modelling of complex interactions between output variables and observed features, such as the interrelations between pixels. By factorizing the probability distribution over different labeling of the random variables, CRFs allow for compact representations and efficient inference [57]. A formal definition of CRFs is presented in section 4.6.

A CRF can be represented as a graph with nodes corresponding to the image pixels, and edges connecting those node pairs and weighted by a pairwise cost. In image processing, CRFs predominantly take on two forms: grid and fully-connected. The grid CRF[101], also known as the TextonBoost model developed by Shotton, et al. [101], considers pairs of pixels that are its immediate neighbors, and therefore only propagates limited information about the overall image context. With fully-connected CRFs (or Dense Random Fields), such as the CRF developed by Krähenbühl and Koltun (2011)

[39], each pixel pair has defined one pairwise term regardless of their mutual distance, and therefore all pairs of variables directly are connected by pairwise potentials [103]. Fully-connected CRFs frequently employ a highly efficient inference algorithm called the "mean field approximation"[39]. Here, the pairwise edge potentials are defined by a linear combination of Gaussian kernels in an arbitrary feature space. The method has been shown to substantially improve segmentation and labeling accuracy [39] [3] [104] [2] [105]. CRFs are also particularly good at modelling the decision boundaries between different pixel classes [105], however these approaches can also be time-consuming and computationally expensive [106].

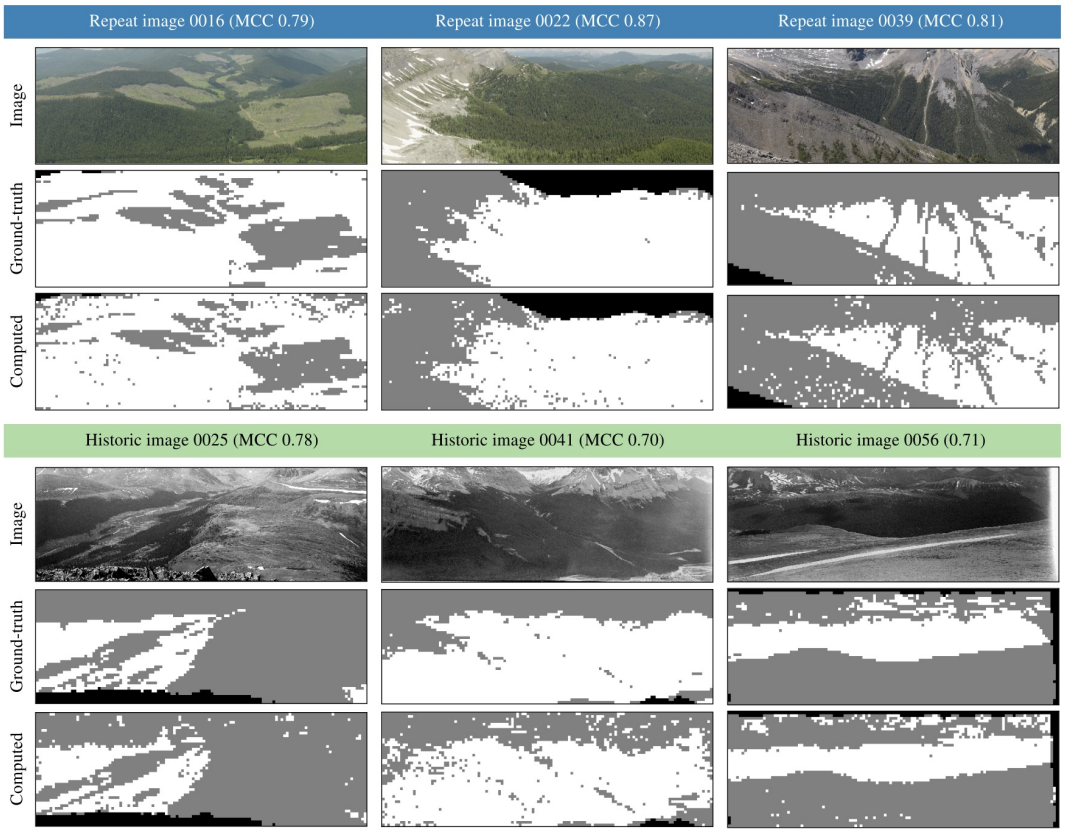
3.3.4 Machine Learning Methods

Machine learning in the form of supervised and unsupervised classifiers has been extensively applied to the problem of segmentation. Unlike techniques that use local features, machine learning attempts to learn about the structure of the image content, and to explicitly model the problem with prior knowledge [107]. For example, while edge detection methods typically require extra processing to form the continuous borders of objects out of the generated image segments, machine learning methods can instead weight the classification of nearby pixels as part of the same object label to automatically form those boundaries.

Random Decision Forests [108] is a supervised learning algorithm that applies ensemble learning using multiple randomized decision tree classifiers, and combines the results into a single classifier. Ensemble learning techniques can train each classifier on a random subspace of the feature space using "bagging", which involves training the trees on random subsets of the training set. Schroff, et al. (2008) [109], for example, used Random Forest classifiers trained on local features to improve performance of pre-learned nearest neighbour matching class models.

K-means clustering, developed by Hartigan and Hartigan (1975) [110] is a form of unsupervised learning that has been adapted for image segmentation. The k-means algorithm randomly places a finite number of centroids in the feature space, and assigns each data point to the nearest centroid, successively shifting the centroid to the center of the cluster. This process continues until a preset threshold is reached. Theiler and Gisler (1997) [111] developed a variant of the standard k-means algorithm that uses both spectral and spatial properties of an image to form regions based on a small number of categories. Mobahi, et al. (2010) [75] treat image segmentation as an energy problem by correctly quantifying only the necessary information needed to encode natural images. This is

FIGURE 3.1: Example segmentations obtained by Jean, et al.[5] using SVM-based piece-wise segmentation. White represents forest, gray represents non-forest, and black represents uncategorized pixels. The MCC value (defined in section 4.7.2) is given for each image.



done using an agglomerative clustering process to find the shortest descriptor length to encode all textures and boundaries in the image.

Support Vector Machines (SVM) have also been adapted for segmentation methods, for example in land-cover analysis [66] and scenery image segmentation [112]. For binary segmentation, the SVM constructs a hyperplane to separate data-points by maximizing the margin from the hyperplane to the two classes. In the multi-class case, a hyperplane between every pair of classes or between each class and the rest classes is created to separate the data-points based on a one-against-one or one-against-all strategy. SVM classifiers learn using only a few critical data-points (the support vectors), that help to eliminate a large number of redundant training samples.

Researchers Jean, et al. (2015) [5] proposed an SVM-based method to segment historic (grayscale) and repeat (colour) mountain landscape images in the MLP collection by land-cover class. The approach extracted square image patches from high resolution photographs, and used SVM classifiers to resolve vegetation classes for each patch. The

training data consisted of feature vectors extracted based on the Histograms of Local Binary Patterns (HLBP) and Histograms of Oriented Gradients (HOG) descriptors, chosen for their invariance to pixel intensity. Colour images were gray-scaled to ensure single-channel consistency in the extracted features. The ground-truth class for a given patch was based on the highest pixel count within the corresponding block in the manual segmentation image. Classification used two separate SVM classifiers trained on the data based on the rationale that each classified separated types of imaging sensors (i.e. glass plate emulsion vs. digital camera). The researchers tested their classifiers on both binary (forest vs non-forest) and multi-category classification of the repeat photos. Given that this grid-based approach only resolves the majority class of a patch, the classification resolution was substantially lower than the image resolution, resulting in coarse output segmentations, as shown in Figure 3.1. The authors note that several thousand feature vectors were extracted to properly encode large texture variations within each image due to perspective projection and illumination conditions (e.g. cloud shadows on the landscape). The MLP dataset proved more resistant to segmentation than other texture databases (e.g. Brodatz[113], CUReT[114]) due to challenging textural intra-variation, and, as mentioned, the piece-wise classification approach results in coarse and noisy segmentations.

3.3.5 Limitations

Many traditional image segmentation methods based on local features frequently produce unsatisfactory results. This stems from two forms of error: under-segmentation and over-segmentation – i.e., generating too few or too many segments – which results in objects that do not accurately represent real-world features [115]. Pixel-based methods, for example, tend to be limited in regard to relative scale, context, and perform poorly on fuzzy or smooth transitions between regions [116]. A limitation of threshold-based classification is that information from surrounding pixels – which may help in correctly identifying the target pixel's class – is largely disregarded. Simple machine-learning classifiers such as Random Forest, SVM, and nearest neighbour clustering also face limitations in that feature extraction requires a good deal of domain-specific engineering or hand-crafted design, and are often time-consuming, in that a large training dataset is often needed for moderate performance. Simple classifiers are also non-flexible in that they do not adapt to different datasets. Given these methods directly extract low level features, such as edges and corners, they lack invariance to image perturbations [117]. In the case of CRFs, they tend to require costly inference computations, which practically deters from modeling complex label dependencies [118].

More generally, traditional methods are not able to model the complex interrelations of an image because they are not able to extract overall semantic information [119]. In the computer vision and image processing community, many researchers have therefore turned to convolutional neural networks, which are concerned with extracting more complex and global (i.e. full-image) features. Convolutional layers downsample the spatial resolution of images while expanding the depth of their feature maps[120], which results in much lower-dimensional and more useful representations of images than traditional methods. These representations tend to be more complex and contextually relevant than those defined by local features [121].

3.4 Deep Learning Methods

3.4.1 Convolutional Neural Networks

For more than two decades, deep learning neural networks – artificial neural networks with several hidden layers – have made significant advances in tackling complex problems across many knowledge domains. Following the landmark work of Krizhevsky, et al. (2012) [122] to develop a deep convolutional neural network for ImageNet, a new paradigm of deep learning algorithms and methods has emerged [120]. In particular, deep convolutional neural networks (DCNNs) have advanced performance in computer vision tasks such as image classification and semantic segmentation well beyond previous state-of-the-art machine learning and engineered methods.

The precursor to the convolutional neural network emerged in 1979 with the neocognitron, a self-organizing multi-layer neural network designed for shift-invariant pattern recognition [123]. The neocognitron was inspired, in part, by studies of cell structures in the mammalian visual cortex by Nobel prize-winning neurologists Hubel and Wiesel in the 1960s [124]. Their findings showed certain "simple" neural cells fire when specifically-oriented edges are recognized in the receptive field, whereas "complex" cells, which have larger receptive fields, respond to different regions of the visual field. These two types of cells were combined to form a cascading model for early pattern recognition [124].

During the 1970s and 1980s, several researchers independently discovered that multi-layer architectures can be trained by simple stochastic gradient descent, where gradients are computed using the backpropagation ("backward propagation of errors") procedure [125]. Building on the work of Rumelhart, et al. (1985) [126], LeCun, et al. (1989) [127] demonstrated in the 1980s that stochastic gradient descent (SGD) via backpropagation proved an effective supervised training method for a new class of convolutional neural networks that extend the neocognitron. SGD consists of computing the outputs and

the errors for a given batch of input examples, then computing the average gradient for those examples, and adjusting the parameter weights of the network accordingly. The technique proved remarkably effective at optimizing parameter weights of networks in comparison with far more elaborate techniques [120].

Although much interest in DCNN applications emerged in the 1990s (for example, in document recognition[128]), the development of other machine learning methods, such as SVM, stalled new developments in deep learning until the mid-2000s [129]. With advances in GPU computing power, the availability of labelled data, and algorithmic advances, neural networks were again brought to the forefront of visual tasks. The multiple layers of nonlinear information processing of DCNNs solve the formidable challenge of feature extraction and transformation, and have made significant advances in pattern analysis, classification, and more recently semantic segmentation.

The basic structure of deep-learning architecture is a stack of several learnable filter modules that successively transform input data to increase both the selectivity and the insensitivity (or "invariance") of the representation [120]. Selectivity here refers to the neural network's ability to select features of the input that are important for discrimination (i.e. maximizes intra-class similarity and inter-class variability); whereas invariance refers to the network's output stability given perturbations and appearance changes in the input (i.e. minimizes intra-class variability and inter-class similarity). Deep learning networks involve what is called "end-to-end learning" of hierarchically abstracted features of an image, that is, successive extractions from simple, low-level shapes through to complex, high-level objects.

A critical advantage over conventional machine learning approaches common to all DCNNs is that they do not require "feature-engineering" or "feature extraction" – the process of transforming or preprocessing the raw input data into features that better represent the underlying problem to the predictive models. Hence, both conventional image filters and neural networks use feature extraction to detect low-level features from the image. However, unlike image filters, which are engineered and fixed, DCNN layers act as learnable filters trained using the backpropagation algorithm [120] [130]. In contrast to previous machine learning approaches, which typically require the user to manually design these discriminative features, neural networks learn features automatically from data when trained with a backpropagation algorithm, such as Stochastic Gradient Descent (SGD).

The architecture of an convolutional neural network typically consists of the following three types of filter layers:

1. **Convolutional layer (C Layer):** Convolutional layers serve the important function of extracting features from the input data to form shift and distortion invariant feature maps [130]. C layers convolve the input space using learnable filters that apply weighted convolution computations across small, multiple regions in the data called receptive fields [120]. Convolutional filters are comparable to spatial filters used in digital image smoothing [130], but instead of smoothing, they serve to identify common and progressively salient visual features such as straight edges, simple colors, and curves, and significantly surpass traditional local feature descriptors in performance [131]. The output of each C layer in the DCNN is referred to as a "feature map", which delineates the locations in the original image for where certain low level features appear. As each feature map unit is connected to local patches in the feature maps of the previous layer through a set of weights, these filters are both inherently localized to their receptive fields (attuned to minute details of the image) and translation-invariant [131]. Feature maps then pass through a non-linear activation function – i.e. typically a rectified linear unit (ReLU), defined as $f(z) = max(z, 0)$. ReLU is a computationally efficient function that effectively operates as a half-wave rectifier [120], such that the activation is thresholded at zero. Analogous to neural activation in biological systems, the non-linear activation function is crucial for modelling complex non-linear dynamical systems. As the rectified outputs from one network layer are passed as inputs to successive layers, the activations gradually represent higher level features. With increasing network depth, the convolutional filters take on larger and larger receptive fields, thus responding to a larger field of information of the original input volume. These responses correspond to higher levels of feature abstraction (i.e. features have more complex shapes), and this progression eventually results in the encoding of the entire image. The result is a distortion of the input space that allows different classes of input data to be separable [120].

2. **Pooling or downsampling layer (P layer)**: The P layer in a DCNN downsamples or pools output feature maps to reduce the spatial resolution and broaden the receptive field. Pooling helps to both improve network invariance to small input perturbations and translations [120] by filtering noisy activations from a lower layer, effectively abstracting feature maps in a receptive field with a single pooled value [48]. Pooling also helps to reduce the number of network parameters (and hence computational cost) to learn the data. However, this broader context comes at the cost of reduced spatial resolution. The P layer normally follows a convolutional layer.

3. **Fully connected layer (FC layer):** The fully connected layer is identical to the traditional multi-layer feed-forward neural networks – in that, it is fully connected

to all output features or units in the previous layer – and serves as the final operator to determine which features of the input activation most correlate to a particular class. Both the input and output of an FC layer are one-dimensional feature vectors of size n, where n is the number of object classes. FC layers are used in image or object classification, and not typically used in semantic segmentation networks [131].

3.4.2 DCNN-based Semantic Segmentation

One of the earliest deep learning approaches to semantic segmentation was the pioneering work of Grangier, et al. (2009) [118], who developed a greedy layer-wise learning algorithm for experiments on the MSRC dataset from Microsoft Research in Cambridge. The method showed promising advantages to CRF-based methods (state-of-the-art at the time), including efficient use of SGD, and the capacity to model arbitrary complex functions from the RGB input image. A few years later, Farabet, et al. (2013) [132] developed a scene parsing system using convolutional layers to make predictions based on raw pixels from image patches taken at multiple scales. The raw input data was transformed through a Laplacian pyramid, and each scale was fed through a three-stage convolutional network, which produces a set of feature maps. The overall technique effectively slides a classification network around an input image to classify the presence of an object within each sliding window area or region of interest. A similar sliding window approach to semantic segmentation using recurrent neural networks was also developed by Pinheiro, et al. [133]. Although competitive at the time on standard scene parsing benchmarks, sliding-window approaches are quite inefficient, as feature maps have to be computed for each rescaled version of the image, and do not reuse shared features between overlapping regions.

An interesting grid-based approach to segmentation proposed by Bayr and Puschmann (2019) [47] – relevant too for our context given its analysis of repeat natural landscape photographs – applied simple DCNN image classification to generate coarse segmentations of different land-cover classes. Unlike previous studies of landscape segmentation, which focus on a single class, this study looked at classifying multiple vegetation types in landscape photographs. The approach is similar to the patch-based approach proposed by Jean, et al. (2015), which used SVM classifiers. DCNN models were trained on 50×50 pixel image tiles representing woody and non-woody vegetation extracted from high-resolution oblique images. These patches proved suitable for DCNNs that require fixed input sizes during training and prediction, and showed invariance to slight differences between the image pairs, which may have occurred due to camera distortion. The DCNN architecture used by Bayr and Puschmann was developed through a

heuristic trial-and-error process, and is notably simpler than other segmentation models. The proposed model was trained from scratch and used both conventional and spatial dropout (see SegNet discussion below) for improved performance [47]. Bayr and Puschmann also developed a second DCNN classifier for all vegetation (woody, herbaceous and grassy vegetation). The proposed approach was capable of identifying clear trends in increasing or decreasing woody vegetation in repeat photographs.

A highly influential paper on a new DCNN architecture published in 2012 by Alex Krizhevsky, with Ilya Sutskever and Krizhevsky's doctoral advisor Geoffrey Hinton [122] is often cited as the pivotal work that reinvigorated recent interest in DCNNs, and set the stage for a new paradigm for deep learning semantic segmentation. In their paper, Krizhevsky, et al. introduced AlexNet, a neural network consisting of convolutional layers, max pooling layers and fully connected layers that has since become a model for modern DCNN architecture. AlexNet was submitted to the 2012 ImageNet Large Scale Visual Recognition Competition (ILSVRC 2012)[134] – the only neural network entered that year – where it achieved top performance by a significant margin.

Once the computer vision community took notice of this achievement, researchers began to work on semantic segmentation methods using similar architectures adapted for the task. Girshick, et al. [135] showed that DCNN architectures designed for image classification on ImageNet, could be modified for dense prediction (to localize and segment objects) using mostly the same layers of AlexNet. They also demonstrated how to boost performance through pretraining – i.e. using a network pretrained on ImageNet to initialize network parameters, and then fine-tuning the model using domain-specific training data.

Image classification networks typically have architectures that progressively downsample the feature maps until a class inference or probability vector is produced from the FC layer. Semantic segmentation, in contrast, involves dense prediction, which aims to produce an output segmentation map of the same dimensions as the input. This entails pixel-level inference, where each label corresponds to the class of the pixel's enclosing object or region. Therefore, the natural progression of dense prediction is localization or detection, where the network encodes and propagates information about both the semantic classes and the spatial location of those classes. This propagation of location and semantic information forms an inherent tension for segmentation networks, as "global information resolves what while local information resolves where" [29]. As DCNN-based segmentation has evolved, much of the work has focused on resolving this tension.

3.4.3 Dense Prediction

A breakthrough notion that helped extend image classification DCNNs to semantic segmentation was that the fully connected layer (FC) is also a convolutional layer, where the filter size can be the same as the size of the input feature map. Applying convolutions at the resolution of the image, however, is computationally prohibitive. Instead, researchers developed networks that downsample (pool) the image using strided convolution, and then upsample the features (i.e. unpool using strided transposed convolution) to reverse the downsampling and increase the feature map size back to the dimensions of the original image. Here, transposed convolution offered the revolutionary concept of upsampling by strided convolution using input-weighted filters, and summed at the overlaps in the output.

A significant success based on this development was the Fully Convolutional Network (FCN), created by Long, et al. (2015) [29]. FCN extends the image classification DCNN in two important respects: (1) the model accepts arbitrary-sized input images for dense prediction, rather than fixed-sized images; and (2) fully connected layers needed for image classification were converted into convolution layers to enable pixel-to-pixel prediction. Long, et al. augmented successful ILSVRC classifiers AlexNet [122], VGGnet [136] and GoogLeNet [137] for end-to-end learned (i.e. all parameters are trained) dense prediction using upsampling and pixelwise loss. "Skip connections" were also added between layers to combine the final prediction layer with earlier layers. This allows the coarse, semantic information of the higher features to fuse with the localized appearance information of the lower features, thus improving the spatial precision of the output. Thus, FCN attempted to resolve the tension between global and local information by combining fine detection layers with coarse ones in order to make fine localized predictions that respect global semantic information[29].

Although Long, et al. achieved state-of-the-art performance, FCN models generate quite coarse segmentation maps due to a loss of spatial information in the max-pooling stages, which degrades segmentation of fine structures and object boundaries [107]. In spite of its skip architecture, FCN disregards potentially useful scene-level information of semantic context [49]. This ability to integrate local and global features for the prediction of local segment labels forms a key challenge for semantic segmentation. Another problem with the FCN approach is that the objects of interest might have different spatial locations within the image, as well as different aspect ratios, necessitating a very large number of regions for accurate prediction.

Noh, et al. [48] sought to improve on the FCN's trade-off between class boundary details (coarseness of the downsampled input) and global semantics with a "deconvolution"

network. Composed of successive upsampling and unpooling layers, the deconvolution network recovers the spatial resolution and location information from the convolutional network to identify pixel-wise class labels and predict segmentation maps. The combined convolution-deconvolution networks form the now familiar encoder-decoder segmentation architecture, where the decoder is tasked with semantically projecting the discriminative features (lower resolution) of the encoder onto the pixel space (higher resolution) to get a dense classification. Later, this became the architecture behind U-net[1], SegNet[51], and DeepLabv3+[2], and other networks.

FCN was adapted by Harbaš, et al. (2018)[42] to classify roadside vegetation for the application of autonomous vehicle navigation. The researchers created their own image database with samples showing various weather and photometric conditions. Optimization of the network involved fine-tuning the VGG16[136] network originally trained for image recognition. The study was notable in that the models extracted features strictly from the visible spectrum (i.e. ordinary camera images) to detect different forms of vegetation. Previous methods for differentiating between different species relied on multispectral and hyperspectral segmentation methods (e.g. remote-sensing data, refer to Khelifi, et al. (2020) [23]), landscape photography is limited to the visible spectrum, which reduces the discriminating information [42].

To better integrate global context into the deep convolutional network layers of FCN, Liu, et al. [49] developed ParseNet, an end-to-end convolutional network that predicts values for all the pixels at one go. ParseNet introduced a technique of concatenating features from the whole image with features from local patches. This helped improve the smoothness and spatial precision of output segmentations. The first step uses a model to generate feature maps which are reduced to a single global feature vector with a pooling layer. This context vector is normalised and then upsampled to produce new feature maps of the same size as the input features. ParseNet also obtained leading scores on the PASCAL-Context challenge and 2012 PASCAL VOC[138] segmentation challenge.

SegNet, developed by Badrinarayanan, et al. [51], is another network based on the encoder-decoder architecture developed to improve FCN. Both SegNet and the Bayesian SegNet [139], use dropout both for training and for normal execution. Dropout[140] refers to the technique of randomly setting the outgoing edges of hidden units (feature map elements) to zero during model training. To eliminate the need for learning to upsample, SegNet's decoder stores pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling in the decoder [51]. SegNet segmentations achieved a higher resolution than that of FCN.

3.4.4 U-net

The U-net architecture, proposed by Ronneberger, et al. [1] in 2015, also sought to modify and improve upon the FCN encoder-decoder network, but additionally aimed to address some of the challenges of DCNN-based semantic segmentation for biomedical applications. The U-net architecture is shown in Figure 5.1 in section 5.3. U-net consists of a "contracting path to capture context and a symmetric expanding path that enables precise localization" [1]. In the contracting path, the receptive field is widened using max-pooling layers, resulting in decreased spatial dimensions, these features are then recovered and spatially expanded with upsampling layers. In the expanding path, U-net has several "feature channels" that pass context information to higher resolution layers [1]. The resulting upsampling path is therefore nearly symmetric to the downsampling path, forming its characteristic U-shape architecture.

The primary advantage of U-net is found in how it resolves the tension between coarse (global) and fine (local) spatial information. Similar to skip connections, the cross-connected "feature channels" copy, crop and concatenate the finely-detailed encoder feature maps to the decoding layers to increase attention to spatial localization. On the other side, the encoder allows for increasing levels of abstraction to take place for coarse localization. U-net also introduced an "overlap-tile strategy" where tiles extracted from the input image are convolved without padding to include a wider region for image context during downsampling. The resulting output segmentation is therefore smaller than the input by a constant border width [1]. To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This technique allows for a seamless segmentation of arbitrarily large images, where, for other networks, image resolution would be limited by available GPU memory.

U-net was specifically developed for segmentation of biomedical images – a dataset that typically lacks sufficient annotated training data for supervised learning. Ronneberger, et al. therefore make extensive use of data augmentation using generated elastic deformations of input examples for efficient use of available training data. Data augmentation also allows the network to learn invariance to such deformations, without the need to see these transformations in the annotated image corpus [1]. U-net is therefore designed to work with very few training images but yield more precise segmentations than FCN. Its simple architecture has made it a popular choice, resulting in many adaptations for different segmentation problems.

U-net was selected as a baseline experimental network for this study. Refer to section 5.3 for details.

3.4.5 Deeplab: Atrous Convolution

The Deeplab series of DCNNs are also encoder-decoder variants designed for dense prediction [141] [2]. Developed by Chen, et al. (2014) and open-sourced by Google, Deeplab has evolved over the past half-decade to become one of the top-performing semantic segmentation models. Deeplab's architecture uses atrous convolution to control the resolution at which feature responses are computed, and "depthwise separable convolution" (i.e., motivated by the Xception model[142] and other developments in efficient separable convolution [143]) to incorporate both spatial and depth dimensions (i.e. number of channels) to the convolution operation. With depthwise convolution, 2D convolution is achieved using two 1D convolutions with separate kernels that are multiplied. This results in a more efficient model with fewer parameters [143]. The later variant DeepLabv3+[2] resulted in top performance on PASCAL VOC 2012[138] and Cityscapes[144] segmentation datasets.

Atrous or dilated convolution was developed by Yu, et al. (2015)[145] as an efficient mechanism to enlarge the field of view of network filters to incorporate larger context while preserving the full spatial dimension. The term "atrous" derives from the French "à trous" meaning "with holes" (also called "algorithme à trous"), which refers to the zeroing of values in the filters. Atrous filters were originally introduced by Holschneider, et al. [146] for wavelet transformations. In segmentation, atrous convolution can be used to control the field-of-view (FOV), and find the best trade-off between accurate localization (narrow FOV) and semantic context (wide FOV) [2]. It can furthermore broaden a filter's receptive field without increasing the number of parameters [105]. The operation applies weights to input feature maps, where the weight values are separated in a kernel with zeros spaced apart according to some specified stride called the "dilation rate". Given an input feature map $\mathbf{x}$, for each location i on an output $\mathbf{y}$, filter weight w, and dilation rate r, we can express this as follows,

$$\mathbf{y}[i] = \sum_{k} \mathbf{x}[i + r + k]\mathbf{w}[k] \tag{3.1}$$

Hence, by varying r, we can modify filter's FOV. Note that when $r = 1$, it is standard convolution.

One of the key advantages of atrous convolution is that it allows DCNNs to handle multi-scaled input. While DCNNs are inherently good at encoding features using global receptive fields [120], many segmentation DCNNs also incorporate multiple-scale features for better performance [132] [133] [29] [2]. Scale invariance is important for segmentation given that images can require different scales of analysis, and objects can appear

at different scales in the same image. The approach taken by Long, et al. [29] and others (e.g. [147]) combines features from the intermediate layers using skip connections. These connected features are inherently multi-scale due to the increasingly large receptive field sizes [148] brought on by downsampling. Other approaches, such as that of Farabet, et al. [132], feed resized input images to a shared deep network and then merge the resulting multi-scale features for pixel-wise classification. Chen, et al. (2016) [148] jointly train a semantic segmentation network with an attention model that learns to softly weight the multi-scale features at each pixel location. They demonstrated performance improvements on average- or max-pooling over scales. In Deeplab, it is atrous convolution that allows the DCNN to systematically merge multi-scale feature maps to tackle spatial resolution loss associated with downsampling (unpooling).

With Deeplabv2 [149], atrous convolution was encapsulated in the atrous spatial pyramid pooling (ASPP) network module that filters at multiple sampling rates and effective FOVs. ASPP is an atrous version of SPP[150] that helps to incorporate different object scales to improve inference accuracy. With DeepLabv3 [151], this module was upgraded to include image-level features drawn from the work of Liu, et al. (2015) for ParseNet[49] and Zhao, et al. for Pyramid Scene Parsing Network[152] to capture longer range information, and added batch normalization[153]. DeepLabv3+[2] extends this network with a decoder module to refine the segmentation results along object boundaries.

DeepLabv3+ was selected as the primary experimental network for this study. Refer to section 5.4 for details.

3.4.6 DCNN + CRF Methods

Hybrid methods that combine DCNN segmentation and CRF modelling have also been shown to significantly improve the accuracy of dense predictions [39] [3] [104] [154] [34] [103] [155]. This can be attributed to the combination of fine-grained CRF modelling power with the representation-learning ability of DCNNs [104]. Techniques for combining these two methods range from introducing CRFs as an independent stage of the segmentation pipeline (i.e. Deeplab [2] [105]) to embedding these models within the network itself (i.e. CRFasRNN [156]).

Buscombe and Ritchie (2018) [34] studied the application of CRFs and DCNNs for the semantic segmentation of natural landscape images. Their aim was to develop a hybrid method for semantic segmentation that uses the memory-efficient mobile DCNN architecture MobileNetV2[157] with CRFs to generate ground-truth data. Used in combination with pretraining, their method trains DCNNs with small datasets, and was tested on satellite images of various natural land covers and landforms, as well as landscape-level

and high-vantage imagery. As well, they proposed a novel segmentation method that hybridizes DCNN-based image classification of small regions in the imagery, with the fine-grained localization of fully-connected CRFs for pixel-level classification [34]. Designed as a "simpler alternative" to the FCN-based networks DCNNs, which are often computationally more demanding to train, this approach is intended for segmentations where the scales of spatially continuous features are larger than the tile size used in the DCNN. They note that spatially isolated features are better classified by FCN-based encoder-decoder models.

3.5 Data Augmentation

Many of the so-called "real world" datasets present a range of challenges for supervised deep learning that can have a significantly negative effect on the performance of segmentations methods. Three main challenges include: (1) Imbalanced semantic classes[158] in the observed data (i.e. datasets with a high ratio of majority to minority samples); (2) High variation in the appearance of objects (e.g. scale, color and texture)[159], and (3) Lack of annotated ground-truth data[1]. In such cases, data augmentation – i.e. extending the dataset with generated data – is commonly used to resolve them by: (1) increasing the size of a limited annotated data; (2) improving DCNN generalization by teaching the expected input appearance variation (improve sensitivity), and (3) augmenting examples with underrepresented classes to mitigate class imbalance.

Though most data augmentation techniques have been developed for image classification or recognition[160], some have been adapted for segmentation – though Chatfield and Zisserman (2014) [161] showed that even simple and generic augmentation benefit all deep architectures. Basic augmentation uses geometric manipulations of image samples, such as horizontal flipping, color space augmentations, and random cropping; more complex transformations include affine (rotation, scaling, translation, and perspective shifts) and elastic deformations. These transformations are both computationally cheap and mostly generic to the application. For ImageNet, Krizhevsky, et al. [122] applied image translations, horizontal reflections and varied the intensities of RGB channels in training images. Yang, et al. [162] applied random horizontal mirror and random scales in for data augmentation during training. Developers of Deeplab applied data augmentation by randomly scaling the input images (from 0.5 to 1.5) during training [2].

Ronneberger, et al. [1] relied heavily on data augmentation to supplement limited annotated biomedical data. They applied elastic deformations to the training images, primarily to introduce shift, intensity, distortion and rotation invariance to the model.

As the authors noted, this form of deformation is particularly important in biomedical segmentation learning, since deformation used to be the most common variation in tissue and realistic deformations can be effectively simulated. In contrast, Long, et al. [29] (FCN) applied random mirroring and "jittering" (short image translations), but reported no noticeable improvement in DCNN performance. This difference in results with Ronneberger, et al. suggests the effectiveness of certain geometric transformations for data augmentation may be domain-specific. Furthermore, image transformations only change depth or scale of image, rather than the label composition, which can improve DCNN performance, but does little to address class imbalance or label distribution [163].

Another form of data augmentation is known as guided-augmentation, where the model learns class-specific transformations from an external source of training data. Dixit, et al. (2017)[164], for example, proposed an attributed-guided augmentation (AGA) which learns a mapping of object features that allows generation of data such that an attribute of a synthesized sample is at a desired value or strength. This approach is suitable for limited labeled data plus an available external source of annotated samples.

Data augmentation is also used to address class imbalance, which can lead to poor DCNN performance and overfitting. Li, et al. (2019) [165] investigated overfitting of neural networks under class imbalance and found that underrepresented samples tend to shift DCNN activations towards the decision boundary, thus losing sensitivity. Models that overfit to training data thus have a bias to under-segment the minority classes on unseen test data. Data augmentation methods that specifically address class imbalance, have primarily been developed for image classification. These can be grouped into two main categories: (1) Sampling-based methods and (2) Algorithm-based methods. Sampling-based methods operate directly on a dataset with the aim to balance its class distribution. SMOTE (synthetic minor oversampling technique) [166], for example, uses k-nearest neighbors similarities to select underrepresented samples to oversample. Given the multi-class composition of individual training samples used in segmentation networks, low-frequency samples are not well-defined for similarity metrics such as nearest neighbor search. Falk, et al. [50] developed an augmentation method for U-net that used a normalized weight map for sampling the spatial location. Weight was given to background regions which were presented to the network only one tenth as frequently as foreground objects, and tiles centered around an ignored region were not selected during training. This method was intended to help networks become robust to translations and to focus on relevant regions. The method ignored the background class, which, in many cases, was homogeneous and easy to learn.

3.6 Loss Functions

Deep learning problems are transformed into optimization problems through the minimization of expected error or loss on the training set, which is estimated using an objective or loss function [167]. The stochastic gradient descent (SGD) optimization algorithm used in DCNN training takes the computed gradient of the loss function to adjust filter weights such that they reduce the loss on the next evaluation [120]. Performance of deep-learning semantic segmentation methods are strongly dependent on the choice of loss function [168], and should reflect the application (e.g. segmentation versus classification), and domain-specific characteristics of the training data (e.g. class imbalanced and/or limited in size).

For semantic segmentation, pixel-wise softmax and cross-entropy (CE) are standard loss functions [132] [29] [2] [1] used to estimate the conditional probabilities of the classes. Cross-entropy is a measure of the difference between two probability distributions, and defined as the negative log-likelihood loss between the empirical distribution defined by the training set, and the probability distribution predicted by the model [167]. A detailed definition of CE is given in section 4.5.4.1. Deeplab experiments, for example, used the sum of equally weighted cross-entropy losses for each spatial position in the DCNN output map[2].

However, CE loss has a number of limitations for segmentation networks. In particular, CE is normally calculated at each pixel independently and then averaged over all pixels – a process that works poorly for dense segmentation maps that contain semantic relations among pixels [119]. For imbalanced datasets, CE asserts equal learning to each pixel, which over-emphasizes high-frequency classes. Furthermore, while segmentation DCNNs learn best when trained on sufficiently large and representative data, datasets with severe class imbalance are quite prone to overfitting, which is largely due to a failure of loss functions, such as CE, to take imbalance into account[165]. Experiments by Li, et al. (2019) [165] on imbalanced datasets showed overfitting can reduce recall (sensitivity), though not affect precision to a large extent. They concluded that models that overfit tend to under-segment the low-frequency minor classes on unseen test data, which they attribute to a shift in logit activations towards the decision boundary.

A popular strategy to improve CE loss for imbalanced data is to assign more importance to the low-frequency labels using weighted cross entropy [29] [1]. Weights are conventionally pre-computed from the ground truth segmentation to compensate using inverse class proportionality as weight values [169]. Weighting loss by simple inverse class frequency can still yield poor performance, and is not frequently employed[170]. To better weight classes in loss computations, the E-Net model [171] uses a "smoothed"

inverse-log variant that prevents blowup from very small classes in the loss calculation. Badrinarayanan, et al. [51] employed median frequency balancing for training SegNet, where class weights are the inverse class frequency scaled by the median of all class frequencies over the whole dataset. Experiments showed that using the scaled weights in the loss function during training resulted in improved average accuracy per class, a moderate increase in mean intersection over union, but a lower global accuracy [51]. Similarly, Cui, et al. (2019) [169] proposed a fully class-balanced loss function that incorporates a prior "data overlap" metric for CE and focal losses based on a calculation of the effective number of samples in the ground truth. This method was not evaluated for segmentation networks.

Segmentation boundaries pose another challenge for loss functions, as boundary pixels form a small fraction of the total image, such that the direct use of the CE loss fails to form sharp contours between classes [168]. To improve sharp edge classification, Ronneberger, et al. [1] proposed an supplementary boundary loss that places a higher weight on thinner edges to force the network to learn boundaries. This scheme improved binary segmentation of cells in biomedical images. Deng, et al. (2018)[168] also proposed a similar boundary loss using the Dice coefficient. Edge detection loss functions were not explored in this thesis, although recommended for future research.

The Sørensen–Dice Similarity Coefficient (abbreviated as "Dice coefficient" and equivalent to the F1 score) is also commonly adapted as a loss function for segmentation networks. Similar to the Jaccard similarity index, or Intersection over Union (IoU), Dice loss quantifies the overlap between a ground-truth and predicted labels normalized to the size of the segmentation map. Milletari, et al. (2016) [172] were the first to introduce a binary Dice coefficient loss layer for the V-net segmentation network. Experimental results for imbalanced medical image segmentation showed better performance using Dice than models trained through the same network using weighted CE loss[172]. Sudre, et al. (2017) [173] proposed a generalized class-weighted Dice loss to improve segmentation training on imbalanced data. Results showed when the level of imbalance increases, loss functions based on overlap measures such as Dice appeared more robust [173]. A detailed definition of Dice loss is given in section 4.5.4.2.

Focal loss was proposed by Tsung-Yi Lin, et al. (2017) [174] for dense object detection to reduce loss for well-classified samples and focus on more uncertain samples that lie near the decision boundary. Focal loss dynamically scales cross entropy loss by a factor that gives lower weight to a class when there is higher confidence in correct prediction [174]. A detailed definition of focal loss is given in section 4.5.4.3.

To better incorporate spatial correlation in ground truth, Kim, et al. (2019) [119] proposed a loss function based on level set theory [175]. Their proposed process first

decomposes multi-class segmentations into binary images – i.e. single class and background, and then converts level set functions into class probability maps. For each class, the method calculates the level set energy and treats the sum as the loss function for training the segmentation network. Though not explored in the current study, further experimentation using this method is recommended.

3.7 Summary of Current Study in Context

This study concerns the problem of multi-class semantic segmentation of different land-cover categories in oblique landscape photography. Using the MLP dataset (i.e. oblique historic (grayscale) and repeat (colour) photography), we aim to evaluate the performance of U-net [1] and DeepLabv3+ [2] architectures for this task. These state-of-the-art deep learning networks were selected based on their proven performance in segmentation for recognized benchmark segmentation datasets (e.g., U-net: [1] [50] [176], Deeplab: [2] [32] [177]). Furthermore, U-net was designed for segmentation of high-resolution images made possible by their overlap-tile strategy, which is adaptable for tile extraction of high-resolution MLP capture images. DeepLabv3+ has been shown to be a high-capacity network designed using atrous convolution (ASPP) to handle multi-scale object classification characteristic of landscape imagery, and in particular for the discrimination of foreground and background visual objects.

To address the severe class imbalance of the MLP dataset (see discussion in section 2.3.3), a novel data augmentation approach was developed and evaluated. The technique is based on the normalized weight map technique developed by Falk, et al. (2019) [50] and SMOTE (synthetic minor oversampling technique) [166]. The proposed method generates samples based on the feature space similarities between existing underrepresented training examples. Pixel class distributions were computed for each tile, and a threshold-based algorithm computed an appropriate sample rate with the objective of minimizing the Jensen–Shannon divergence with an ideal balanced distribution. An inverse-log loss weighting scheme proposed by Paszke, et al. (2016) [171] was used for CE and Dice losses, which is bounded as the probability approaches zero.

Performance was also evaluated for the proposed models based on the use of three different segmentation loss functions: Weighted cross-entropy loss [1] [2] [50], Dice loss [172] and Focal Loss[174]. The evaluation was centred on performance metrics for underrepresented classes using loss functions that address the limitations of CE for imbalanced and limited datasets.

Finally, we present an evaluation of fully-connected CRFs[39] [3] as a post-processing performance boost using the DCNN output partial probabilities as unary potentials. This follows reported improvements to fine segmentations shown by Arnab, et al. [104], Chen, et al. [2], Buscombe and Ritchie (2018) [34], among others.

