# Land Disturbance Extraction in Alberta Oil Sands Satellite Imagery

by

Bingxu Hu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Geography

Waterloo, Ontario, Canada, 2021

## Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Tracking and mapping land disturbances from oil and gas development are critical to environmental assessments and land reclamation. Manual extraction of land disturbances is costly, time-consuming, and requires updating given rapid development. Through this study, a methodology and model for oil and gas land disturbance extraction along with an error-correcting algorithm solving rural mapping deficiencies is proposed. Chapter 3 titled "Deep-learning Extraction of Land Disturbances Arising from Oil and Gas Development" explores and finds an optimal strategy for land disturbance extraction methodology. Outlining the 3 findings in Chapter 3 for land disturbance extraction: (1) road and well-pad extraction should be integrated into a single task, (2) land disturbance extraction task should be segmented between forest and farmland backgrounds, and (3) RGB outperforms NDVI in land disturbance extraction.Chapter 4 titled "Maintaining Accurate Maps of Rural Land Disturbances: A Deep-Learning Automatic Change Detection Algorithm" introduces and tests the proposed error-correcting algorithm and explores its hyperparameters. Results in Chapter 4 show that the proposed automated error-correcting algorithm improves performance by $8.3\% - 15.4\%$ compared to baseline. Key findings in Chapter 4 explore how hyperparameters affect model performance: (1) alpha and beta in the AEC algorithm need to be carefully selected, (2) careful selection of alpha and beta can reduce the number of transitory artifacts introduced, (3) running the AEC algorithm a few times can greatly improve model performance, (4) adding a threshold to when the AEC algorithm begins stabilizes model performance. Combining findings in Chapters 3 and 4, an accurate fully automatic alternative to manual mapping for oil and gas land disturbance extraction is proposed. The combined model proposed in Chapter 4 is a ready solution to tracking, mapping, and managing land disturbances in the Alberta oil sands for purposes of environmental assessments and land reclamation.

## Acknowledgements

Thanks to Professor Jonathan Li for all his help and support. Without his guidance, none of this would be possible. Thanks to my parents for their love, advice, and support. Without my loving parents, I would not be here today. Thanks to Hongjie He and Hongzhang Xu from the Geospatial Sensing and Data Intelligence Group for their help reviewing and their comments on my thesis. Thanks to Dr. Jonathan Li, Dr. Ke Yang, Hongjie He, and Hongzhang Xu for sharing with me research resources. Thanks to Hongjie He, Hongzhang Xu, Siyu Li, Wenxuan Zhu, Yiqing Wu, Yuxiang Fang, Longxiang Xu, and Charlotte Pan for gathering and labelling the Alberta oil sands road and well-pad dataset used in my thesis. Thanks to Alan Anthony for all the support he has provided me throughout my graduate studies. Thanks to all my friends, loved ones, and everyone who has helped me on my journey here today. Finally thanks to the gracious committee examiners Dr. LinLin Xu, Dr. Michael Chapman, and Dr. Jonathan Li for taking the time to read and review my thesis.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

**AA** Average Pixel Accuracy 23, 24, 28, 31, 32, 53, 54, 59, 60, 64, 68

**AEC** Automated Error-Correcting Algorithm 7, 37, 40–45, 47–49, 51–55, 59, 60, 62, 64, 66, 68, 70, 71

**ASPP** Atrous Spatial Pyramid Pooling 9, 20

**cIoU** Class Intersection Over Union 24, 40, 41, 43, 47, 48, 52, 64

**CNN** Convolutional Neural Networks 3, 10, 11, 16, 35, 37, 66–68

**CNT** Cognition Network Technology 12

**CRF** conditional random fields 10

**CVA** Change Vector Analysis 11, 12

**DCNN** Deep Convolutional Neural Networks 9

**DL** Deep-learning 3, 5–7, 10, 13, 14, 16, 22, 25, 33, 35, 37, 41, 42, 67, 68, 71

**EC** Error-Correcting 40, 41, 43, 47, 48, 51, 52, 55, 64

**FCN** Fully Convolutional Network 9, 12, 20

**GTE** Graph Tensor Encoding 10, 11

**IoU** Intersection Over Union 24

**mIoU** Mean Intersection Over Union 24, 28, 31, 32, 53–55, 59, 60, 62, 64, 68

**ML** Machine-learning 3, 5, 67

**NDVI** Normalized Difference Vegetation Index 12, 17, 27, 33, 68

**NIR** Near-Infrared 16, 27

**RNN** Sequential or Recurrent Neural Network 10

**SSD** Single Shot MultiBox Detector 12

**YOLO** You Only Look Once 12

# Chapter 1

# Introduction

## 1.1 Motivations

Geospatial information on land disturbances related to mining development is essential to assess potential environmental impacts. The removal of soil and vegetation, and the development of infrastructure for production and transportation (Figure 1.1), lead to transformative and cumulative impacts on the environment (Yang et al., 2018). Land disturbances due to mining development and activity exacerbate the natural processes of erosion. Sustainable local and regional mining development can only be achieved once mining land-disturbances are identified, mapped, tracked, and monitored for potential risks and impacts (Erzurumlu and Erzurumlu, 2015).

The Alberta oil sands are the third-largest oil reserve in the world, after Venezuela and Saudi Arabia (Figure 1.2). The development of the oil sands in Alberta creates jobs and tax revenue supporting social programs and infrastructure projects. Sustainably developing the oil sands in Alberta is an important task with local, Canada-wide, and global ramifications. Understanding the ecological impacts of mining activities requires accurate rural maps of oil sands development. Land-disturbances from oil sands development can be divided into two categories: roads, and well-pads. The Alberta oil sands provide a diverse dataset on land-disturbances from development ranging a wide array of circumstances (Figure 1.3). This research will focus on extracting roads and well-pads from oil sands satellite imagery.

Two central problems need to be solved for the successful extraction of roads and well-

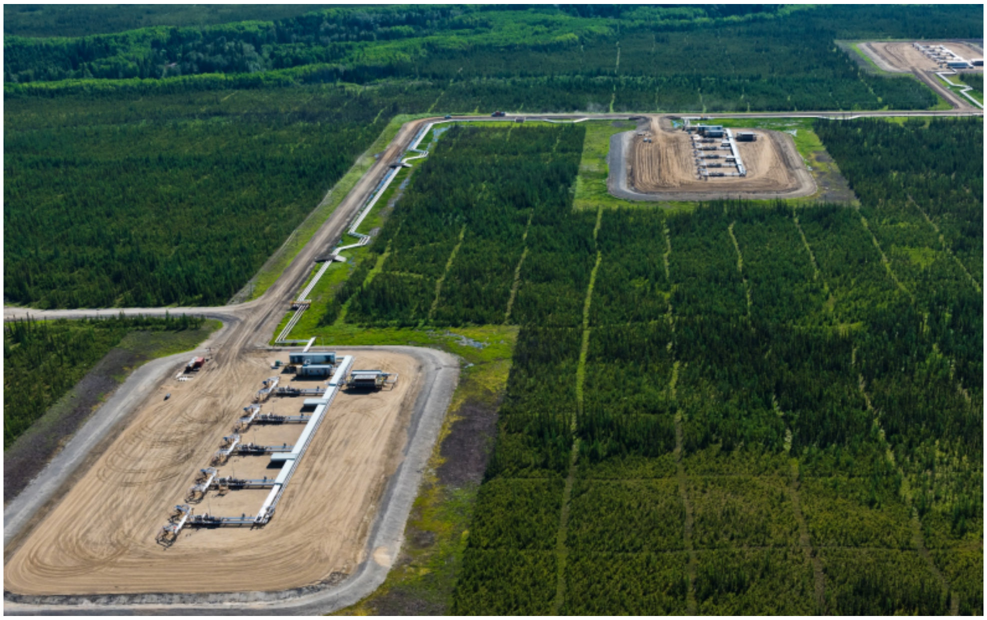

Figure 1.1: Aerial Image of a Well-Pad and Roads

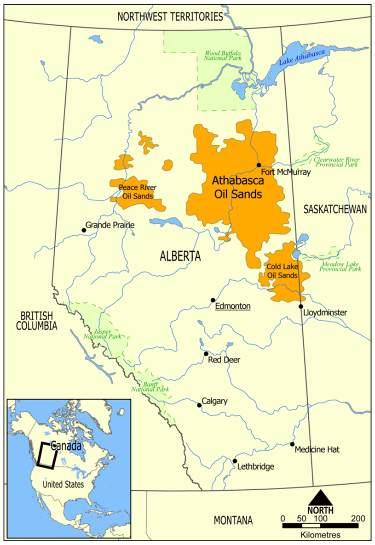

Figure 1.2: The Oil Sands in Alberta

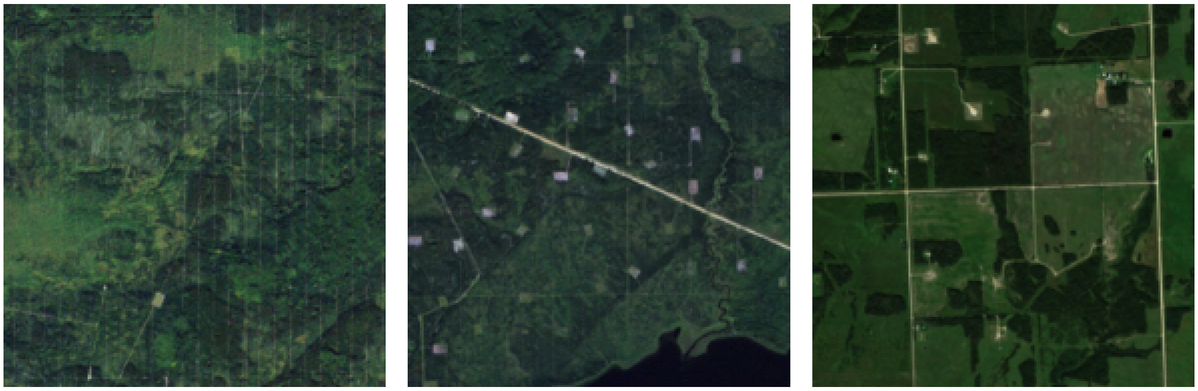

Figure 1.3: A wide range of land-disturbances in rural and farmland Alberta

pads. The first problem is that creating accurate road maps, in general, is challenging. Aerial imagery combined with deep learning provides an alternative to manual labour in creating high-quality and accurate rural maps (Bastani et al., 2018). Manual labelling of roads and well-pads is time-consuming, error-prone, and expensive (Yang et al., 2018). Algorithmic generation of roads and well-pads is also difficult given error vectors generated from the occlusion of trees, shadows, and noise (Figure 1.4). Machine-learning (ML) and Deep-learning (DL) are best suited for rural road and well-pad extraction in terms of accuracy and simplicity (Ball et al., 2017). With increasing popularity, Convolutional Neural Networks (CNN) have shown the best results in terms of accuracy on remote-sensing tasks such as road-extraction (Bastani et al., 2018; He et al., 2020), and building extraction (Sun et al., 2017). CNNs can be used as an effective automatic tool to extract roads and well-pads. The second central problem arises from utilizing CNNs as a tool for extraction in the procurement of high-quality data for model training and testing.

In some contexts, DL and CNN performance rely heavily on the quality of the data used for training and testing. DL models in some cases can perform well even when trained on low-quality data (Yeh et al., 2020). Reducing deep-learning model input into a series of

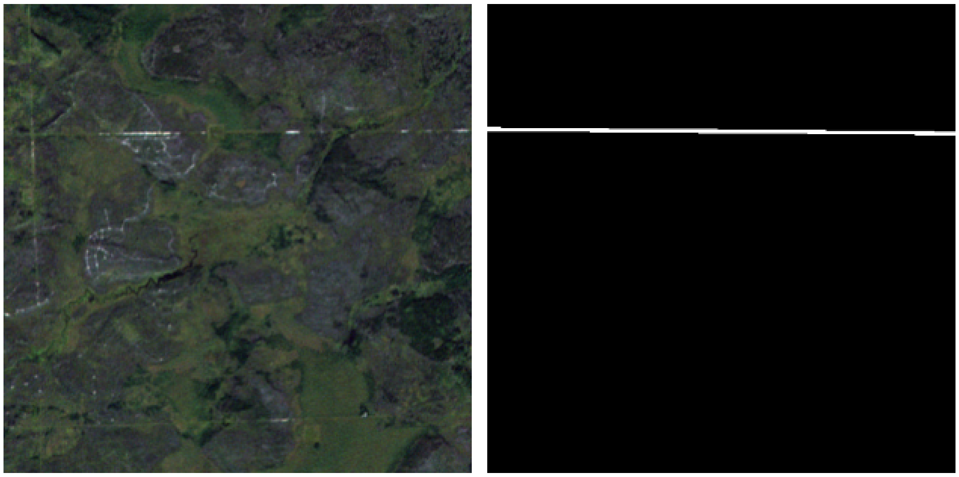

Figure 1.4: A noisy image with its respective ground truth

features $\{f_1, ..., f_n\}$ and predictions into a series of variables $\{v_1, ..., v_m\}$ where each unique $f_i$ is necessary to correctly predict a unique $v_k$, different contexts for deep-learning can be created. In this reduction of deep-learning models, additional noise or errors pollutes some features in the input and reduces the overall number of useable features.