Chapter 4

Methodology

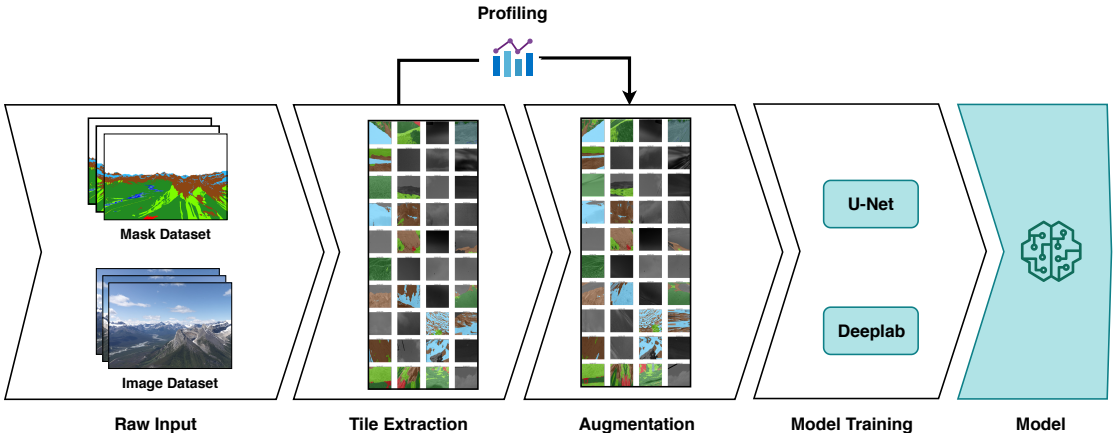
4.1 Overview

This chapter summarizes the proposed method of this study to both assess the effectiveness of DCNN-based methods for land cover classification of oblique photographs, as well as address the domain-specific challenges described in Chapter 2. This chapter begins with a brief description of the objectives of this study in Section 4.2. Next, methods for the preliminary preparation of the raw dataset (Section 4.3) and image preprocessing (Section 4.4) are introduced. Third, the proposed semantic segmentation methodology (Section 4.5 is given, including a discussion of the conditional random fields as a post-processing step (Section 4.6). Finally, the evaluation framework used to assess experimental results (Section 4.7) is presented.

4.2 Objectives

In outline, this study proposes to evaluate, through experiments, the performance of two state-of-the-art convolutional neural network architectures, U-net [1] and DeepLabv3+ [2], optimized for pixel-based semantic segmentation of oblique imagery at a landscape scale. This evaluation compares the predicted segmentation maps of the DCNN network models with the manually-created ground truth segmentation maps. An experiment is defined here as a series of trials, where each trial includes the training, validation, and evaluation (testing) of a network model for a specific configuration of parameters, as specified in Chapter 5. Model architectures were selected based on their proven performance in the generalization and accuracy of pixel classification for recognized benchmark segmentation datasets [1] [2] [177] [121], and served as controlled test architectures for experiments.

FIGURE 4.1: Image preprocessing and model training pipeline.



Three experiments were conducted as follows:

1. **Data Augmentation**: Evaluated performance metrics for various sizes and compositions of extracted and augmented databases, as outlined in the preprocessing steps in Section 4.4;

2. **Loss Functions**: Tested different types, configurations, and ratios of loss functions as outlined in the preprocessing steps in Section 4.5.4;

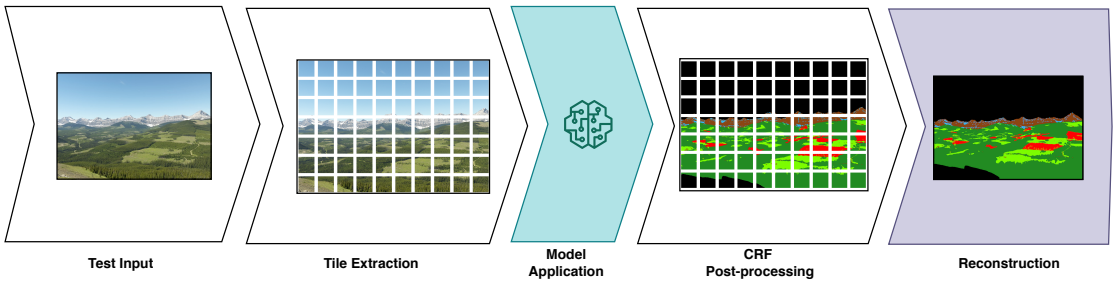
3. **Conditional Random Fields** Test variables included the post-processing application of a trained conditional random field filter to network output unary data, as outlined in Section 4.6.

Other tested configuration settings, including input and network layer normalization, activations functions, and model hyperparameters (i.e., learning rate, batch size, and the number of training epochs), were part of a preliminary parameter tuning, and were not explicitly included or evaluated in the experimental results (see discussion in Section 5.2). Experiments required the training of separate historic and repeat models, not only to avoid adapting input channel dimensions in the model, but also because of differences in image quality, including artifacts and noise in historic photographs. Specific differences between historic and repeat capture images are discussed in Background.

4.3 Dataset

Training images and their corresponding segmentation masks were sourced from two datasets sampled from the Mountain Legacy Project collection (see DST.A.1 and DST.B.1

FIGURE 4.2: Model application and CRF post-processing pipeline.



in Appendix A for a full list of the images). Each image has a corresponding colour-coded segmentation mask created manually by field experts that labels the land cover regions of interest. Images were previously selected for precise manual segmentation according to criteria outlined in Fortin et al. [6]. Input image pairs in the datasets have furthermore been aligned through affine transformations (rotation, scaling, and translation) applied to the historic images, as part of an image registration pre-processing step. Refer to Section 2.3.1 for further information on the collection and prior preprocessing of the raw input images, as well as the manual segmentation procedure for creating ground-truth segmentation masks. DST.A and DST.B were each divided into historic (grayscale) and repeat (colour) image captures. Note that historic images were also augmented by gray-scaled versions of the repeat tile database, which greatly expanded the size of the historic database.

Masks in DST.A use a different land cover classification scheme than those in DST.B, labelled LCC.A and LCC.B, respectively. These schemes are summarized in Table 2.1 in Chapter 2. To train the segmentation model using the combined datasets (DST.A.1 and DST.B.1), a common or "merged" categorization scheme was used that combined categories, B and S, used in DST.B to form Broadleaf-Mixedwood (B-MW) and Herbaceous/Shrub (H-S) categories. This merged scheme is equivalent to the scheme used by Jean et al. [5] for DST.A. Using the more compact LCC.A scheme helped to mitigate the class imbalance by merging two severely underrepresented classes and simplify model evaluation. Also note that foreground pixels in DST.A and DST.B were classified as "not classified" (see discussion in Section 2.3.3) – i.e. this class was not ignored during training, which allowed the model to discriminate foreground pixels of a given class from those at greater distances from the camera (see also: Section 4.7.3).

4.4 Image Preprocessing

Image preprocessing is a necessary step in supervised learning in which the raw input image data is prepared as suitable training data for the network model. Preprocessing for this study is summarized in Figure 4.1 proceeded as follows: (1) Tiles (subimages) were extracted from the full-sized high resolution MLP images; (2) Data augmentation techniques were applied to both increase the size of the training dataset and mitigate class imbalance; (3) Segmentation network models (U-net and DeepLabv3+) were trained on the extracted and/or augmented tile database. The testing method shown in Figure 4.2 proceeded as follows: (1) Tiles were extracted based on a modified application of the symmetric overlap-tile strategy[1]; (2) Segmentation maps were created for each tile using the trained networks; (3) Optimized fully-connected conditional random fields[3] were used to boost the accuracy of the DCNN output; (4) Output tiles were reconstructed into a full-sized segmentation map by averaging class probabilities over tile overlaps; (5) Predicted segmentation maps were evaluated against ground-truth masks for accuracy.

The following subsections present further details to the training preprocessing: (1) **Extraction**: To accommodate GPU memory limits, the full-sized, high-resolution input images and corresponding masks were partitioned into smaller square tiles. The use of tiles scales the input size for the DCNN models, but also allows for high-resolution dense predictions; (2) **Data Augmentation** Extracted tiles were augmented with perturbed copies of the originals (i.e. generated affine transformations). Augmentation contributes three main performance enhancements: (1) it enlarges the annotated dataset; (2) improves model sensitivity; and (3) improves class imbalance. To calculate effective sample rates for data augmentation, a novel threshold-based algorithm was developed that uses the class distribution of each tile. This statistical data was also used to calculate class loss weights for model training (See Section 3.5).

Note that an important design attribute followed in the implementation was the minimization of *a priori* information needed to train and enhance the DCNN model. Though the model requires external metadata to adjust losses for class imbalance (i.e. a class distribution for each tile), this metadata was computed in the preprocessing step, and was therefore integrated in the training pipeline.

4.4.1 Tile Extraction

Given the very high resolution of the raw MLP input images (20 to 40 megapixels), it was not feasible to train the DCNN models using full-sized samples. Instead, smaller

subimages or "tiles" were extracted and stored in a high-performance database for fast retrieval. Extracted tiles were selected as square 512 × 512 pixel subimages, which allowed for (on average) each raw full-sized image to be decomposed into approximately 100 tiles. Furthermore, in addition to the full-sized images, extractions of downsampled versions, scaled by 0.2 and 0.5, were also added to the database. These additional tiles were intended to improve the classification of regions of an arbitrary scale (see related discussion in Section 3.4.5). Extracted tiles were also augmented using data augmentation (see below). See Table 5.1 for a summary of the the extraction and augmented databases.

Note that the stride of the tile extraction was model-dependent: cropping, for example, was inherent to the encoding phase of the U-net model[1], whereas tile dimensions were preserved in the output masks for the DeepLab model. Specific extraction settings are outlined for each model below.

1. **U-net** The U-net network model accepts only fixed input size for training samples. Square tiles of size 512 × 512 (for an output size was 324 × 324 pixels) were extracted from raw images as input, based on the original tile size proposed by the U-net creators [1]. This allowed for the processing of a batch size of 10 during training. This tile-based training was adapted from the overlapping tile strategy as described by Ronneberger, et al. for the U-net model [1].

2. **DeepLabv3+** Input tiles for the Deeplab model were not cropped during the encoding phase of training, and therefore an overlap was not required for the training dataset tiles. During model testing, however, a stride of 256 pixels was chosen to facilitate blending during image stitching after testing. Tile reconstruction is described in Section 4.5.5.

4.4.2 Data Augmentation

4.4.2.1 Overview

Data augmentation is the process of extending the dataset with generated data. Often this involves introducing perturbed copies of the original input images, where perturbations may include affine transformations such as rotation, scaling, flipping, translation, as well as cropping, and changes to the brightness of the pixels (see Section 3.5). The process forms an integral part of DCNN-based methods employed to overcome overfitting on models, improve sample sensitivity, and boost overall performance. Supervised DCNN network sensitivity and performance strongly depend on the number of

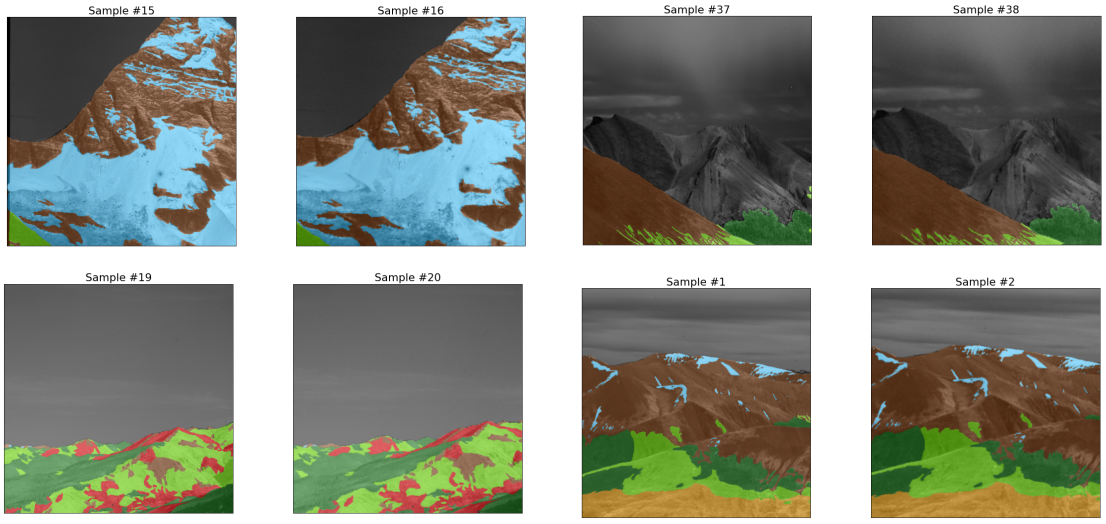
training samples, where more samples per class allows the model to better extract the discriminating features among various classes, and thus increase classification accuracy [178]. Augmentation has also been widely shown to make more efficient use of the available annotated data in semantic segmentation, where unbalanced semantic label distribution could negatively affect segmentation accuracy. [122] [1] [163]. Previous studies have demonstrated that increasing the size of the training dataset using different data-augmentation techniques increases performance and makes the learning of DCNNs models robust to changes in scales, brightness and geometrical distortions [179] [180].

For the current experiments, perturbed versions of input tiles were generated and added to the training dataset image. The distortions were limited to affine and/or perspective deformations that do not significantly alter the natural appearance and composition of the landscape segments. It was hypothesized that elastic deformations used for data augmentation in previous case studies, such as for biomedical image segmentation [1], would introduce training samples not naturally found in oblique landscape photos. Instead, transformation matrices were generated that apply randomized perspectives, translations (vector addition), and scale operations (linear transformations), and applied to the image and mask data using the OpenCV libraries [181]. The perspective transformation in OpenCV requires a 3×3 transformation matrix. Straight lines remain straight after the transformation. Four points on the input image (three of which must be non-collinear) and corresponding points on the output image were used to generate the transformation matrix. Augmented samples were also randomly flipped along the vertical axis. Cropping was applied to gaps created by perspective-shifted tiles. To maintain the correct mask colour palette (LCC.A), nearest-neighbour pixel interpolation was used for augmentation transformations. Refer to Figure 4.3 for examples of applied tiles deformations.

4.4.2.2 Segmentation Oversampling Method

To address class imbalance, an oversampling method was developed and implemented, inspired by SMOTE (synthetic minor oversampling technique) [166] [182], and based on the normalized weight map method developed by Falk et al.[50] (see Section 3.5). The proposed method generates sample rates for each tile using a thresholding algorithm applied to its class distribution. Optimal scaling parameters used to compute sample rates are computed by minimizing the Jensen-Shannon Divergence (JSD) of the empirical distribution of the dataset with an ideal balanced distribution.

FIGURE 4.3: Sample extraction (left) and augmented (right) tiles showing segmentation mask overlays. Randomized affine deformations to the image data were calibrated to ensure natural landscape features were not overly distorted.



Given an image dataset $\mathcal{D}$ defined previously, let the overall pixel-class probability vector of $\mathcal{D}$ be $\mathbf{p}_\mathcal{D}$, defined over the sequence of m semantic classes $\mathcal{L}$. Also let $\mathbf{p}_i$ be the pixel-class probability vector for an image $\mathbf{x}_i \in \mathcal{D}$ (i.e., tile subimage). The ideal balanced pixel-class probability distribution p_b can be defined as the case where all pixel-class counts are equal, and thus each class has probability $\frac{1}{m}$.

An oversampling score can be calculated for each sample image $\mathbf{x}_i \in \mathcal{D}$ defined as the square-root of the dot-product of the image pixel-class probability vector with the inverse dataset pixel-class probability vector:

$$s_i = \sqrt{\mathbf{p}_i \cdot \frac{1}{\mathbf{p}_\mathcal{D}}} \tag{4.1}$$

This score measures the degree to which oversampling the image $\mathbf{x}_i$ will contribute to overall class balancing of $\mathcal{D}$ by inversely scaling each pixel-class probability by its corresponding probability in the dataset. Augmented pixels from minor classes are therefore given greater weight in the score, whereas dominant classes are weighted less. Consequently, images dominated by minor pixels classes will score higher than images dominated by dominant classes. For scores greater than some threshold τ, the augmentation sample rate is defined as $r = \rho s_i$, where ρ is a tunable scaling factor and s_i is the image sample rate score. The expected augmented probability vector p_α can be determined as the aggregation of each pixel distribution of $\mathbf{x}_i \in \mathcal{D}$ scaled by its calculated oversample rate:

$$\mathbf{p}_\alpha = \mathbf{r}_i \odot \mathbf{p}_i \tag{4.2}$$

Optimal values for τ and ρ can be estimated through a grid search that minimizes the Jensen-Shannon distance (JSD) between the predicted $\mathbf{p}_\alpha$ of the augmented dataset and $\mathbf{p}_b$ for the ideal balanced distribution. JSD is a metric used to quantify the difference between two probability distributions. The metric is defined as the square-root of the Jensen-Shannon Divergence ($\sqrt{JS}$):

$$JSD = \sqrt{JS(\mathbf{p}_\alpha \| \mathbf{p}_b)} = \sqrt{\frac{1}{2}(KL(\mathbf{p}_\alpha \| M) + KL(\mathbf{p}_b \| M))} \tag{4.3}$$

where $M(i) = \frac{1}{2}[p_{\alpha,i} + p_{b,i}]$ and KL is the Kullback-Leibler divergence, defined as:

$$KL(\mathbf{p}_\alpha \| M) = \sum_{i \in \mathcal{L}} p_{\alpha,i} \log_2 \frac{p_{\alpha,i}}{M(i)} \tag{4.4}$$

JSD provides a smoothed and normalized version of KL divergence, with scores between 0 (identical distributions) and 1 (maximally different distributions), when using the base-2 logarithm. Therefore, the aim is to score a JSD as close to zero to improve distribution balance.

A second metric used to evaluate the final class imbalance, the $M2$ Gibbs index [183], gives variance of a multinomial distribution, where $p_{\mathcal{D}}(l) = f_i/m$ is the probability the pixel belongs to some class $l \in \mathcal{L}$ semantic classes:

$$M2 = \frac{m}{m-1}\left(1 - \sum_{i \in \mathcal{L}} p_{\mathcal{D}}[i]^2\right) \tag{4.5}$$

The value of $M2$ ranges from 0.0 to 1.0, where 1.0 indicates a uniform distribution. $\frac{m}{m-1}$ is a standardization factor.

4.5 Multi-class Semantic Segmentation

4.5.1 Definition

Semantic image segmentation for multiple classes can be formally defined as the nonlinear transformation of an input image tensor $\mathbf{x}$, with axes corresponding to image rows, image columns, and channels (red, green, blue or single-channel intensities for grayscale images), to an output tensor $\hat{\mathbf{y}}$ that is a probability distribution over semantic labels for each pixel. The output tensor has axes corresponding to mask rows, mask columns, and the different semantic classes [167], which, in the case of the MLP dataset, correspond to the different land cover categories defined in 2.3.1.

Each input dataset used for the experiments was modeled as a set of n image tensors $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ that have pixel coordinates (x_1, x_2) on a discrete grid of size $d_1 \times d_2$ and channel intensities $\mathbf{x}_i(x_1, x_2) \in \mathcal{V} \subset \mathbb{Z}^c$ where c is the number of image channels. For three-channel RGB images, $\mathcal{V} \in \mathbb{Z}^3$ and for grayscale $\mathcal{V} \in \mathbb{Z}^1$.

Furthermore, each image $\mathbf{x}_i \in \mathcal{D}$ has a corresponding ground-truth segmentation mask tensor $\mathbf{y}_i$ of the same grid size as $\mathbf{x}_i$. Each pixel x in $\mathbf{y}_i$ can be assumed to map to a single class in $\mathbf{y}_i(x) \in \mathcal{L} \subset \mathbb{Z}$, where $\mathcal{L}$ defines a set of m semantic class labels $\{l_1, ..., l_m\}$.

Following a similar formal model to that presented by Novikov[184], each $\mathbf{y}_i$ can be mapped to an equivalent sequence of binary masks over m classes: $\{\mathbf{B}_{i,0}, ..., \mathbf{B}_{i,m},\} \in \mathcal{B} = \mathbf{B}_{d_1 \times d_2}(\{0, 1\})$, where $\mathcal{B}$ is the space of all binary matrices of size $d_1 \times d_2$. If $\mathcal{Y} = \mathbf{B}_{g_1 \times g_2}(\{1, m\})$ is defined as the space of all matrices with values $\{1, ..., m\}$ that index semantic class labels in $\mathcal{L}$, a well-defined mapping $g : \mathcal{B} \to \mathcal{Y}$ can be expressed as,

$$\mathbf{y}_i = g(\mathbf{B}_i) = \sum_{l=1}^{m} l \cdot \mathbf{B}_{i,l} \tag{4.6}$$

Therefore, for a given image tensor $\mathbf{x}_i \in \mathcal{D}$ and corresponding ground-truth matrix $\mathbf{y}_i \in \mathcal{Y}$, the inverse mapping of g for some semantic class $l \in \mathcal{L}$ is the projection of $\mathbf{y}_i$ onto $\mathcal{B}$ defined as $\pi_l(\mathbf{y}_i) = \mathbf{B}_{i,l}$ where $\pi_l : \mathcal{Y} \to \mathcal{B}$. This inverse projection is equivalent to the one-hot or "one-of-K" encoding of the categorical index of l over $\{1, ..., m\}$.

Model training and evaluation were distinct processes that required disjoint partitions of the input dataset $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ defined as $\mathcal{D}_{train}$, $\mathcal{D}_{eval}$ and $\mathcal{D}_{test}$. To prevent overfitting of the model, it was crucial that the training algorithm did not observe the validation set [167]. All data was assumed to be independent from each other and that the training set and test set were drawn from the same probability distribution.

Model evaluation (or training validation) was applied to the input dataset $\mathcal{D}_{eval}$; whereas model testing was applied to the dataset $\mathcal{D}_{test}$, each consisted of the forward pass through the network of unseen images defined by the input $\mathbf{x} \in \mathcal{D}_{eval}$ and $\mathbf{x} \in \mathcal{D}_{test}$, respectively.

4.5.2 Model inference (forward pass)

In the model's forward pass, the input tensor $\mathbf{x}_i$ is filtered through the network layers to compute a vector of weights assigned to each semantic class, and for each pixel in $\mathbf{x}_i$. The output prediction tensor, $\hat{\mathbf{y}}_i$, has the same dimensions (or same aspect ratio) as $\mathbf{x}_i$ – i.e. a dense prediction. Each image tensor $\mathbf{x}_i \in \mathcal{D}$ therefore produces a corresponding multi-class segmentation mask $\hat{\mathbf{y}}_i$ prediction, which can be defined by a nonlinear functional mapping $F : \mathcal{D} \rightarrow \mathcal{Y}$. F derives for each pixel of $\mathbf{x}_i$ its semantic class $l \in \mathcal{L}$ with some probability. For simplicity, this mapping can be represented as a nonlinear matrix function $F(\mathbf{x}) \in \mathcal{Y}$.

At each network layer t in the forward pass, the total input to each unit $\mathbf{z}_t$ is computed, which is equal to the weighted sum of the outputs of the units in the layer below (and the input at $t = 0$ is $\mathbf{x}_i$), such that $\mathbf{z}_t = \mathbf{W}\mathbf{z}_{t-1}$, where $\mathbf{W}$ is the layer filter weights, as defined by the network architecture [120]. (Note that the bias term $\mathbf{b}_t$ is omitted here for simplicity.) The output was then passed through a nonlinear rectified linear unit function $\text{ReLU}(\mathbf{z}_t) = max(0, \mathbf{z}_t)$. ReLU is a standard activation function for deep neural networks [185]. For U-net experiments, the Scaled Exponential Linear Units (SELU) was also used, defined as follows,

$$\text{SELU}(\mathbf{z}_t) = \lambda \begin{cases} \mathbf{z}_t & \mathbf{z}_t = 0 \\ \alpha e^{\mathbf{z}_t} - \alpha & \mathbf{z}_t \leq 0 \end{cases}$$

Network input tensors consist of a batch of size $b \ll n$ examples defined as subset $\mathbf{K} \in \mathcal{D}$. Batch Normalization (BN) was applied after each convolutional layer and before the activation function. BN normalizes, re-centres and re-scales the input layer to improve variance. Given the tensor input $\mathbf{x}$ (a mini-batch of 2D inputs with additional channel dimension), the output $norm(\mathbf{x})$ of batch normalization were defined as,

$$norm(\mathbf{x}) = \gamma \frac{\mathbf{x} - \overline{\mathbf{x}}}{\sqrt{\sigma(\mathbf{x}) + \epsilon}} + \beta \tag{4.7}$$

where the mean and standard-deviation (σ) are calculated per-dimension over the mini-batches, and γ and β were learnable parameter vectors of size equal to the input dimensions. A small-valued ϵ was added for numerical stability.

Normalizing layer inputs has been shown to improve training speed by reducing the phenomenon of internal covariate shift, defined as the "change in the distribution of network activations due to the change in network parameters during training." [153] By integrating normalization into the model architecture such that it is applied to each training mini-batch, BN allows for higher learning rates, and thus reduces the number of training steps, as well as add sensitivity to parameter initialization [153]. BN is also the most widely used DCNN normalization method for natural semantic segmentation [60].

Given the model's final output logits, defined as $\hat{\mathbf{z}}$, the Softmax function was applied to compute the class probabilities assigned to each semantic category, which gave the final probabilities of $\hat{\mathbf{y}}$. The Softmax function was also indirectly required to compute losses for multi-class semantic segmentation. Given the tensor of raw (non-normalized) predictions, or logits $\hat{\mathbf{z}}_\mathbf{j}$ generated by the network model, $\hat{\mathbf{z}}_\mathbf{j}$ was normalized using the Softmax function over m semantic classes:

$$\hat{\mathbf{y}}_\mathbf{j} = \text{Softmax}(\hat{\mathbf{z}}_\mathbf{j}) = \frac{e^{\hat{\mathbf{z}}_\mathbf{j}}}{\sum_i^m e^{\hat{\mathbf{z}}_\mathbf{j}}} \tag{4.8}$$

The Softmax function returns a tensor of normalized semantic class probabilities of the same dimensions as $\hat{\mathbf{y}}_\mathbf{j}$, where each pixel was associated with a distribution of m probability values. The index j of the most likely classes over $\{1, ..., m\}$ for each pixel was then extracted from either $\hat{\mathbf{z}}$ or $\hat{\mathbf{y}}$ to give the predicted mask $\hat{\mathbf{m}} \in \mathcal{Y}$:

$$\hat{\mathbf{m}}(\hat{\mathbf{z}}) = \underset{j}{\text{argmax}}\, \hat{\mathbf{z}}_j = \underset{j}{\text{argmax}}\, \hat{\mathbf{y}}_j. \tag{4.9}$$

During model training, the computed unary potentials $\hat{\mathbf{y}}_i = F(\mathbf{x}_i)$ were instead used to derive an accuracy estimate over the validation dataset (see Section 4.7 below).

4.5.3 Model training (back-propagation)

During training, each model took as its input tensor a mini-batch of size b defined as subset $\mathbf{K} \in \mathcal{D}_{train}$, which was filtered through the forward pass. As discussed in the preceding, for an example $\mathbf{x}_i \in \mathbf{K}$, each pixel location was assigned a probability vector that scores each semantic class $l \in \mathcal{L}$. Therefore, each of the m classes of the forward pass transformation $F(\mathbf{x}_i)$ corresponds to an $d_1 \times d_2$ probability density distribution. Given these pixel classification scores, model training aims to maximize the probability of the true class l' for each pixel in $\mathbf{x}_i$. This optimization aims to minimize the distance

between $\hat{\mathbf{y}}_i = F(\mathbf{x}_i)$ and $\mathbf{y}_i$. A generalized objective function J is therefore defined as the minimization of:

$$J : \mathcal{D} \times \mathcal{Y} \rightarrow \mathbb{R} \tag{4.10}$$

that estimates the error of the network outcome from the ground-truth.

Let a training dataset set of n image tensors be defined as $\mathcal{D}_{train} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ with empirical distribution $p_{\mathcal{D}}$. Let $p_{model}(\mathbf{x}; \boldsymbol{\Theta})$ be a parametric family of probability distributions over the same space indexed by $\boldsymbol{\Theta}$. In other words, $p_{model}(\mathbf{x}; \boldsymbol{\Theta})$ maps any configuration $\mathbf{x}$ to a real number estimating the true probability $p_{\mathcal{D}}$ [167]. The maximum likelihood estimator for $\boldsymbol{\Theta}$ is then defined as

$$\Theta_{MLE} = \text{argmax}_{\boldsymbol{\Theta}} \sum_{i=1}^{m} log[P(\mathbf{y}|\mathbf{x}; \boldsymbol{\Theta})] \tag{4.11}$$

that estimates the error of the model prediction from the desired ground-truth. In the case of deep learning, the estimator Θ_{MLE} is defined as back-propagation, discussed in the following section.

In the backward pass of the model, the back-propagation algorithm [120] [167] was used to minimize model loss (model convergence to local minima) by computing the gradient of the objective (loss) function $J(\mathbf{x}, \mathbf{y}, \boldsymbol{\Theta})$, given an input tensor $\mathbf{x}$, ground-truth mask $\mathbf{y}$, and network model parameters $\boldsymbol{\Theta}$. This gradient is calculated over n input samples with respect to $\boldsymbol{\Theta}$, expressed as:

$$\nabla J(\mathbf{x}, \mathbf{y}, \boldsymbol{\Theta}) = \frac{1}{n} \sum_{i=1}^{n} \nabla J_i(\mathbf{x}, \mathbf{y}, \boldsymbol{\Theta}). \tag{4.12}$$

This gradient vector indicates by what amount the error would increase or decrease if a given weight is increased by a small differential.

The most common procedure for this calculation is stochastic gradient descent (SGD), also used in the proposed experiments (see discussion in Related Work). SGD averages the loss functions for each example in the training dataset, using a computationally-efficient approximation of the loss gradient used to optimize J. Whereas batch gradient descent computes the gradient using the whole dataset, SGD computes the gradient of a single instance or mini-batch.

For the current experiments, small mini-batch SGD was employed to provide better model generalization [186]. At each iteration of SGD, a mini-batch $\mathbf{K}$ of input examples

of size $k \ll n$ is uniformly sampled to use as random data instances. Model outputs and errors are first computed in the forward pass, which are then used to compute the average gradient for the batch. Once computed, the method then propagates these gradients back through the DCNN layers to adjust the filter weights accordingly [120]. At each network layer the error gradient was computed with respect to the output of each unit, which is the weighted sum of the error derivatives with respect to the total inputs to the units in the layer above. That is, the gradient averaged over the mini-batch was used to update network parameters $\mathbf{\Theta}$ in steps, as follows,

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \frac{\eta}{n} \sum_{i}^{k} \nabla J_i(\mathbf{x}, \mathbf{y}, \mathbf{\Theta}^{(t)}) \tag{4.13}$$

where η is the learning rate, a positive scalar that determines the size of the update step, selected based on a grid search. Effective learning rates are discussed in Implementation. Note that the parameter weights $\mathbf{\Theta}^{(t+1)}$ are adjusted in the opposite direction to the gradient vector. This procedure was repeated until a defined set of stopping criteria are fulfilled – i.e. once the average of the objective function stops decreasing.

To improve the performance of SGD, two deep learning heuristics were also introduced:

1. Step Scheduling was used to lower the learning rate η by a fixed amount after each epoch. The specific step value is discussed in Implementation. Step decay of the learning rate is claimed to help accelerate SGD convergence and reduce overfitting of the model [2].

2. The Adam (Adaptive Moment Estimation) algorithm [187] was used to optimize learning rates. Adam is an adaptive learning rate optimization algorithm developed for training deep neural networks that computes individual learning rates for different parameters. Adam was developed to bring together the advantages of the optimization algorithm RMSprop [188] [125] and SGD with momentum, and computes running averages of both the gradients and the second moments of the gradients to adjust learning rate.

4.5.4 Loss Functions

In the proposed approach, three loss functions were used and evaluated for the back-propagation computation: (1) **Cross-entropy loss**; (2) **Sørensen–Dice similarity coefficient**; and (3) **Focal Loss**. Loss is an indicator of the error incurred in the learned trainable parameters defined by $\mathbf{\Theta}$ in Equation 4.13.

4.5.4.1 Cross-entropy loss

Cross-entropy is a measure of the difference between two probability distributions, and can be defined as a loss function, L_{CE}, equal to the negative log-likelihood loss between the empirical distribution $\mathbf{y}$ defined by the training set, and the probability distribution $\hat{\mathbf{y}}$ predicted by the model [167]. L_{CE} loss increases as the predicted probability diverges from the actual label.

Let $\hat{\mathbf{y}}$ be the normalized predicted probabilities for each class $l \in \mathcal{L}$ for an input tensor $\mathbf{x}$, where $\mathcal{L}$ defines a set of m semantic class labels $\{l_1, ..., l_m\}$. Similarly, let $\mathbf{y}$ be the corresponding ground-truth class-encoding where $\mathbf{y} \in \mathcal{L} \subset \mathbb{Z}$. For multi-class classification, the ground-truth can be represented as a one-hot encoded tensor: a value of one for the true class and zeros for all other classes. L_{CE} loss is then defined as:

$$L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i}^{m} y_i \cdot log(\hat{y}_i) \tag{4.14}$$

Note that it is typically assumed that the tensor $\hat{\mathbf{y}}$ has unnormalized log probabilities that are normalized using the Softmax function (Equation 4.8), and so the cross-entropy loss for class i can be expressed as follows,

$$L_i = -\log\left(\frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}\right) \qquad \text{or equivalently} \qquad L_i = -\hat{y}_i + \log\sum_j e^{\hat{y}_j} \tag{4.15}$$

When using cross-entropy loss, the probability distribution of the pixel classes can greatly influence training accuracy [165], such that a balanced dataset generally provides improved overall classification performance compared to an imbalanced dataset [182]. L_{CE} assumes identical weight to all the samples and classes, and therefore requires a large training set with balanced classes to achieve good generalization. For unbalanced data, L_{CE} typically results in unstable training and leads to decision boundaries biased towards the dominant classes [169]. As discussed in Background, training sets sampled from DST.A and DST.B do exhibit severe class imbalance, with large variations in pixel counts in each class. Following Badrinarayanan, et al. [51], the variation in the loss calculation can be reduced through *class balancing*: applying weights to the computed loss differently based on the true class; hence, dominant classes in the training set given less weight than smaller, minor classes. The weighted cross-entropy loss, L_{WCE}, is defined as,

$$L_{WCE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i}^{m} w_i \cdot y_i \cdot log(\hat{y}_i) \tag{4.16}$$

where weight vector w of length m scales the losses by a formula proportionate to the pixel-class frequency in the dataset $\mathcal{D}$, defined by,

$$\mathbf{w}_{\mathcal{D}} = \frac{1}{log(1.02 + \mathbf{p}_{\mathcal{D}})} \tag{4.17}$$

where $\mathbf{p}_{\mathcal{D}}$ is the pixel-class probability vector of dataset $\mathcal{D}$. The value is then normalized by the largest weight value. This formula is based on results of the E-Net semantic segmentation model [171], and the analysis by Cui, et al. [169] on the effects of using inverse class proportionality as weight values, which is conventionally applied [169]. A "smoothed" inverse-log variant prevents blowup from very small classes in the loss calculation. These weights were computed for each training dataset beforehand during a pre-processing step, as discussed in Section 4.4.2.

4.5.4.2 Sørensen–Dice similarity coefficient loss

The second loss function uses a statistical validation metric called the Sørensen–Dice similarity coefficient (DSC), which is equivalent to the $F1$ score. DSC measures the similarity of two sets and is commonly used in semantic segmentation to quantify the overlap of outputs segmentations with ground-truth. Dice coefficient was introduced to computer vision researchers by Milletari et al. in 2016 for 3D medical image segmentation [172]. Similar to the widely-used Jaccard similarity index, or Intersection over Union (IoU), it quantifies the overlap between a ground-truth and predicted label mask normalized to the size of the mask.

Let $\mathbf{y}$ be the ground-truth mask where $\mathbf{y}(x) \in \mathcal{L} \subset \mathbb{Z}$, and $\mathcal{L}$ defines a set of m semantic class labels $\{l_1, ..., l_m\}$. Let $\hat{\mathbf{y}}$ be the corresponding predicted mask, with probabilities for each class $l \in \mathcal{L}$. Precision is defined as $P = |\mathbf{y} \cdot \hat{\mathbf{y}}|/|\mathbf{y}|$, and recall as $R = |\mathbf{y} \cdot \hat{\mathbf{y}}|/|\hat{\mathbf{y}}|$ of the pixel prediction. DSC is defined as the harmonic mean of P and R, expressed as,

$$\text{DSC}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2}{1/P + 1/R} = \frac{2TP}{2TP + FP + FN} = \frac{2|\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y}| + |\hat{\mathbf{y}}|} \tag{4.18}$$

where TP are True Positives, FP are False Positives, and FN are False Negatives. DSC weights each pixel error inversely proportionally to the size of the selected or relevant set rather than treating them equally. Hence, the Dice coefficient not only quantifies positives overlaps, but it also penalizes false positives, in a way similar to precision.

DSC is used to define a loss function known as soft Dice loss, since the predicted Softmax probabilities are used instead of first thresholding and converting them into a mask. Soft Dice loss is defined as unity minus the scalar mean DSC value summed over all classes $c \in \mathcal{L}$, denoted as L_{DSC}, as follows,

$$L_{DSC}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{1}{m} \sum_{c=1}^{m} \frac{2 \sum_i y_{ci} \hat{y}_{ci}}{\sum_i {y_{ci}}^2 + \sum_i \hat{y}_{ci}^2} \tag{4.19}$$

4.5.4.3 Focal loss

Focal loss (L_F), originally developed by Tsung-Yi Lin et al. [174] for dense object detection, was adapted for semantic segmentation and used as an additional loss function for the current experiments. Focal loss adds a weighting term $-\alpha_t(1 - p_t)^\gamma$ to the standard cross-entropy loss, where p_t represents the estimated probability for a predicted semantic class $t \in \mathcal{L}$, γ is a scalar focus constant, and $-\alpha_t$ represents a scaling constant. Focal loss can therefore be expressed as follows,

$$L_F = -\alpha_t(1 - p_t)^\gamma L_{CE} \tag{4.20}$$

The chief purpose of using focal loss is to counteract problems encountered with cross entropy loss becoming ineffective during model training using highly imbalanced datasets [174]. In such cases, easily classified negatives comprise the bulk of the losses and dominate the gradient. L_F attempts to down-weight the contribution of easy examples so that the DCNN focuses more on hard examples.

4.5.4.4 Weighted Multi-loss

Given loss functions L_{WCE}, L_{DSC} and L_F, the proposed back-propagation uses a combined loss value that weights each loss by coefficients α, β, γ, respectively, as follows:

$$\mathrm{L} = \alpha \cdot L_{CE}\left(\mathbf{y}, \hat{\mathbf{y}}\right) + \beta \cdot L_{DSC}\left(\mathbf{y}, \hat{\mathbf{y}}\right) + \gamma \cdot L_F\left(\mathbf{y}, \hat{\mathbf{y}}\right) \tag{4.21}$$

This combined loss is intended to compensate for some of the limitations in using cross-entropy for semantic segmentation – the conventional approach. Specifically, cross-entropy calculates loss at each pixel independently, which is not suitable for dense probability maps that contain semantic relations among pixels [119]. DSC loss compensates by considering the total number of segment pixels at global scale (denominator), while the also considering the overlap between the two sets at local scale (numerator).

4.5.5 Model testing

To evaluate model performance (see evaluation criteria and methods described in Section 4.7), test segmentation outputs were generated from trained models using tiles extracted from individual images in DST.A.2, DST.B.2 and DST.C (see Appendix A). Test images were selected to be within 20 to 40 megapixels in size. The proposed testing method consisted of the following steps:

1. **Tile extraction**: Input images were resized slightly, and top pixels cropped, in order to adjust the image to multiples of the tile dimensions. Extraction (unfolding) also used a 256 pixel stride that allowed for a significant overlap between tiles. This overlap was required to allow for the interpolation of class probabilities between adjacent tiles during image reconstruction.

2. **Model inference**: Segmentation mask tiles were generated using the trained U-net and Deeplabv3+ models.

3. **Mask reconstruction**: Predicted tiles were assembled into an integrated mask of the same dimensions as the test image. Since each tile prediction was evaluated independently of the predictions of adjacent tiles, important contextual information of neighbouring pixel classes was lost during training; hence, simple edge-to-edge stitching of the tiles resulted in discontinuities at tile boundaries. A weighted average of the probabilities over the overlap area allowed for a blending or smoothing of final pixel classifications. This approach is based on the U-net overlap-tile strategy, which employs a similar method to adjust for the padded cropping of input images during the encoding phase.

4. (Optional) **CRF post-processing**: Conditional Random Fields post-processing is discussed in Section 4.6.

4.6 Conditional random fields

4.6.1 Overview

Conditional Random Fields (CRFs) are a class of discriminative probabilistic graphical models used for structured prediction [101]. For semantic segmentation applications, CRFs are particularly effective at modelling the decision boundaries between different pixel classes. On their own, CRFs have attained top performance in previous segmentation competitions, such as PASCAL VOC 2010 [68]. CRF modelling has also been shown

to significantly improve the accuracy of DCNN dense predictions [39] [3] [104] [154] [34] [103] [155], which can be attributed to the combination of CRF modelling power with the representation-learning ability of DCNNs [104]. Techniques for combining these two methods range from introducing CRFs as an independent stage of the segmentation pipeline to embedding these models within the network itself, such as Deeplab [2] [105]. In the current experiments, CRFs were used as post-processing filters applied to the reconstructed output segmentation unary data to boost performance of the model.

4.6.2 Definition

Consider an input image tensor $\mathbf{x}$, where each pixel in the image is associated with a segmentation mask $\mathbf{y}$ that ranges over a finite set of m pixel-level semantic class labels $\mathcal{L} = \{l_1, ..., l_m\}$. The joint output variable $\mathbf{Y} \in L$ can be defined over variables $\{Y_1, ..., Y_N\}$ that ranges over the possible labels, and a random field $\mathbf{X}$ defined over variables $\{X_1, ..., X_N\}$ that ranges over possible input images of size N. For any pixel u, X_u is taken to be the pixel's color vector, and Y_u is the label assigned to the pixel.

These random variables $\mathbf{X}$ and $\mathbf{Y}$ are jointly distributed, and are used to construct a conditional model $P(\mathbf{Y}|\mathbf{X})$ from paired input images $\mathbf{x}$ and semantic labels $\mathbf{x}$ [102]. Let $G(E, V)$ be an undirected graph with vertices V and edges E, such that $\mathbf{Y} = \mathbf{Y}_v(v \in V)$, which means $\mathbf{Y}$ is indexed by the vertices in G, the values of which range over the possible discrete labels in $\mathcal{L}$. It is also assumed that $\mathbf{Y}$ obeys the Markov property which holds that the conditional probability distribution given its adjacent nodes is independent of the rest of the nodes in the graph [107]. By conditioning the graph G on $\mathbf{X}$ globally, a conditional random field $(\mathbf{Y}|\mathbf{X})$ can be defined such that the values of random variables in $\mathbf{X}$ are fixed or given, and all of the random variables in Y follow the Markov property $P(\mathbf{Y}_u|\mathbf{X}, \mathbf{Y}_v, u \neq v) = P(\mathbf{Y}_u|\mathbf{X}, \mathbf{Y}_u, u \sim v)$ [102], where $u \sim v$ indicates that u is a neighbour of v.

The CRF $(\mathbf{Y}|\mathbf{X})$ can then be characterized as a Gibbs distribution where each clique $c \in G$ in a set of cliques $C_G \in G$ induces some cost ϕ_c [39], such that

$$P_\phi(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} exp\Big(- \sum_{c \in C_G} \phi_c(\mathbf{Y}_c|\mathbf{X}) \Big) \tag{4.22}$$

where ϕ_c is an unnormalized measure and $Z(\mathbf{X})$ is a normalization constant [102].

The Gibbs energy of an image labeling $\mathbf{y} \in \mathcal{L}$ can be expressed as,

$$E(\mathbf{y}|\mathbf{X}) = exp\Big(- \sum_{c \in C_G} \phi_c(\mathbf{Y}_c|\mathbf{X}) \Big) \tag{4.23}$$

For a given pixel u, CRFs incur two forms of costs or potentials: unary and pairwise. Unary costs represent relations between labels and local image features, and the CRF uses the pixel class probabilities produced by model as its unary cost. A unary cost $\phi_u(Y_u = y|\mathbf{x}_i)$ is therefore incurred to assign category y for some image $\mathbf{x}$ [107] [103]. A pairwise cost is used by the CRF to model relationships between pixels, and is incurred to assign a pair of labels (y_u, y_v) to a pair of pixels (u, v), where $u \in V$. Given an image $\mathbf{x}$, the pairwise cost function is defined as $\phi_{u,v}(\mathbf{Y}_u = y_u, \mathbf{Y}_v = y_v|\mathbf{X})$. Pairwise potentials introduce penalties for nearby similar pixels that are assigned different labels.

Unary and pairwise costs can be combined to reframe the model as an energy minimization problem for CRFs, where the energy is given by:

$$E(\mathbf{y}, \mathbf{X}) := \sum_u \phi_u(\mathbf{Y}_u = y|\mathbf{X}) + \sum_{u \neq v} \phi_{u,v}(\mathbf{Y}_u = y_u, \mathbf{Y}_v = y_v|\mathbf{X}) \tag{4.24}$$

where the objective is to optimize the CRF model by minimizing the costs incurred in pixel misclassification. CRF potentials incorporate smoothness terms that maximize label agreement between similar pixels, and can also include more complex terms that model contextual relationships between semantic classes [39].

4.6.3 Dense CRF

With a fully-connected CRF model (or Dense Random Fields), each pair of image pixels u and v has one pairwise term regardless of their mutual distance; therefore, the graph G is fully-connected. Such models therefore have all pairs of variables directly connected by pairwise potentials. The fully-connected CRF developed by Krähenbühl and Koltun (2011) [39], which was also integrated in DeepLabv3+ [2] was selected for experimentation based on evidence that dense connectivity at the pixel level substantially improves segmentation and labeling accuracy [39] [3] [104] [2] [105].

This model first computes the unary costs $\phi_u(x_u) = -log(P(x_u))$, where $P(x_u)$ is the label assignment probability at pixel u as computed by DCNN. The second term is the pairwise potential defined as,

$$\phi_{u,v}(u, v) = \mu(u, v) \sum_{m=1}^{m} w^{(m)} k^{(m)}(\mathbf{f}_u, \mathbf{f}_v) \tag{4.25}$$

where $k^{(m)}$ is the Gaussian kernel

$$k^{(m)}(\mathbf{f}_u, \mathbf{f}_v) = \exp\left(-\frac{1}{2}(\mathbf{f}_u - \mathbf{f}_v))^T \Lambda^{(m)}(\mathbf{f}_u - \mathbf{f}_v) \right) \tag{4.26}$$

where $\mathbf{f}_u$ and $\mathbf{f}_v$ are feature vectors extracted for pixels u and v in an arbitrary feature space, and which are weighted by linear combination $w^{(m)}$. Finally, $\mu(u, v)$ is a label compatibility function to select non-self pixels. At its most basic, $\mu(x_u, x_v) = 1$ if $x_u \neq x_v$ and zero otherwise, as defined by the Potts model[189]. Each kernel $k^{(m)}$ is also characterized by a symmetric, positive-definite precision matrix $\Lambda^{(m)}$, which defines its shape [39] [3]. This introduces a penalty for nearby similar pixels that are assigned different labels. However, the simple Potts model is insensitive to compatibility between labels [3]. For example, this model will penalize a pair of nearby pixels assigned labels "Coniferous" and "Herbaceous / Shrub" to the same extent as pixels labeled "Coniferous" and "Water". To improve segment classification, the CRF model can learn a more complext data-specific symmetric compatibility function $\mu(x_u, x_v)$ that accounts for different interactions between labels.

For multi-class image segmentation and labeling, Krähenbühl and Koltun [39] [3] proposed contrast-sensitive two-kernel potentials, defined in terms of the color vectors I_u and I_v and positions p_u and p_v:

$$k^{(m)}(\mathbf{f}_u, \mathbf{f}_v) = w^{(1)} \exp\left(-\frac{|p_u - p_v|^2}{2\theta_\alpha^2} - \frac{|I_u - I_v|^2}{2\theta_\beta^2}\right) + w^{(2)} \exp\left(-\frac{|p_u - p_v|^2}{2\theta_\gamma^2}\right) \tag{4.27}$$

The first term is labelled the "appearance kernel," which factors in the observation that neighbouring pixels with similar color are likely to be in the same class. The degrees of nearness and similarity are controlled by parameters θ_α and θ_β, the pixel location and RGB standard deviations, respectively. The second term, labelled the "smoothness kernel," attempts to remove small isolated regions. Parameters θ_β and θ_α can be optimized through a grid search based on the Dice loss of inferred filtered masks with ground-truth across the validation set $\mathcal{D}_{valid}$. The compatibility parameters $\mu(u, v)$ can be learned using L-BFGS algorithm[190] to maximize the log-likelihood $\ell(\mu : \mathcal{D}_{valid}, \mathcal{L}_{valid})$ of the model where $\mathcal{D}_{valid}$ is a validation dataset with corresponding ground-truth masks $\mathcal{L}_{valid}$. L-BFGS requires the computation of the gradient of ℓ, but which requires the mean field approximation described in Krähenbühl and Koltun [39] [3], in which the following approximation of the gradient for each training image is derived:

$$\frac{\partial}{\partial \mu(u, v)} \ell(\mu : \mathcal{D}^{(n)}, \mathcal{L}^{(n)}) \approx -\sum_i \mathcal{L}_i^{(n)}(u) \sum_{j \neq i} k^{(m)}(\mathbf{f}_u, \mathbf{f}_v) \mathcal{L}_j^{(n)}(v) + \sum_i Q_i(u) \sum_{j \neq i} k^{(m)}(\mathbf{f}_u, \mathbf{f}_v) Q_j(v) \tag{4.28}$$

where $(\mu : \mathcal{D}^{(n)}, \mathcal{L}^{(n)})$ is a single image and associated mask, and binary mask $\mathcal{L}_j^{(n)}(v) \in \mathcal{B} = \mathbf{B}_{d_1 \times d_2}(\{0, 1\})$.

4.7 Evaluation Criteria and Metrics

The evaluation of the proposed approach outlined above is based on a comparison of the predicted segmentation maps generated from the network models with the manually-created ground truth segmentation maps in the test datasets (DST.A.2, DST.B.2 and DST.C). This comparison uses multiple metrics for the measurement of model performance and quality for the task of dense classification of landscape imagery. Specifically, the evaluation of the experimental results applies a set of four criteria each with corresponding metrics: (**1**) Generalization, (**2**) Accuracy, (**3**) Sensitivity, and (**4**) Efficiency. These criteria were used both to assess the DCNN model performance, and to provide an interpretative framework for the experimental results. Given the distinct characteristics and specific visual features of the Mountain Legacy Project datasets (DST.A and DST.B), standard segmentation benchmarks, such as the PASCAL Visual Object Classes (PASCAL VOC) [191] were not used to evaluate model performance.

4.7.1 Generalization [EVAL.1]

Generalization describes the property of a model to automatically learn the discriminative features for classification in new, unseen images with minimal engineering of the feature extraction method. For a properly generalized DCNN model, the gap between empirical and expected error goes to zero with increasing size of the training set [192]. Greater generalization is indicated if the model is: (1) able to minimize training error; and (2) able to minimize the discrepancy between training and validation error.

Loss values were also used to diagnose model overfitting, underfitting or anomalies in the training process, using the following heuristics to assess performance: **Underfitting** is indicated when a low training error cannot be attained. The threshold minimum loss for underfitting was determined empirically by multiple experiments. **Overfitting** is indicated when the gap between the training error and validation error is too large. The expected validation error should instead be greater than or close to the expected value of training error [167]. Overfitting indicates the model is conforming to the training data as opposed to generalizing from patterns or features observed in the data. Consequently, overfitting leads to loss of generality. When only a small training dataset is available, overfitting can become a critical issue [165].

Metrics: During model training, losses were recorded at regular intervals to gauge how well the model was learning (encoding and decoding) the features of the input data (see Section 4.5 for a discussion on model loss functions). The following metrics were recorded and averaged over the entire training and validation partitions of the extracted and augmented databases (Note that training/validation dataset proportions follow an 80/20 split):

1. **Weighted Cross-entropy Loss (L_{WCE})**: See Section 4.5.4.1
2. **Dice Similarity Coefficient Loss (L_{DSC})**: See Section 4.5.4.2
3. **Focal Loss (L_F)**: See Section 4.5.4.3

4.7.2 Accuracy [EVAL.2]

Accuracy encompasses metrics that gauge how well the model is able to output fine-grained dense predictions that correctly infer every pixel class. Accuracy was evaluated during both model validation and testing using multiple metrics for the average similarity of output segmentations, however model performance metrics reported are based on outputs from test images in DST.A.2, DST.B.2 and DST.C, which were reserved from the main datasets. Note that the evaluation and analysis of individual image segmentations focused specifically on the outputs of DeepLabv3+ network architectures for the specified model configurations.

Metrics: The overall classification accuracy of models was evaluated using standard metrics of precision and recall (defined below). Precision and recall are sensitive to over and under-segmentation, where over-segmentation results in low precision scores, while under-segmentation results in low recall scores. More refined accuracy metrics for segmentations were also used to measure the number of pixels common between the target and prediction masks normalized by the total number of pixels in both masks. To evaluate average accuracy over the test dataset, F1 Scores were calculated for each class and then averaged over all classes; averaged frequency-weighted Intersection over Union ($fIoU$) Matthews correlation coefficient (MCC)[193] were also computed. Since an averaged F1 score can hide challenging cases, per-image scores were also measured to provide a representative range of example cases. Per-image scores reduce the bias with respect to dominant pixel classes, as missing or incorrectly segmented small objects have a lower impact on the global confusion matrix.

1. **Standard metrics**:

(a) True Positives (TP): The number of correctly classified pixels belonging to a certain class l.

(b) True Negatives (TN): The number of correctly classified pixels that do not belong to the class l.

(c) False Positives (FP): The number of pixels that do not belong to the class l but are predicted as that class.

(d) False Negatives (FN): The number of pixels that belong to the class l but are not predicted as that class.

(e) Precision (P): The ratio of the number of true positive class predictions to the number of times the model predicted a positive label in total.

$$\mathrm{P}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{TP}{TP + FP} \in [0, 1] \tag{4.29}$$

(f) Recall (R): The ratio of the number of true positive class predictions to the number of actual positives predicted by the model plus the number of positives incorrectly predicted as negative by the model.

$$\mathrm{R}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{TP}{TP + FN} \in [0, 1] \tag{4.30}$$

(g) Weighted average of precision, recall and F1-scores where the weights are the support values

(h) Support (per class): Number of occurrences (pixels) of each class in the ground-truth data.

2. **F1 Score ($F1$) or Dice similarity coefficient** F1 score, or the equivalent Dice similarity coefficient (DSC), was used as a statistical validation metric to evaluate the performance of both the reproducibility of manual segmentations, and the spatial overlap accuracy of automated probabilistic fractional segmentation of MLP dataset images. During validation, the F1 score is calculated for each class individually and then averaged over all classes to provide a global, mean score.

$$\mathrm{F1}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2TP}{2TP + FP + FN} = \frac{2|\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y}| + |\hat{\mathbf{y}}|} \in [0, 1] \tag{4.31}$$

Refer to Section 4.5.4.2 for a formal definition of DSC. During model testing, mean F1 scores are computed for each class and over all classes. Per-image testing computes class/overall mean $F1$ scores for that image.

3. **Frequency-weighted Intersection over Union ($fIoU$)**: The $fIoU$ metric uses inverse frequency weighting where pixel counts for a given class are weighted by the inverse of the class frequency. Given that a global, mean DSC measure is biased

towards object incidences that cover a large image area, $fIoU$ is more sensitive to classification of less frequent (minor) classes.

$$fIoU = \frac{1}{\sum_{i=1}^{k} t_i} \sum_{i=1}^{k} \frac{t_i \cdot n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \in [0, 1] \tag{4.32}$$

where n_{ii} denotes the number of pixels in class i predicted to be in class i (true positives); n_{ij} denotes the number of pixels of the class i predicted to belong to class j (false positives); and t_i is the total number of pixels of class i in ground truth segmentation [29][68]. Per-image testing computes class/overall mean $fIoU$ scores for that image.

4. **Matthews correlation coefficient (MCC)** The MCC [193] is used in machine learning as a balanced measure of the quality of multi-class segmentations that incorporates TP and TN counts appropriate for class imbalanced datasets. The value of MCC ranges between -1 and +1, where a coefficient value of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. MCC is defined as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \in [1, -1] \tag{4.33}$$

4.7.3 Sensitivity [EVAL.3]

Sensitivity describes the neural network's ability to extract deep features that are robust to input perturbations [194], which includes image variation due to weather and photometric conditions, occlusions, noise and photographic distortions, as described in Chapter 2.3.3. To assess model sensitivity, segmentations generated during model testing using images sampled from DST.A.2, DST.B.2 and DST.C datasets were compared to the ground-truth masks.

For the purposes of this study, sensitivity is measured by error rates for two main classification errors:

1. **Intra-class difference**: This error occurs when the same pixel classes imaged under different conditions may appear different. For example, cloud shade can significantly alter the texture and intensity of common land covers; similarly, fog or cloud-obscured regions can create intra-class variation. This difference is particularly relevant in evaluating the model's ability to classify close-range foreground

pixels as NC (not classified), while also classifying the same land-cover class when present in the background.

2. **Inter-class similarity** This error occurs when different pixel classes imaged under certain conditions appear similar. For example, the gradual transition from perennial species to woody species characterized by regrowth vegetation makes it difficult for experts to distinguish different classes of forest vegetation [47][195]. Similarly, discrimination between regenerating areas, broadleaf/mixedwood (B-MW) and grassland herbaceous areas (H-S), has previously proven difficult when manually segmenting images [12].

Separate classification procedures, and therefore separate segmentation models were developed for historic single-channel images, and for repeat colour images. Not only did this design decision reflect the distinct visual and technological conditions of each capture type (e.g., historic glass plate emulsion versus modern digital camera) [5], MLP research had previously demonstrated that colour information from the modern images appeared to enhance segmentation discrimination in pairs of grayscale images [12]. Inversely, stripping colour from repeat images introduced significant error in the classification of two key areas: the broadleaf/mixedwood (B-MW) and grassland herbaceous (H-S) areas [12]. A model invariant to colour channel information was therefore not investigated. However, grayscaled repeat images proved an effective source of training samples for the historic model, since conversion from RGB to single channel provides sufficient channel information for class segmentation.

Metrics: Intra-class difference and inter-class similarity were evaluated using two main metrics: (1) Performance metrics defined in EVAL.2; and (2) Visual inspection of test samples that exemplify "hard cases", or that illustrate how the model accurately predicts specific pixel classes, or misclassifies classes in a systematic way. Examples for visual inspection apply the following sources of error or perturbation. Note that these sources are not intended to exhaustively cover all possible image errors and distortions, but to highlight some of the observed conditions surrounding misclassifications.

1. **Ground-truth Errors**: Misclassifications due to manual segmentation errors or inconsistencies. See Section 2.3.3 for further information.

2. **Photographic Conditions**

   (a) **Photometric:** Due to variation in illumination (i.e. sunlight or shadows), or weather conditions that can introduce both intra-class and inter-class errors that affect classification.

(b) **Occlusion** Due to opaque or semitransparent objects in the field of view that obstruct, obscure or divide continuous segments. Such occlusions can occur due to prevailing conditions (e.g., smoke or fog), or foreground vegetation [6]. The model is trained to ignore small and semitransparent occlusions.

(c) **Scale:** Due to differences in scale of same pixel class texture (e.g., foreground versus background representations). Vegetation classes at different scales can introduce both intra-class and inter-class variation that can affect proper classification. (See discussion under "Foreground/Background Representation" in Section 2.3.3).

(d) **Rotation (Not Evaluated)** Due to tilting of the camera or digital scanner. This attribute is not evaluated, however camera azimuth, elevation, and elevation angle measurements are available for future analysis. Tilting of the digital scanner is assumed to be negligible for samples in the test dataset.

(e) **Translation (Not Evaluated):** Due to slight panning of camera or distortion introduced during image registration. This attribute is not evaluated as DCNN networks are assumed to be intrinsically translation-invariant, given that their basic components (convolution, pooling, and activation functions) operate on local input regions, and depend only on relative spatial coordinates [29].

(f) **Perspective (Not Evaluated):** Due to variation in lens position, viewpoint or view angle (azimuth) in oblique photographs. This attribute is not evaluated, however azimuth, elevation, and elevation angle measurements are available for future analysis.

(g) **Blur (Not Evaluated)** Due to rapid movement of objects or camera, or to long exposure. Camera exposure time, foreground/background focus, and visibility can relate to weather conditions [196] [197]. This attribute is not evaluated as camera blur is assumed to be negligible in test image samples.

3. **Image Quality**

(a) **Noise:** Due to photographic artefacts, digitization artefacts, scratches, localized emulsion defects, "salt and pepper", and other occasional but visible damages to the glass plate negatives for the historic photos [5]. These errors are problematic for historic photographs but negligible for repeat images.

(b) **Poor Resolution (Not Evaluated):** Due to low resolution that leads to the disappearance of the textural characteristics of pixel classes needed for classification [197]. Sufficient image resolution is critical for accurate classification using high-resolution input MLP models. Given that the very-high resolution images (20 to 40 megapixels) in the test dataset, low resolution

was not a significant factor in model sensitivity, and so this attribute was not evaluated.

4.7.4 Efficiency [EVAL.4]

Reducing computational complexity and memory usage of a neural network is important to minimize training and inference time and power consumption for devices. For this study, evaluation of model efficiency is simply based on training time to convergence (backward pass on the learning model), and for model testing, the latency of inference (forward pass on the fully trained model) given standard-sized test images. Though efficiency was not a primary consideration in the experimental design, some measurements of elapsed training and inference times were recorded for comparison of different input parameters and training datasets.

Metrics:

1. **Training Latency:** Elapsed time (hours) to train DCNN model of a given database to convergence with minimum validation loss over n_{iter} iterations of 20 batches, where each batch has 8 tile samples.
2. **Average Inference Latency:** Elapsed time (seconds/tile) to complete model inference and mask reconstruction for a given input image per sample tile of a fixed size. Note: does not include CRF inference time.

4.8 Summary

The proposed method for this study uses deep convolutional neural networks (DCNNs) to map land cover types in oblique ground-level photographs. The training dataset consists of historic photographs and modern repeats sampled from the Mountain Legacy Project collection, a vast archive of ground-level photographs of Canada's western mountains. To mitigate limited labeled data, semantic class imbalance, and appearance variation of the images, a novel threshold-based data augmentation was introduced as a pre-processing step. As well, the application of three different loss functions was evaluated for improvements to the prediction of underrepresented classes. The application of a conditional random fields (CRFs) model was evaluated as a post-processing step to boost accuracy of segments in DCNN outputs. The evaluation framework used applies four criteria (Generalization, Accuracy, Sensitivity, and Efficiency) in the comparison the predicted segmentation maps of the DCNN network models with the manually-created ground truth segmentation maps.

Chapter 5

Implementation

5.1 Overview

This chapter presents the implementation specification of the experiments. As discussed in Methodology, the aim of these experiments was to optimize two deep learning architectures for the semantic segmentation of oblique landscape-level photography. Section 5.2 presents the experimental setup for image preprocessing, training and testing stages of the experiments. Separate segmentation experiments were performed on historic (grayscale) and repeat (colour) capture images. Sections 5.1 and 5.2 present the configuration specifications of the tested U-net [1] and Deeplabv3+ [2] network architectures, respectively. Section 5.5 describes the parameter optimization and inference steps to the conditional random field post-processing.

5.2 Experimental Setup

5.2.1 Development and Testing Environment

All DCNN models and preprocessing utilities were implemented in PyTorch, an open source Python [198] library based on the Torch library and developed by Facebook's AI Research lab [199], and OpenCV, a library of programming functions developed for computer vision. [181].

All training was performed on Compute Canada's Cedar GPU cluster located at Simon Fraser University.[1] For most experiments, model training used four NVIDIA P100 Pascal (12G HBM2 memory) GPUs. Training time for models ranged from approximately 16-24 hours for up to 40 epochs (see also latency measurements in Table 6.10).

All test results were computed on a 2.2 GHz Intel Core i7 (Kabylake) MacBook Pro; single processor with six cores. Conditional random field tests were also computed on the MacBook processor.

5.2.2 Preprocessing: Tile Extraction

Tile subimages were extracted and stored in a high-performance database for model training. Extracted tiles were selected as square 512 × 512 pixel subimages, which allowed for (on average) each raw full-sized image to be decomposed into approximately 100 tiles. In addition to the full-sized images, extractions of downsampled versions, scaled by 0.2 and 0.5, were incorporated in the database. Extracted tiles were also augmented using data augmentation. See Table 5.1 for a summary of the the extraction and augmented databases.

Each DCNN model implementation loads image tile batches using the Hierarchical Data Format (HDF5) binary data format accessed using the h5py Python library interface [200]. HDF is particularly suited for storing multi-dimensional arrays together with metadata in a hierarchical structure, and supports parallel I/O for efficient reads of high volume and complex data. The HDF Single-Writer/Multiple-Reader (SWMR) feature further enables multiple data file reads without requiring locks or inter-process communication.

Earlier diagnostic tests confirmed that database reads and stores presented a critical performance bottleneck during model training. To improve performance, a parallelized image data buffer was designed and implemented resulting in a speed-up of approximately 60% over hash-based retrieval. The buffer can load up to 1,000 training samples (tiles) in a single memory allocation, which were defaulted to contiguous allocation (laid out on disk in traditional C order) enabling fast sequential access. Direct reads from an HDF5 dataset into a NumPy array also avoids creating intermediate copies of the data [200].

5.2.3 Preprocessing: Data Augmentation

Data augmentation was applied to the training data using the proposed oversampling method described in Section 4.4.2.2. An analysis of the pixel class distributions of the combined extraction dataset (i.e., DST.A.1 and DST.B.1 extractions) helped to identify minor (underrepresented) and dominant (overrepresented) semantic classes. This analysis is reported in Background, and summarized in Figures 2.2 and 2.3, as well as Tables 5.2 and 5.3 below. The between-class imbalance shown in the charts highlights three

Capture	ID	Database	Size	JSD [4.4.2.2]	M2 [4.4.2.2]
Historic	**H-Ext**	Extracted	8393	0.2133	0.7127
	H-Aug	Augmented	12358	0.1169	0.8593
	H-Mrg	Merged	24396	0.1165	0.8584
Repeat	**R-Ext**	Extracted	8108	0.2252	0.7023
	R-Aug	Augmented	12038	0.1224	0.8559

TABLE 5.1: MLP training and testing image databases. Databases are created from tile extractions of training images in datasets DST.A.1 and DST.B.1.

deficiencies of the dataset that typically prove problematic for supervised learning: (1) undersampling of critical minor classes; (2) oversampling of the non-categorized class; and (3) limited labeled data.

For the historic capture images, severe undersampling of minor classes B-MW, WL, WT, S-I ($< 1\%$) was found, as well as oversampling of the noncategorized (NC) class ($> 50\%$), which accounts for more than half of the pixels in the training samples (see Figure 2.2). repeat capture images showed a very similar profile with the same minor and dominant classes (see Figure 2.3).

Class	H-Ext	H-Mrg	% Change	Weight
Non-categorized (NC)	0.5580	0.3944	-29.2	0.0708
Broadleaf/Mixedwood (B-MW)	0.0035	0.0339	855.4	0.4677
Coniferous (C)	0.1832	0.2332	27.2	0.1088
Herbaceous/Shrub (H-S)	0.0860	0.0757	-12.1	0.2687
Sand/Gravel/Rock (S-G-R)	0.1170	0.1078	-7.9	0.2041
Wetland (WL)	0.0093	0.0418	348.3	0.4094
Water (WT)	0.0011	0.0049	340.6	1.0000
Snow/Ice (S-I)	0.0234	0.03411	42.2	0.4660
Regenerating Area (RA)	0.0183	0.0742	305.8	0.2727

TABLE 5.2: Class probability distributions for the historic capture databases (extracted versus combined DST.A, DST.B, merged with grayscaled repeat captures, augmented). Weights calculated using Equation 4.17. Total number of H-Mrg samples: 24,396. LCC.A categorization.

Sample rates for augmented tiles were calculated for each training database (see Section 4.4.2.2). Given an augmentation size limit of 4,000, and a maximum sample rate of 7, the repeat extracted database (R-Ext) was found to have a τ of 1.45 and ρ coefficient of 4, resulting in 3930 augmented samples. This reduced the distribution JSD value in the resultant database (R-Aug) from 0.2252 to 0.1223 (a 45.7% reduction), while increasing the M2 normalized variance from 0.7023 to 0.8559 (a 21.8% increase). The historic extracted database (H-Ext) was found to have a τ of 1.3 and a ρ coefficient of 4, resulting in 3965 augmented samples. Data augmentation reduced the distribution JSD value from 0.2133 to 0.1165 (a 45.4% reduction) for H-Aug, while increasing the M2 normalized variance from 0.7127 to 0.8593 (a 20.6% increase). With the addition of

Class	R-Ext	R-Aug	% Change	Weight
Non-categorized (NC)	0.5568	0.3909	-29.8	0.0695
Broadleaf/Mixedwood (B-MW)	0.0095	0.0493	419.4	0.3572
Coniferous (C)	0.2190	0.2431	11.0	0.1025
Herbaceous/Shrub (H-S)	0.0613	0.0638	4.1	0.2974
Sand/Gravel/Rock (S-G-R)	0.1155	0.1149	-0.5	0.1891
Wetland (WL)	0.0082	0.0388	370.8	0.4190
Water (WT)	0.0008	0.0042	401.0	1.0000
Snow/Ice (S-I)	0.0080	0.0203	155.9	0.6060
Regenerating Area (RA)	0.02103	0.0748	255.5	0.2643

TABLE 5.3: Class probability distributions for the repeat capture databases (extracted versus augmented database for combined DST.A, DST.B). Weights calculated using Equation 4.17. Total number of R-Aug samples: 12,038. LCC.A categorization.

the grayscaled copies of R-Aug to H-Aug resulted in a negligible shift in the JSD and M2 values.

5.2.4 Model training

Two network architectures: U-net [1] and Deeplabv3+ [2], were trained on combined DST.A.1 and DST.B.1 extraction and augmented databases. Different input database versions were tested to benchmark performance with and without data augmentation (see Table 5.1). Both models use several hyperparameters that govern the training process. Optimization of these values required a grid search over multiple test experiments. The following is a brief description of each hyperparameter:

1. **Learning Rate (η)**: The step size in the gradient descent algorithm. This parameter controls adjustments to the network weights according to the loss gradient. Problems occur when the learning rate is too small or too large. In practice a suitable learning rate is often found only after multiple experiments.

2. **Batch Size:** The number of sample tiles used in a single training iteration. Batch size is calibrated to the available GPU memory, since training the network using fewer samples requires less memory. Smaller batch sizes can lead to less accurate results, as variance is greater than large mini-batches.

3. **L2 Weight Decay:** L2 regularization is a classic method to reduce overfitting that adds a penalty to the weight updates θ equal to the sum of the squares of the parameter weights $\mathbf{\Theta}$, scaled by a weight decay coefficient μ_d. Hence, μ_d effectively subtracts a diminishing portion of the weight at each step.

4. **Dropout** Dropout layers [140] prevent overfitting and enhance the performance of a CNN classifier. Two dropout layers have been used between the fully connected networks to avert overfitting with probability p = 0.2.

5. **Kernel Size** The default (original) U-net and Deeplabv3+ kernel sizes have been used in both network architectures. Kernel sizes determine the number of trainable parameters which in turn affects the performance of a CNN architecture.

The following specifications apply to all experiments:

Networks were trained from scratch using the tile extraction database described in Section 4.4. For all experiments, the training/validation dataset split ratio was consistently set to 80/20.

Input image data was normalized between $[0, 1]$ in order to centre the dataset pixel intensities with a mean at 0, which can help avoid exploding or disappearing gradients. A z-score normalization for pixel intensities used the following pixel mean ($\bar{x}$) and standard deviation (σ) values:

1. **H-Ext** $\bar{x}$: 142.22, σ: 25.73
2. **H-Mrg** $\bar{x}$: 142.01, σ: 23.66
3. **R-Ext** [RGB] $\bar{x}$: [143.00, 156.63, 164.54], σ: [23.07, 20.12, 17.10]
4. **R-Aug** [RGB] $\bar{x}$: [132.47, 144.47, 149.45] σ: [24.85, 22.04, 18.77]

Batch Normalization[153] (BN) was applied after each convolutional layer. BN also maintains running estimates of its computed mean and variance during training, which are then used for normalization during evaluation. The running estimates are kept with a default momentum of 0.1.

5.2.5 Model testing

Segmentation maps were generated for test datasets DST.A.2, DST.B.2, and DSC.C using trained Pytorch models. Model performance metrics were calculated on the final thresholded outputs of the reconstructed versions.

5.3 U-net implementation

The U-net model is a standard DCNN structure for biomedical semantic segmentation, first developed using only a limited dataset of annotated images that can train within a reasonable time [1] (Refer to Section 3.4.4 for a background discussion). Suitable for very high-resolution images, U-net employs an overlap-tile strategy that was adapted to the MLP dataset for classification of oblique landscape images using modified versions of existing Pytorch library implementations available online. The proposed U-net implementation was derived from an existing Pytorch implementation available on Github[2]. The architecture is shown in Figure 5.1. Note that U-net's feedforward DCNN architecture is restricted to a fixed-sized input mapped to a fixed-sized output (including cropping during the forward pass). The total number of trainable network parameters was approximately 28M.

The following U-net network configuration was used for both historic and repeat image databases.

- Network weights were initialized according to the method described in He, K. et al. (2015) [201] using a uniform distribution.
- The models were trained for up to 40 epochs, depending on early stopping criteria (i.e. stagnant validation accuracy).
- Adam learning rate (LR) optimization [187] was used with an initial LR determined through grid search for each database. The initial learning rate varied from 5e-05 to 0.001, depending on the size of the training dataset. A step learning rate scheduler with a learning rate decay of $\gamma = 0.9$ was applied at every epoch.
- A batch normalization layer [153] was applied after each convolution and before activation.
- Convolutional layers used ReLU nonlinear rectification, although some experiments used SELU.
- Batch sizes were limited to 10 due to GPU memory constraints on the 512 pixel $\times$ 512 pixel input tiles.
- The training was regularized by weight decay (the L2 penalty multiplier set to 0.0005) and dropout regularization (dropout ratio set to 0.5).
- The applied loss function was computed as the sum of the weighted cross-entropy loss L_{WCE}, mean Dice coefficient loss L_{DSC}, and focal loss L_F for each spatial position in the DCNN segmentation map (See discussion on losses in Section 4.12).

Experiments tested weight factors of for L_{WCE}, L_{DSC} and L_F, as discussed in Chapter 6.

- Class weights were calculated using the profile discussed in Section 4.4 and applied on some experiments.
- Network parameter weights were clipped using a gradient norm computed over all gradients together, as if they were concatenated into a single vector, and modified in-place. This helped to mitigate exploding gradients.

FIGURE 5.1: U-net architecture [1]

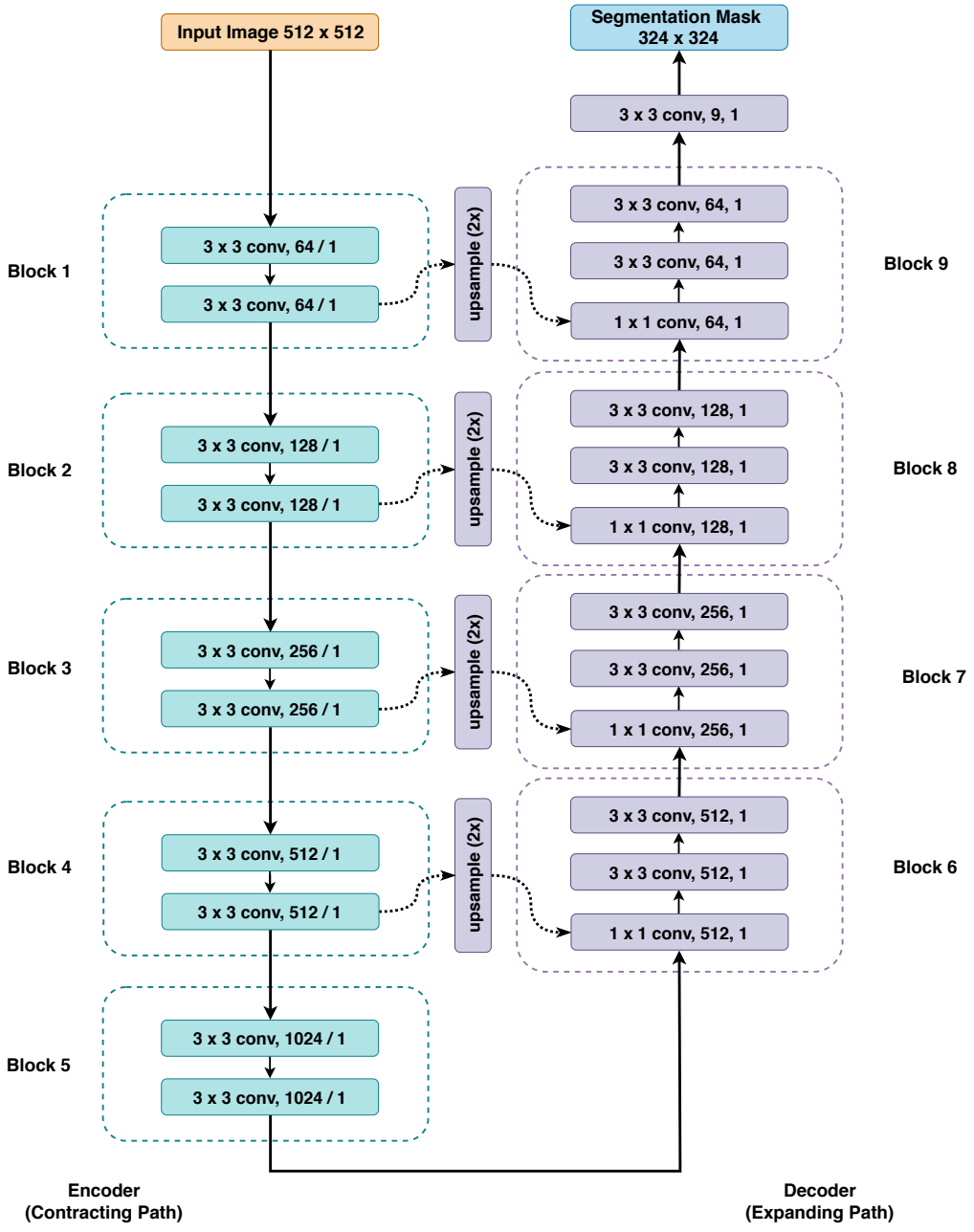


TABLE 5.4: U-net Architecture - Detailed specification.

	Layer	Channels		Kernel Size	Stride
		Input	Output		
Encoder					
Block 1	conv1	3	64	3×3	1×1
	conv2	64	64	3×3	1×1
Block 2	conv1	64	128	3×3	1×1
	conv2	128	128	3×3	1×1
Block 3	conv1	128	256	3×3	1×1
	conv2	256	256	3×3	1×1
Block 4	conv1	256	512	3×3	1×1
	conv2	512	512	3×3	1×1
Block 5	conv1	512	1024	3×3	1×1
	conv2	1024	1024	3×3	1×1
Decoder					
Block 6	upsample $\times 2$				
	conv1	1024	512	1×1	1×1
	conv2	1024	512	3×3	1×1
	conv3	512	512	3×3	1×1
Block 7	upsample $\times 2$				
	conv1	512	256	1×1	1×1
	conv2	512	256	3×3	1×1
	conv3	256	256	3×3	1×1
Block 8	upsample $\times 2$				
	conv1	256	128	1×1	1×1
	conv2	256	128	3×3	1×1
	conv3	128	128	3×3	1×1
Block 9	upsample $\times 2$				
	conv1	128	64	1×1	1×1
	conv2	128	64	3×3	1×1
	conv3	64	64	3×3	1×1
Block 10	conv1	64	9	1×1	1×1

5.4 Deeplabv3+ implementation

The proposed implementation of Deeplabv3+ was derived from a Pytorch implementation of Deeplabv3+[3] with a ResNet101 [202] backbone was adapted for experiments. The architecture is shown in Figure 5.2. The total number of trainable network parameters was approximately 59M. Refer to Section 3.4.5 for a discussion of the Deeplab series of segmentation architectures.

The following Deeplab network configuration was used for both historic and repeat image databases.

- The Deeplab ResNet101 encoder was initialized with a pretrained model, trained on ImageNet database [203].
- A batch normalization layer [153] was applied after each convolution and before activation.
- Convolutional layers used exclusively ReLU nonlinear rectification.
- Batch sizes were limited to 8 due to GPU memory constraints given the 512 pixel $\times$ 512 pixel input tiles.
- Adam learning rate (LR) optimization [187] was used with an initial LR determined through grid search for each database. The initial learning rate varied from 5e-05 to 0.001, depending on the size of the training dataset. A step learning rate scheduler with a learning rate decay of $\gamma = 0.9$ was applied at every epoch.
- Models were trained up to 40 epochs, depending on early stopping criteria (i.e. stagnant validation accuracy).
- The training was regularized by weight decay (the L2 penalty multiplier set to 0.0005) and dropout regularization (dropout ratio set to 0.5).
- The applied loss function was computed as the sum of the weighted cross-entropy loss L_{WCE}, mean Dice coefficient loss L_{DSC}, and focal loss L_F for each spatial position in the DCNN segmentation map (See discussion on losses in Section 4.12). Experiments tested weight factors of for L_{WCE}, L_{DSC} and L_F, as discussed in Chapter 6.
- Class weights were calculated using the profile discussed in Section 4.4 and applied on some experiments.