In the case where $n > m$, the input contains more than enough information to correctly predict all variables in the output. As $n > m$, using lower-quality data with additional noise and errors to some factor less than the difference $n - m$ will not impact deep-learning performance if the ratio $m/n \leq 1$ remains with the lower-quality input. Alternatively, where $n \leq m$, lower-quality training and testing data can further reduce the number of features compared to the number of variables. Using lower-quality data will further increase the ratio of $m/n$ larger than 1 and will result in further limits to model performance and learning speed. Given the subjective and complex nature of road labelling from satellite imagery, as evident in comparing urban road maps (Kearney et al., 2020; Bastani et al., 2018), the assumption can be made that rural land disturbance extraction task has $n \leq m$

even with perfect data. This means that additional steps and techniques need to be utilized to remove noise and fix errors thus reducing the ratio $m/n$ for the land disturbance extraction task.

Sourcing accurate "ground truth" data for roads and well-pads for our rural study area is difficult. Given the challenges in sourcing accurate "ground truth" data even for well-travelled urban areas, "ground truth" data for rural area road-networks is not reliable (Kearney et al., 2020; Bastani et al., 2018). Road and well-pad maps in rural areas can be subjective and prone to error (Figure 1.5) as routine mapping is often lacking (Prendes et al., 2019; Workman et al., 2016). Rural road networks and well-pads are always changing with rapid new infrastructure development (Laurence and Balmford, 2013).

Apart from noise and errors diminishing model performance, statistically skewed or unbalanced data can also negatively affect model performance. Fundamentally, current ML and DL models rely on the assumption of densely and uniformly sampled training data (Kailkhura et al., 2019). When learning from data with skewed classes, incorrect inferences can be made across classes that weakens model generalization (Kailkhura et al., 2019). Combined with data noise and errors, skewed classes can exacerbate inaccuracies in model performance. Compared with urban road and object extraction, rural road and well-pad extraction have a greater data bias. As urban areas are much denser and uniform, a greater amount of information with less variation can be leveraged through the same DL techniques compared to extraction in rural imagery. Given the difference between urban and rural areas, urban road and object extraction methods and data are not easily applied to the study area. Additional considerations and steps need to be taken to minimize the effects of errors and noise in the data given the nature of the study area.

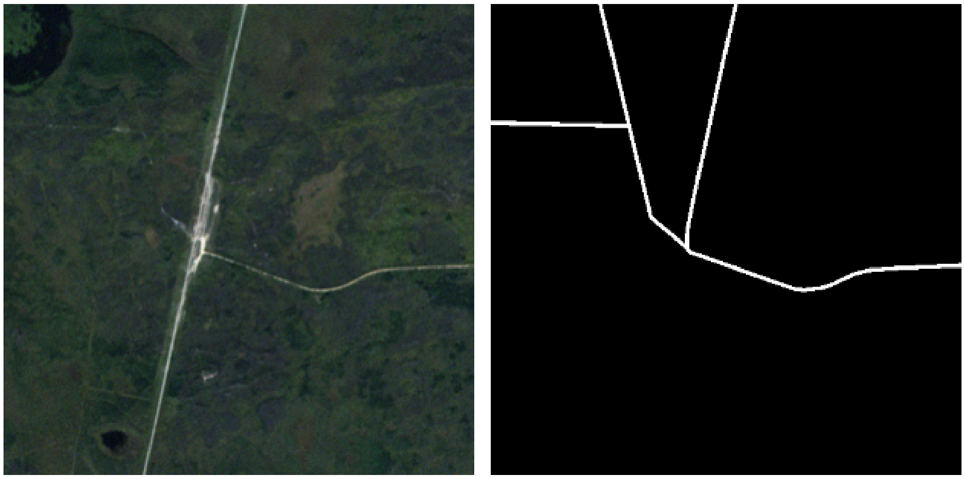

Figure 1.5: Image and "ground truth" pair with missing or wrong labels

## 1.2 Objectives of the Study

The objective of this study is to outline solutions solving the following two problems: (1) how to best leverage DL for road and well-pad extraction in our study area, and (2) how to minimize the effects of noise and errors in the data. By solving both problems, the combined model can automatically generate and update road and well-pad maps for the study area in the Alberta oil sands.

Addressing problem (1), this study shows the efficacy of DL in road and well-pad extraction and optimizes a methodology. There are three main axes that the extraction methodology can be configured in: the background landscape, the objects of interest, and the spectral information used. In this study, the land disturbance extraction results of using different methodology are compared and analyzed. Through this study, an effective methodology will be outlined for road and well-pad extraction in the study area.

In addition to optimizing the methodology, this study also outlines a solution to problem

(2) by proposing a change and error detection and correction algorithm (Automated Error-Correcting Algorithm (AEC)). This study shows that the proposed algorithm improves model performance in road and well-pad extraction. Additionally, this study introduces hyperparameters for AEC and investigates their effects on model performance. Through this study, an automatic algorithm for detecting and correcting change or error in data in a DL framework is outlined, tested, and analyzed.

## 1.3 Structure of the Thesis

This thesis is structured following the University of Waterloo manuscript-option format where manuscript articles prepared for submission are presented in standalone chapters. Chapter 1 introduces the motivation, research objectives, and structure of this study. Chapter 2 provides a literature review regarding DL road and well-pad extraction techniques. Chapter 3 presents the manuscript entitled ”Deep-learning Extraction of Land Disturbances Arising from Oil and Gas Development”, detailing the methodology, results, and findings of the study. Chapter 4 presents the manuscript entitled ”Maintaining Accurate Maps of Rural Land Disturbances: A Deep-Learning Automatic Change Detection Algorithm”, detailing the methodology, results, and findings of the study. Lastly, Chapter 5 summarizes the contributions and limitations of both studies and presents recommendations for future work.

# Chapter 2

# Related Studies

## 2.1 Road Extraction from Satellite Imagery

Road extraction is a well-studied area in the field of remote sensing. Road maps are critical in many applications. Location-based mobile services and autonomous vehicles are creating a necessity for high-quality road maps and networks (Bastani et al., 2018). Accurate road network maps are required for other extraction tasks such as vehicle extraction, building extraction, autonomous vehicles, and potentially oil-pad extraction (Bastani et al., 2018). Manual creation and maintenance of digital maps are tedious, expensive, and prone to error. Automated solutions of road extraction using satellite imagery, aerial imagery, and GPS have been shown to be innovative in generating road maps. Even given the success of the many proposed automated road extraction methods, road network extraction remains a tricky task due to the diversity and complexity of the road networks (He et al., 2020).

Road extraction has two main categories: pixel-wise segmentation- and graph-based approaches (He et al., 2020). Segmentation-based approaches first assign semantic scores to each pixel in satellite imagery and then extract a road network using heuristics. On the other hand, graph-based approaches directly construct a road network from satellite imagery (Bastani et al., 2018; Chu et al., 2019; Li et al., 2019). Segmentation-based approaches rely on wide/global receptive fields, generate non-graph intermediate representations, and utilize heuristics to generate road-network graphs. Errors and noise in the non-graph intermediaries can propagate and limit the accuracy of the final road network in segmentation-based approaches (He et al., 2020). Developing a reliable post-processing heuristic for segmentation-based approaches can also be challenging given the error, noise, and complexity of the intermediaries (He et al., 2020). In contrast, graph-based approaches rely on local information in a narrow receptive field and an iterative process to generate road-network graphs directly from imagery. Recent works (Chu et al., 2019; Li et al., 2019) have proposed improvements to the performance of graph-based approaches through incorporating global information through a sequential generative approach. The performance of graph-based approaches relies heavily on local information in a narrow receptive field and can drop in imagery with noisy backgrounds. Recent research (Batra et al., 2019a; Chu et al., 2019; Li et al., 2019) in both segmentation-based and graph-based approaches have focused on shoring up on their respective limitations and shortcomings.

Deep Convolutional Neural Networks (DCNN) (Zhou et al., 2018; Batra et al., 2019b; Simonyan and Zisserman, 2015; He et al., 2015; Szegedy et al., 2016) have shown to be highly effective on many visual recognition tasks. In terms of semantic segmentation, Fully Convolutional Network (FCN) (Long et al., 2015) architecture is most efficient and has achieved state-of-the-art performances. Zhang et al. (2018) demonstrated the effectiveness of the FCN architecture to extract roads by utilizing UNet with residual connections. Zhou et al. (2018) builds on Linknet (Chaurasia and Culurciello, 2017), an efficient semantic segmentation encoder-decoder network that utilizes skip-connections and residual blocks, to create D-Linknet for road extraction. D-Linknet employs additional dilated convolutional layers which enlarges the receptive field while still preserving detailed local information (Zhou et al., 2018). DeepLabV3+ (Chen et al., 2018) utilizes atrous spatial pyramid pooling along with an encoder-decoder structure to incorporate multi-scale contextual information in semantic segmentation tasks. Atrous Spatial Pyramid Pooling (ASPP) probes imagery with filters or pooling operations at multiple rates and effective field-of-views (Chen et al., 2018). DeepLabV3+ is a highly promising architecture to be used for road extraction and other semantic segmentation tasks.

Like segmentation-based approaches, graph-based approaches such as RoadTracer (Bastani et al., 2018), Sat2Graph (He et al., 2020), and DeepRoadMapper (Máttyus et al., 2017) have also shown promising results for road extraction. Wegner et al. (Wegner et al., 2015) laid the groundwork for graph-based approaches through creating a minimum-cost path algorithm utilizing conditional random fields (CRF) to solve for road networks. DeepRoadMapper (Máttyus et al., 2017) builds on roads extracted through segmentation by solving a shortest-path problem using A*-search. DeepRoadMapper (Máttyus et al., 2017) improves extracted road networks by ensuring and enhancing connectedness in road topology. RoadTracer (Bastani et al., 2018) is a purely graph-based approach that iteratively builds road networks one segment at a time. RoadTracer (Bastani et al., 2018) leverages DL by training a CNN the iterative process of adding one road at a time. In essence, RoadTracer trains a CNN not only how to identify road segments but also how to complete A* search. Due to its iterative nature, RoadTracer is extremely sensitive to noise and errors as small errors made early in the algorithm can quickly propagate into large errors (Bastani et al., 2018). Recent works (Chu et al., 2019; Li et al., 2019) advances ideas proposed by RoadTracer by applying Sequential or Recurrent Neural Network (RNN). Sequential models allow for the inclusion of greater contextual information compared to RoadTracer (He et al., 2020). On the other hand, the non-recurrent and holistic nature of RoadTracer to generate road networks in one shot allows it to be easier trained and extended (He et al., 2020). Sat2Graph (He et al., 2020) proposes a Graph Tensor Encoding (GTE) that allows CNNs to be trained without an iterative or recurrent approach. Sat2Graph (He et al., 2020) improves performance over segmentation-based approaches in properly identifying stack roads that are not by segmentation-based approaches due to their intermediary representation. Sat2Graph trains a CNN like segmentation-based approaches but replaces a traditional pixel encoding with GTE labels (He et al., 2020). GTE being more restrictive and less expressive compared to pixel encoded labels means that noise and errors in labels will quickly erode model performance.

## 2.2 Well-pad extraction from Satellite Imagery

Well-pad extraction is not commonly studied in computer vision. Understanding the short- and long-term environmental and ecological impacts of well-pads requires accurate detection of well-pads. Like road extraction, manual extraction of well-pads from satellite or aerial imagery can be time-consuming and costly. Automating well-pad extraction utilizing deep-learning is promising given the success of deep-learning approaches with other similar problems.

There are two approaches to well-pad extraction: segmentation-based detection, or knowledge/heuristics-based detection. Segmentation-based detection extracts well-pads by assigning semantic scores to each pixel in satellite imagery. On the other hand, knowledge/heuristics-based extracts well-pads by identifying and labelling clusters of pixels. Much existing research into well-pad extraction has proposed utilizing knowledge-based and heuristics-based classification schemes. Jiao et al. (2015) showed promising results for well-pad extraction utilizing image pre-processing techniques in a knowledge-based approach. Jiao et al. (2015) utilized three texture measures (grey level co-occurrence matrix mean, mean deviation from mean and variance, and radar vegetation index) to construct a tiered segmentation scheme to distinguish oil sands infrastructure from the background. Zhang et al. (2014) proposed a heuristic-based well-pad extraction approach using Change Vector Analysis (CVA) with satellite imagery. CVA is a methodology based on the estimation of radiometric differences between images. Utilizing a time-series of images over the same scene, Zhang et al. (2014) applies the CVA methodology on the Normalized Difference Vegetation Index (NDVI) and a proposed regrowth index to extract and track well-pad development. Zhang et al. (2017) shows similar success in extracting and tracking well-pad development by using spectral analysis and Cognition Network Technology (CNT). CNT is a proprietary object-based image analysis method. As they rely heavily on algorithms and assumptions on data, knowledge/heuristics-based approaches do not generalize well across all possibilities, can be prone to errors caused by noise, can be dependent on hyper-parameters, and can require manual manipulation.

Segmentation-based deep-learning approaches have been very successful in object detection tasks. In computer vision, the primary object detection objective involves identifying and labelling a bounding box for a localized object in an image. The object detection objective can be extended to also identifying all pixels in an image for each object effectively creating a mask rather than a bounding box. With these two object detection objectives in mind, researchers utilize either a one- or two-stage detector. One-stage detectors, like the You Only Look Once (YOLO) family of detectors (Redmon et al., 2016; Redmon and Farhadi, 2016, 2018; Bochkovskiy et al., 2020), prioritizes detection speed through generating a bounding box without generating an object mask. RetinaNet (Lin et al., 2018)., and Single Shot MultiBox Detector (SSD) (Liu et al., 2016) are also examples of powerful one-stage detectors. Two-stage detectors focus on accuracy over speed by generating regions of interest and outputting object masks instead of bounding boxes. Popular two-stage detectors include Faster R-CNN (Ren et al., 2016), Mask R-CNN (He et al., 2018) and Cascade R-CNN (Cai and Vasconcelos, 2017). In addition to one- and two-stage detectors, FCNs also show promise in the semantic segmentation task given in the extended object detection objective. DeepLabV3+ has shown excellent performance in semantic segmentation tasks (Chen et al., 2018). Deep learning approaches show great potential in automating the extraction of well-pads.