- Network parameter weights were clipped using a gradient norm computed over all gradients together, as if they were concatenated into a single vector, and modified in-place. This helped to mitigate exploding gradients.

FIGURE 5.2: Deeplabv3+ network architecture with ResNet101 backbone.

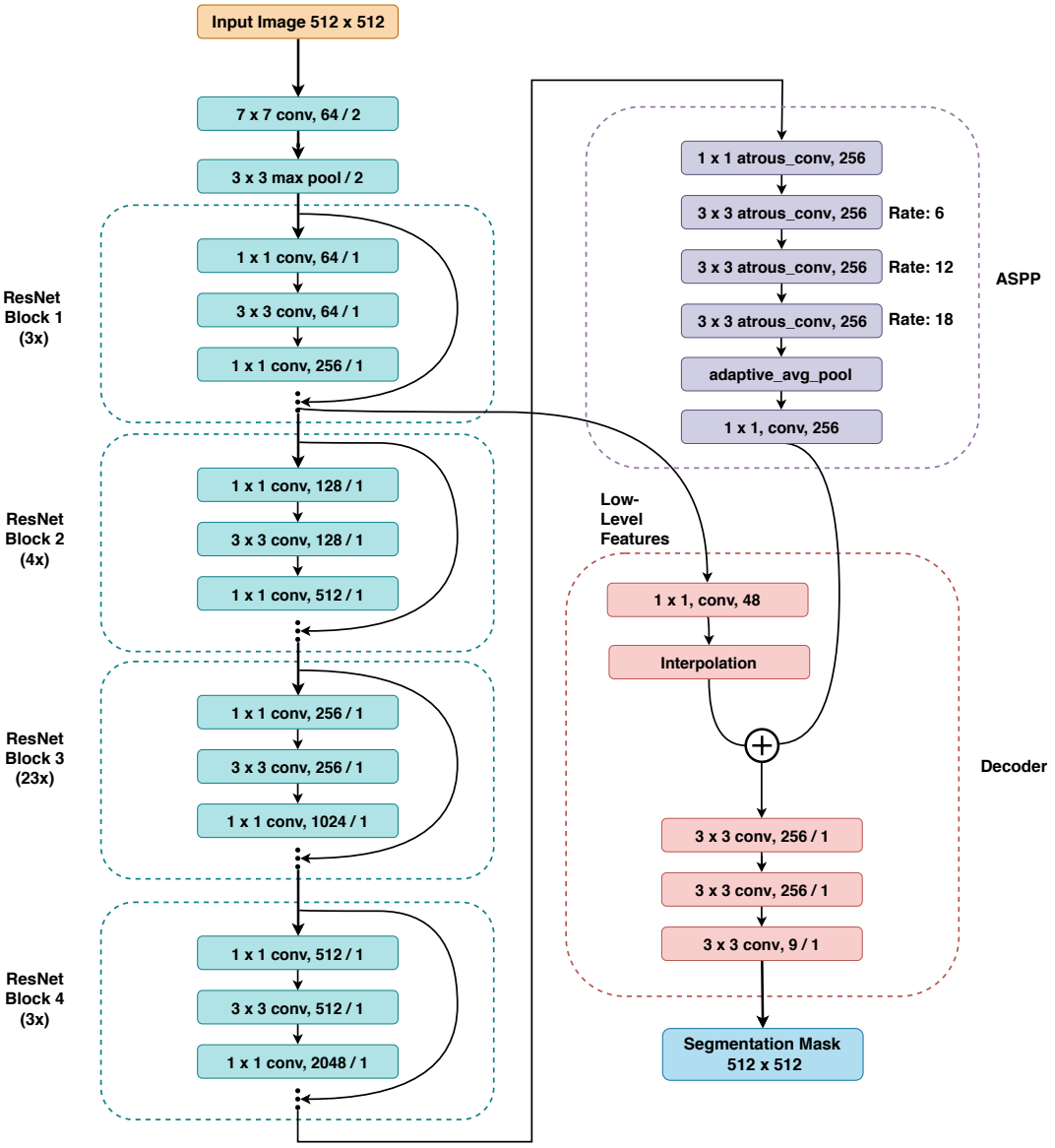


TABLE 5.5: Deeplabv3+ Architecture - Detailed specification.

	Layer	**Channels**		**Kernel**	**Stride**	**Padding**	**Dilation**
		Input	Output				
ResNet101							
	conv1	3	64	7×7	2×2	3×3	
	maxpool	3	2	3×3	2×2	1×1	1×1
Block 1	conv1	64	64	1×1	1×1		
	conv2	64	64	3×3	1×1	1×1	
	conv3	64	256	1×1	1×1		
Downsample	conv1	64	256	1×1	1×1		
Bottleneck	conv1	256	64	1×1	1×1		
($\times 2$)	conv2	64	64	3×3	1×1	1×1	
	conv3	64	256	1×1	1×1		
Block 2							
Bottleneck	conv1	256	128	1×1	1×1		
	conv2	128	128	3×3	2×2	1×1	
	conv3	128	512	1×1	1×1		
Downsample	conv1	256	512	1×1	2×2		
Bottleneck	conv1	512	128	1×1	1×1		
($\times 3$)	conv2	128	128	3×3	1×1	1×1	
	conv3	128	512	1×1	1×1		
Block 3							
Bottleneck	conv1	512	256	1×1	1×1		
	conv2	256	256	3×3	2×2	1×1	
	conv3	256	1024	1×1	1×1		
Downsample	conv1	512	1024	1×1	2×2		
Bottleneck	conv1	1024	256	1×1	1×1		
($\times 22$)	conv2	256	256	3×3	1×1	1×1	
	conv3	256	1024	1×1	1×1		
Block 4							
Bottleneck	conv1	1024	512	1×1	1×1		
	conv2	512	512	3×3	1×1	2×2	2×2
	conv3	512	2048	1×1	1×1		
Downsample	conv1	1024	2048	1×1	1×1		
Bottleneck	conv1	2048	512	1×1	1×1		
($\times 2$)	conv2	512	512	3×3	1×1	4×4	4×4
ASPP	atrous-conv	2048	256	1×1	1×1		
	atrous-conv	2048	256	3×3	1×1	6×6	6×6
	atrous-conv	2048	256	3×3	1×1	12×12	12×12
	atrous-conv	2048	256	3×3	1×1	18×18	18×18
	conv1	2048	256	1×1	1×1		
	conv2	1280	256	1×1	1×1		
Decoder							
	conv1	256	48	1×1	1×1		
	conv2	304	256	3×3	1×1	1×1	
	conv3	256	256	3×3	1×1	1×1	
	conv4	256	9	1×1	1×1		

5.5 Conditional Random Fields

Python and C++ code used to implement CRF processing on model output data was directly adapted from Krähenbühl and Koltun's work on inference using fully connected CRFs [3] [39]. Notably, the released version of the code computes the gradient of the permutohedral lattice directly, instead of the general Gauss Transform, as presented in the ICML 2013 paper. A Python wrapper called PyDenseCRF[4] was also used to interface with Pytorch segmentation data for inference computations. Unary potentials were computed as the softmax probabilities of the Deeplabv3+ model output tensors.

For the CRF inference, two pairwise potentials were computed that correspond to the kernels in Equation 4.27: (1) Gaussian pairwise potential adds the color-independent term (i.e. features are the locations only) to the "smoothness kernel"; (2) Bilateral pairwise potential adds the color-dependent terms to the "appearance kernel". Parameter values were optimized through a grid search based on the Dice loss of inferred filtered masks with ground-truth across the validation set $\mathcal{D}_{valid}$. The grid search involved a coarse-to-fine search over ranges for θ_γ, θ_α and θ_β. Once a coarse range has been optimized, a smaller range around the optimal value is then used to further refine the grid search. Optimization was based on the maximization of F1 scores and MCC values parameters over $\mathcal{D}_{valid}$. The optimized values are as follows (default values provided in the original paper [3] shown in brackets): $\theta_\gamma = 3(3)$, $\theta_\alpha = 10(80)$ and $\theta_\beta = 5(13)$.

The parameters of the label compatibility function $\mu(x_u, x_v)$ were also learned using images and masks from the validation dataset. As described in Section 4.6, the learned compatibility parameters incorporate information about the interactions between semantic classes into the function [3]. The default function is the Potts model (i.e. $\mu(x_u, x_v) = -w[u = v]$, where w is a weight given a value of 10.

Experiment #3 evaluated the use of CRF inference on the unary potentials of the DCNN (Deeplabv3+) model. Tests were carried out for both the default settings for the Potts label compatibility function recommended in the original paper, as well as learned parameters (μ parameters form a weight vector of size m), and parameters determined through a gridsearch for the general symmetric compatibility function $\mu(x_u, x_v)$ trained on a validation dataset. The algorithm for learning parameters to optimize the CRF, provided by Krähenbühl and Koltun (2013) [3] has all parameters estimated jointly, thus capturing dependencies between them. Parameter learning followed a three-part training procedure: (1) Optimized the CRF for unary potentials; (2) Optimized for unary and pairwise potentials; (3) Optimized the full CRF (kernel parameters). The algorithm uses L-BFGS[190] to optimize the performance of the mean field inference in the model.

Results of the CRF inferences were evaluated against the ground-truth masks, as well as the Deeplabv3+ predicted segmentation masks.

5.6 Summary

The implementation for this study employs two state-of-the-art segmentation architectures: U-net [1] and Deeplabv3+ [2], modified from available Pytorch implementations to train on tiles extracted from high resolution images. Separate segmentation models were developed for both historic (grayscale) and repeat (colour) capture images. Image preprocessing first extracts square tiles from the originals, and then applies a threshold-based sampling algorithm to augment specific tiles based on class distribution. In the post-processing step, an optimized fully-connected conditional random fields (CRF) model[39] is applied to the DCNN unary outputs to improve prediction accuracy, where CRF parameters were learned using the L-BFGS algorithm.

Chapter 6

Experimental Results and Analysis

6.1 Overview

In this chapter, results of the DCNN semantic segmentation experiments on the U-Net [section 5.1] and Deeplabv3+ [section 5.2] models are reported and evaluated. The chapter begins with some key takeaways from the results summarized in In Section 6.4. In Section 6.3, the loss curves for model training and validation are analyzed. Next, the three main experiments are described (see also Table B.1 in Appendix B). The evaluation of these results applies the framework outlined in Section 4.7, and specifically compares model dense classification predictions against the ground truth segmentation maps in the test dataset. The first experimental results in Section 6.4 report on the application of data augmentation optimized for the semantic class imbalance of the data; this involved a proposed selective augmentation method described in Section 4.4. Section 6.5 reports on the application of three different loss functions for model training: (1) Cross-entropy loss (L_{CE}); (2) Dice similarity coefficient loss (L_{DSC}); and (3) Focal loss (L_F), as defined in Section 4.5.4. Section 6.6 reports on the application of a conditional random field (CRF) models to the DCNN output segmentation masks as a post-processing step. Section 6.7 then presents an examination of systematic misclassifications that arose from the data. Next, section 6.8 reports on the measured training and inference latency. Section 6.9 reviews some of the important limitations that limit the confidence and reproducibility of the results.

For a more detailed analysis of the results data, refer to Appendix B. Example segmentation map results are presented in Appendix C.

6.2 Key Findings

Deeplabv3+ was found to perform significantly better over U-Net (F1 Scores $> 15-25\%$) across the evaluation metrics for both historic and repeat image captures on test image datasets (DST.A.2, DST.B.2, and DST.C). Historic models showed a top mean F1 score of 0.839 (frequency-weighted IoU of 0.753 and MCC of 0.742), for model DLAB.H.2.3 trained on H-Mrg, and an equal weighting of losses L_{CE} and L_{DSC} with no focal loss L_F; repeat models achieved a top mean F1 score of 0.909 (frequency-weighted IoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F. Models with the top aggregate accuracy did not consistently produce individual output segmentations of high sensitivity metrics and qualitative assessment (see discussion in Section 6.7), but in many cases model sensitivity was highly dependent on the input image. This suggests an ensemble of models may improve performance. As found with Buscombe and Ritchie (2018) [34], overall class accuracy is less informative than the prediction performance for each class, in which case fine-tuning of model hyperparameters is required to reduce the misclassification of classes with high inter-class similarity.

Given the extent to which the Deeplabv3+ architecture outperformed the U-Net models in all experiments, the focus of much of this evaluation was given to the Deeplabv3+ experiments, whereas only overall performance metrics are reported for the U-Net. Evaluation of the experimental results uses the framework outlined in Section 4.7.

6.3 Model Training/Validation

Training and validation loss values were recorded at regular intervals to track the model's ability to determine correct predictions for all weights and biases for the labeled data (see Sections 4.5 and 4.7.1). Each model corresponds to different weightings of loss functions, as summarized in Tables 6.7 and 6.8. Note that model DLAB.H.1.3 used the same configuration as DLAB.H.2.1.

In summary, L_{DSC} provided the slowest, but most effective loss convergence during training and validation. For historic H-Mrg models, L_{CE}, and in some cases L_F, showed overfitting after 30,000 iterations, whereas L_{CE} continued to drop. However, low L_{DSC} loss gaps did not necessarily translate into top performance for EVAL.2 (accuracy) on the test dataset, which includes images from different surveys (see DST.C in Appendix A). Given the specific sensitivity of L_{DSC} to accurate segmentation overlap (see Section 4.5.4.2), and the attendant disadvantages of L_{CE} as discussed in Section 4.5.4.4, this was expected.

6.3.1 Loss Curves for Historic Models

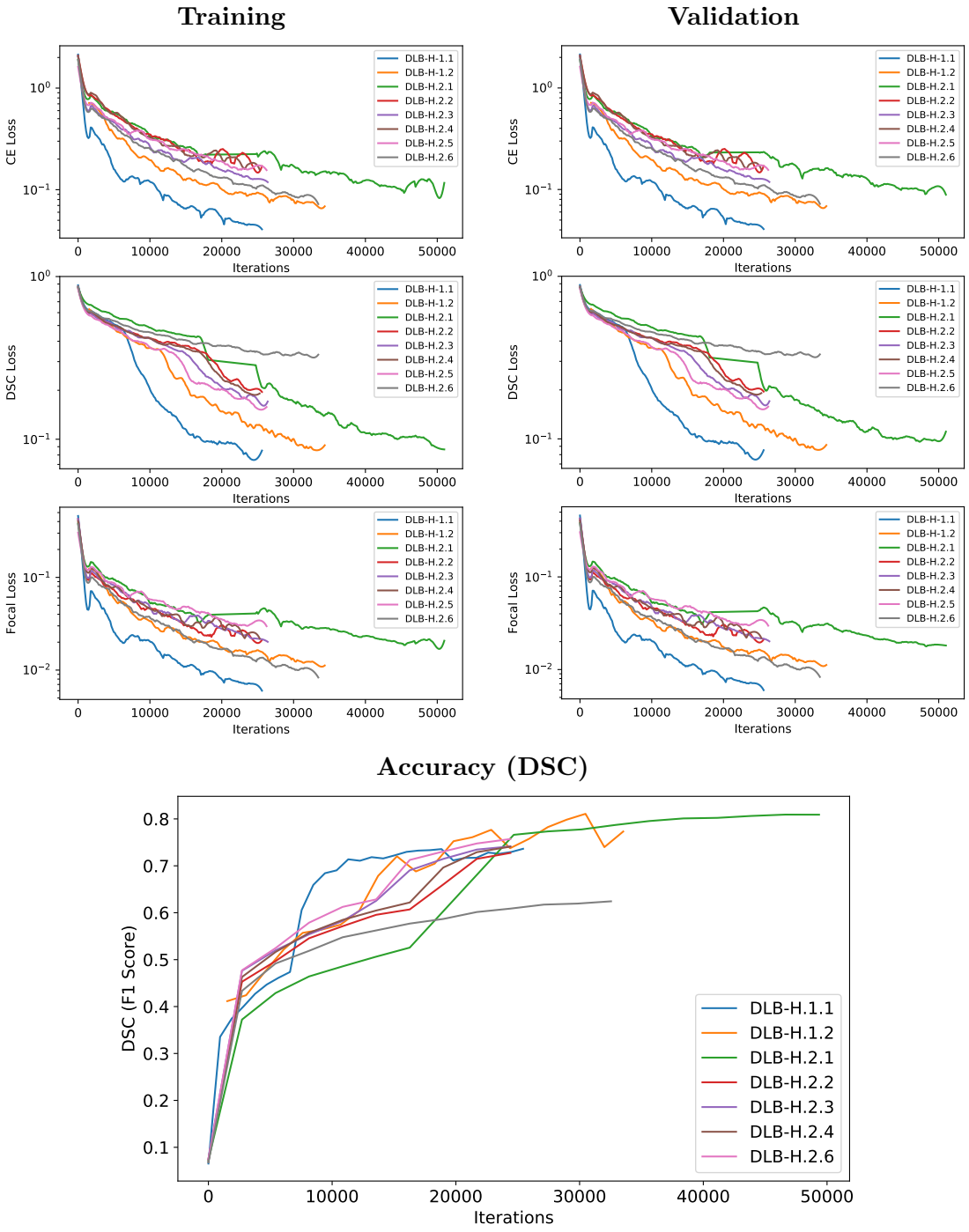


TABLE 6.1: Model training and validation losses (L_{CE}, L_{DSC}, L_F) for historic captures (logarithmic scale) and validation accuracy. Each model corresponds to different gradient weightings of loss functions, as summarized in Table 6.7. See also Sections 4.5 and 4.7.1.

6.3.2 Loss Curves for Repeat Models

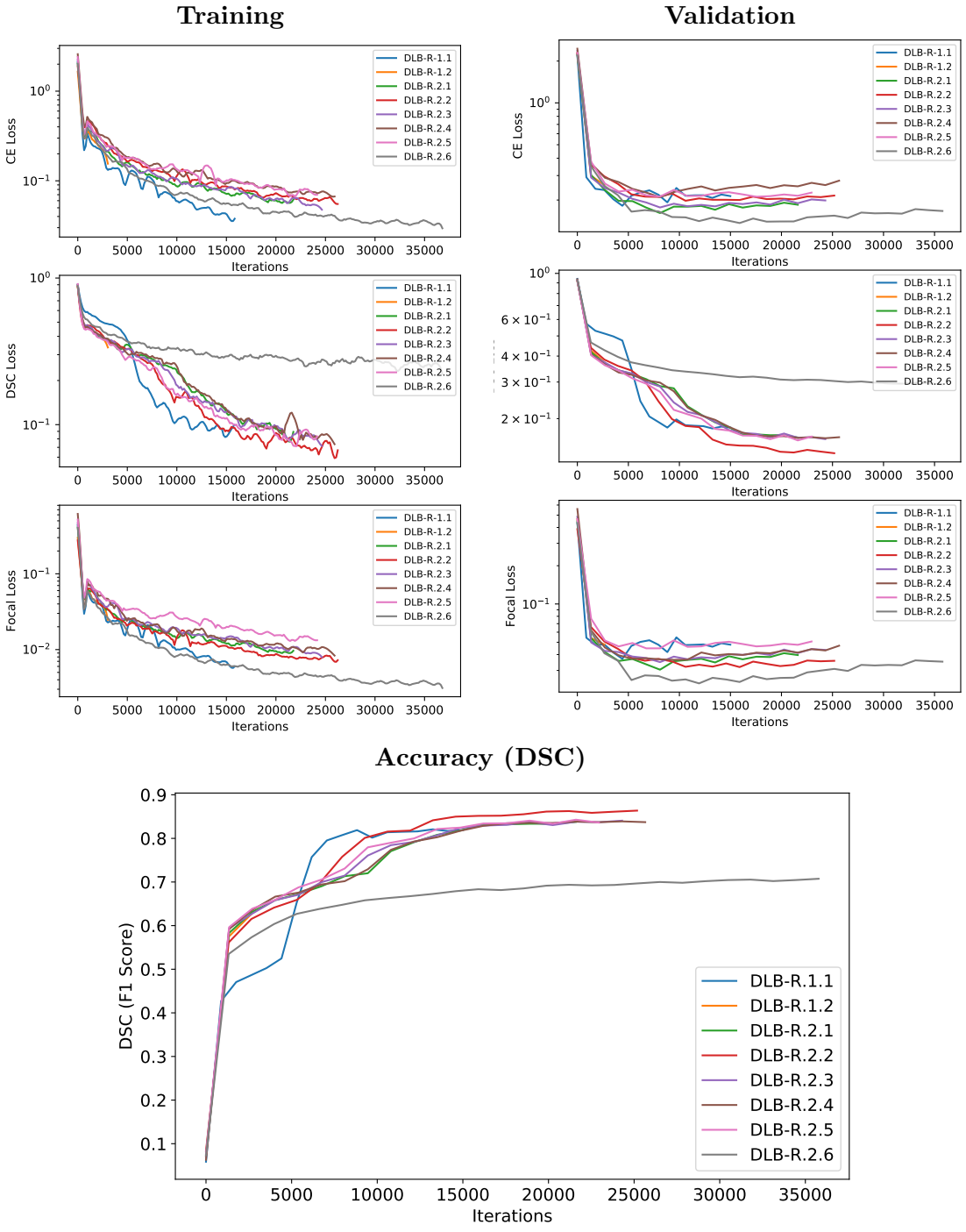


TABLE 6.2: Model training and validation losses (L_{CE}, L_{DSC}, L_F) for repeat captures (logarithmic scale) and validation accuracy. Each model corresponds to different gradient weightings of loss functions, as summarized in Table 6.8. See also Sections 4.5 and 4.7.1.

6.4 Experiment 1: Data Augmentation

Experiment #1 tested the use of data augmentation to address semantic class imbalance, using a custom oversampling method (see Section 4.4)., which was shown to marginally improve class imbalance (see Figures 5.2 and 5.3). For historic data augmentation results, summarized in Table 6.3, performance impact by data augmentation compared three overlapping training databases: (1) H-Ext (extracted), (2) H-Aug (augmented), and (3) H-Mrg (H-Aug merged with grayscaled R-Aug). For repeat data augmentation results, summarized in Table 6.5, performance for two training databases was evaluated: (1) R-Ext (extracted), and (2) R-Aug (augmented). Normalized confusion matrix visualizations are presented for historic captures in Table 6.4, and for repeat captures in Table 6.6. Each value in the cells of the confusion matrix visualizations is equal to the number of observations known to be in semantic class i but predicted to be in class i normalized by the support $n_{\mathcal{D}}$ (total pixel count for i). All experiments used an equal weighting of loss functions, and weighted cross entropy loss used class weights, as defined in Section 4.5.4.1.

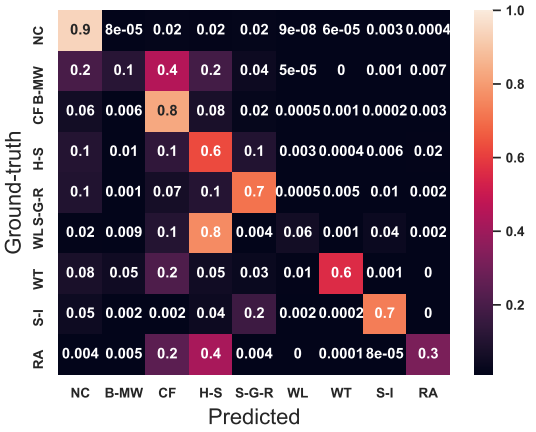
6.4.1 Historic Models

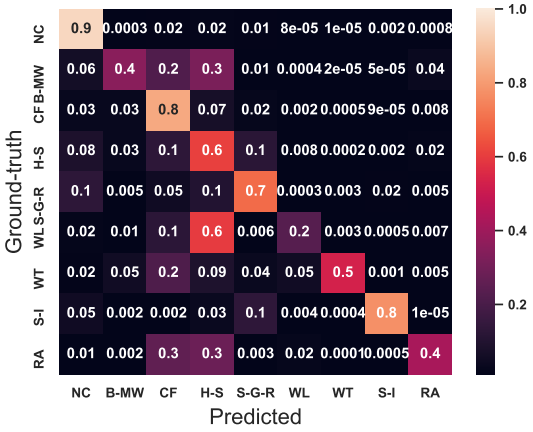
For historic captures, data augmentation (H-Aug) gave a moderate overall performance improvement for DLAB.H.1.2 (F1: +3.0%, fIoU: +4.0%, MCC: +3.6%). Augmentation further showed significant performance improvements for minor classes B-MW (F1: +125.0%) and WL (F1: +218.3%), and noticeable improvements to classes WT (F1: +3.5%) and S-I (+10.6%), without any significant performance loss for other classes. Augmentation by grayscaled repeat images (H-Mrg) did not noticeably alter DLAB.H.1.2 average performance (F1: −0.2%, fIoU: +0.4%, MCC: +0.4%), with increased accuracy across minor classes except for B-MW (F1: −24.4%), RA (F1: −10.9%). All DLAB.H models showed better classification accuracy (F1: $>$ 0.800%) for dominant classes NC and C.

Data Augmentation Results: Historic Models

Model	Database	$F1$ Score	$fIoU$	MCC
UNET.H.1.1	H-Ext	0.612	0.563	0.522
UNET.H.1.2	H-Aug	0.631	0.572	0.552
UNET.H.1.3/2.1	H-Mrg	0.654	0.603	0.587
DLAB.H.1.1	H-Ext	0.815	0.727	0.713
DLAB.H.1.2	H-Aug	0.820	0.735	0.718
DLAB.H.1.3/2.1	H-Mrg	**0.835**	**0.755**	**0.737**

TABLE 6.3: Experiment H.1: Results summary for data augmentation (historic captures).





Class	Prec	Rec	F1	Sup
NC	0.950	0.946	0.948	0.585
B-MW	0.718	0.221	0.338	0.042
C	0.807	0.853	0.829	0.203
H-S	0.490	0.618	0.547	0.073
S-G-R	0.684	0.731	0.707	0.067
WL	0.711	0.274	0.396	0.012
WT	0.767	0.542	0.635	0.002
S-I	0.675	0.794	0.730	0.008
RA	0.299	0.418	0.349	0.008
cAvg	0.678	0.600	0.609	1.000
wAvg	0.849	0.844	0.839	1.000
Accuracy:		0.844	**wIoU:**	0.754
F1 Score:		0.839	**MCC:**	0.742

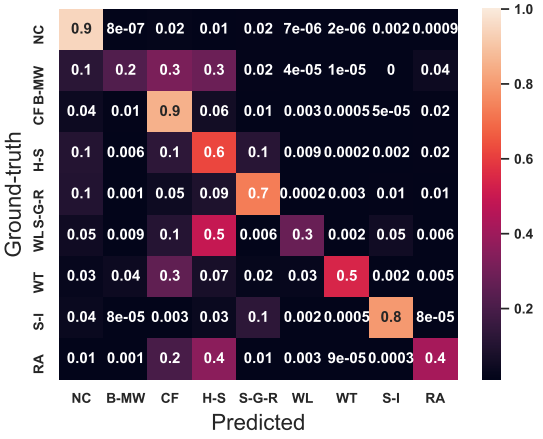


TABLE 6.4: Confusion matrices for models DLAB.H.1.1 (top left), DLAB.H.1.2 (top right), and DLAB.H.1.3/2.1 (bottom right) trained on H-Ext, H-Aug and H-Mrg databases, respectively (see Table 5.1). Class accuracy results for DLAB.H.1.3/2.1 are also shown. "Accuracy" is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class. Support total: 449,533,967 pixels. For detailed analysis, see Appendix B.3.1).

6.4.2 Repeat Models

Results of model training with data augmentation for repeat (colour) captures are summarized in Table 6.5. UNET.R and DLAB.R models performed considerably better than their historic counterparts, as it is speculated that the additional RGB channels provided more information for feature learning – i.e. the DCNN is better able to automatically discover the representations needed for segmentation of the raw data. Classification of repeat capture images achieved a top mean F1 score of 0.909 (fIoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F. Trained on the augmented database (R-Aug), the model showed modest improvement over the average F1 score (+1.0%) over the extracted database (R-Ext), but also exhibited a small uptick in the weighted F1 score (+0.6%). Moreover, data augmentation showed improved F1 scores for minor classes B-MW (+9.3%), WL (+8.2%), WT (+19.7%) and RA (+8.4%), and $< 0.5\%$ change in class S-I. As with the historic captures, these improvements correspond to the augmented minor classes summarized in Table 5.3. All DLAB.R models showed very good classification accuracy for dominant classes NC, C, S-G-R, and RA (F1: > 0.800).

Model	Database	$F1$ Score	$fIoU$	MCC
UNET.R.1.1	R-Ext	0.686	0.632	0.665
UNET.R.1.2	R-Aug	0.711	0.689	0.701
DLAB.R.1.1	R-Ext	0.904	0.837	0.855
DLAB.R.1.2 / 2.2	R-Aug	**0.909**	**0.844**	**0.864**

TABLE 6.5: Summary of accuracy metrics for extracted (H-Ext), augmented (H-Aug) and merged (H-Mrg) training databases. See Section 4.7.2 for metric definitions.

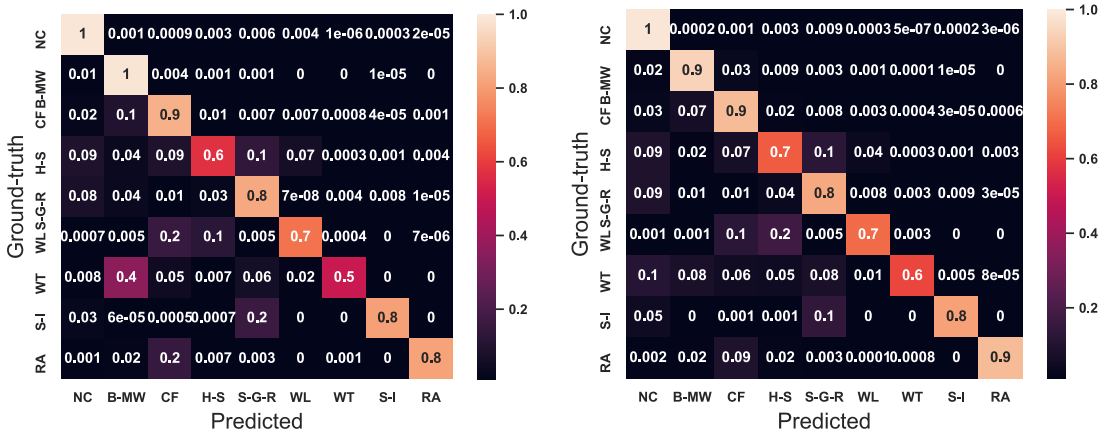


TABLE 6.6: Confusion matrices for models DLAB.R.1.1 and DLAB.R.1.2 trained on R-Ext and R-Aug databases, respectively (see Table 5.1). Support total: 281,240,583 pixels. For detailed analysis, see Appendix B.3.2).

6.5 Experiment 2: Loss Functions

Experiment #2 evaluated model performance based on the type and configuration of training and validation loss functions. Three loss functions were investigated: (1) L_{CE}: Cross-entropy loss; (2) L_{DSC}: Dice similarity coefficient loss; and (3) L_F: Focal loss, as defined in Section 4.5.4. Additionally, the impact of using weighted cross-entropy loss (L_{WCE} on model performance was evaluated, as indicated by the w superscript).

6.5.1 Historic Models

Historic models achieved a top mean F1 score of 0.839 (frequency-weighted IoU of 0.753 and MCC of 0.742), for model DLAB.H.2.3 trained on H-Mrg. Both model back-propagation gradients used an equal weighting of losses L_{CE} and L_{DSC} with no focal loss L_F. All DLAB.H models showed better classification accuracy (F1: > 0.800) for dominant classes NC and C. Semantic segmentation of repeat (colour) capture images achieved a top mean F1 score of 0.909 (frequency-weighted IoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F.

In general, results for both historic and repeat captures showed moderate variation in accuracy for different model loss functions. As discussed in Section 6.3, models that use L_{DSC} had more accurate segmentation overlap (see Section 4.5.4.2), which allowed for convergence at a lower error rate than other loss functions. This was anticipated given the advantages of directly measuring segmentation overlap, as well as the limitations of using L_{CE} pixel-wise losses as discussed in Section 4.5.4.4. Focal loss (L_F) did not appear to significantly improve convergence or accuracy. However, the scaling parameter L_F was determined empirically, and other values of the scaling parameter γ may improve these results. Hossain et al.[204], for example, have proposed an adaptive focal loss for semantic segmentation, where γ is a learnable parameter tuned through backpropagation, rather than empirical tuning.

6.5.2 Repeat Models

For repeat (colour) models (DLAB.R), segmentation outputs achieved a top mean F1 score of 0.909 (frequency-weighted IoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F. Unlike historic models, the weighted cross-entropy and focal loss appeared to marginally improve average accuracy. As illustrated in Figure 6.1 with

		EVAL.1 [4.7.1]						EVAL.2 [4.7.2]		
Model ID	Database	L_{CE}		L_{DSC}		L_F		$F1$	$fIoU$	MCC
UNET.H.2.1	H-Mrg	0.320	*	0.280	*	0.106	*	0.640	0.541	0.520
UNET.H.2.2	H-Mrg	0.327	$*^w$	0.260	*	0.110	*	0.609	0.512	0.592
UNET.H.2.3	H-Mrg	0.409	*	0.212	*	0.124		0.634	0.553	0.535
UNET.H.2.4	H-Mrg	0.339	$*^w$	0.307	*	0.120		0.601	0.529	0.508
UNET.H.2.5	H-Mrg	0.389		0.323	*	0.117	*	0.584	0.603	0.587
DLAB.H.2.1	H-Mrg	0.224	*	0.186	*	0.046	*	0.827	0.745	0.731
DLAB.H.2.2	H-Mrg	0.259	$*^w$	0.198	*	0.055	*	0.835	**0.755**	0.737
DLAB.H.2.3	H-Mrg	0.269	*	0.196	*	0.059		**0.839**	0.753	**0.742**
DLAB.H.2.4	H-Mrg	0.385	$*^w$	0.216	*	0.058		0.834	**0.755**	0.737
DLAB.H.2.5	H-Mrg	0.283		0.191	*	0.063	*	0.830	0.744	0.733
DLAB.H.2.6	H-Mrg	0.274	*	0.375		0.053	*	0.827	0.742	0.727

TABLE 6.7: Experiment 2.1 Results Summary: Loss functions (historic captures). The asterisk ($*$) indicates inclusion of the loss in the gradient computation, and the w superscript indicates class weights were applied to the CE loss. Multi-loss computations used an equal weighting of the loss value (i.e. $L_{total} = \alpha L_{CE} + \beta L_{DSC} + \gamma L_F$, where $\alpha, \beta, \gamma \in [0, 1]$, see Section 4.5.4.4). Model trained on H-Mrg database for up to 20 epochs. Support is 449,533,967 pixels.

DST.B.R.2.6, repeat models showed much less prediction variance than historic models, and much of that was concentrated on minor classes, whereas dominant classes were well classified.

		EVAL.1 [4.7.1]						EVAL.2 [4.7.2]		
Model	DB	L_{CE}		L_{DSC}		L_F		$F1$	$fIoU$	MCC
UNET.R.2.1	R-Aug	0.322	*	0.267	*	0.096	*	0.664	0.621	0.567
UNET.R.2.2	R-Aug	0.387	$*^w$	0.290	*	0.102	*	0.688	0.603	0.587
UNET.R.2.3	R-Aug	0.302	*	0.240	*	0.104		0.620	0.630	0.577
UNET.R.2.4	R-Aug	0.356	$*^w$	0.252	*	0.097		0.613	0.585	0.564
DLAB.R.2.1	R-Aug	0.176	*	0.140	*	0.038	*	0.908	0.841	0.861
DLAB.R.2.2	R-Aug	0.215	$*^w$	0.136	*	0.036	*	**0.909**	**0.844**	**0.864**
DLAB.R.2.3	R-Aug	0.178	*	0.136	*	0.040		0.898	0.826	0.848
DLAB.R.2.4	R-Aug	0.236	$*^w$	0.140	*	0.038		0.875	0.833	0.852
DLAB.R.2.5	R-Aug	0.204		0.129	*	0.046	*	0.905	0.836	0.857
DLAB.R.2.6	R-Aug	0.167	*	0.292		0.035	*	0.907	0.840	0.860

TABLE 6.8: Experiment 2.2 Results Summary: Loss functions (repeat captures). The asterisk ($*$) indicates inclusion of the loss in the gradient computation, and the w superscript indicates class weights were applied to the CE loss. Multi-loss computations used an equal weighting of the loss value (i.e. $L_{total} = \alpha L_{CE} + \beta L_{DSC} + \gamma L_F$, where $\alpha, \beta, \gamma \in [0, 1]$ - see Section 4.5.4.4. Model trained on R-Aug database for up to 20 epochs. Support is 449,533,967 pixels.

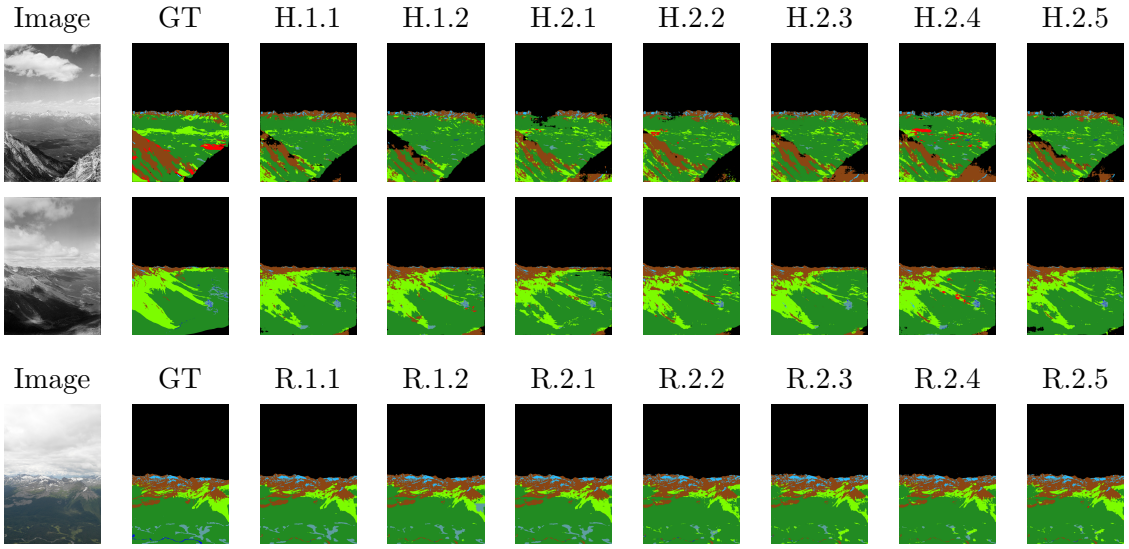


FIGURE 6.1: Comparison of output segmentations for images DST.A.H.2.1 (top), DST.B.H.2.4 (middle) on DLAB.H (historic) models, and image DST.B.R.2.6 (bottom) on DLAB.R (repeat) models.

6.6 Experiment 3: Conditional Random Fields

Experiment #3 tested model performance with the use of an optimized fully-connected conditional random fields (CRF) model, as described by Krähenbühl and Koltun [39] [3], applied as a post-processing enhancement to DCNN outputs. CRF inference and parameter optimization is described in Sections 4.6 and 5.5. Note that only results for DLAB.H were produced for this study.

CRF post-processing produced mixed results as far as improvements to accuracy of DCNN outputs, and in many cases resulted in little to no change in F1 scores (F1: $< 1.0\%$ change). CRF inference results showed a high degree of variability given small changes in parameter values for the label compatibility function (LCF) $\mu(x_u, x_v)$. LCF parameters were computed as a vector using the L-BFGS algorithm[190], which proved time-consuming for high-resolution, multi-class segmentation over even the small dataset used. Furthermore, it is not clear that the limited validation dataset provided sufficient data to optimize parameters. However, some test images did bear accuracy improvements for low-frequency minor classes, as well as minor segmentation corrections for dominant classes. Figure C.9, for example, shows improved resolution of S-I and S-G-R of distant mountains in DST.A.H.2.2, and better classification of WT and RA classes in DST.B.H.2.7.

Model	$F1$ Score	$fIoU$	MCC	% Change (F1)
DLAB.H.2.2	0.836	0.755	0.743	0.11%
DLAB.H.2.3	**0.841**	**0.759**	**0.751**	0.23%
DLAB.H.2.4	0.839	0.749	0.737	0.60%

TABLE 6.9: Experiment 3: Conditional Random Fields Filter (historic models)

6.7 Evaluation of Model Sensitivity

Model sensitivity (see Evaluation Methods in Section 4.7.3) was evaluated to identify systematic classification errors observed in the results. As anticipated from the discussion in Section 2.3.3, discrimination between vegetation classes proved the most challenging for classification, whereas the classification boundary between vegetation and non-vegetation was substantially more distinct. The difficulty of classifying vegetation classes is compounded for the minor (low-frequency) classes that include R-A, B-MW and H-S areas, which had previously proven difficult for manual segmentation [12]. With the historic extraction database (H-Ext), for example (see confusion map in Figure B.4), systematic misclassification of R-A regions split between C and H-S was observed. Models trained on the augmented databases (H-Aug) – as indicated in Figure B.5 confusion map – corrected many NC misclassifications and significantly improved recall for classes B-MW (+60.0%), WL (+233.1%) and RA (+31.7%), but with lower precision. However, inter-class similarity misclassification for vegetation classes are less of a problem for change detection than errors that mistake non-vegetation classes, suggesting augmentation improved the "quality" of misclassifications.