## 2.3 Ground Truth Error Correction

Ground truth error correction is not commonly studied in computer vision. In most cases and studies, the ground truth data is carefully curated to remove errors and biases. In most experiments, the curation of ground truth data is done before the experiments are ran. Most research in the field begin with the assumption of high-quality data that is accurate, complete, and unbiased (Sessions and Valtorta, 2006). Data preparation is usually done separate before DL experiments are ran. Usually the scope of DL studies are controlled by the scope of available high-quality data. With the availability of urban road data through OpenStreetMaps and other similar datasets, urban road extraction with machine-learning has seen an influx of studies (Vargas-Munoz et al., 2021; Bastani et al., 2018; He et al., 2020). On the other hand, in DL tasks where high-quality datasets are not easily accessible, the data curation process becomes a barrier for DL research to occur. Much of ground truth error correction happens during the data preparation phase and is apart of the data curation process.

There are two main approaches to ground truth data curation: manual, and machine-assisted. When it comes to data curation, the four characteristics of DL data need to be considered: Volume, Velocity, Variety, and Value (Katal et al., 2013). The volume and velocity of DL data pose a problem to researchers in curating high-quality data. Given the manual curatation of data, the volume and velocity of data can overwhelm researchers lacking resources and can greatly delay research. Given a high volume and velocity of data, there is a tradeoff between resources spent and time required to curate the necessary data for experiments. Given machine-assisted curation, the data curation process can become easier compared to a fully manual data curation process. The challenge to machine-assisted curation is the creation of the heuristical pipeline or algorithms required. Given the necessary expertise, algorithmic, or heuristical knowledge, some data curation tasks can be fully automated (Girard et al., 2019). For most cases, given the variety of DL data, machine-assisted curation process cannot be applied between different DL tasks (Girard et al., 2019; Máttyus et al., 2017). Given a heuristical approach to machine-assisted curation, costly manual manipulation can still be required in edge cases given high volume and velocity of data. There is no generic heuristic or manual approach to data curation for all DL tasks.

Apart from data curation, certain deep-learning techniques can be used to reduce the effects of low-quality data. The effects of low-quality data can be catorgorised as contributing to model underfitting or overfitting. Viewing errors as contradictions, DL models that tries to learn a task given this data and these contradictions will not learn efficiently nor correctly. The speed that the model learns the task will be greatly reduced as the model loss for all iterations will have an average higher than with perfect data. Through increasing the capacity of the model to fix the problem of underfitting with low-quality data, the model may have a new problem of overfitting. With low-quality data, a model with high capacity can learn generalizations that are not correct and be limited in overall performance. Techniques exist in DL to reduce model overfitting such as regularization, focal loss, weight constraint, dropout, noise, and early stopping. Given random errors and noise in low-quality data, these techniques can be helpful to smooth out model loss and for the model to settle in a better global extrema. On the other hand, these techniques are not helpful in removing contradictions and errors in the data. There is a need for processes and algorithms built into DL to correct ground truth errors and contradictions so that higher quality data is output given a lower quality input.

# Chapter 3

# Deep-learning Extraction of Land Disturbances Arising from Oil and Gas Development

## 3.1 Introduction

The development of oil and gas infrastructure has caused widespread alterations to landscapes spanning the world. Being essential to our daily lives, oil and natural gas development require responsible management, monitoring, and tracking (Zhang et al., 2017). Sustainable and responsible management and monitoring of oil and gas development footprints is the basic requirement for successful land reclamation and impact assessments (Antoniuk et al., 2009; Jordaan et al., 2009).

Bitumen can be extracted either by surface mining or *in-situ* production methods (Jordaan et al., 2009). As most petroleum deposits are too deep and intermittent to be extracted by surface mining, *In-situ* extraction of bitumen has become the predominant method for petroleum deposit exploitation (Zhang et al., 2017). As a result, land-disturbances from typical oil and gas development extend over large geographic regions and take the form of numerous well-pads and access roads. Mapping and monitoring of well-pads and roads in a region are essential for quantitative environmental assessments (Zhang et al., 2017; Rokosh et al., 2012; ERCB, 2012). Through the monitoring of both surface mining and *in-situ* oil and gas development, the responsible exploitation of resources and eventual successful land reclamation can occur.

With the increasing availability of high-resolution multispectral satellite imagery, new methods and models for land disturbance extraction have been developed (Zhang et al., 2014). DL methods especially CNN have become increasingly popular and effective in pixel- and object-based extraction (Bastani et al., 2018; Máttyus et al., 2017; Zhang et al., 2018; He et al., 2020; Chen et al., 2018; Zhou et al., 2018). Utilizing high-resolution RapidEye imagery, this study explores methodologies for oil and natural gas land disturbance extraction. With the oil sands of Alberta as a study area, the data contains a diverse set of background landscapes and land disturbances. Given the various backgrounds, land disturbances, and spectral information, DL and CNN models perform differently given various methodology configurations. The objective of this work is to compare and analyze the performance of DL land disturbance extraction given different methodology configurations of the study area. The structure of the paper is as follows: in Section 3.2, descriptions of the study area data, and preprocessing is provided. In Section 3.3, descriptions of the DL model used, evaluation metrics, and Experiment parameters are provided. Results and discussions on the performance of models, given different methodologies, are provided in Section 3.4. Conclusions and further discussions are provided in Section 3.5.

## 3.2 Data and Preprocessing

The dataset consists of 600 patches with size 500x500px generated from RapidEye satellite imagery with a spatial resolution of 5m. Each image consists of 5 spectral resolution bands; blue, green, red, red-edge, and Near-Infrared (NIR). The dataset consists of 400 patches with forested backgrounds and 200 patches with farm backgrounds. Each patch

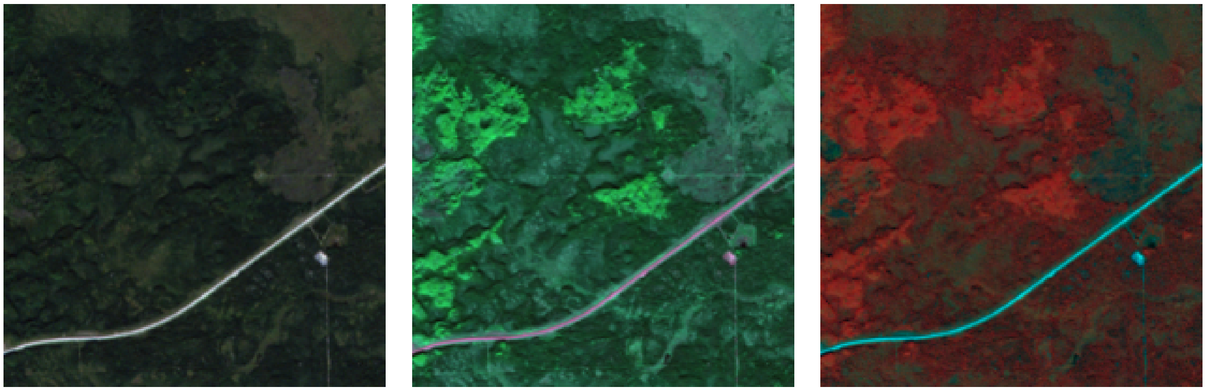

Figure 3.1: Left: RGB, Center: Red-Red-Edge-NIR, Right: Clipped NDVI

has respective road and well-pad labels.

For each of the 600 patches, three 3-banded images are created (Figure 3.1). The first type of image is RGB. The second type of image is Red-Red-edge-NIR. The third type of image is a clipped NDVI scheme. All created images have pixel values normalized between 0 and 1 using equation (4.1).

$$Norm(p_i, img = \{p_1, p_2, ..., p_i, ..., p_n\}) = \frac{p_i}{max(img)} \tag{3.1}$$

NDVI is a powerful index to measure live green vegetation. NDVI is used to quickly identify vegetated areas and quantify their condition. NDVI has values ranging between -1 to 1. Negative values of NDVI approaching -1 correspond to water. Values close to 0 generally represent barren areas containing rock, sand, or snow. Low positive values (0.2 to 0.4) of NDVI represent shrub and grassland, whereas high positive values approaching 1 represent temperate and tropical rainforests. Equation (3.2) shows how NDVI is calculated. Equation (3.3) shows how the NDVI value is modified to produce a clipped NDVI value.

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{3.2}$$

$$ClippedNDVI = max(0, NDVI - 0.2) \tag{3.3}$$

The clipped NDVI has a range between 0 and 0.8. The clipped NDVI value is used in a 3-band scheme ClippedNDVI-Green-Blue. Utilizing this clipped NDVI scheme, only vegetated areas (NDVI values 0.2 to 1) are pigmented red. In this clipped NDVI scheme (clipped NDVI), the green and blue bands are still visible for areas with NDVI between -1 to 0.2. Clipping NDVI is a useful heuristic that further contrasts land-disturbances from vegetation (Figure 3.2).

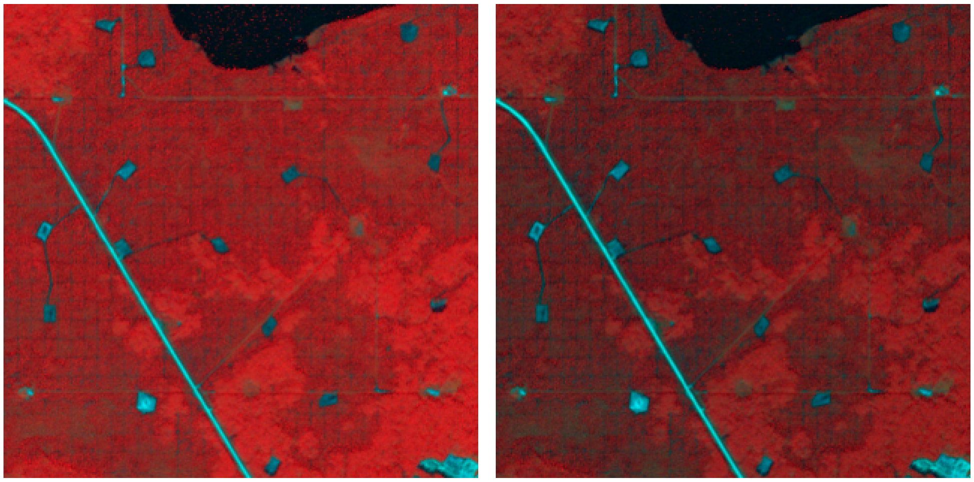

Figure 3.2: Land-disturbances are more visible in the clipped NDVI (Right) compared to the unclipped NDVI scheme

Ground truth road and well-pads are labelled separately for each image patch. Roads and well-pads are stored in a shapefile (.shp) and are converted to 500x500px black and white images. Three ground truth label images are created for each of the 600 patches (Figure 3.3): labels containing just roads, labels containing just well-pads, and labels containing both roads and well-pads. The dataset is divided into training and testing sets by using a 10-fold algorithm. As the patches are sequentially numbered respective to their geographic positioning, a 10-fold division of training and testing sets ensures a relatively unbiased uniform distribution for both sets. Of 600 patches, 60 patches are used for testing and 540 patches are used for training.

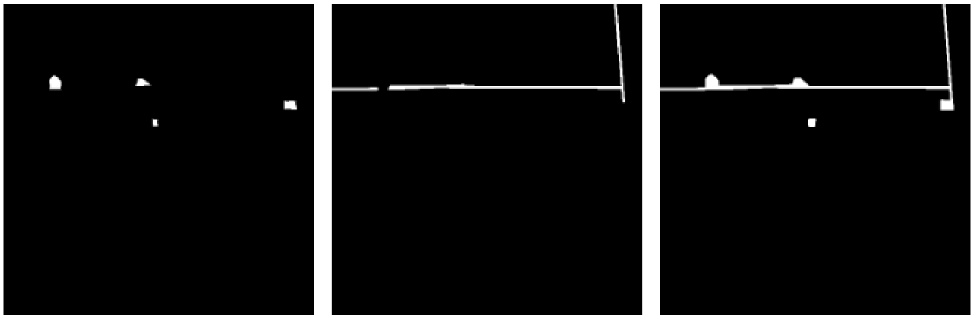

Figure 3.3: Well-Pad Label (Left), Road Label (Center), and Combined Label (Right)

## 3.3 Methods

### 3.3.1 DeepLabV3+

A segmentation-based approach is used for both road and well-pad extraction. Not only is a segmentation-based approach best for both tasks but it also makes integrating both extraction tasks very easy. Given noisier data when compared to urban imagery, using graph-based approaches to extract road networks from rural imagery becomes less feasible. Due to the lack of man-made structures and the additional randomness of rural background