6.7.1 Photometric Invariance

Overall, experimental results show both historic and repeat models have very good invariance to photometric differences in images. Cloud shade across continuous segments of land cover classes, for example image DST.B.H.2.2 in Figure C.4, did not significantly affect classification accuracy of the affected vegetation classes C, H-S and RA. Fog or cloud-obscured regions where visibility of landscape features was reduced did, however, appear to negatively impact accuracy, as for example in images DST.A.H.2.2 and DST.A.H.2.2 in Figure C.1, where the definition between classes S-G-R and C or H-S is somewhat less well-defined for distant than for clear images, such as image DST.B.H.2.3 in Figure C.3.

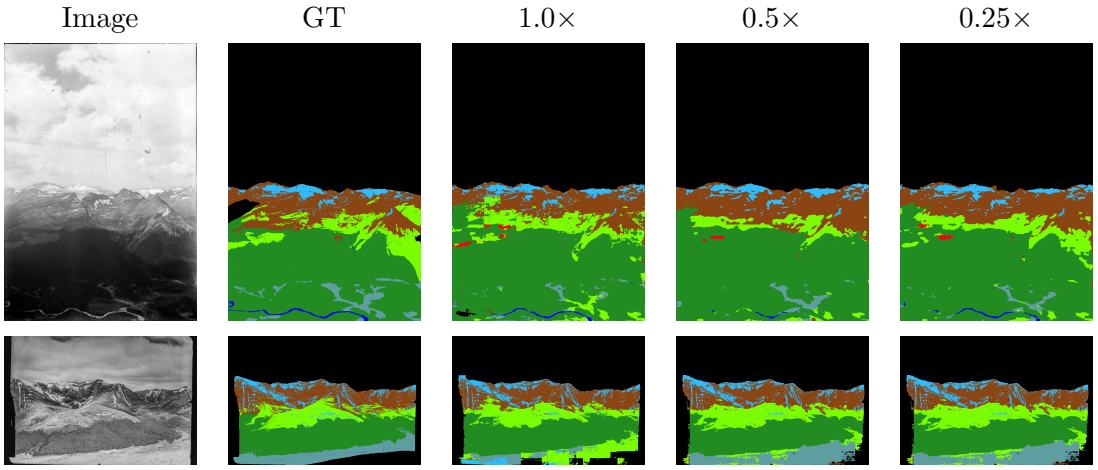


FIGURE 6.2: Comparison of output segmentations for resampled versions of images DST.B.R.2.6 (top) and DST.B.H.2.3 (bottom) on model DLAB.H.2.3. From left, prediction masks shown are at full-sized (1.0×), scaled to 0.5×, and scaled to 0.25×.

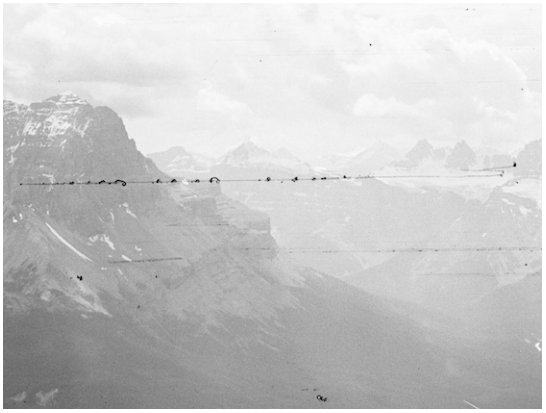
6.7.2 Scale Invariance

Differences in scale of same pixel class texture introduce both intra-class and inter-class variation that impact classification (See discussion under "Foreground/Background Representation" in Section 2.3.3). For instance, in Figure 6.1, output segmentations for DST.A.H.2.1 show the classification of the left foreground varies between barren rock and herbaceous/shrub (S-G-R or H-S) or not classified (NC). These areas may be classified as NC (not classified), depending on an unknown threshold learned by the model that encodes the boundary between close-range foreground and pixels in the normal classification range.

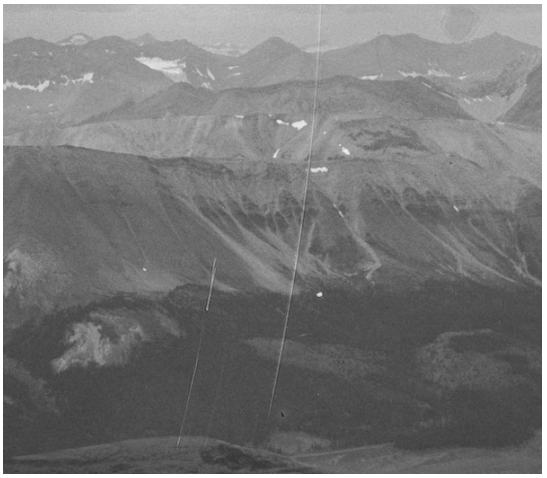
More generally, prediction variance is closely tied to image scale, as illustrated in Figure 6.2, where the classification of minor classes, in particular, varies significantly at different resampled sizes – e.g., differences in H-S (herbaceous) and WT (water) classification for different resampled versions of image DST.B.H.2.3. As well, the classification of foreground pixels (e.g., the improvement to WT (wetland) classification in the lower portion of image DST.B.H.2.3), is highly dependent on image size and resolution. These observations suggest that averaging results over an ensemble of outputs at different scales for the same model might improve performance. Furthermore, adjusting model testing to input image size and resolution presents another area of model parameter tuning that requires further study, as higher-resolution tiles will naturally correspond to higher resolution (and larger) images. This consideration applies to both historic and repeat models.

6.7.3 Image Noise Invariance

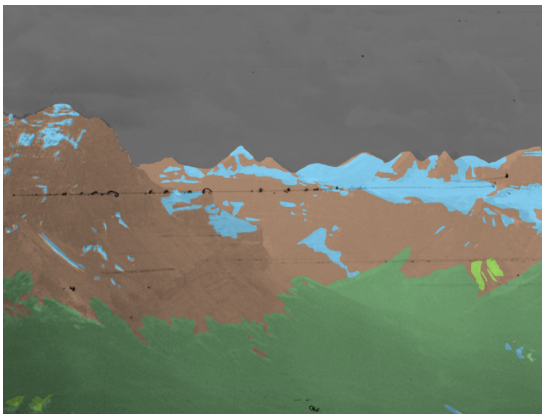
For the historic photos specifically, model results were marginally affected by photographic and digitization artefacts, such as scratches and localized emulsion defects along photo edges of input images, as well as other occasional but visible damages to the glass plate negatives for the historic photos [5]. (These errors are negligible on photo images from the repeat dataset.) Edge artefacts seen in images DST.A.H.2.4 [C.2], DST.B.H.2.3 [C.3], DST.C.H.1.8 [C.7], and DST.C.H.1.5 [C.8] were correctly categorized as NC. However, Figure 6.3 shows horizontal marks in image DST.A.H.2.2 and vertical scratches in image DST.H.B.2.1 that appear to affect the classification of pixels surrounding where the marks superimpose on the masks. It is possible that, given a larger training dataset, invariance to noise would improve. Adding artificial perturbations to the augmented samples, such as added Gaussian noise, speckles, or random illumination to the grayscaled repeat images, offers a potential approach to improve noise invariance not explored in this thesis.



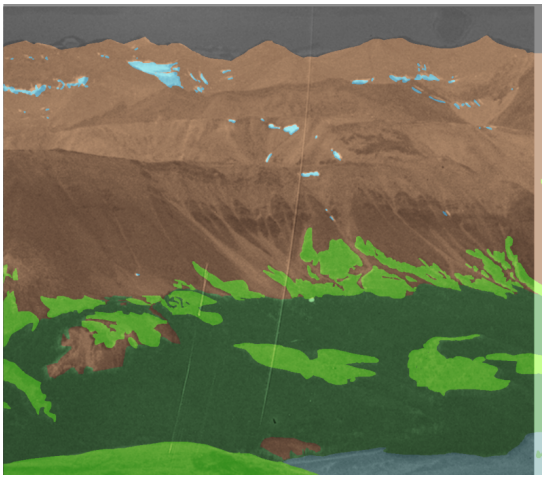
DST.A.H.2.2



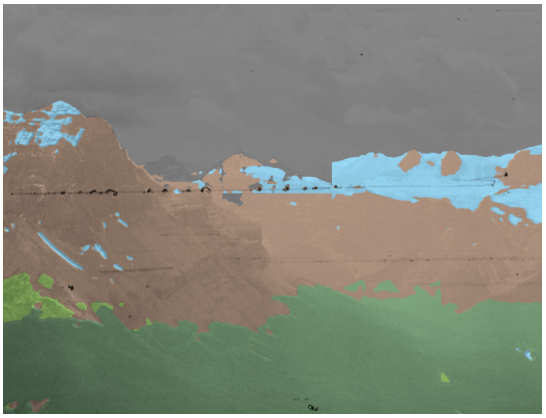
DST.B.H.2.1



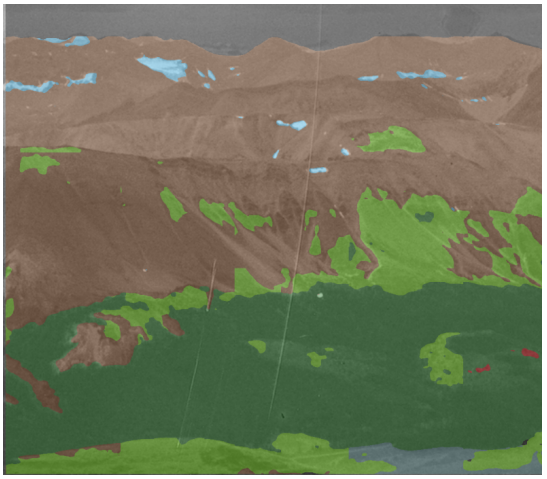
Ground-truth



Ground-truth



Predicted



Predicted

FIGURE 6.3: (Top) Extractions from images DST.A.H.2.2 and DST.B.H.2.1 showing markings and/or scratches from glass plate negatives; (Middle) Ground-truth mask extraction; (Bottom) Segmentation output from model DLAB.H.8.3.

6.8 Evaluation of Model Latency

As defined in EVAL.4 (Efficiency) in Section 4.7.4, an efficient or "compact" model converges for a small training dataset, and within a reasonable training time. Inference latency measures the time to generate segmentation outputs given an input test image. The average training time to process tiles of size 512 × 512 pixels was 2.10 batches/sec for Deeplabv3+ and 1.09 batches/sec for U-Net. The average inference time for a given input image tile of size 512 × 512 pixels was 1.62sec/tile for U-Net and 1.90sec/tile for Deeplabv3+. A 25 megapixel image will have approximately 330 tiles for a total elapsed time of 8.9 minutes for U-Net, and 10.45 minutes for Deeplabv3+. Image reconstruction takes an average of 0.08 seconds per tile for grayscale images, and 0.11 seconds per tile for colour images.

Elapsed Training and Inference Times			
Model	Database	Training (hrs)	Inference (sec/tile)
UNET.H.1.1	H-Ext	12.3	1.62
UNET.H.1.2	H-Aug	16.3	1.61
UNET.H.1.3/2.1	H-Mrg	25.6	1.60
DLAB.H.1.1	H-Ext	11.3	1.87
DLAB.H.1.2	H-Aug	17.4	1.88
DLAB.H.1.3/2.1	H-Mrg	26.3	1.92
DLAB.R.1.1	H-Ext	11.2	1.89
DLAB.R.1.2	H-Aug	16.8	1.90
DLAB.R.1.3	H-Mrg	27.4	1.91

TABLE 6.10: Summary of training and inference times for historic and repeat models. U-Net network totaled approximately 28M trainable parameters; Deeplabv3+ network totaled approximately 59M. Database sizes are listed in Table 5.1

6.9 Limitations

The results reported in this chapter have the following caveats that restrict the confidence and place conditions on the reported accuracy measurements.

1. **Limited dataset** Results shown are specific to datasets DST.A, DST.B, and DST.C, which together represent only a very small fraction ($< 0.1\%$) of the complete MLP collection (see Appendix A for a detailed image list). Model generalization and sensitivity for the wider selection of input images has not been tested. Test images, though excluded from the DST.A.1 and DST.B.1 training/validation datasets, nonetheless were drawn from the same surveys – e.g., Wheeler (1923, 1924) and Miller (1928) – and therefore may present similar features for classification. (Note, however, that some images from DST.C were drawn from separate surveys.) It is expected that this may have skewed test accuracy for those images.

2. **Ground-truth errors** Errors and inconsistencies in the ground-truth segmentation masks (Section 2.3.3), entail problematic labeling that can contribute to poor supervised learning. Ground-truth errors were furthermore more concentrated in minor classes that have high inter-class similarity, such as broadleaf-mixedwood (B-MW), coniferous forest (C), herbaceous/shrub (H-S), and regenerating areas (RA). Similarly, in some cases predictions by trained models appear to correct misclassifications in the ground-truth, however, these corrections require verification.

3. **Foreground/background classification:** Accuracy was significantly affected by inconsistencies in the ground-truth labeling of foreground and distant background pixels. For example, distant mountains in images DST.C.H.1.1 and DST.C.H.1.2 in Figure C.6, and DST.C.H.1.6 in Figure C.7, were not included in the ground-truth mask, but predicted in the output segmentations. In the case of foreground regions (see Section 2.3.3), close-range objects were largely excluded from manual classifications to avoid pixel class representation distortions of close-range objects. Given that foreground exclusion is a standard practice for land-cover classification, models were trained to label foreground objects as NC, and several of the models did accurately exclude foreground segments. However, no systematic exclusion of foreground pixels based on measurable criteria was used in the creation of ground-truth masks, but was dependent on the discretion of the field expert. Infrequently, relatively close-range objects were classified in some of the ground-truth masks, which contributed to both uncertainty during model training and to inaccurate test metrics.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Recent advances in deep learning neural networks (DCNNs) have been applied in new robust methods to classify land cover types in time-series remote-sensing data (aerial and satellite orthoimagery) for change detection. However, deep learning techniques for oblique photography used in repeat photography – an underutilized technique to detect landscape change – remain comparatively undeveloped. Through repeat photography, oblique images represent a deeper historical record of ecological and geomorphological change than that available from the more recent remote-sensing data[18]. Still, current methods used to create oblique landscape segmentation maps of sufficient accuracy and detail require costly and time-consuming manual labor [6], and therefore cannot be applied for the analysis of very large datasets.

The primary goal of this study was to evaluate the performance of DCNNs for the task of land cover classification for oblique ground-level photography. Two state-of-the-art neural networks, U-net[1] and Deeplabv3+[2], were trained on high resolution historic grayscale and (in separate models) modern colour landscape images sampled from the Mountain Legacy Project collection[37]. A novel threshold-based method for data augmentation was developed and implemented to address limited annotated data and class imbalance in the dataset. Furthermore, three independent loss functions – weighted cross-entropy loss, dice coefficient loss, and focal loss – were also evaluated. Though deep learning applications strive for some degree of generalization, this project illustrates the practical considerations of the "no free lunch theorem,"[205] for statistical inference. The theorem claims that no *a priori* distinctions exist between learning algorithms, so we can infer no generalized deep learning approach can be guaranteed to work well in all cases, or for all datasets. In the case of the MLP collection, analysis of

the inherent class imbalance and other characteristics of the dataset informed decisions on how to adapt the model to better fit the problem.

Results obtained from a series of experiments showed top overall F1 scores of 0.839 for historic models, and 0.909 for repeat models. The proposed data augmentation method showed modest improvements to overall accuracy (+3.0% historic / +1.0% repeat), but much larger gains for under-represented classes over the extracted database (H-Ext). All tested models showed excellent classification accuracy (F1: > 0.80) for dominant classes NC and CF. It was noted that a comparison of this method with a randomized augmentation methods would better test the efficacy of the approach. Furthermore, models trained using soft Dice loss (e.g., in combination with cross-entropy and focal loss) were also shown to produce the most accurate segmentation maps. As expected from the discussion on segmentation challenges in Section 2.3.3, discrimination between the vegetation classes (C, B-MW, H-S, RA) proved the most challenging for classification, whereas the classification boundary between vegetation and non-vegetation was substantially more distinct. In other words, predicting the spatial extent of vegetation land cover can be estimated more accurately than the detection of different vegetation classes. As well, in some cases predictions by trained models appear to correct misclassifications in the ground-truth (such as a mislabeled region of snowpack), however, such cases require further verification. Evaluation of model selectivity showed a strong invariance to input noise and perturbations, but moderately high prediction variance for segmentations given input at different scales.

The use of conditional random fields as a post-processing performance boost produced only marginal improvements (F1: $< 2\%$ change) for DLAB.H and DLAB.R models. Improvements were furthermore highly dependent on the input image and parameter optimization. However, specific accuracy improvements were found for low-frequency minor classes, as well as minor segmentation corrections for dominant classes. Figure C.9, for example, shows improved resolution of S-I and S-G-R of distant mountains in DST.A.H.2.2, and better classification of WT and RA classes in DST.B.H.2.7. More investigation is needed to explore the possibilities of multi-class CRF modelling for the MLP dataset. In particular, relatively recent work by Zheng et al. [156] has reformulated the CRF mean-field approximate inference with Gaussian pairwise potentials as a recurrent neural network. By integrating the CRF filter into the DCNN training, parameter learning can be transformed into another convolutional filter, where learning the weights of this filter is equivalent to learning the LCF.

7.2 Future Work and Challenges

Due to its speed, throughput and performance, DCNN-based landscape classification has the potential to revolutionize the analysis of landscape change in large datasets, such as the MLP collection. Further study of deep learning approaches can expect to profit from advances in DCNN architecture and algorithms that improve model accuracy, reduce training times, and perhaps most importantly, address the bottleneck of limited ground truth data. DCNN architectures have primarily been designed and trained on large generic image libraries, such as ImageNet[203], COCO and Pascal VOC [191], or texture databases such as Brodatz[113] and CUReT[114], and primarily for application on close-range and relatively low resolution imagery[206]. Similar large-scale generic datasets for annotated landscape imagery are not available. As noted, the reported experimental results are based on training data that samples only a very small fraction ($< 0.1\%$) of the complete MLP collection. This limited dataset does not provide good representation of the data, and limits model sensitivity. More robust models will therefore require that a much wider range of historic photographs be incorporated in the training data.

Resolving limited annotated data will therefore either involve new methods to extend the ground truth data, or that do more with less (or no) annotated data. Buscombe and Ritchie (2018) [34], for example, developed a hybrid method for semantic segmentation that uses the memory efficient mobile DCNN architecture MobileNetV2[157] with CRFs to generate ground-truth data. However, DCNN supervised learning is computationally intensive and often requires millions of examples to train from scratch, and specialized optimizations for domain-specific applications. Emerging semi-supervised, weakly-supervised and unsupervised approaches to DCNN-based segmentation also offer a novel way to reduce or eliminate the need for this labeled training data. Hong et al. (2015)[207], for instance, decoupled classification and segmentation to learn separate networks for each task; Hung et al. (2018)[208] applied adversarial learning to the problem of segmentation. Unsupervised segmentation approaches, such as the iterative self-training procedure developed by Zou et al. (2018) [209], promise to eliminate the need for manual data altogether. The use of generative adversarial networks (GANs) to generate realistic labeled images is another emerging approach to extend data samples for DCNN segmentation networks. GANs are themselves DCNNs that can be trained to generate realistic images by learning the ground-truth dataset distribution in a zero-sum game framework (see: Goodfellow, et al. (2014) [210]). Liu, et al. (2019) [163], used GANs to improve both the segmentation accuracy on classes with imbalanced data, and overall accuracy. Their proposed approach generates supplementary data with pixel-level annotation labels to balance data-distribution within the dataset. GANs can also

be trained to discriminate between ground-truth and predicted, allowing it to detect and correct higher-order inconsistencies[211].

Another future direction to this research is in the use of ensemble learning to combine predictions from multiple models. The aim of this approach would be to reduce prediction variance over different models, as illustrated in Figures 6.1 and 6.2, and based on the evaluation of selectivity in Section 6.7. Given that different models will usually not make all the same errors on the test set[167], ensemble methods attempt to average or "best fit" segmentation output probabilities over multiple outputs. Ensemble training for deep learning segementation has been extensively applied in biomedical science – e.g., Codella, et al. [212]. Accuracy could alternatively be improved by tuning and training models to classify one or two classes and combining the results. Further investigation would be required to determine the relationship between accuracy for a particular class, and the model's configuration and training database. Furthermore, adjusting model testing to input image size and resolution presents another area of model parameter tuning that requires further study, as higher-resolution tiles will naturally correspond to higher resolution (and larger) images. This consideration applies to both historic and repeat models.

Appendix A

Mountain Legacy Project Image Datasets

Training Datasets DST.A.H.1/DST.A.R.1

Historic ID	Year	Surveyor	Station	Repeat ID	Year	Width	Height
DST.A.H.1.1	1913	Bridgland	Stn. 27 Dutch Creek Head No. 1	DST.A.R.1.1	2008	3804	5318
DST.A.H.1.2	1913	Bridgland	Stn. 27 Dutch Creek Head No. 1	DST.A.R.1.2	2008	3850	5329
DST.A.H.1.3	1913	Bridgland	Stn. 27 Dutch Creek Head No. 1	DST.A.R.1.3	2008	3820	5308
DST.A.H.1.4	1913	Bridgland	Stn. 27 Dutch Creek Head No. 1	DST.A.R.1.4	2008	3838	5328
DST.A.H.1.5	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.5	2008	5354	3684
DST.A.H.1.6	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.6	2008	5388	3716
DST.A.H.1.7	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.7	2008	5365	3705
DST.A.H.1.8	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.8	2008	5364	3712
DST.A.H.1.9	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.9	2008	5359	3695
DST.A.H.1.10	1913	Bridgland	Stn. 43 Grassy Ridge	DST.A.R.1.10	2008	5354	3684
DST.A.H.1.11	1913	Bridgland	Stn. 54 Sentinel Pass West No. 2	DST.A.R.1.11	2008	5349	3698
DST.A.H.1.12	1913	Bridgland	Stn. 54 Sentinel Pass West No. 2	DST.A.R.1.12	2008	5362	3700
DST.A.H.1.13	1913	Bridgland	Stn. 54 Sentinel Pass West No. 2	DST.A.R.1.13	2008	5359	3690
DST.A.H.1.14	1913	Bridgland	Stn. 58 Willow Creek No. 2	DST.A.R.1.14	2008	5364	3698
DST.A.H.1.15	1913	Bridgland	Stn. 58 Willow Creek No. 2	DST.A.R.1.15	2008	5374	3694
DST.A.H.1.16	1913	Bridgland	Stn. 6 Bolton No. 1	DST.A.R.1.16	2008	5503	3846
DST.A.H.1.17	1913	Bridgland	Stn. 6 Bolton No. 1	DST.A.R.1.17	2008	5465	3834
DST.A.H.1.18	1913	Bridgland	Stn. 6 Bolton No. 1	DST.A.R.1.18	2008	5469	3809
DST.A.H.1.19	1913	Bridgland	Stn. 12 Boundary No. 2A	DST.A.R.1.19	2008	5473	3830
DST.A.H.1.20	1913	Bridgland	Stn. 12 Boundary No. 2A	DST.A.R.1.20	2008	5490	3840
DST.A.H.1.21	1913	Bridgland	Stn. 12 Boundary No. 2A	DST.A.R.1.21	2008	5471	3821
DST.A.H.1.22	1913	Bridgland	Stn. 12 Boundary No. 2A	DST.A.R.1.22	2008	5488	3841
DST.A.H.1.23	1913	Bridgland	Stn. 12 Boundary No. 2A	DST.A.R.1.23	2008	5499	3857
DST.A.H.1.24	1919	Bridgland	Stn. 235	DST.A.R.1.24	2009	4920	3365
DST.A.H.1.25	1919	Bridgland	Stn. 235	DST.A.R.1.25	2009	4925	3370
DST.A.H.1.26	1922	Bridgland	Stn. 35	DST.A.R.1.26	2008	3462	4985

TABLE A.1: Image and segmentation mask training dataset DSC.A.1. Segmentation masks use land cover classes (see LCC.A classes in Table 2.1) employed in a Landsat-based map of the same area (see Jean et al.[5]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca.

Training Datasets DST.A.H/DST.A.R (continued)

Historic ID	Year	Surveyor	Station	Repeat ID	Year	Width	Height
DST.A.H.1.27	1927	Bridgland	Stn. 443	DST.A.R.1.27	2011	3547	5101
DST.A.H.1.28	1927	Bridgland	Stn. 443	DST.A.R.1.28	2011	5114	3531
DST.A.H.1.29	1927	Bridgland	Stn. 443	DST.A.R.1.29	2011	5112	3537
DST.A.H.1.30	1927	Bridgland	Stn. 443	DST.A.R.1.30	2011	5134	3562
DST.A.H.1.31	1927	Bridgland	Stn. 443	DST.A.R.1.31	2011	5088	3556
DST.A.H.1.32	1927	Bridgland	Stn. 443	DST.A.R.1.32	2011	5118	3545
DST.A.H.1.33	1927	Bridgland	Stn. 444	DST.A.R.1.33	2011	4922	3372
DST.A.H.1.34	1927	Bridgland	Stn. 446	DST.A.R.1.34	2011	4927	3358
DST.A.H.1.35	1927	Bridgland	Stn. 446	DST.A.R.1.35	2011	4917	3358
DST.A.H.1.36	1927	Bridgland	Stn. 446	DST.A.R.1.36	2011	4967	3432
DST.A.H.1.37	1927	Bridgland	Stn. 445	DST.A.R.1.37	2011	5144	3604
DST.A.H.1.38	1927	Bridgland	Stn. 445	DST.A.R.1.38	2011	5123	3553
DST.A.H.1.39	1927	Bridgland	Stn. 445	DST.A.R.1.39	2011	5160	3601
DST.A.H.1.40	1896	Wheeler	Moose Mt Centre	DST.A.R.1.40	2008	3923	5627
DST.A.H.1.41	1897	Wheeler	Forget-me-not-ridge	DST.A.R.1.41	2008	5628	3957
DST.A.H.1.42	1901	Wheeler	Stn. 25 Napoleon	DST.A.R.1.42	2011	6022	4206
DST.A.H.1.43	1901	Wheeler	Stn. 37 Abbot Ridge No. 1	DST.A.R.1.43	2011	3900	5597
DST.A.H.1.44	1901	Wheeler	Stn. 37 Abbot Ridge No. 1	DST.A.R.1.44	2011	3920	5639
DST.A.H.1.45	1901	Wheeler	Stn. 37 Abbot Ridge No. 1	DST.A.R.1.45	2011	3899	5618
DST.A.H.1.46	1901	Wheeler	Stn. 37 Abbot Ridge No. 1	DST.A.R.1.46	2011	3928	5647
DST.A.H.1.47	1901	Wheeler	Stn. 37 Abbot Ridge No. 1	DST.A.R.1.47	2011	3924	5604
DST.A.H.1.48	1911	Wheeler	Ptarmigan Peak No. 1	DST.A.R.1.48	2011	3896	5578
DST.A.H.1.49	1911	Wheeler	Ptarmigan Peak No. 1	DST.A.R.1.49	2011	3897	5604
DST.A.H.1.50	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.50	2011	4852	3406
DST.A.H.1.51	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.51	2011	4921	3425
DST.A.H.1.52	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.52	2011	4910	3443
DST.A.H.1.53	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.53	2011	4870	3412
DST.A.H.1.54	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.54	2011	4877	3428
DST.A.H.1.55	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.55	2011	4876	3422
DST.A.H.1.56	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.1.56	2011	4868	3418

Testing Datasets DST.A.H.2/DST.A.R.2

Historic ID	Year	Surveyor	Station	Repeat ID	Year	Width	Height
DST.A.H.2.1	1922	Bridgland	Stn. 34	DST.A.R.2.1	2008	3453	4940
DST.A.H.2.2	1927	Bridgland	Stn. 443	DST.A.R.2.2	2011	5112	3553
DST.A.H.2.3	1927	Bridgland	Stn. 443	DST.A.R.2.3	2011	5114	3548
DST.A.H.2.4	1924	Wheeler	Stn. 289 - Casket : Coffin Mtn.	DST.A.R.2.4	2011	4946	3415

TABLE A.2: Image and segmentation mask test dataset DSC.A.2 (Testing). Segmentation masks use land cover classes (see LCC.A classes in 2.1) employed in a Landsat-based map of the same area (see Jean et al.[5]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca.

Training Datasets DST.B.H.1/DST.B.R.1

Historic ID	Year	Surveyor	Station	Reference ID	Repeat ID	Year	Width	Height
DST.B.H.1.1	1927	Lambart	Stn. 5	24	DST.B.R.1.1	2014	10821	7672
DST.B.H.1.2	1927	Lambart	Stn. 15	5	DST.B.R.1.2	2014	5558	3948
DST.B.H.1.3	1927	Lambart	Stn. 33	12114	DST.B.R.1.3	2011	5558	3948
DST.B.H.1.4	1927	Lambart	Stn. 35	261	DST.B.R.1.4	2007	3922	5576
DST.B.H.1.5	1927	Lambart	Stn. 36	268	DST.B.R.1.5	2016	4852	6880
DST.B.H.1.6	1927	Lambart	Stn. 37	280	DST.B.R.1.6	2007	3956	5603
DST.B.H.1.7	1927	Lambart	Stn. 39	293	DST.B.R.1.7	2016	7274	4912
DST.B.H.1.8	1927	Lambart	Stn. 42	320B	DST.B.R.1.8	2007	4014	5614
DST.B.H.1.9	1927	Lambart	Stn. 51	390	DST.B.R.1.9	2007	5556	3924
DST.B.H.1.10	1928	Miller	Stn. 1	819	DST.B.R.1.10	2014	6069	4226
DST.B.H.1.11	1928	Miller	Stn. 2	821	DST.B.R.1.11	2014	6008	4192
DST.B.H.1.12	1928	Miller	Stn. 4.5	840	DST.B.R.1.12	2014	5359	3718
DST.B.H.1.13	1928	Miller	Stn. 5	843	DST.B.R.1.13	2014	5380	3736
DST.B.H.1.14	1928	Miller	Stn. 7	859	DST.B.R.1.14	2014	5388	3757
DST.B.H.1.15	1928	Miller	Stn. 8	864	DST.B.R.1.15	2014	5364	3729
DST.B.H.1.16	1928	Miller	Stn. 9	876	DST.B.R.1.16	2014	5383	3722
DST.B.H.1.17	1928	Miller	Stn. 10	880	DST.B.R.1.17	2014	6037	4203
DST.B.H.1.18	1928	Miller	Stn. 13	908	DST.B.R.1.18	2014	6089	4238
DST.B.H.1.19	1928	Miller	Stn. 14	912	DST.B.R.1.19	2014	5397	3741
DST.B.H.1.20	1928	Miller	Stn. 16	924	DST.B.R.1.20	2014	6102	4246
DST.B.H.1.21	1928	Miller	Stn. 18	949	DST.B.R.1.21	2014	6063	4324
DST.B.H.1.22	1928	Miller	Stn. 19	953	DST.B.R.1.22	2014	6084	4200
DST.B.H.1.23	1928	Miller	Stn. 20	957	DST.B.R.1.23	2014	5310	3695
DST.B.H.1.24	1944	Nidd	Stn. 11	44136	DST.B.R.1.24	2012	3866	2895

TABLE A.3: Image and segmentation mask training dataset DSC.B.1. Segmentation masks use land cover classes ((see LCC.B classes in 2.1) employed in a Landsat-based map of the same area (see Fortin et al.[6]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca

Training Datasets DST.B.H.1/DST.B.R.1 (continued)

Historic ID	Year	Surveyor	Station	Reference ID	Repeat ID	Year	Width	Height
DST.B.H.1.25	1944	Nidd	Stn. 12	44167	DST.B.R.1.25	2012	3605	2703
DST.B.H.1.26	1944	Nidd	Stn. 15	42064	DST.B.R.1.26	2012	3971	2973
DST.B.H.1.27	1944	Nidd	Stn. 19	3	DST.B.R.1.27		3980	2963
DST.B.H.1.28	1944	Nidd	Stn. 21	2	DST.B.R.1.28		3970	2941
DST.B.H.1.29	1946	Nidd	Stn. 8	77	DST.B.R.1.29	2016	7250	4912
DST.B.H.1.30	1946	Nidd	Stn. 12	121	DST.B.R.1.30	2016	7001	4891
DST.B.H.1.31	1947	Nidd	Stn. 2	402	DST.B.R.1.31	2016	6732	4820
DST.B.H.1.32	1953	Nidd	Stn. 5	24	DST.B.R.1.32	2016	7360	4912
DST.B.H.1.33	1923	Wheeler	Stn. 248	213	DST.B.R.1.33	2007	3525	5063
DST.B.H.1.34	1923	Wheeler	Stn. 249	219	DST.B.R.1.34	2007	3381	4878
DST.B.H.1.35	1923	Wheeler	Stn. 251	235	DST.B.R.1.35	2007	3430	4886
DST.B.H.1.36	1923	Wheeler	Stn. 254	256	DST.B.R.1.36	2007	3456	4920
DST.B.H.1.37	1924	Wheeler	Stn. 289	5	DST.B.R.1.37	2011	4870	3412
DST.B.H.1.38	1924	Wheeler	Stn. 297	64	DST.B.R.1.38	2011	4888	3417
DST.B.H.1.39	1924	Wheeler	Stn. 309	173	DST.B.R.1.39	2011	4928	3436

Test Datasets DST.B.H.2/DST.B.R.2 (continued)

Historic ID	Year	Surveyor	Station	Reference ID	Repeat ID	Year	Width	Height
DST.B.H.2.1	1928	Miller	Stn. 12	901	DST.B.R.2.1	2014	5368	3725
DST.B.H.2.2	1928	Miller	Stn. 6	852	DST.B.R.2.2	2014	5391	3756
DST.B.H.2.3	1953	Nidd	Stn. 3	500	DST.B.R.2.3	2016	7360	4912
DST.B.H.2.4	1923	Wheeler	Stn. 245	183	DST.B.R.2.4	2007	3513	4993
DST.B.H.2.5	1923	Wheeler	Stn. 247	202	DST.B.R.2.5	2007	4932	3480
DST.B.H.2.6	1923	Wheeler	Stn. 250	222	DST.B.R.2.6	2007	3414	4918
DST.B.H.2.7	1924	Wheeler	Stn. 292	30	DST.B.R.2.7	2011	4870	3379

TABLE A.4: Image and segmentation mask test dataset DSC.B.2. Segmentation masks use land cover classes ((see LCC.B classes in 2.1) employed in a Landsat-based map of the same area (see Fortin et al.[6]). All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca

Test Datasets DST.C.H/DST.C.R (continued)

Historic ID	Year	Surveyor	Station	Reference ID	Repeat ID	Year	Width	Height
DST.C.H.1.1	1916	Nichols	Stn. 39	20378-2	-	-	6000	4152
DST.C.H.1.2	1916	Nichols	Stn. 39	20686	-	-	4800	3318
DST.C.H.1.3	1945	Nidd	Stn. 1	0267	DST.C.R.1.3	2016	6742	4771
DST.C.H.1.4	1946	Nidd	Stn. 10	739	DST.C.R.1.4	2016	5393	3816
DST.C.H.1.5	1940	Parlee	Stn 14	4140	-	-	4799	3443
DST.C.H.1.6	1940	Parlee	Stn 29	4276	-	-	4855	3719
DST.C.H.1.7	1895	Wheeler	Stn. 15	5	-	-	5580	3905
DST.C.H.1.8	1897	Wheeler	Stn. 14	7	-	-	6419	4478

TABLE A.5: Image and segmentation mask test dataset DSC.C. Segmentation masks use land cover classes ((see LCC.C classes in 2.1) employed in a Landsat-based map of the same area. All Mountain Legacy Project data is available freely under a Creative Commons license for non-commercial use at mountainlegacy.ca

Appendix B

Experimental Results and Discussion

B.1 Overview

This appendix is intended to provide a more detailed analysis of the experimental results presented in Chapter 6. The report focuses on an evaluation of the results of DCNN semantic segmentation experiments on the U-Net [section 5.1] and Deeplabv3+ [section 5.2] models. These results are grouped under the following three experiments, also summarized in Table B.1:

6.4 **Data Augmentation**: Application of data augmentation optimized for the semantic class imbalance of the data; this involved a proposed selective augmentation method described in Section 4.4. Tested variables included the size and composition of the prepared input dataset, as outlined in the preprocessing steps;

6.5 **Loss Functions**: Application of three different loss functions for model training: (1) Cross-entropy loss (L_{CE}); (2) Dice similarity coefficient loss (L_{DSC}); and (3) Focal loss (L_F), as defined in Section 4.5.4. Test variables included the type, configuration, and ratio of loss functions as outlined in Section 4.5.4;

6.6 **Conditional Random Fields**: Application of a Conditional random fields (CRF) model to the DCNN output segmentation masks as a post-processing step. CRF inference parameters were optimized according to the method described in Section 4.6.

TABLE B.1: Summary of experiments and model descriptors.

Experiment	Capture	Model	Descriptor
1. Data Augmentation (section 6.4)	Historic (H)	U-Net	UNET.H.1
		Deeplabv3+	DLAB.H.1
	Repeat (R)	U-Net	UNET.R.1
		Deeplabv3+	DLAB.R.1
2. Loss Functions (section 6.5)	Historic (H)	U-Net	UNET.H.2
		Deeplabv3+	DLAB.H.2
	Repeat (R)	U-Net	UNET.R.2
		Deeplabv3+	DLAB.R.2
3. Conditional Random Fields (section 6.6)	Historic (H)	U-Net	UNET.H.3
		Deeplabv3+	DLAB.H.3
	Repeat (R)	U-Net	UNET.R.3
		Deeplabv3+	DLAB.R.3

Experimental results were evaluated according to the Evaluation Criteria and Methods outlined in Section 4.7. Given the extent to which the Deeplabv3+ architecture outperformed the U-Net models in all experiments, the focus of much of this evaluation was given to the Deeplabv3+ experiments, whereas only overall performance metrics are reported for the U-Net. Evaluation of criteria EVAL.1 was based on analysis of training and validation loss metrics, as defined in Section 4.5.4. Evaluation of accuracy criteria EVAL.2 (section 4.7.2), presented in the summarized experimental results (see Sections 6.4], 6.5, 6.6), was based on model test metrics defined in that Section , which include F1 scores, frequency weighted intersection over union ($fIoU$), Matthew's correlation coefficient (MCC), as well as standard precision and recall metrics over the test dataset. Confusion map visualizations were also generated to indicate class-specific accuracy performance, and to help determine systematic misclassifications due to inter-class similarities, for example, misclassifications among coniferous forest (CF), herbaceous/shrub (H-S), and regenerating areas (RA). Evaluation of sensitivity criteria in EVAL.3 (section 4.7.3), presented in Section 6.7, includes a descriptive analysis of sample segmentations of Deeplabv3+ test images, drawn from the test datasets DST.A.2, DST.B.2, DST.C, to highlight observations relevant to the EVAL.3 criteria, as well as EVAL.2 metrics. Evaluation of model efficiency (EVAL.4), including training and inference latency, is presented in Section 6.8.

B.1.1 Key Findings

The following summarizes the key conclusions from the chapter's evaluation of the experimental results.

1. Deeplabv3+ was found to perform significantly better (F1 Scores $> 15-25\%$) over U-Net across the evaluation metrics for both historic and repeat image captures on test image datasets DST.A.2, DST.B.2, and DST.C. Experimental results give promising evidence that, provided selective data augmentation and careful tuning of model parameters, Deeplabv3+ architecture can offer a feasible approach for dense classification of terrestrial oblique photography.

2. Semantic segmentation of historic (grayscale) achieved a top mean F1 score of 0.839 (frequency-weighted IoU of 0.753 and MCC of 0.742), for model DLAB.H.2.3 trained on H-Mrg. Both model back-propagation gradients used an equal weighting of losses L_{CE} and L_{DSC} with no focal loss L_F.

3. Semantic segmentation of repeat (colour) capture images achieved a top mean F1 score of 0.909 (frequency-weighted IoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F.