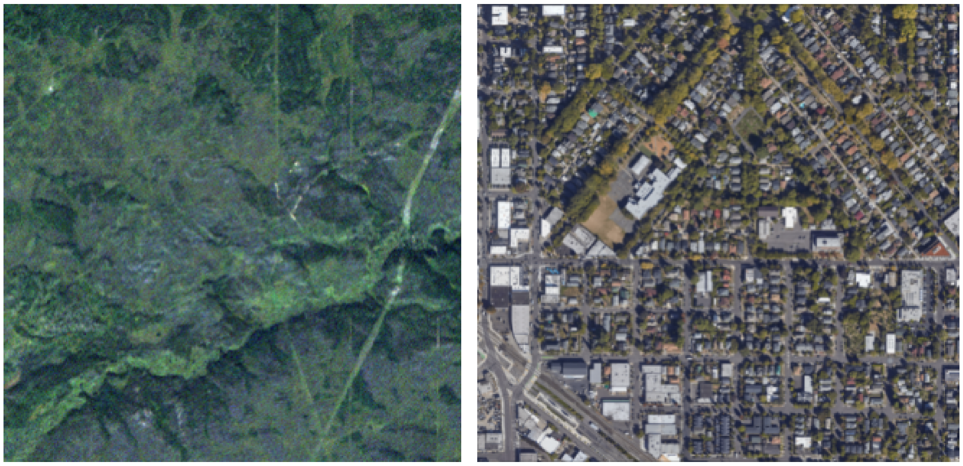

Figure 3.4: Rural Imagery (left) and Urban Imagery (right) for Road Extraction

scenes (Figure 3.4), graph-based approaches for road extraction become difficult to train due to their reliance on local data and narrow receptive fields. Given the various shapes of well-pads (Figure 3.5), an extended object-detection task is best suited for well-pad extraction. Pixel segmentation is more useful compared to bounding boxes in quantifying land disturbances caused by variously sized well-pads. Two-stage detectors or FCNs are best suited for an extended object-detection task such as well-pad extractions. Given both road and well-pad extraction tasks can be completed by FCNs, both tasks can be combined by integrating labels in a generalized land disturbance extraction task.

DeepLabV3+ is a powerful semantic-segmentation model that can complete both road and well-pad extraction tasks (Chen et al., 2018). The DeepLabV3+ model combines ASPP with the encoder-decoder structure of the previous DeepLabV3 model (Chen et al., 2018). ASPP allows DeepLabV3+ to utilize a wider receptive field and leverage greater contextual background information compared to DeepLabV3 (Figure 3.7) (Chen et al., 2018). DeepLabV3+ proposes a new decoder module and uses DeepLabV3 as an encoder module

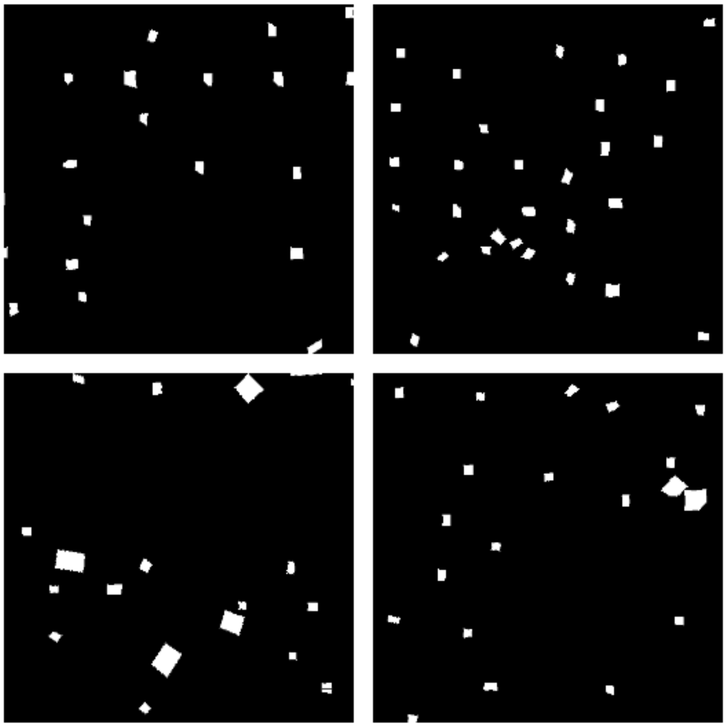

Figure 3.5: Example Well-pads of different shapes

(Figure 3.6). Aside from DeepLabV3+, other semantic segmentation models may also be suitable for this task. DeepLabV3+ is selected for it's ability to incorporate together both local- and global-receptive fields. DeepLabV3+ model is chosen as it performs extremely well on the PASCAL VOC 20212 and Cityscapes dataset. Other semantic segmantation models may also be suitable for this task. The goal of this research is not to determine the best DL model for rural land disturbance extraction but to provide a methodology for extraction for any given DL model.

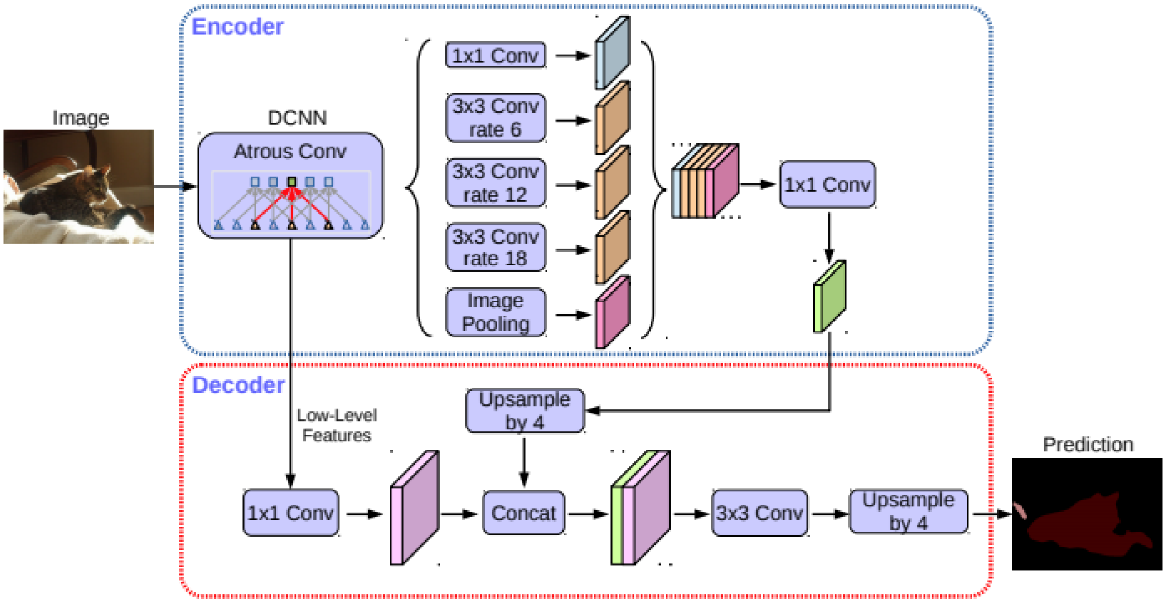

Figure 3.6: The DeepLabV3+ Architecture (Chen et al., 2018)