4. For historic captures, data augmentation (H-Aug) gave a moderate overall performance improvement for DLAB.H.1.2 (F1: +3.0%, fIoU: +4.0%, MCC: +3.6%). Augmentation further showed significant performance improvements for minor classes B-MW (F1: +125.0%) and WL (F1: +218.3%), and noticeable improvements to classes WT (F1: +3.5%) and S-I (+10.6%), without any significant performance loss for other classes. Augmentation by grayscaled repeat images (H-Mrg) did not noticeably alter DLAB.H.1.2 average performance (F1: −0.2%, fIoU: +0.4%, MCC: +0.4%), with increased accuracy across minor classes except for B-MW (F1: −24.4%), RA (F1: −10.9%). All DLAB.H models showed excellent classification accuracy (F1: > 0.80) for dominant classes NC and CF.

5. For repeat captures, data augmentation (R-Aug) only slightly increased both the average F1 score (+1.0%) and the weighted F1 score (+0.6%) over R-Ext. However, augmentation led to improved F1 scores for minor classes B-MW (+9.3%), WL (+8.2%), WT (+19.7%) and RA (+8.4%), and $< 0.5\%$ change in class S-I. All DLAB.R models showed excellent classification accuracy (F1: > 0.80) for classes NC, CF, S-G-R, and RA.

6. Models with the top aggregate accuracy (EVAL.2) did not consistently produce individual output segmentations of high EVAL.3 sensitivity metrics and qualitative assessment (see discussion in Section 6.7), but in many cases EVAL.3 performance was highly dependent on the input image. This suggests an ensemble of models may improve performance. As found with Buscombe and Ritchie (2018) [34], overall class accuracy is less informative than the prediction performance for each

class, in which case fine-tuning of model hyperparameters is required to reduce the misclassification of classes with high inter-class similarity.

7. Conditional random fields as a post-processing step produced only modest performance (F1: $< 2\%$ change) improvements for DLAB.H and DLAB.R models. Improvements were furthermore highly dependent on the input image and parameter optimization. CRF results are presented in Table B.11.

B.2 Model Training/Validation Losses

Training and validation loss values were recorded at regular intervals to track the model's ability to determine correct predictions for all weights and biases for the labeled data (see Sections 4.5 and 4.7.1). Average losses indicate how poorly the model's prediction was on a series of input example batches. Each model corresponds to different gradient weightings of loss functions, as summarized in Tables 6.7 and 6.8. Note that model DLAB.H.1.3 used the same configuration as DLAB.H.2.1.

Based on the heuristics described in EVAL.1 (generalization) in Section 4.7.1, all training and validation losses showed improved generalization over datasets DST.A.1 and DST.B.1. Training losses reported in Tables 6.1 and 6.2 showed fast convergence for extracted training databases (H-Ext, R-Ext), slower convergence for the larger augmented databases (H-Aug, R-Aug), and even slower convergence for the much larger merged database (H-Mrg). For H-Ext and R-Ext, minimum validation CE loss (L_{CE}) was reached at approximately 7,000 (epoch 5) and 5,000 (epoch 5) iterations respectively (where one iteration is equal to 20 batches of 8 examples). However, DSC loss (L_{DSC}) continued to decrease beyond these minima, and peak accuracy is obtained at approximately 20,000 iterations (epoch 29) and 12,000 (epoch 9) respectively. For the augmented and merged databases (H-Aug, H-Mrg, R-Aug), convergence was reached earlier for L_{CE} and L_F, with the exception of DLAB.H.2.1, which saw decreasing L_{DSC} to converge at 50,000 iterations (epoch 39). The discrepancy or gap between training and validation error for both historic and repeat databases, as shown in Table B.2, also approaches near zero at convergence with increasing training database size. Oscillations in the loss gap curves indicate random shuffling of the dataset at each epoch was restricted to the size of the database buffer (1,000 images).

From the foregoing, it can be concluded that L_{DSC} provided the slowest, but most effective loss convergence during training and validation. Given the specific sensitivity of L_{DSC} to accurate segmentation overlap (see Section 4.5.4.2), and the attendant disadvantages of L_{CE} as discussed in Section 4.5.4.4, this was expected. For historic H-Mrg

models, L_{CE}, and in some cases L_F, showed overfitting after 30,000 iterations, whereas L_{CE} continued to drop. It is important to note that all models used some weighting of L_{DSC} as an objective function. However, as shown in the following Section , low L_{DSC} loss gaps did not necessarily translate into top performance for EVAL.2 (accuracy) on the test dataset, which includes images from different surveys (see DST.C in Appendix A).

B.2.1 Loss Gap Measurements for Both Models

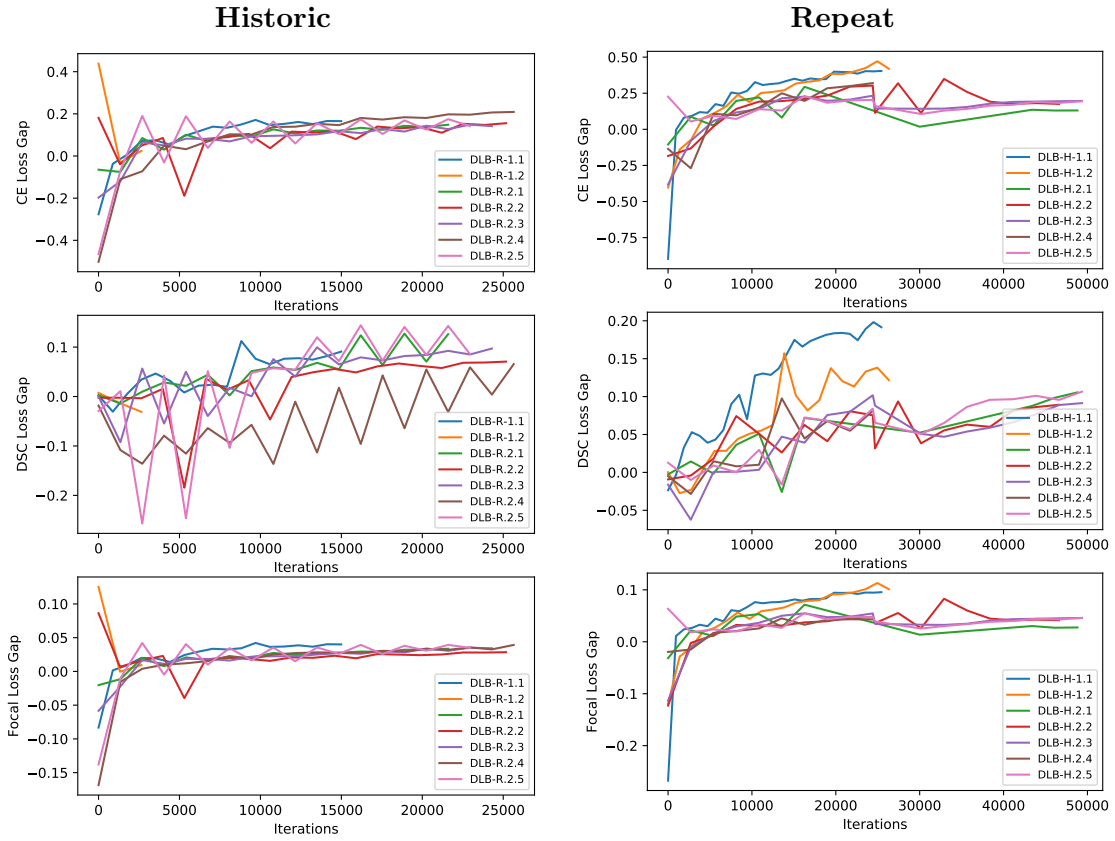


TABLE B.2: Measured gaps between training and validation losses (L_{CE}, L_{DSC}, L_F) for historic and repeat models. See also Sections 4.5 and 4.7.1.

B.3 Experiment 1: Data Augmentation

Experiment #1 tested the use of data augmentation to address semantic class imbalance, using a custom augmentation method, as described in Section 4.4. Imbalanced data, underrepresented data, and severe class distribution skews, can significantly compromise the performance of DCNN learning algorithms [182]. Data augmentation is frequently used to address label imbalance in the training dataset. The sampling method described in Section 4.4.2.2 proposed a procedure for selecting appropriate training samples for augmentation based on the number of minor class pixels, and was shown to marginally improve class imbalance (see Figures 5.2 and 5.3).

For historic data augmentation results, summarized in Table B.3, performance impact by data augmentation compared three overlapping training databases: (1) H-Ext (extracted), (2) H-Aug (augmented), and (3) H-Mrg (H-Aug merged with grayscaled R-Aug). For repeat data augmentation results, summarized in Table B.7, performance for two training databases was evaluated: (1) R-Ext (extracted), and (2) R-Aug (augmented). Classification reports and normalized confusion matrix visualizations are presented for historic captures in Tables B.4, B.5, and B.6, and for repeat captures in Tables B.8 and B.9. Each value in the cells of the confusion matrix visualizations is equal to the number of observations known to be in semantic class i but predicted to be in class i normalized by the support $n_{\mathcal{D}}$ (total pixel count for i). All experiments used an equal weighting of loss functions, and weighted cross entropy loss used class weights, as defined in Section 4.5.4.1.

Note that a stronger case could be made for the efficacy of data augmentation through a comparison with randomized augmentation (randomly selected tiles) sampling algorithm (section 4.4.2.2).

B.3.1 Historic Image Models

Results of model training with data augmentation for historic (grayscale) captures is summarized in Table B.3. A substantial performance gap between U-Net and Deeplab models was observed, which is speculated to be the result of using a deep residual network pre-trained on ImageNet to initialize the DeepLab ResNet encoder (see Section 5.2).

For historic captures, the augmented database (H-Aug, model:DLAB.H.1.2) showed moderately improved model generalization and accuracy over the extraction database (H-Ext, model DLAB.H.1.1), with scores F1: +3.0%, fIoU: +4.0%, MCC: +3.6%. Broken down by class, improvements in accuracy for B-MW (F1: +125.0%), WL (F1:

+218.3%), WT (F1: +3.5%) and S-I (+10.6%) were observed, without any significant performance loss for other classes. Minor class RA showed little accuracy improvement with augmentation, however fewer misclassifications, as shown in the confusion map. With the exception of RA, classes with improved accuracy correspond to those the greatest increase in augmentation shown in Table 5.2. Augmentation by grayscaled repeat images (H-Mrg) did not noticeably alter DLAB.H.1.2 average performance (F1: −0.2%, fIoU: +0.4%, MCC: +0.4%), with increased accuracy across minor classes except for B-MW (F1: −24.4%), RA (F1: −10.9%). All DLAB.H models showed excellent classification accuracy (F1: > 0.80) for dominant classes NC and CF.

Data Augmentation Results: Historic Models				
Model	Database	$F1$ Score	$fIoU$	MCC
UNET.H.1.1	H-Ext	0.612	0.563	0.522
UNET.H.1.2	H-Aug	0.631	0.572	0.552
UNET.H.1.3/2.1	H-Mrg	0.654	0.603	0.587
DLAB.H.1.1	H-Ext	0.815	0.727	0.713
DLAB.H.1.2	H-Aug	0.820	0.735	0.718
DLAB.H.1.3/2.1	H-Mrg	**0.835**	**0.755**	**0.737**

TABLE B.3: Experiment H.1: Results summary for data augmentation (historic captures).

Class	Prec	Rec	F1	Sup
NC	0.940	0.940	0.940	0.585
B-MW	0.670	0.116	0.198	0.042
C	0.776	0.829	0.801	0.203
H-S	0.463	0.625	0.532	0.073
S-G-R	0.634	0.708	0.669	0.067
WL	0.644	0.060	0.109	0.012
WT	0.650	0.556	0.600	0.002
S-I	0.632	0.730	0.678	0.008
RA	0.503	0.319	0.391	0.008
cAvg	0.657	0.543	0.546	1.000
wAvg	0.830	0.827	0.816	1.000
Accuracy:		0.827	**wIoU:**	0.727
F1 score:		0.816	**MCC:**	0.713

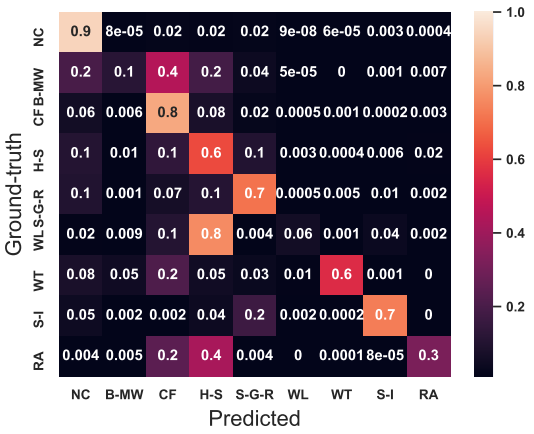


TABLE B.4: Accuracy metrics and confusion matrix for model DLAB.H.1.1 trained on the extracted database (H-Ext, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

Class	Prec	Rec	F1	Sup
NC	0.958	0.946	0.952	0.585
B-MW	0.609	0.354	0.447	0.042
C	0.827	0.831	0.829	0.203
H-S	0.442	0.607	0.511	0.073
S-G-R	0.660	0.692	0.676	0.067
WL	0.691	0.231	0.347	0.012
WT	0.749	0.531	0.621	0.002
S-I	0.726	0.775	0.750	0.008
RA	0.368	0.420	0.392	0.008
cAvg	0.670	0.599	0.614	1.000
wAvg	0.849	0.841	0.841	1.000
Accuracy:		0.841	**wIoU:**	0.757
F1 Score:		0.841	**MCC:**	0.739

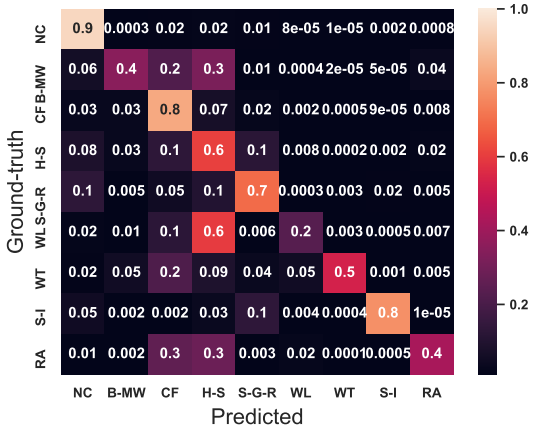


TABLE B.5: Accuracy metrics and confusion matrix for model DLAB.H.1.2 trained on the augmented database (H-Aug, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

Class	Prec	Rec	F1	Sup
NC	0.950	0.946	0.948	0.585
B-MW	0.718	0.221	0.338	0.042
C	0.807	0.853	0.829	0.203
H-S	0.490	0.618	0.547	0.073
S-G-R	0.684	0.731	0.707	0.067
WL	0.711	0.274	0.396	0.012
WT	0.767	0.542	0.635	0.002
S-I	0.675	0.794	0.730	0.008
RA	0.299	0.418	0.349	0.008
cAvg	0.678	0.600	0.609	1.000
wAvg	0.849	0.844	0.839	1.000
Accuracy:		0.844	**wIoU:**	0.754
F1 Score:		0.839	**MCC:**	0.742

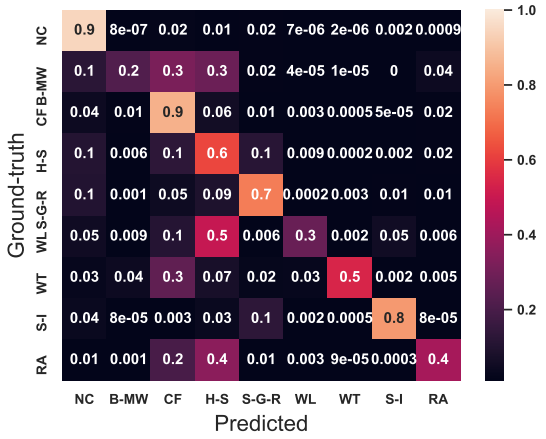


TABLE B.6: Class accuracy and confusion matrix for model DLAB.H.1.3/2.1 trained on the merged database (H-Mrg, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

B.3.2 Repeat Image Models

Results of model training with data augmentation for repeat (colour) captures is summarized in Table B.7. UNET.R and DLAB.R models performed considerably better than their historic counterparts, as it is speculated that the additional RGB channels provided more information for feature learning – i.e. the DCNN is better able to automatically discover the representations needed for segmentation of the raw data. The role of colour in representing textural features remains controversial in the literature [213]. Sun et al. (2014) [117] claim it is not clear whether color offers useful information for classification, especially under varying illumination conditions, and may only make a minor influence on classification. However, the above results for colour repeat images are supported by similar findings in the work of Jean et al. [56] specifically on DST.A using textural analysis techniques.

Classification of repeat capture images achieved a top mean F1 score of 0.909 (fIoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F. Trained on the augmented database (R-Aug), the model showed modest improvement over the average F1 score (+1.0%) over the extracted database (R-Ext), but also exhibited a small uptick in the weighted F1 score (+0.6%). Moreover, data augmentation showed improved F1 scores for minor classes B-MW (+9.3%), WL (+8.2%), WT (+19.7%) and RA (+8.4%), and $<$ 0.5% change in class S-I. As with the historic captures, these improvements correspond to the augmented minor classes summarized in Table 5.3. All DLAB.R

models showed very good classification accuracy for dominant classes NC, CF, S-G-R, and RA (F1: > 0.800).

Model	Database	$F1$ Score	$fIoU$	MCC
UNET.R.1.1	R-Ext	0.686	0.632	0.665
UNET.R.1.2	R-Aug	0.711	0.689	0.701
DLAB.R.1.1	R-Ext	0.904	0.837	0.855
DLAB.R.1.2	R-Aug	**0.912**	**0.848**	**0.868**

TABLE B.7: Summary of accuracy metrics for extracted (H-Ext), augmented (H-Aug) and merged (H-Mrg) training databases. See Section 4.7.2 for metric definitions.

Class	**Prec**	**Rec**	**F1**	**Sup**
NC	0.962	0.985	0.973	0.504
B-MW	0.570	0.984	0.722	0.045
C	0.955	0.853	0.901	0.260
H-S	0.766	0.572	0.655	0.055
S-G-R	0.863	0.830	0.847	0.105
WL	0.638	0.708	0.671	0.018
WT	0.676	0.499	0.574	0.003
S-I	0.826	0.808	0.817	0.006
RA	0.823	0.815	0.819	0.003
cAvg	0.786	0.784	0.775	1.000
wAvg	0.913	0.903	0.904	1.000
Accuracy:		0.903	**wIoU:**	0.837
F1 Score:		0.904	**MCC:**	0.855

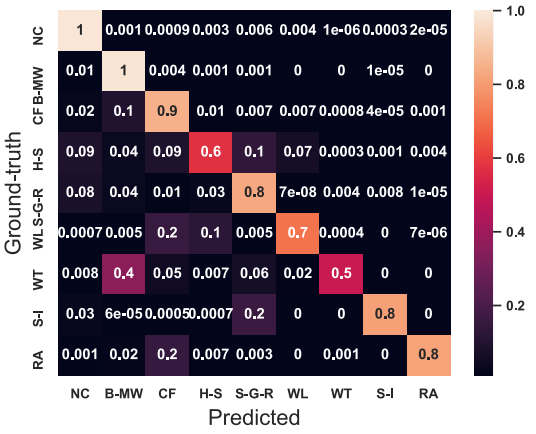


TABLE B.8: Class Accuracy and confusion matrix for model DLAB.R.1.1 trained on extracted database (R-Ext, see Table 5.1). Support total: 281,240,583 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

Class	Prec	Rec	F1	Sup
NC	0.955	0.986	0.970	0.504
B-MW	0.682	0.936	0.789	0.045
C	0.958	0.876	0.915	0.260
H-S	0.720	0.661	0.689	0.055
S-G-R	0.856	0.828	0.842	0.105
WL	0.764	0.692	0.726	0.018
WT	0.773	0.618	0.687	0.003
S-I	0.817	0.812	0.814	0.006
RA	0.905	0.873	0.888	0.003
cAvg	0.826	0.809	0.813	1.000
wAvg	0.915	0.913	0.912	1.000
Accuracy:		0.913	**wIoU:**	0.848
F1 Score:		0.912	**MCC:**	0.868

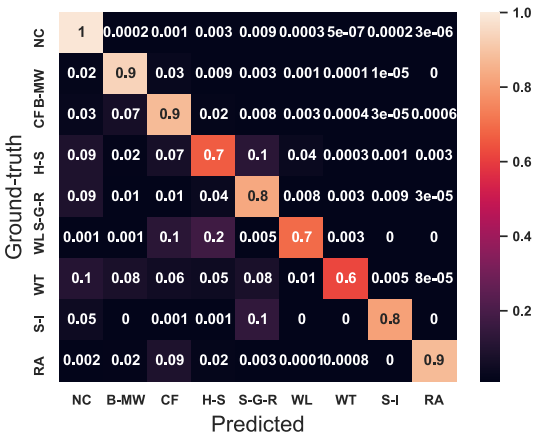


TABLE B.9: Class Accuracy and confusion matrix for model DLAB.R.1.2 trained on augmented database (R-Aug, see Table 5.1). Support total: 281,240,583 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

B.4 Experiment 2: Loss Functions

Experiment #2 evaluated model performance based on the type and configuration of training and validation loss functions. Three loss functions were investigated: (1) L_{CE}: Cross-entropy loss; (2) L_{DSC}: Dice similarity coefficient loss; and (3) L_F: Focal loss, as defined in Section 4.5.4. Additionally, the impact of using weighted cross-entropy loss (L_{WCE} on model performance was evaluated, as indicated by the w superscript).

In general, results for both historic and repeat captures showed only moderate variation in accuracy for different model loss functions. As discussed in Section 6.3, models that use L_{DSC} had more accurate segmentation overlap (see Section 4.5.4.2), which allowed for convergence at a lower error rate than other loss functions. This was anticipated given the advantages of directly measuring segmentation overlap, as well as the limitations of using L_{CE} pixel-wise losses as discussed in Section 4.5.4.4. Focal loss (L_F), which is intended to place more weight on hard examples (low frequency classes), did not appear to significantly improve convergence or accuracy. However, the scaling parameter L_F was determined empirically, and other values of the scaling parameter γ may improve these results. Hossain et al.[204], for example, have proposed an adaptive focal loss for semantic segmentation, where γ is a learnable parameter tuned through backpropagation, rather than empirical tuning.

B.4.1 Historic Image Models

For historic (grayscale) models (DLAB.H), the top mean F1 score of 0.841 (frequency-weighted IoU of 0.757 and MCC of 0.744) was attained using model DLAB.H.1.2 trained on the augmented database (H-Aug); and a top mean F1 score of 0.839 (frequency-weighted IoU of 0.753 and MCC of 0.742), for model DLAB.H.2.3 trained on the merged database (H-Mrg). Both model back-propagation gradients used an equal weighting of losses L_{CE} and L_{DSC} with no focal loss L_F. Class weights did not appear to significantly improve accuracy.

In contrast to the comparable average accuracy scores across models, a comparison of different model outputs for two sample images in Figure 6.1 shows significant prediction variance. For example, output segmentations for DST.A.H.2.1 for models DLAB.H.2.1, DLAB.H.2.2, and DLAB.H.2.5 show misclassification of regions of the background mountain range as NC, whereas models DLAB.H.1.1, DLAB.H.1.2, and DLAB.H.2.3 show a much more accurate, clearer, and unbroken classification of the same region. Similarly, classification of the left foreground in the same image as either barren rock and herbaceous/shrub (S-G-R or H-S) or not classified (NC) differs considerably between models. On the other hand, DST.B.H.2.4 shows much less variation, due perhaps to the clarity and even illumination of the photo, and prediction variance is more concentrated on the minor classes, such as WT and H-S.

To reduce prediction variance, ensemble learning is recommended to combine the predictions from multiple models. It appears possible that accuracy could be substantially improved by training models to classify one or two classes and combine the results. Further investigation is required to determine the relationship between accuracy for a particular class, and the model's configuration and training database.

B.4.2 Repeat image models

For repeat (colour) models (DLAB.R), segmentation outputs achieved a top mean F1 score of 0.909 (frequency-weighted IoU of 0.844 and MCC of 0.864) for model DLAB.R.2.2 trained on R-Aug using an equal weighting of losses L_{WCE} (weighted by class frequency), L_{DSC} and L_F. Unlike historic models, the weighted cross-entropy and focal loss appeared to marginally improve average accuracy. As illustrated in Figure 6.1 with DST.B.R.2.6, repeat models showed much less prediction variance than historic models, and much of that was concentrated on minor classes, whereas dominant classes were well classified.

B.5 Experiment 3: Conditional Random Fields

Experiment #3 examined the impact to model performance based on the use of an optimized fully-connected conditional random fields (CRF) model, as described by Krähenbühl and Koltun [39] [3], applied as a post-processing enhancement to DCNN outputs. CRF inference and parameter optimization is described in Sections 4.6 and 5.5. Note that only results for DLAB.H were produced for this study.

CRF post-processing produced mixed results as far as improvements to accuracy of DCNN outputs, and in many cases resulted in little to no change in F1 scores. CRF inference results, furthermore, showed a high degree of variability given small changes in parameter values for the label compatibility function (LCF) $\mu(x_u, x_v)$. The LCF captures the compatibility between different pairs of labels, which is particularly useful for imbalanced datasets, as it provides a way to weight the potentials of minor and dominant classes. In this experiment, LCF parameters were computed as a vector using the L-BFGS algorithm[190], which proved time-consuming for high-resolution, multi-class segmentation over even the small dataset used. Furthermore, it is not clear that the limited validation dataset provided sufficient data to optimize parameters.

However, some test images did bear accuracy improvements for low-frequency minor classes, as well as minor segmentation corrections for dominant classes. Figure C.9, for example, shows improved resolution of S-I and S-G-R of distant mountains in DST.A.H.2.2, and better classification of WT and RA classes in DST.B.H.2.7.

More investigation is needed to explore the possibilities of multi-class CRF modelling for the MLP dataset. In particular, relatively recent work by Zheng et al. [156] has reformulated the CRF mean-field approximate inference with Gaussian pairwise potentials as a recurrent neural network. By integrating the CRF filter into the DCNN training, parameter learning can be transformed into another convolutional filter, where learning the weights of this filter is equivalent to learning the LCF.

Model	$F1$ Score	$fIoU$	MCC	% Change (F1)
DLAB.H.2.2	0.836	0.755	0.743	0.11%
DLAB.H.2.3	**0.841**	**0.759**	**0.751**	0.23%
DLAB.H.2.4	0.839	0.749	0.737	0.60%

TABLE B.10: Experiment 3: Conditional Random Fields Filter (historic captures)

Class	Prec	Rec	F1	Sup
NC	0.963	0.921	0.948	0.585
B-MW	0.481	0.390	0.434	0.042
C	0.807	0.843	0.818	0.203
H-S	0.490	0.587	0.494	0.073
S-G-R	0.684	0.756	0.709	0.067
WL	0.823	0.430	0.563	0.012
WT	0.767	0.653	0.680	0.002
S-I	0.679	0.824	0.721	0.008
RA	0.384	0.472	0.423	0.008
cAvg	0.662	0.642	0.641	1.000
wAvg	0.823	0.844	0.839	1.000
Accuracy:		0.851	**wIoU:**	0.754
F1 Score:		0.841	**MCC:**	0.749

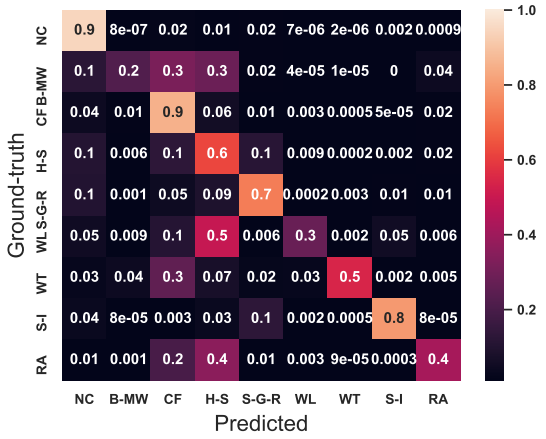


TABLE B.11: Class accuracy and confusion matrix for model DLB-H.2.3 + CRF trained on the merged database (H-Mrg, see Table 5.1). Support (Sup) total: 449,533,967 pixels. Accuracy is the overall pixel accuracy of predictions with respect to the ground-truth. *cAvg* values equal the unweighted mean per class, and *wAvg* equal the support-weighted mean per class.

B.6 Evaluation of Model Sensitivity

In this Section , model sensitivity is evaluated (see Evaluation Methods in Section 4.7.3), and some of the systematic and dataset-specific classification errors observed in the results are described. Sensitivity describes the robustness of model classification to image variation and noise. In general, effective segmentation allows for the features of segment interiors to be invariant to appearance change so as to form homogeneous pixels clusters, while the features of boundaries should be selective to slight appearance change to distinguish objects and surrounding pixels.

Evaluation criteria under consideration for test dataset (DST.A.1, DST.B.1, and DST.C) focus on errors that largely fall under two categories of classification errors: intra-class difference and inter-class similarity. Intra-class differences occur when the same pixel classes imaged under different conditions may appear different. Inter-class similarity occurs when different pixel classes imaged under certain conditions may appear similar. Image samples are presented to demonstrate model sensitivity to these errors based on illumination or photometric variation (EVAL.3.3.a), occlusion (EVAL.3.3.b), scale (EVAL.3.3.c), and noise (EVAL.3.4.a).

In general, and as anticipated from the discussion on segmentation challenges in Section 2.3.3, discrimination between vegetation classes proved the most challenging for classification, whereas the classification boundary between vegetation and non-vegetation was substantially more distinct. The difficulty of classifying vegetation classes is compounded for the minor (low-frequency) classes that include R-A, B-MW and H-S areas,

which had previously proven difficult for manual segmentation [12]. With the historic extraction database (H-Ext), for example (see confusion map in Figure B.4), systematic misclassification of R-A regions was split between CF and H-S. Models trained on the augmented databases (H-Aug) – as indicated in Figure B.5 confusion map – corrected many NC misclassifications and significantly improved recall for classes B-MW (+60.0%), WL (+233.1%) and RA (+31.7%), but with lower precision. However, inter-class similarity misclassification for vegetation classes are less of a problem for change detection than errors that mistake non-vegetation classes, suggesting augmentation improved the "quality" of misclassifications.

B.6.1 Photometric Invariance

Overall, experimental results show both historic and repeat models have very good invariance to photometric differences in images. Cloud shade across continuous segments of land cover classes, for example image DST.B.H.2.2 in Figure C.4, did not significantly affect classification accuracy of the affected vegetation classes CF, H-S and RA. Fog or cloud-obscured regions where visibility of landscape features was reduced did, however, appear to negatively impact accuracy, as for example in images DST.A.H.2.2 and DST.A.H.2.2 in Figure C.1, where the definition between classes S-G-R and CF or H-S is somewhat less well-defined for distant than for clear images, such as image DST.B.H.2.3 in Figure C.3.

B.6.2 Multi-scale Invariance

Detecting objects at different scales is a critical attribute for landscape image segmentation. DCNNs, however, are not inherently scale invariant, and much work has been done to improve multi-scale object classification (see Section 3.4.5). For MLP segmentation, robust multi-scale classification was found to be critical for two primary reasons: (1) foreground regions were typically omitted from landcover analysis; (2) input images were digitized at different resolutions. Deeplabv3+ tackles scale by merging features of multiple scales through Atrous Spatial Pyramid Pooling (ASPP)[2]. ASPP aggregates features from many intermediate convolutional layers and the input feature map, exploits multi-scale features by employing multiple parallel filters with different rates [2]. Given that objects of the same class can have different scales in the image, ASPP helps to account for the different scales, which can improve accuracy.

Differences in scale of same pixel class texture – for example, foreground versus background representations of coniferous forest (CF) or barren rock (S-G-R), or classification of low-resolution images – can introduce both intra-class and inter-class variation that

can affect proper classification. (See discussion under "Foreground/Background Representation" in Section 2.3.3). For instance, in Figure 6.1, output segmentations for DST.A.H.2.1 show the classification of the left foreground varies between barren rock and herbaceous/shrub (S-G-R or H-S) or not classified (NC). These areas may be classified as NC (not classified), depending on an unknown threshold learned by the model that encodes the boundary between close-range foreground and pixels in the normal classification range.

More generally, prediction variance is closely tied to image scale, as illustrated in Figure 6.2, where the classification of minor classes, in particular, varies significantly at different resampled sizes – e.g., differences in H-S (herbaceous) and WT (water) classification for different resampled versions of image DST.B.H.2.3. As well, the classification of foreground pixels (e.g., the improvement to WT (wetland) classification in the lower portion of image DST.B.H.2.3), is highly dependent on image size and resolution. These observations suggest that averaging results over an ensemble of outputs at different scales for the same model might improve performance. Furthermore, adjusting model testing to input image size and resolution presents another area of model parameter tuning that requires further study, as higher-resolution tiles will naturally correspond to higher resolution (and larger) images. This consideration applies to both historic and repeat models.

B.6.3 Image noise Invariance

For the historic photos specifically, model results were marginally affected by photographic and digitization artefacts, such as scratches and localized emulsion defects along photo edges of input images, as well as other occasional but visible damages to the glass plate negatives for the historic photos [5]. (These errors are negligible on photo images from the repeat dataset.) Edge artefacts seen in images DST.A.H.2.4 [C.2], DST.B.H.2.3 [C.3], DST.C.H.1.8 [C.7], and DST.C.H.1.5 [C.8] were correctly categorized as NC. Figure 6.3 shows horizontal marks in image DST.A.H.2.2 and vertical scratches in image DST.H.B.2.1 that appear to affect the classification of pixels surrounding where the marks superimpose on the masks. It is possible that, given a larger training dataset, invariance to noise would improve. Adding artificial perturbations to the augmented samples, such as added Gaussian noise, speckles, or random illumination to the grayscaled repeat images, offers a potential approach to improve noise invariance not explored in this thesis.

B.7 Evaluation of Tile Reconstruction

As discussed in Section 4.5.5, reconstruction of the DCNN output tiles to full-sized masks, involves a process of overlapping and blending adjacent tiles by averaging partial probabilities over the overlapped regions. This technique helps to soften the edge discontinuities between tiles, but can still result in sharp horizontal and vertical discontinuities, as seen in DST.C.H.1.5 (Figure C.8. These artefacts are the result of the predicted tile classified out of context with its neighbouring tiles, where this context allows the model to graduate the transitions between classes. For example, a fully coniferous (CF) tile located next to a transition tile to herbaceous/shrub (H-S) will not account for the proximity to H-S, resulting in a much lower unary potential for H-S at the boundaries.

Two proposed solutions to the problem of tile discontinuities – not explored in this thesis – are to (1) merge different scales of segmentation outputs (for example, combining outputs resampled at 1.0× 0.5× and 0.25×, as shown in Figure 6.2); and (2) use ensemble learning to combine the predictions from multiple neural network models to reduce prediction variance.

Appendix C

Example Segmentation Maps

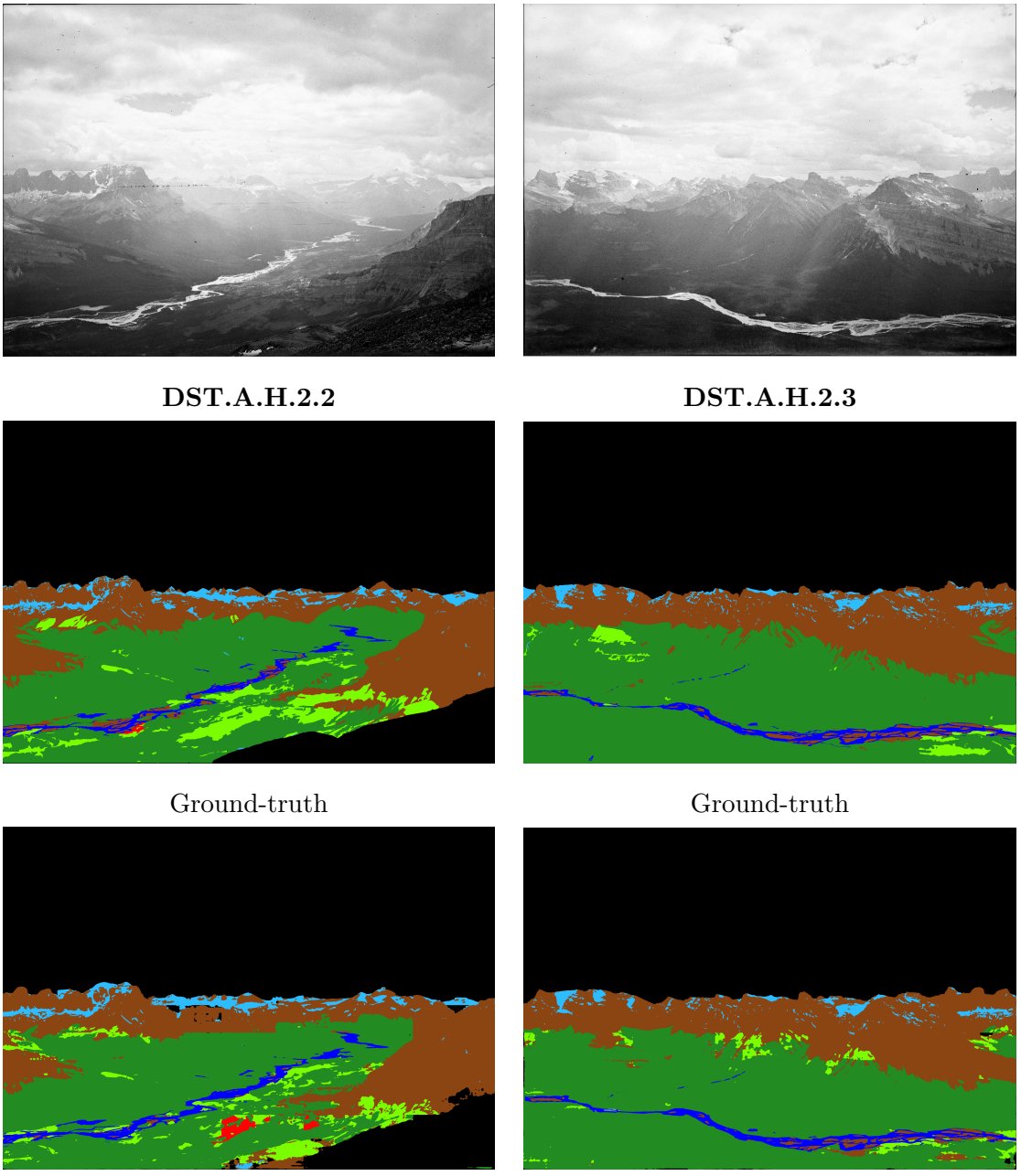


FIGURE C.1: DST.A.H.2.2 Image shows poor background visibility due to cloudiness. Scratches and marks visible across the middle of the photo. Bottom-right corner foreground pixels not classified. Moderate representation of minor classes [WT, S-I, RA]. DST.A.H.2.3 Image shows some visibility occlusion due to cloudiness and rain. Photo is grainy. Moderate representation of minor classes. [WT, S-I, H-S].

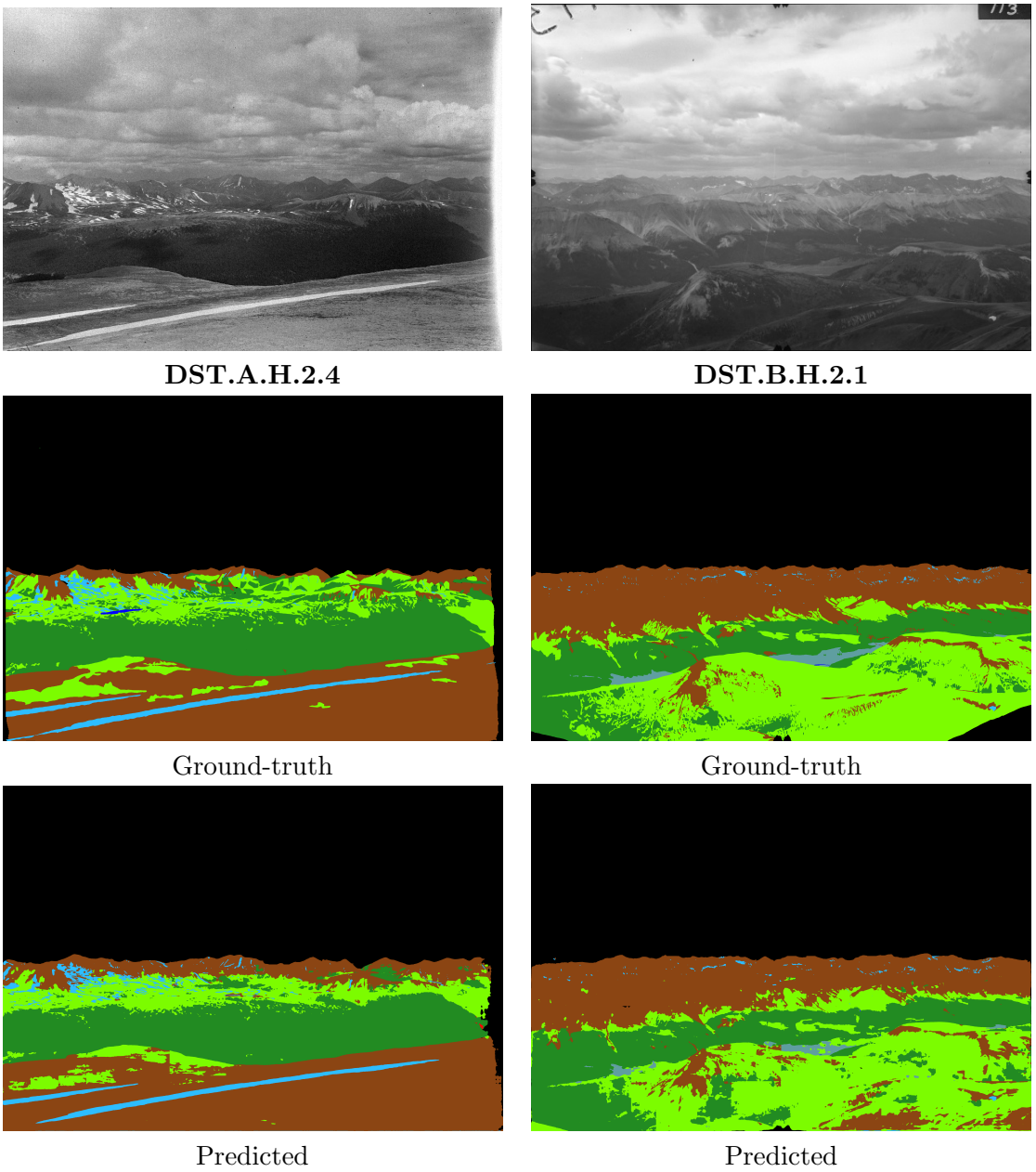


FIGURE C.2: DST.A.H.2.4: Some visibility occlusion due to cloudiness and rain. Emulsion edges are washed out (not classified). Photo is grainy. [S-I]. DST.B.H.2.1: Photo catalog markings visible on corners and along edges. Bottom corner regions contain foreground pixels not classified. [WL, H-S]. Output segmentations shown from DLAB.H.2.5.

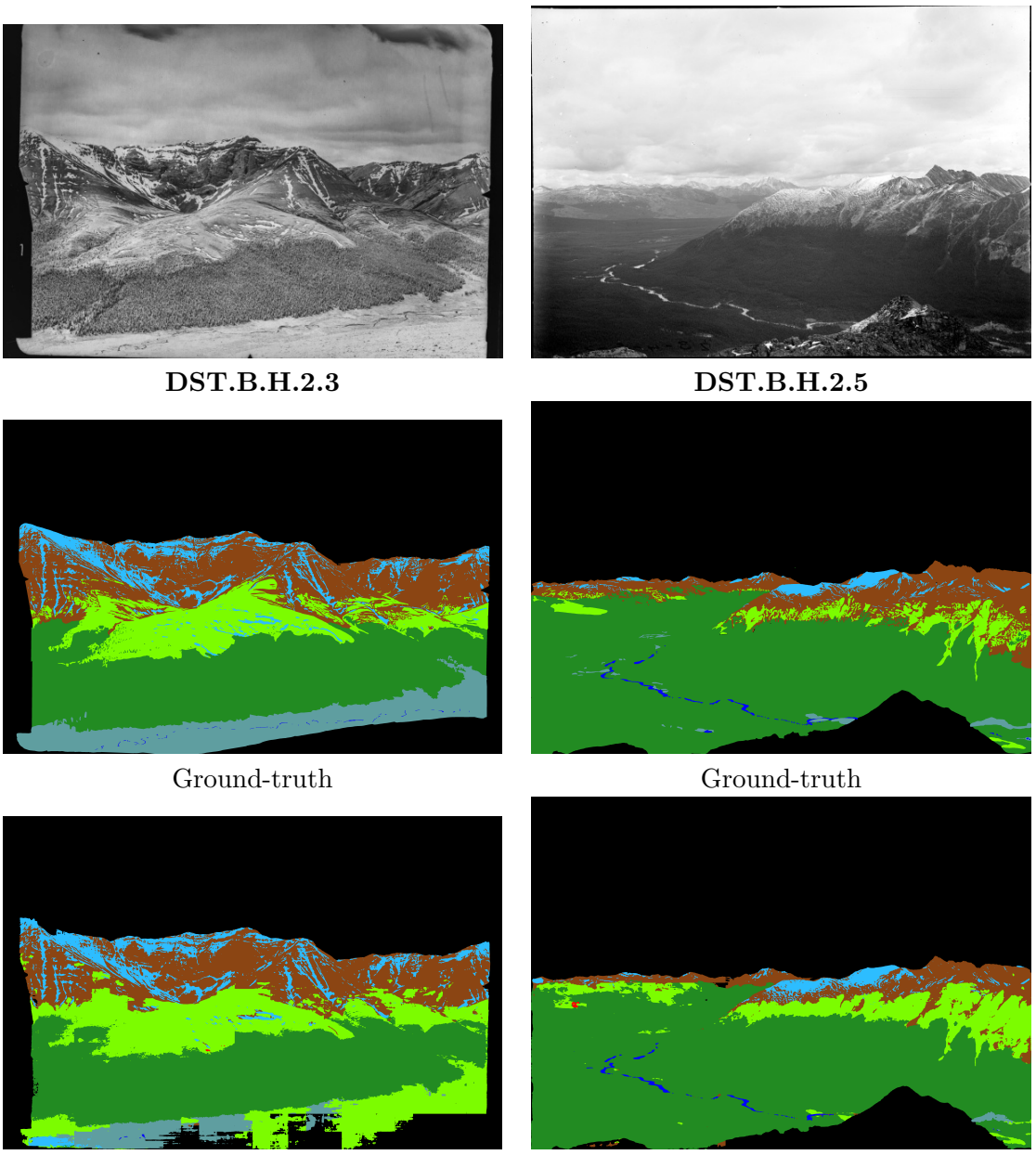


FIGURE C.3: DST.B.H.2.3: Photo image is distorted by plate inconsistencies. Emulsion edges are washed out with a wide gradient. Photo is also grainy. Significant border around the image is not classified. [WL, WT, H-S, S-I]. DST.B.H.2.5: Some visibility occlusion due to cloudiness and rain. Emulsion defects at edges. Photo is grainy. Large bottom-centre foreground region is not classified. [WL, WT, H-S, S-I]

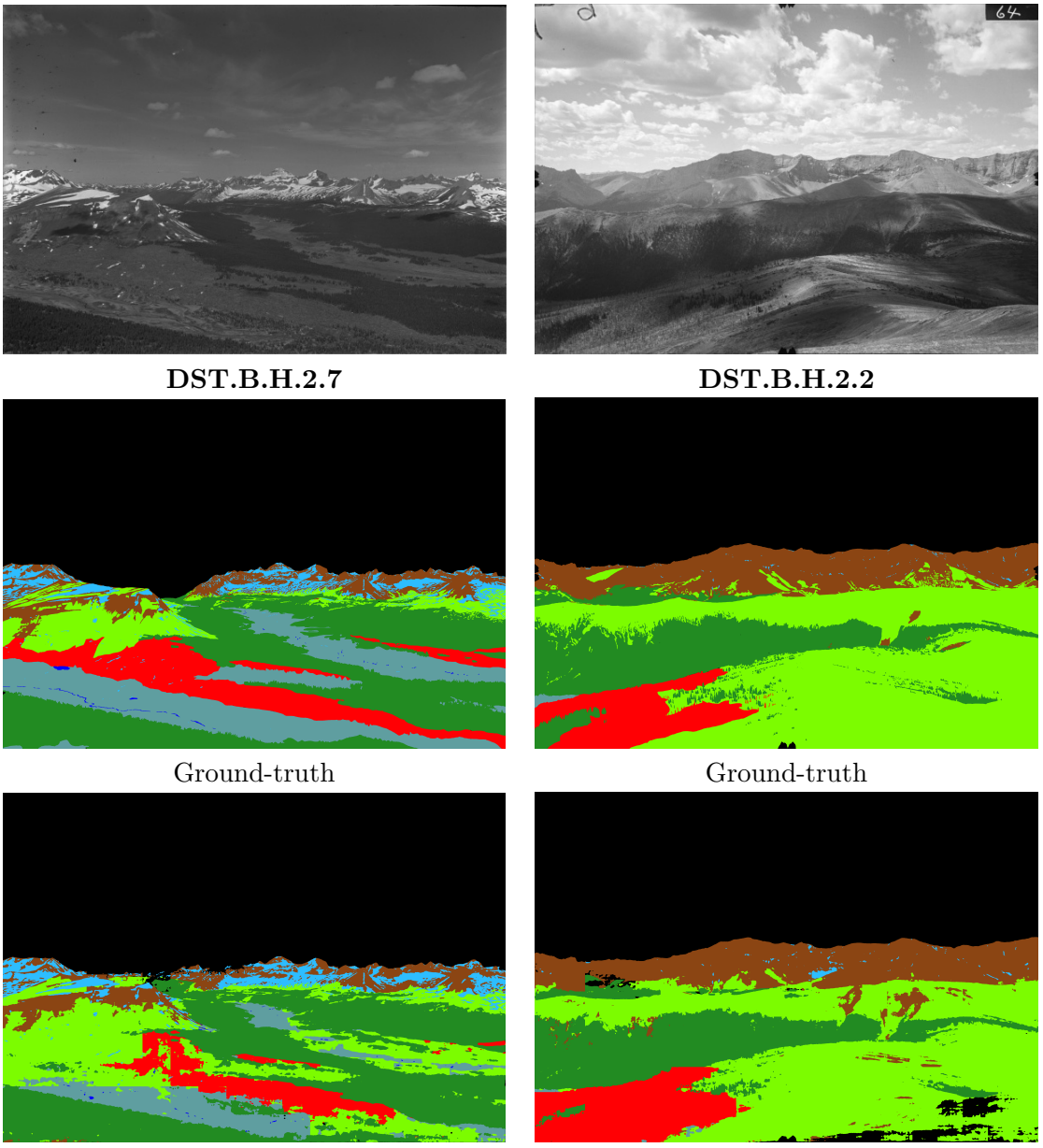


FIGURE C.4: DST.B.H.2.7: Relatively clear visibility. Some scratches and marks visible across the top left of the photo. [WT, WL, S-I, H-S, RA]. DST.B.H.2.2: Photo catalog markings visible on corners and along edges. Cloud shadows add significant photometric variation to forest cover. [WL, H-S, RA].

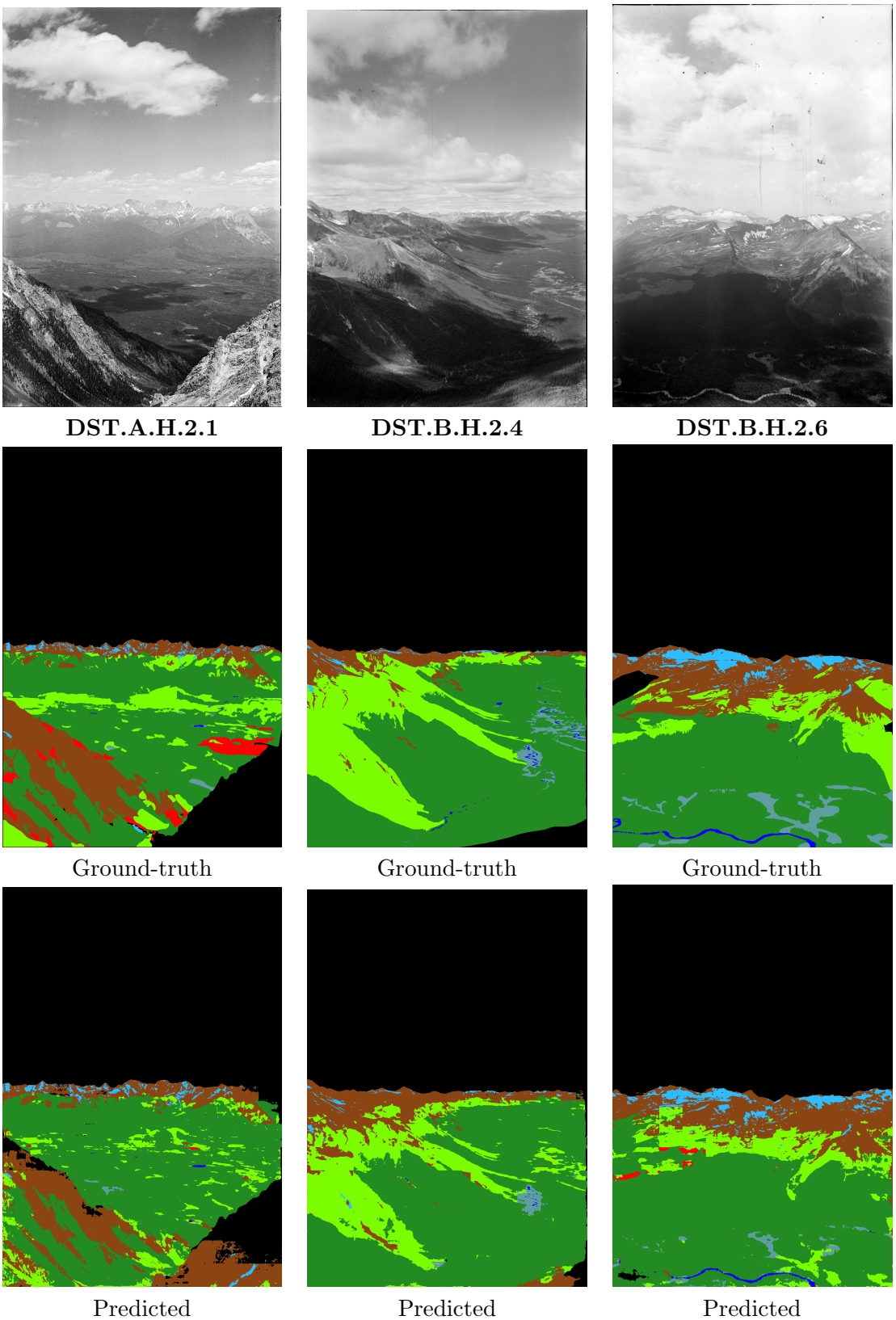


FIGURE C.5: DST.A.H.2.1. Image shows some visibility occlusion in distant background. Cloud shadows add photometric variation to forest cover. Bottom-right corner foreground pixels not classified. [S-I, RA]. Output segmentations shown from DLAB.H.2.5. DST.B.H.2.4: Cloud shadows add significant photometric variation to forest cover. Bottom-right corner foreground pixels not classified. [WL, WT, H-S, S-I]. DST.B.H.2.6: Scratches and marks visible across the middle of the photo. Some visibility occlusion due to cloudiness and rain, and due to emulsion defects. Middle-left region is not classified. [WL, WT, H-S, S-I].

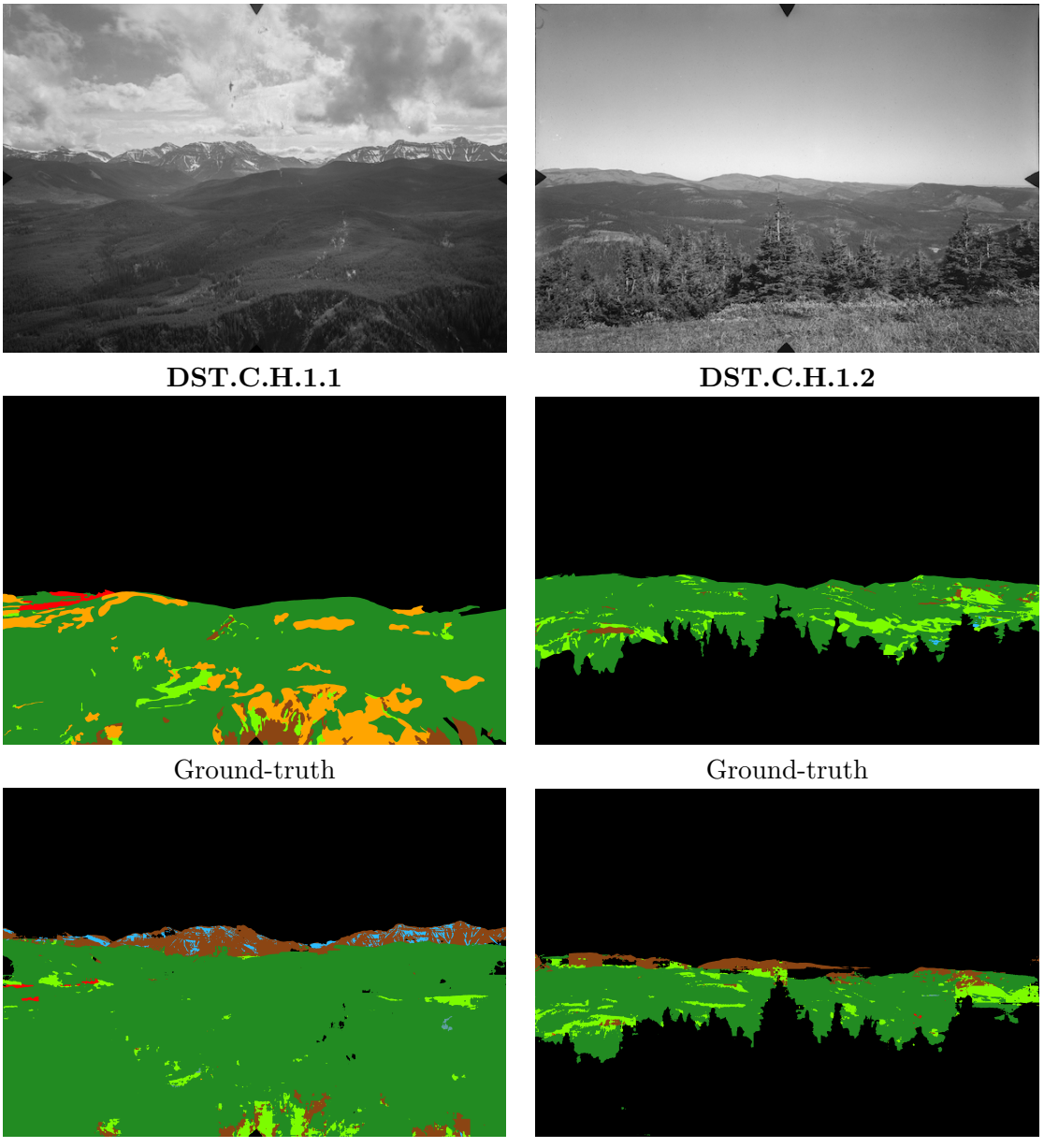


FIGURE C.6: Example historic model segmentation results form images DST.C.H.1.1, DST.C.H.1.2

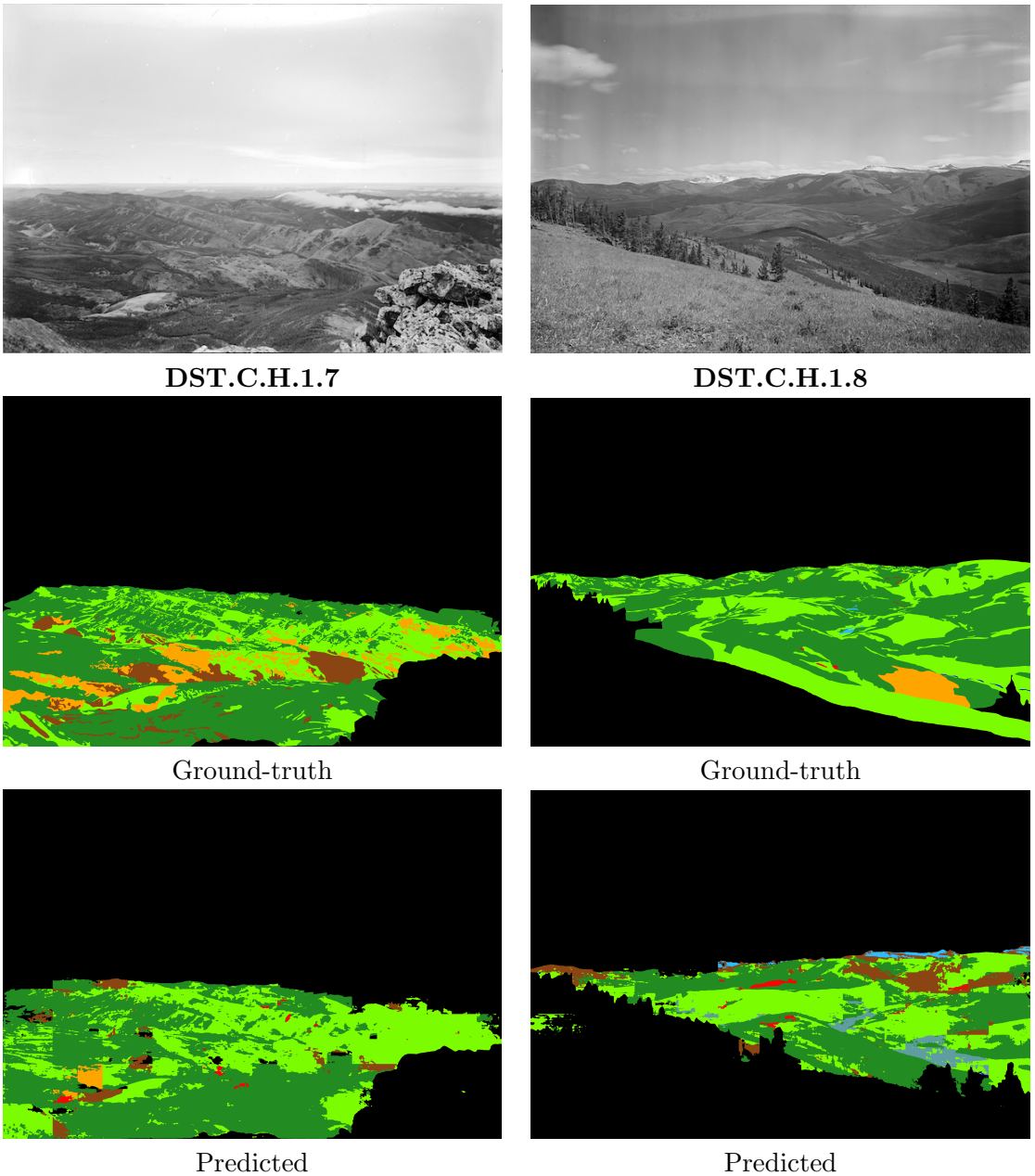
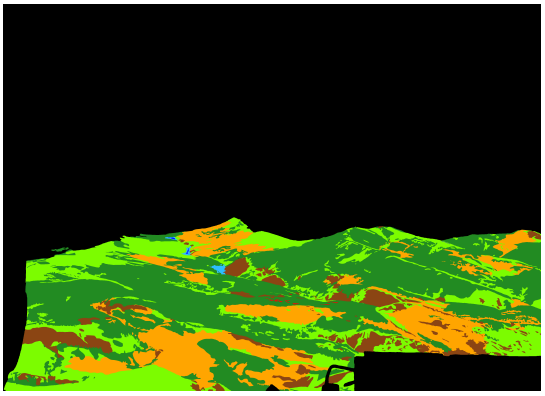


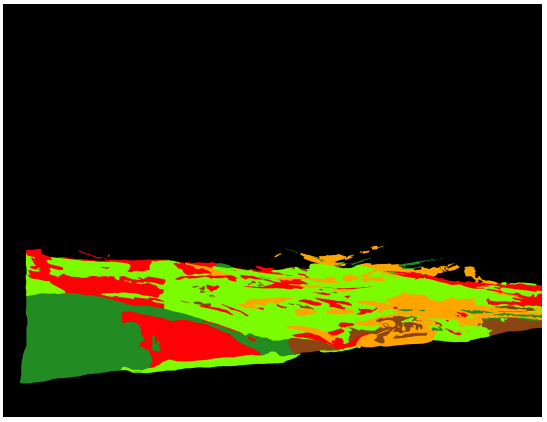
FIGURE C.7: Example historic model segmentation results form images DST.C.H.1.7, DST.C.H.1.8

DST.C.H.1.5

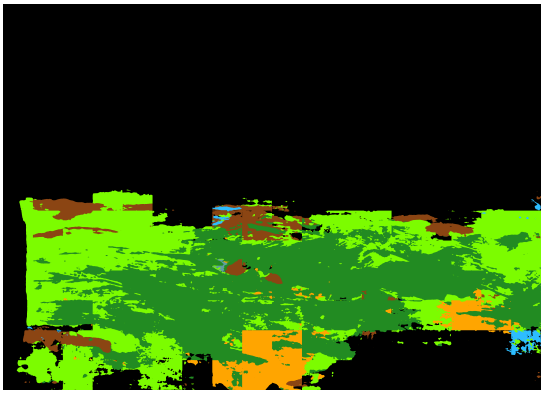
DST.C.H.1.6



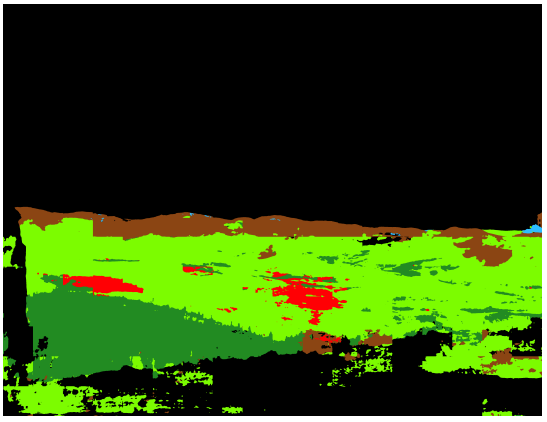
Ground-truth



Ground-truth

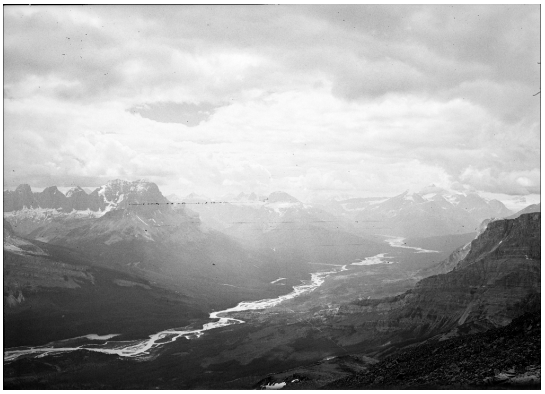


Predicted



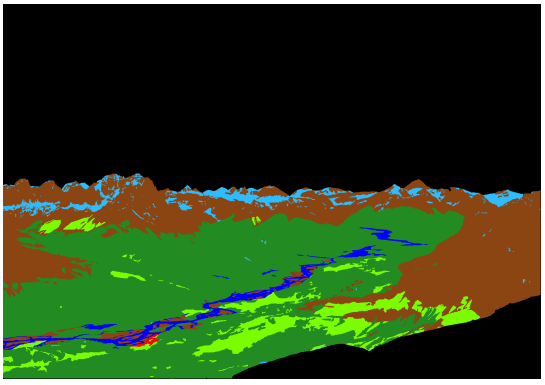
Predicted

FIGURE C.8: Example historic model segmentation results for images DST.C.H.1.5, DST.C.H.1.6

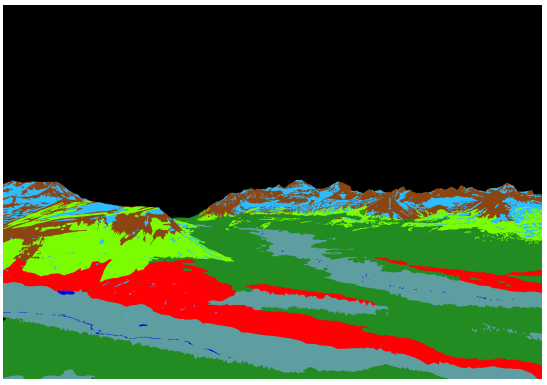


DST.A.H.2.2

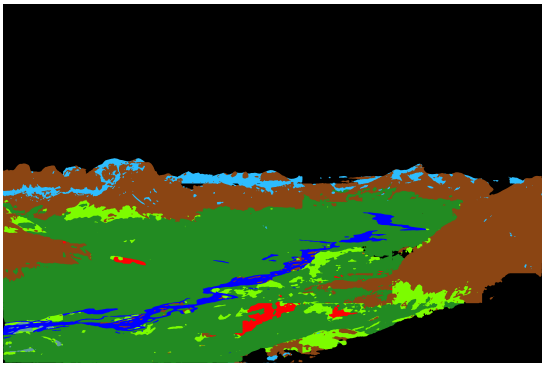
DST.B.H.2.7



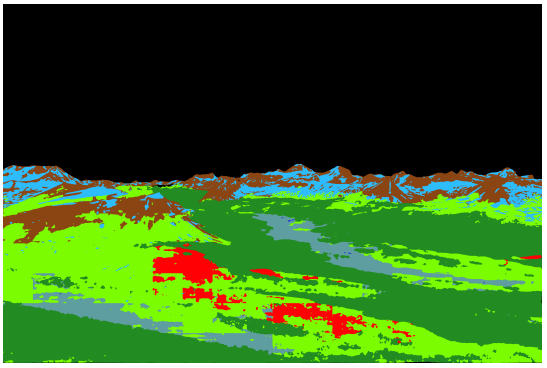
Ground-truth



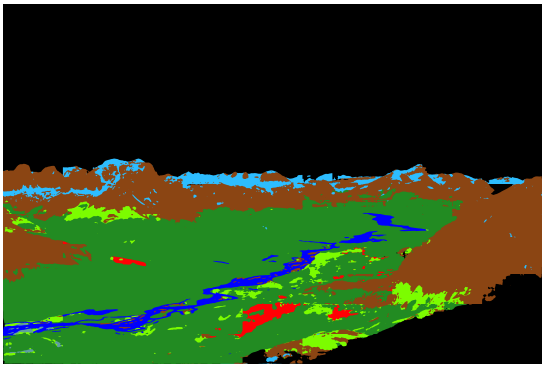
Ground-truth



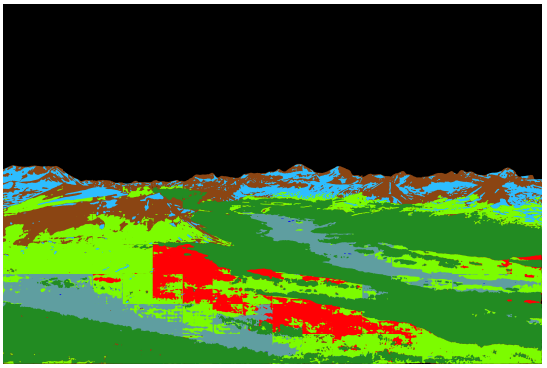
DCNN output



DCNN output



CRF output



CRF output

FIGURE C.9: Example results using conditional random fields (CRF) for post-processing images. DST.A.H.2.2 shows improved classification of distant mountains and correction of NC classified regions; DST.B.H.2.7 shows significant correction of foreground minor class RA (regenerating area) and WT (wetlands) regions.

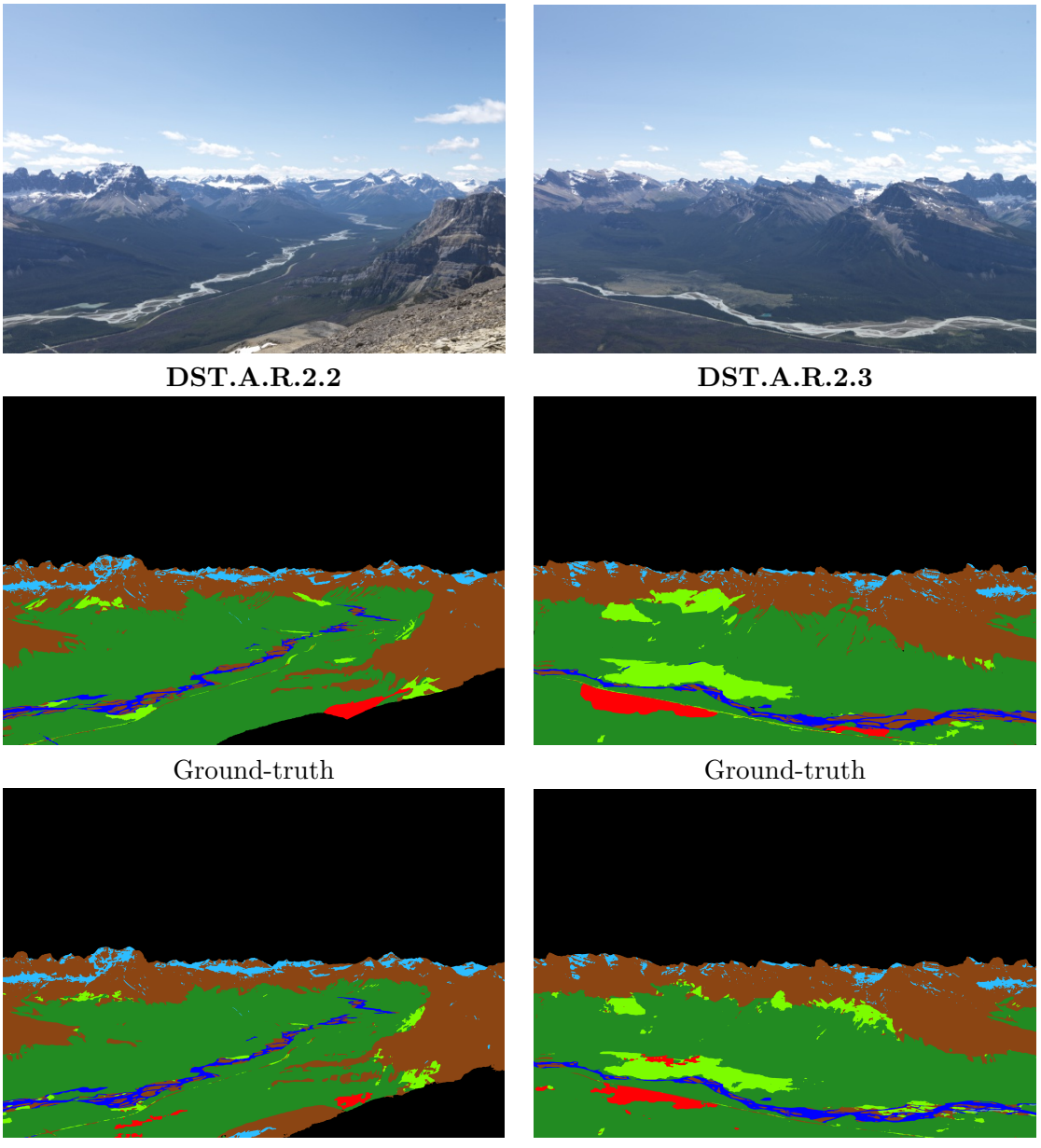


FIGURE C.10: Example repeat model segmentation results for images DST.A.R.2.2, DST.A.R.2.3

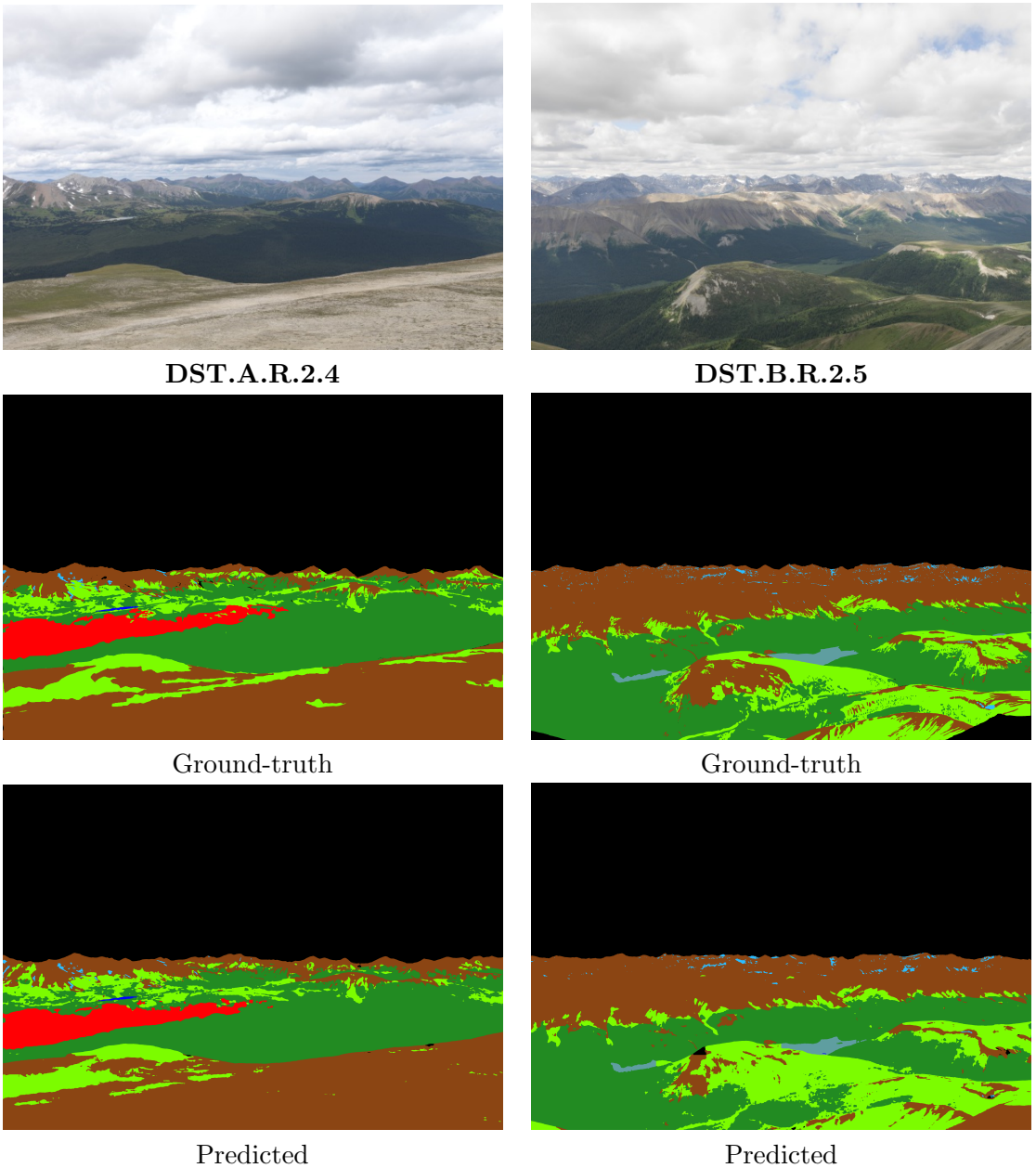
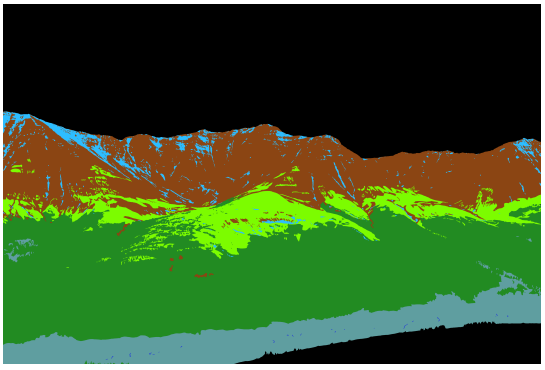


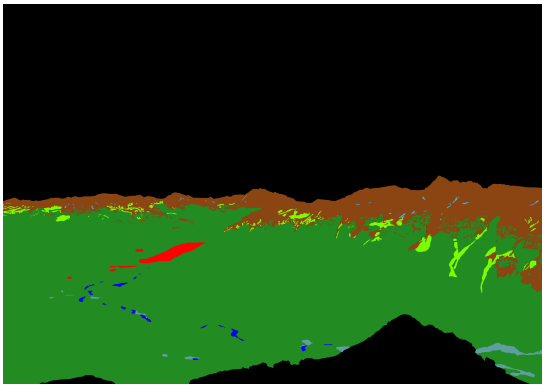
FIGURE C.11: Example repeat model segmentation results for images DST.A.R.2.4, DST.B.R.2.5

DST.B.R.2.3

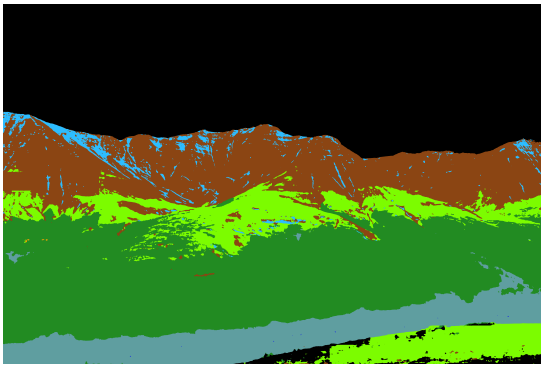
DST.B.R.2.5



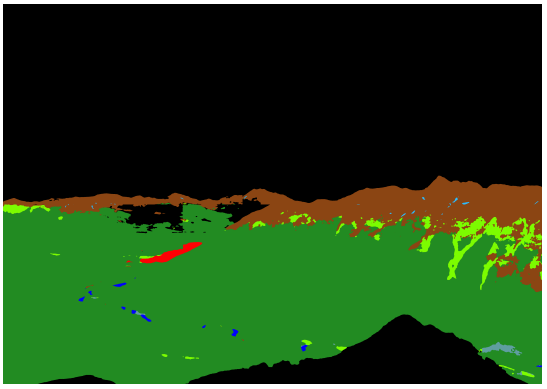
Ground-truth



Ground-truth



Predicted



Predicted

FIGURE C.12: Example repeat model segmentation results for images DST.B.R.2.3, DST.B.R.2.5

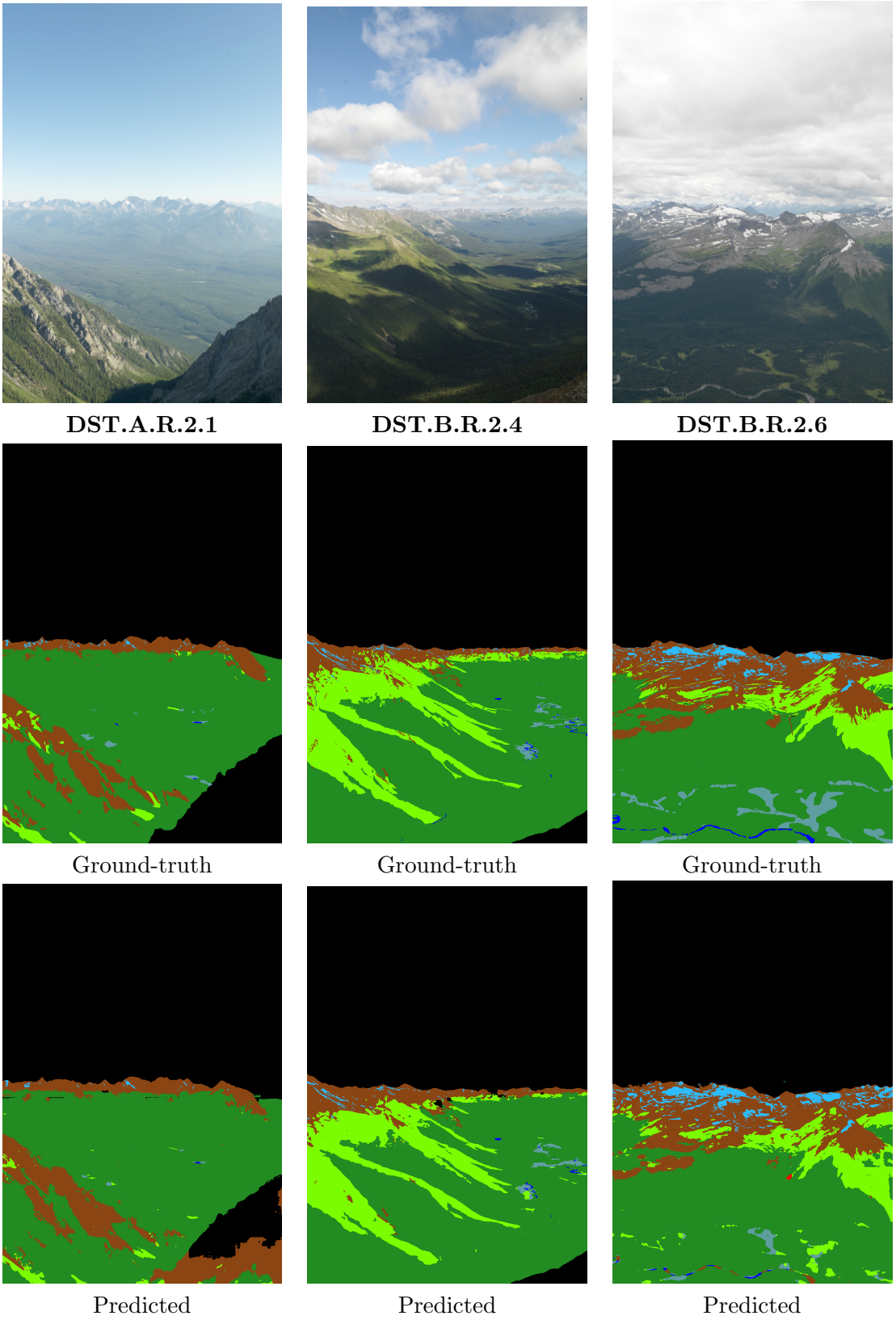


FIGURE C.13: Example repeat model segmentation results for images DST.A.R.2.1, DST.B.R.2.4, DST.B.R.2.6

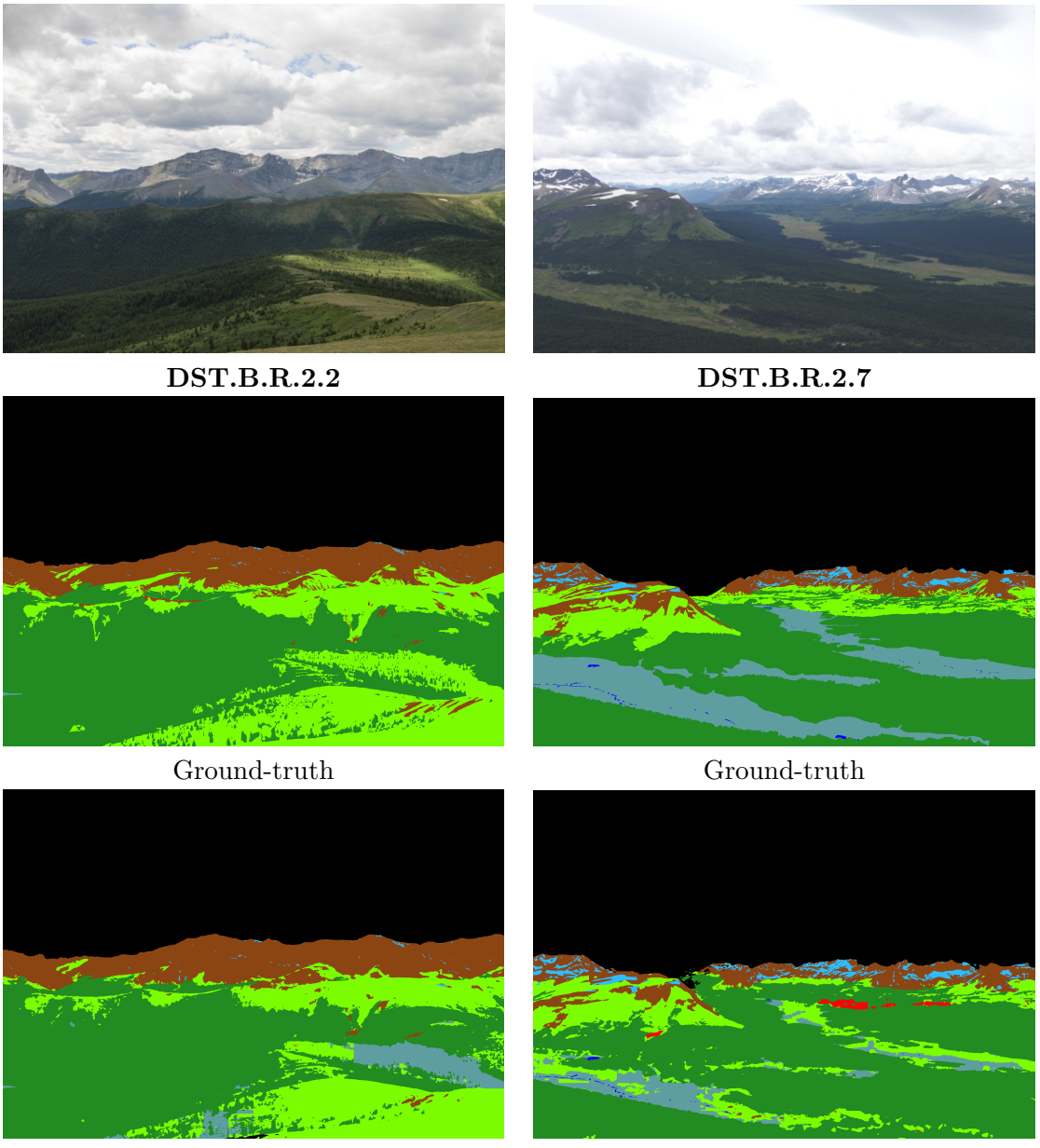


FIGURE C.14: Example repeat model segmentation results for images DST.B.R.2.2, DST.B.R.2.7

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351:234–241, 2015. ISSN 16113349. doi: 10.1007/978-3-319-24574-4_28. URL http://lmb.informatik.uni-freiburg.de/.

[2] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. ISSN 01628828. doi: 10.1109/TPAMI.2017.2699184.

[3] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. *30th International Conference on Machine Learning, ICML 2013*, 28(PART 2):1550–1558, 2013.

[4] Julie Fortin. *Lanscape and biodiversity change in the Willmore Wilderness Park through Repeat Photography*. PhD thesis, University of Victoria, 2015.

[5] Frédéric Jean, Alexandra Branzan Albu, David Capson, Eric Higgs, Jason T. Fisher, and Brian M. Starzomski. The mountain habitats segmentation and change detection dataset. *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, pages 603–609, 2015. doi: 10.1109/WACV.2015.86.

[6] Julie A. Fortin, Jason T. Fisher, Jeanine M. Rhemtulla, and Eric S. Higgs. Estimates of landscape composition from terrestrial oblique photographs suggest homogenization of Rocky Mountain landscapes over the last century. *Remote Sensing in Ecology and Conservation*, 5(3):224–236, 2019. ISSN 20563485. doi: 10.1002/rse2.100.

[7] Eric Higgs, Mary Ellen Sanseverino, Michael James Whitney, and Julie Fortin. Advances in Visual Applications: Visualizing & quantifying landscape change in

SW Alberta using Mountain Legacy Project photography Prepared. Technical report, University of Victoria, Victoria, 2020.

[8] GJ McDermid, RJ Hall, GA Sanchez-Azofeifa, SE Franklin, GB Stenhouse, T Kobliuk, and EF LeDrew. Remote sensing and forest inventory for wildlife habitat assessment. *Forest Ecology and Management*, 257(11):2262–2269, 2009.

[9] Eric Higgs, Donald A Falk, Anita Guerrini, Marcus Hall, Jim Harris, Richard J Hobbs, Stephen T Jackson, Jeanine M Rhemtulla, and William Throop. The changing role of history in restoration ecology. *Frontiers in Ecology and the Environment*, 12(9):499–506, 2014.

[10] Yan Boulanger, Anthony R Taylor, David T Price, Dominic Cyr, Elizabeth McGarrigle, Werner Rammer, Guillaume Sainte-Marie, André Beaudoin, Luc Guindon, and Nicolas Mansuy. Climate change impacts on forest landscapes along the canadian southern boreal forest transition zone. *Landscape Ecology*, 32(7):1415–1431, 2017.

[11] NIJOS and OECD. *Agricultural impacts on landscapes : Developing indicators for policy analysis*. 2002. ISBN 8274643089. URL http://www.skogoglandskap.no/filearchive/nettrapport07-08.pdf.

[12] Tanya Taggart-Hodge. *A century of landscape: level changes in the Bow watershed, Alberta, Canada, and implications for flood management*. PhD thesis, University of Victoria, 2016. URL https://dspace.library.uvic.ca//handle/1828/7655.

[13] W. Roush, J. S. Munroe, and D. B. Fagre. Development of a spatial analysis method using ground-based repeat photography to detect changes in the alpine treeline ecotone, Glacier National Park, Montana, U.S.A. *Arctic, Antarctic, and Alpine Research*, 39(2):297–308, 2007. ISSN 15230430. doi: 10.1657/1523-0430(2007)39[297:DOASAM]2.0.CO;2.

[14] Tommaso Julitta, Edoardo Cremonese, Mirco Migliavacca, Roberto Colombo, Marta Galvagno, Consolata Siniscalco, Micol Rossini, Francesco Fava, Sergio Cogliati, Umberto Morra di Cella, and Annette Menzel. Using digital camera images to analyse snowmelt and phenology of a subalpine grassland. *Agricultural and Forest Meteorology*, 198-199:116–125, 2014. ISSN 01681923. doi: 10.1016/j.agrformet.2014.08.007. URL http://dx.doi.org/10.1016/j.agrformet.2014.08.007.

[15] NP Gillett, AJ Weaver, FW Zwiers, and MD Flannigan. Detecting the effect of climate change on canadian forest fires. *Geophysical Research Letters*, 31(18), 2004.

[16] Brian H Aukema, Allan L Carroll, Jun Zhu, Kenneth F Raffa, Theodore A Sickley, and Stephen W Taylor. Landscape level analysis of mountain pine beetle in british columbia, canada: spatiotemporal development and spatial synchrony within the present outbreak. *Ecography*, 29(3):427–441, 2006.

[17] Ashbindu Singh. Review Articlel: Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10(6):989–1003, 1989. ISSN 13665901. doi: 10.1080/01431168908903939.

[18] Jeanine M. Rhemtulla, Ronald J. Hall, Eric S. Higgs, and S. Ellen Macdonald. Eighty years of change: Vegetation in the montane ecoregion of Jasper National Park, Alberta, Canada. *Canadian Journal of Forest Research*, 32(11):2010–2021, 2002. ISSN 00455067. doi: 10.1139/x02-112.

[19] Christopher A. Stockdale, Claudio Bozzini, S. Ellen Macdonald, and Eric Higgs. Extracting ecological information from oblique angle terrestrial landscape photographs: Performance evaluation of the WSL Monoplotting Tool. *Applied Geography*, 63:315–325, 2015. ISSN 01436228. doi: 10.1016/j.apgeog.2015.07.012. URL http://dx.doi.org/10.1016/j.apgeog.2015.07.012.

[20] Carmelo Riccardo Fichera, Giuseppe Modica, and Maurizio Pollino. Land cover classification and change-detection analysis using multi-temporal remote sensed imagery and landscape metrics. *European journal of remote sensing*, 45(1):1–18, 2012.

[21] Jordi Inglada, Arthur Vincent, Marcela Arias, Benjamin Tardy, David Morin, and Isabel Rodes. Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1):95, 2017.

[22] Mary Ann Cunningham. Accuracy assessment of digitized and classified land cover data for wildlife habitat. *Landscape and Urban Planning*, 78(3):217–228, 2006.

[23] Lazhar Khelifi and Max Mignotte. Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis. *arXiv preprint arXiv:2006.05612*, 2020.

[24] Lena Halounová. Textural classification of B&W aerial photos for the forest classification. *New Strategies for European Remote Sensing*, 10:1–7, 2005. URL http://www.earsel.org/symposia/2003-symposium-Ghent/pdf/C04.pdf.

[25] Emilio Guirado, Siham Tabik, Domingo Alcaraz-Segura, Javier Cabello, and Francisco Herrera. Deep-learning Versus OBIA for scattered shrub detection with Google Earth Imagery: Ziziphus lotus as case study. *Remote Sensing*, 9(12):1–22, 2017. ISSN 20724292. doi: 10.3390/rs9121220.

[26] Philip G. Brodrick, Andrew B. Davies, and Gregory P. Asner. Uncovering Ecological Patterns with Convolutional Neural Networks. *Trends in Ecology and Evolution*, 34(8):734–745, 2019. ISSN 01695347. doi: 10.1016/j.tree.2019.03.006. URL https://doi.org/10.1016/j.tree.2019.03.006.

[27] C. M.R. Caridade, A. R.S. Marçal, and T. Mendonça. The use of texture for image classification of black and white air photographs. *International Journal of Remote Sensing*, 29(2):593–607, 2008. ISSN 13665901. doi: 10.1080/01431160701281015.

[28] Robert H Webb. *Repeat photography: methods and applications in the natural sciences*. Island Press, 2010.

[29] 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.

[30] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie, and Amy Loutfi. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, 8(4), 2016. ISSN 20724292. doi: 10.3390/rs8040329.

[31] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE transactions on geoscience and remote sensing*, 56(5):2811–2821, 2018.

[32] 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. 2018. URL http://arxiv.org/abs/1805.12219.

[33] Tayeb Alipourfard, Hossein Arefi, and Somayeh Mahmoudi. A novel deep learning framework by combination of subspace-based feature extraction and convolutional neural networks for hyperspectral images classification. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018-July:4780–4783, 2018. doi: 10.1109/IGARSS.2018.8518956.

[34] Daniel Buscombe and Andrew C. Ritchie. Landscape classification with deep neural networks. *Geosciences (Switzerland)*, 8(7):1–23, 2018. ISSN 20763263. doi: 10.3390/geosciences8070244.

[35] Wang Li, Robert Buitenwerf, Michael Munk, Peder Klith Bøcher, and Jens-Christian Svenning. Deep-learning based high-resolution mapping shows woody vegetation densification in greater maasai mara ecosystem. *Remote Sensing of Environment*, 247:111953, 2020.

[36] Benjamin Lucas, Charlotte Pelletier, Daniel Schmidt, Geoffrey I Webb, and François Petitjean. A bayesian-inspired, deep learning, semi-supervised domain adaptation technique for land cover mapping. *arXiv preprint arXiv:2005.11930*, 2020.

[37] Andrew J Trant, Brian M Starzomski, and Eric Higgs. A publically available database for studying ecological change in mountain ecosystems. *Frontiers in Ecology and the Environment*, 13(4):187–187, 2015.

[38] Mary Ellen Sanseverino, Michael James Whitney, and Eric Stowe Higgs. Exploring Landscape Change in Mountain Environments With the Mountain Legacy Online Image Analysis Toolkit. *Mountain Research and Development*, 36(4):407–416, nov 2016. ISSN 0276-4741. doi: 10.1659/mrd-journal-d-16-00038.1. URL http://www.bioone.org/doi/10.1659/MRD-JOURNAL-D-16-00038.1.

[39] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with Gaussian edge potentials. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, pages 1–9, 2011.

[40] Andreas H Schweiger, Isabelle Boulangeat, Timo Conradi, Matt Davis, and Jens-Christian Svenning. The importance of ecological memory for trophic rewilding as an ecosystem restoration approach. *Biological Reviews*, 94(1):1–15, 2019.

[41] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[42] Iva Harbaš, Pavle Prentašić, and Marko Subašić. Detection of roadside vegetation using Fully Convolutional Networks. *Image and Vision Computing*, 74:1–9, 2018. ISSN 02628856. doi: 10.1016/j.imavis.2018.03.008.

[43] Conedera Bozzini, Marco Conedera, and Patrik Krebs. A new monoplotting tool to extract georeferenced vector data and orthorectified raster data from oblique non-metric photographs. *international Journal of Heritage in the Digital era*, 1 (3):499–518, 2012.

[44] Natalia Kolecka, Jacek Kozak, Dominik Kaim, Monika Dobosz, Christian Ginzler, and Achilleas Psomas. Mapping secondary forest succession on abandoned agricultural land with LiDAR point clouds and terrestrial photography. *Remote Sensing*, 7(7):8300–8322, 2015. ISSN 20724292. doi: 10.3390/rs70708300.

[45] Joanne C White and Michael A Wulder. The landsat observation record of canada: 1972–2012. *Canadian Journal of Remote Sensing*, 39(6):455–467, 2014.

[46] Antoine Lefebvre, Thomas Corpetti, and Laurence Hubert-Moy. Object-oriented approach and texture analysis for change detection in very high resolution images. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 4(1):IV – 663–IV – 666, 2008. doi: 10.1109/IGARSS.2008.4779809.

[47] Ulrike Bayr and Oskar Puschmann. Automatic detection of woody vegetation in repeat landscape photographs using a convolutional neural network. *Ecological Informatics*, 50(December 2018):220–233, 2019. ISSN 15749541. doi: 10.1016/j.ecoinf.2019.01.012. URL https://doi.org/10.1016/j.ecoinf.2019.01.012.

[48] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:1520–1528, 2015. ISSN 15505499. doi: 10.1109/ICCV.2015.178.

[49] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. ParseNet: Looking Wider to See Better. 2015. URL http://arxiv.org/abs/1506.04579.

[50] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, Alexander Dovzhenko, Olaf Tietz, Cristina Dal Bosco, Sean Walsh, Deniz Saltukoglu, Tuan Leng Tay, Marco Prinz, Klaus Palme, Matias Simons, Ilka Diester, Thomas Brox, and Olaf Ronneberger. U-Net: deep learning for cell counting, detection, and morphometry. *Nature Methods*, 16(1):67–70, 2019. ISSN 15487105. doi: 10.1038/s41592-018-0261-2.

[51] 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.

[52] É. Deville. Photographic surveying: including the elements of descriptive geometry and perspective. Technical report, Government Printing Bureau, Canada, 1895.

[53] MP Bridgland. Photographic surveying in canada. *Geographical Review*, 2(1): 19–26, 1916.

[54] M. P. Bridgland. Photographic Surveying, 1924.

[55] Christopher Gat. Feature-based matching in historic repeat photography: an evaluation and assessment of feasibility. Master's thesis, Rijksuniversiteit Groningen, Victoria, British Columbia, Canada, 2011.

[56] Frédéric Jean, Alexandra Branzan Albu, David Capson, Eric Higgs, Jason T. Fisher, and Brian M. Starzomski. The Mountain Habitats Segmentation and

Change Detection Dataset. November 2014. doi: 10.5281/zenodo.12590. URL https://doi.org/10.5281/zenodo.12590.

[57] Hongyuan Zhu, Fanman Meng, Jianfei Cai, and Shijian Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, 2016. ISSN 10959076. doi: 10.1016/j.jvcir.2015.10.012.

[58] Masroor Hussain, Dongmei Chen, Angela Cheng, Hui Wei, and David Stanley. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:91–106, 2013. ISSN 09242716. doi: 10.1016/j.isprsjprs.2013.03.006. URL http://dx.doi.org/10.1016/j.isprsjprs.2013.03.006.

[59] Shichao Yang, Daniel Maturana, and Sebastian Scherer. Real-time 3d scene layout from a single image using convolutional neural networks. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2183–2189. IEEE, 2016.

[60] Xiao Yun Zhou and Guang Zhong Yang. Normalization in training U-Net for 2-D biomedical semantic segmentation. *IEEE Robotics and Automation Letters*, 4(2): 1792–1799, 2019. ISSN 23773766. doi: 10.1109/LRA.2019.2896518.

[61] Nabil Ibtehaz and M. Sohel Rahman. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Networks*, 121: 74–87, 2020. ISSN 18792782. doi: 10.1016/j.neunet.2019.08.025.

[62] Saturnino Maldonado-Bascón, Sergio Lafuente-Arroyo, Pedro Gil-Jimenez, Hilario Gómez-Moreno, and Francisco López-Ferreras. Road-sign detection and recognition based on support vector machines. *IEEE transactions on intelligent transportation systems*, 8(2):264–278, 2007.

[63] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In *Advances in neural information processing systems*, pages 424–432, 2014.

[64] Michael Treml, José Arjona-medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, Bernhard Nessler, and Sepp Hochreiter. Speeding up Semantic Segmentation for Autonomous Driving. *NIPS 2016 workshop MLITS*, (Nips): 1–7, 2016. URL https://openreview.net/pdf?id=S1uHiFyyg{%}0Ahttps://openreview.net/forum?id=S1uHiFyyg.

[65] Suvash Sharma, John E. Ball, Bo Tang, Daniel W. Carruth, Matthew Doude, and Muhammad Aminul Islam. Semantic segmentation with transfer learning for off-road autonomous driving. *Sensors (Switzerland)*, 19(11):1–21, 2019. ISSN 14248220. doi: 10.3390/s19112577.

[66] C. Huang, L. S. Davis, and J. R.G. Townshend. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4):725–749, 2002. ISSN 01431161. doi: 10.1080/01431160110040323.

[67] Timothy G. Whiteside, Guy S. Boggs, and Stefan W. Maier. Comparing object-based and pixel-based classifications for mapping savannas. *International Journal of Applied Earth Observation and Geoinformation*, 13(6):884–893, 2011. ISSN 15698432. doi: 10.1016/j.jag.2011.06.008. URL http://dx.doi.org/10.1016/j.jag.2011.06.008.

[68] Martin Thoma. A Survey of Semantic Segmentation. pages 1–16, 2016. URL http://arxiv.org/abs/1602.06541.

[69] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5302 LNCS(PART 1): 30–43, 2008. ISSN 03029743. doi: 10.1007/978-3-540-88682-2-4.

[70] Samarth Brahmbhatt, Henrik I. Christensen, and James Hays. StuffNet: Using 'Stuff' to improve object detection. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, (Figure 1):934–943, 2017. doi: 10.1109/WACV.2017.109.

[71] David A. Forsyth, Jitendra Malik, Margaret M. Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding pictures of objects in large collections of images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1144:335–360, 1996. ISSN 16113349. doi: 10.1007/3-540-61750-7_36.

[72] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012. ISSN 01628828. doi: 10.1109/TPAMI.2012.28.

[73] Carolyn Burnett and Thomas Blaschke. A multi-scale segmentation/object relationship modelling methodology for landscape analysis. *Ecological modelling*, 168 (3):233–249, 2003.

[74] Andrew Rabinovich and Alexander C Berg. ParseNet: Looking wider to see better. pages 1–11, 2016.

[75] Hossein Mobahi, Shankar R Rao, Allen Y Yang, Shankar S Sastry, and Yi Ma. Segmentation of natural images by texture and boundary compression. *International journal of computer vision*, 95(1):86–98, 2011.

[76] Robert K McConnell. Method of and apparatus for pattern recognition, January 28 1986. US Patent 4,567,610.

[77] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *European Conference on Computer Vision*, pages 404–420. Springer, 2000.

[78] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 09205691. doi: 10.1023/B:VISI.0000029664.99615.94.

[79] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[80] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[81] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, 1 (c):10–17, 2003. doi: 10.1109/iccv.2003.1238308.

[82] Simona Caraiman and Vasile I Manta. Histogram-based segmentation of quantum images. *Theoretical Computer Science*, 529:46–60, 2014.

[83] Jun Zhang, Jimin Liang, and Heng Zhao. Local energy pattern for texture classification using self-adaptive quantization thresholds. *IEEE Transactions on Image Processing*, 22(1):31–42, 2013. ISSN 10577149. doi: 10.1109/TIP.2012.2214045.

[84] James H Elder and Steven W Zucker. Computing contour closure. In *European conference on computer vision*, pages 399–412. Springer, 1996.

[85] Theo Gevers and Arnold WM Smeulders. Color based object recognition. In *International Conference on Image Analysis and Processing*, pages 319–326. Springer, 1997.

[86] Shiping Zhu, Xi Xia, Qingrong Zhang, and Kamel Belloulata. An image segmentation algorithm in image processing based on threshold segmentation. In *2007 third international IEEE conference on signal-image technologies and internet-based system*, pages 673–678. IEEE, 2007.

[87] Rishi R Rakesh, Probal Chaudhuri, and CA Murthy. Thresholding in edge detection: a statistical approach. *IEEE Transactions on Image Processing*, 13(7): 927–936, 2004.

[88] Alan J Danker and Azriel Rosenfeld. Blob detection by relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):79–92, 1981.

[89] Olivier Monga. An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(03n04): 351–375, 1987.

[90] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[91] Bela Julesz. Visual Pattern Discrimination. *IRE Transactions on Information Theory*, 8(2):84–92, 1962. ISSN 21682712. doi: 10.1109/TIT.1962.1057698.

[92] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From BoW to CNN: Two Decades of Texture Representation for Texture Classification. *International Journal of Computer Vision*, 127(1):74–109, 2019. ISSN 15731405. doi: 10.1007/s11263-018-1125-z. URL https://doi.org/10.1007/s11263-018-1125-z.

[93] Laurens Van Der Maaten and Eric Postma. Texton-based texture classification. *Belgian/Netherlands Artificial Intelligence Conference*, pages 213–220, 2007. ISSN 15687805.

[94] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep Filter Banks for Texture Recognition, Description, and Segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016. ISSN 15731405. doi: 10.1007/s11263-015-0872-3.

[95] Bidyut Baran Chaudhuri and Nirupam Sarkar. Texture segmentation using fractal dimension. *IEEE Transactions on pattern analysis and machine intelligence*, 17 (1):72–77, 1995.

[96] Xiuwen Liu and De Liang Wang. Image and texture segmentation using local spectral histograms. *IEEE Transactions on Image Processing*, 15(10):3066–3077, 2006. ISSN 10577149. doi: 10.1109/TIP.2006.877511.

[97] Jiangye Yuan, Deliang Wang, and Anil M. Cheriyadat. Factorization-Based Texture Segmentation. *IEEE Transactions on Image Processing*, 24(11):3488–3497, 2015. ISSN 10577149. doi: 10.1109/TIP.2015.2446948.

[98] Dennis Dunn and William E Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on image processing*, 4(7):947–964, 1995.

[99] Yang Chen and Runsheng Wang. Texture segmentation using independent component analysis of gabor features. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 147–150. IEEE, 2006.

[100] Mihran Tuceryan and Anil K. Jain. Texture analysis. *Pattern Recognition and Computer Vision*, pages 207–248, 1998. ISSN 1470-7330. doi: 10.1102/1470-7330.2010.0021. URL http://www.ncbi.nlm.nih.gov/pubmed/20667507.

[101] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pages 1–15. Springer, 2006.

[102] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[103] Tong Liu, Xiutian Huang, and Jianshe Ma. Conditional Random Fields for Image Labeling. *Mathematical Problems in Engineering*, 2016, 2016. ISSN 15635147. doi: 10.1155/2016/3846125.

[104] Anurag Arnab and Philip H.S. Torr. Bottom-up instance segmentation using deep higher-order CRFs. *British Machine Vision Conference 2016, BMVC 2016*, 2016-Septe:9.1–9.12, 2016. doi: 10.5244/C.30.19.

[105] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. ISSN 01628828. doi: 10.1109/TPAMI.2017.2699184.

[106] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018.

[107] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-paredes, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip Torr. Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation. *Cvpr*, XX(Xx):1–15, 2018. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.308.8889{&}rep=rep1{&}type=pdf{%}0Ahttp://dx.doi.org/10.1109/CVPR.2012.6248050.

[108] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[109] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object class segmentation using random forests. In *BMVC*, pages 1–10, 2008.

[110] John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[111] James P Theiler and Galen Gisler. Contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation. In *Algorithms, devices, and systems for optical information processing*, volume 3159, pages 108–118. International Society for Optics and Photonics, 1997.

[112] Yuan-Hui Yu and Chin-Chen Chang. Scenery image segmentation using support vector machines. *Fundamenta Informaticae*, 61(3-4):379–388, 2004.

[113] Phil Brodatz. *Textures: a photographic album for artists and designers, by Phil Brodatz*. Dover publications, 1966.

[114] KRISTIN J Dana, B van Ginneken, SK Nayar, and JJ Koenderink. Curet: Columbia utrecht reflectance and texture database, 1999.

[115] Markus Möller, Leo Lymburner, and Martin Volk. The comparison index: A tool for assessing the accuracy of image segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 9(3):311–321, 2007.

[116] T. Blaschke. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):2–16, 2010. ISSN 09242716. doi: 10.1016/j.isprsjprs.2009.06.004. URL http://dx.doi.org/10.1016/j.isprsjprs.2009.06.004.

[117] Xiangping Sun. *Robust texture classification based on machine*. PhD thesis, Deakin University, 2014.

[118] David Grangier, Léon Bottou, and Ronan Collobert. Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3, page 109. Citeseer, 2009.

[119] Youngeun Kim, Seunghyeon Kim, Taekyung Kim, and Changick Kim. CNN-based semantic segmentation using level set loss. *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pages 1752–1760, 2019. doi: 10.1109/WACV.2019.00191.

[120] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015. ISSN 14764687. doi: 10.1038/nature14539.

[121] Xiaolong Liu, Zhidong Deng, and Yuhan Yang. Recent progress in semantic image segmentation. *Artificial Intelligence Review*, 52(2):1089–1106, 2019. ISSN 15737462. doi: 10.1007/s10462-018-9641-3. URL https://doi.org/10.1007/s10462-018-9641-3.

[122] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[123] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. ISSN 03401200. doi: 10.1007/BF00344251.

[124] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968. ISSN 14697793. doi: 10.1113/jphysiol.1968.sp008455. URL http://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1968.sp008455/abstract?rss=1.

[125] Yann N. Dauphin, Harm De Vries, and Yoshua Bengio. Equilibrated adaptive learning rates for non-convex optimization. *Advances in Neural Information Processing Systems*, 2015-January:1504–1512, 2015. ISSN 10495258.

[126] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[127] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[128] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[129] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. ISSN 1530888X. doi: 10.1162/NECO_a_00990.

[130] Chih-Cheng Hung, Enmin Song, and Yihua Lan. *Image Texture Analysis*. 2019. ISBN 9783030137724. doi: 10.1007/978-3-030-13773-1.

[131] Mircea Cimpoi. *Recognizing Describable Attributes and Materials of Textures in the Wild and Clutter Supervisor : Recognizing Describable Attributes and Materials of Textures in the Wild and Clutter*. PhD thesis, University of Oxford, 2015.

[132] Clement Farabet, Camille Couprie, Laurent Najman, and Yann Lecun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2012.231.

[133] Pedro O. Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. *31st International Conference on Machine Learning, ICML 2014*, 1:151–159, 2014.

[134] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[135] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:580–587, 2014. ISSN 10636919. doi: 10.1109/CVPR.2014.81.

[136] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[137] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2014.

[138] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[139] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[140] 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.

[141] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[142] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[143] Vincent Vanhoucke. Learning visual representations at scale. *ICLR invited talk*, 1:2, 2014.

[144] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[145] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[146] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.

[147] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. Object Instance Segmentation and Fine-Grained Localization Using Hypercolumns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):627–639, 2017. ISSN 01628828. doi: 10.1109/TPAMI.2016.2578328.

[148] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[149] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[150] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[151] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[152] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6230–6239, 2017. doi: 10.1109/CVPR.2017.660.

[153] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, 1:448–456, 2015.

[154] Marvin T. T. Teichmann and Roberto Cipolla. Convolutional CRFs for Semantic Segmentation. 2018. URL http://arxiv.org/abs/1805.04777.

[155] Shuai Zhao, Boxi Wu, Wenqing Chu, Yao Hu, and Deng Cai. Correlation Maximized Structural Similarity Loss for Semantic Segmentation. 2019. URL http://arxiv.org/abs/1910.08711.

[156] S Zheng, S Jayasumana, B Romera-Paredes, V Vineet, Z Su, D Du, C Huang, and PHS Torr. Conditional random fields as recurrent neural networks. arxiv 2015. *arXiv preprint arXiv:1502.03240*.

[157] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[158] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. ISSN 18792782. doi: 10.1016/j.neunet.2018.07.011.

[159] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong Tian. Selectivity or Invariance: Boundary-aware Salient Object Detection. 2018. URL http://arxiv.org/abs/1812.10066.

[160] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

[161] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 2014. doi: 10.5244/c.28.6.

[162] Zhengeng Yang, Hongshan Yu, Qiang Fu, Wei Sun, Wenyan Jia, Mingui Sun, and Zhi-Hong Mao. Real time backbone for semantic segmentation. 14(8):1–6, 2019. URL http://arxiv.org/abs/1903.06922.

[163] Shuangting Liu, Jiaqi Zhang, Yuxin Chen, Yifan Liu, Zengchang Qin, and Tao Wan. Pixel Level Data Augmentation for Semantic Image Segmentation Using Generative Adversarial Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019-May:1902–1906, 2019. ISSN 15206149. doi: 10.1109/ICASSP.2019.8683590.

[164] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7455–7463, 2017.

[165] Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Overfitting of Neural Nets Under Class Imbalance: Analysis and Improvements for Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11766 LNCS:402–410, 2019. ISSN 16113349. doi: 10.1007/978-3-030-32248-9_45.

[166] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN 10769757. doi: 10.1613/jair.953. URL https://arxiv.org/pdf/1106.1813.pdf{%}0Ahttp://www.snopes.com/horrors/insects/telamonia.asp.

[167] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[168] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–578, 2018.

[169] Yin Cui, Menglin Jia, Tsung Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:9260–9269, 2019. ISSN 10636919. doi: 10.1109/CVPR.2019.00949.

[170] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Application of Decision Rules for Handling Class Imbalance in Semantic Segmentation. 2019. URL http://arxiv.org/abs/1901.08394.

[171] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. pages 1–10, 2016. URL http://arxiv.org/abs/1606.02147.

[172] Fausto Milletari, Nassir Navab, and Seyed Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *Proceedings* -

2016 4th International Conference on 3D Vision, 3DV 2016, pages 565–571, 2016. doi: 10.1109/3DV.2016.79.

[173] 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.

[174] 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.

[175] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[176] Fan Jia, Jun Liu, and Xue-Cheng Tai. A regularized convolutional neural network for semantic image segmentation. *Analysis and Applications*, pages 1–19, 2020. ISSN 0219-5305. doi: 10.1142/s0219530519410148.

[177] Boyuan Ma, Xiaojuan Ban, Haiyou Huang, Yulian Chen, Wanbo Liu, and Yonghong Zhi. Deep learning-based image segmentation for Al-La alloy microscopic images. *Symmetry*, 10(4):1–13, 2018. ISSN 20738994. doi: 10.3390/sym10040107.

[178] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014. ISSN 1932-8346. doi: 10.1561/2000000039. URL http://dx.doi.org/10.1561/2000000039.

[179] Siham Tabik, Daniel Peralta, Andrés Herrera-Poyatos, and Francisco Herrera. A snapshot of image pre-processing for convolutional neural networks: case study of mnist. *International Journal of Computational Intelligence Systems*, 10(1):555–568, 2017.

[180] Daniel Peralta, Isaac Triguero, Salvador García, Yvan Saeys, Jose M Benitez, and Francisco Herrera. On the use of convolutional neural networks for robust classification of multiple fingerprint captures. *International Journal of Intelligent Systems*, 33(1):213–230, 2018.

[181] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[182] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. ISSN 10414347. doi: 10.1109/TKDE.2008.239.

[183] Jack P Gibbs and Dudley L Poston Jr. The division of labor: Conceptualization and related measures. *Social Forces*, 53(3):468–476, 1975.

[184] Alexey A. Novikov, Dimitrios Lenis, David Major, Jiri Hladuvka, Maria Wimmer, and Katja Buhler. Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. *IEEE Transactions on Medical Imaging*, 37(8):1865–1876, 2018. ISSN 1558254X. doi: 10.1109/TMI.2018.2806086.

[185] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[186] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. ExFuse: Enhancing feature fusion for semantic segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11214 LNCS:273–288, 2018. ISSN 16113349. doi: 10.1007/978-3-030-01249-6_17.

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

[188] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

[189] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:21–37, 2016. ISSN 16113349. doi: 10.1007/978-3-319-46448-0_2.

[190] Galen Andrew and Jianfeng Gao. Scalable training of l1-regularized log-linear models. In *International Conference on Machine Learning*, January 2007. URL https://www.microsoft.com/en-us/research/publication/scalable-training-of-l1-regularized-log-linear-models/.

[191] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[192] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Karthik Sridharan, Brando Miranda, Noah Golowich, and Tomaso Poggio. Musings on Deep Learning: Properties of SGD. *CBMM Memo*, (067), 2017. URL https://dspace.mit.edu/bitstream/handle/1721.1/107841/CBMM-Memo-067.pdf?sequence=1.

[193] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[194] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[195] Karen A Harper, S Ellen Macdonald, Philip J Burton, Jiquan Chen, Kimberley D Brosofske, Sari C Saunders, Eugenie S Euskirchen, DAR Roberts, Malanding S Jaiteh, and Per-Anders Esseen. Edge influence on forest structure and composition in fragmented landscapes. *Conservation biology*, 19(3):768–782, 2005.

[196] Anna Pyataeva. Dynamic texture recognition under adverse lighting and weather conditions for outdoor environments. *E3S Web of Conferences*, 75, 2019. ISSN 22671242. doi: 10.1051/e3sconf/20197501008.

[197] Søren Skovsen, Mads Dyrmann, Anders Krogh Mortensen, Kim Arild Steen, Ole Green, Jørgen Eriksen, René Gislum, Rasmus Nyholm Jørgensen, and Henrik Karstoft. Estimation of the botanical composition of clover-grass leys from RGB images using data simulation and fully convolutional neural networks. *Sensors (Switzerland)*, 17(12), 2017. ISSN 14248220. doi: 10.3390/s17122930.

[198] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

[199] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS*, 2017.

[200] Andrew Collette. *Python and HDF5*. O'Reilly, 2013.

[201] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[202] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, 2016. ISSN 10636919. doi: 10.1109/CVPR.2016.90.

[203] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[204] Md Sazzad Hossain, Andrew P Paplinski, and John M Betts. Adaptive class weight based dual focal loss for improved semantic segmentation. *arXiv preprint arXiv:1909.11932*, 2019.

[205] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[206] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[207] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in neural information processing systems*, pages 1495–1503, 2015.

[208] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

[209] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[210] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[211] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

[212] Noel CF Codella, Q-B Nguyen, Sharath Pankanti, David A Gutman, Brian Helba, Allan C Halpern, and John R Smith. Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM Journal of Research and Development*, 61 (4/5):5–1, 2017.

[213] Claudio Cusano, Paolo Napoletano, and Raimondo Schettini. Evaluating color texture descriptors under large variations of controlled lighting conditions. *Journal of the Optical Society of America A*, 33(1):17, 2016. ISSN 1084-7529. doi: 10.1364/josaa.33.000017.