The output stride DeepLabV3+ hyper-parameter determines the output resolution of the DeepLabV3 encoder (Chen et al., 2018). The output of the encoder along with the ASPP MobileNet backbone output is then fed to the proposed decoder to produce a segmentation prediction (Chen et al., 2018) (Figure 3.6). MobileNet is selected in favour of ResNet for model backbone given its faster speed and lower memory requirement, albeit

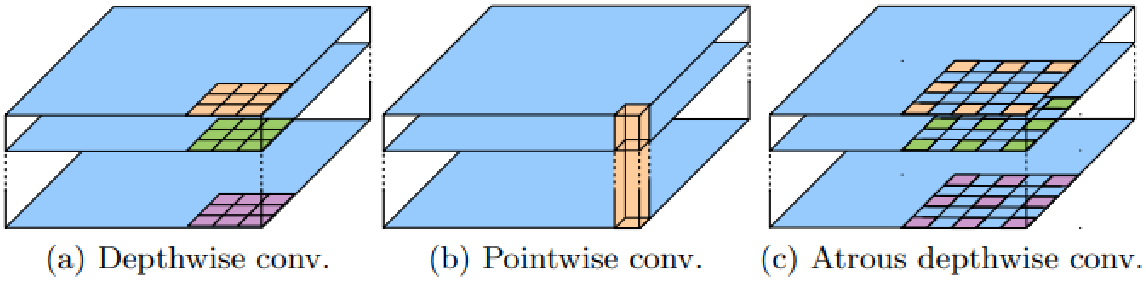
(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

Figure 3.7: ASPP allows for a wider receptive field (Chen et al., 2018)

lower performance (Chen et al., 2018). An output stride of 16 is chosen in favour of 8 or 32 as a balance between performance and speed. Focal loss is utilized in favour of cross-entropy to mitigate class imbalance within ground truth labels (Lin et al., 2018). Focal loss dynamically negatively scales cross-entropy loss with class confidence (Lin et al., 2018). The remaining parameters are left as default to DeepLabV3+ with a learning rate of 0.01, polynomial decay schedule, and momentum set to 0.9. A standard training batch size of 32 is chosen to balance speed and performance and a validation/test batch size of 8 is used. The same random seed is used throughout all experiments. All experiments are conducted using Cuda 11.1 on a GeForce RTX 3090 GPU and an AMD Ryzen 9 3900x CPU.

### 3.3.2 Evaluation Metrics

#### Average Pixel Accuracy (AA)

Average accuracy represents the average pixel accuracy of segmentation compared to ground truth labels (Akosa, 2017). Pixel accuracy is defined as the percentage of pixels correctly predicted by the model (Akosa, 2017). Equation (3.4) shows how Average Pixel Accuracy (AA) is calculated.

$$AA = Avg(pixelAccuracy_(img_1), ..., pixelAccuracy_(img_n)) \tag{3.4}$$

Given (3.4) and $n$ the total number of images, $pixelAccuracy_(img_i)$ defines the pixel accuracy for the $i^{th}$ image, and the function $Avg(...)$ produces the average value of all its operands. The main problem of AA is bias due to class imbalance (Akosa, 2017). The AA value can be skewed by a model that correctly classifies a minority of class making up most of the ground truth but misclassifies a majority of other classes. In our case where one class makes up over 95% of the ground truth, it is important to consider AA bias and to use other metrics as well.

### Intersection Over Union (IoU)

Intersection Over Union (IoU) is a commonly used metric in segmentation (Rahman and Wang, 2016). Equation (3.5) shows how IoU is calculated.

$$IoU = \frac{|Y^* \cap Y|}{|Y^* \cup Y|} \tag{3.5}$$

Where $Y^*$ and $Y$ in (3.5) is the predicted segmentation and the ground truth respectively (Rahman and Wang, 2016). IoUs can be taken for each class. These Class Intersection Over Union (cIoU) are useful in measuring the segmentation accuracy for each specific class. Taking the average of all cIoUs, we can generate the Mean Intersection Over Union (mIoU). mIoU equally penalizes inaccuracies in segmentation for each class regardless of frequency in ground truth making it unbiased due to class imbalance. Both cIoUs and mIoU are useful in measuring the accuracy of segmentation models.

### 3.3.3 Methodology Configurations

The objective of this work is to compare and analyze the performance of DL land disturbance extraction given different methodology configurations of the study area. Three main axes of configuration exist for the methodology: the objects of interest, the background landscape, and the spectral range. The experiments in this paper all run for 30k iterations.

#### Objects of Interest

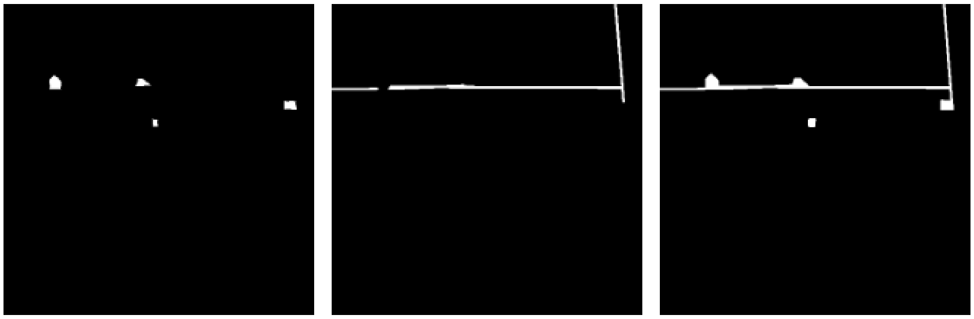

Figure 3.8: Well-Pad Label (Left), Road Label (Center), and Combined Label (Right)

There are two types of land disturbances of interest to this study: roads, and well-pads. Although roads and well-pads are similar in terms of a pixel-wise comparison, they are different in shape and structure (Figure 3.8). Given that both well-pads and roads are typically made of crushed gravel, soil, and asphalt, RapidEye imagery of both will have similar pixel-wise comparisons (Zhang et al., 2017). The objective of configuring data in terms of object of interest is to determine whether performance can be improved by combining road and well-pad extraction as one integrated task. Table 3.1 highlights the three experiments conducted to determine changes in model performance given different objects of interest.

Table 3.1: Object of Interest Experiments

| # | Background | Objects of Interest | Spectral Range |
|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB |
| 2 | Farmland | Roads | RGB |
| 3 | Farmland | Well-Pads | RGB |

### Background Landscape

As oil and natural gas resources often span large geographical areas, land disturbances such as roads and well-pads can have various backgrounds as well. In the study area dataset, there are two distinct backgrounds: farmland, and forest. Figure 3.9 provides a typical comparison between the two distinct backgrounds in the dataset.

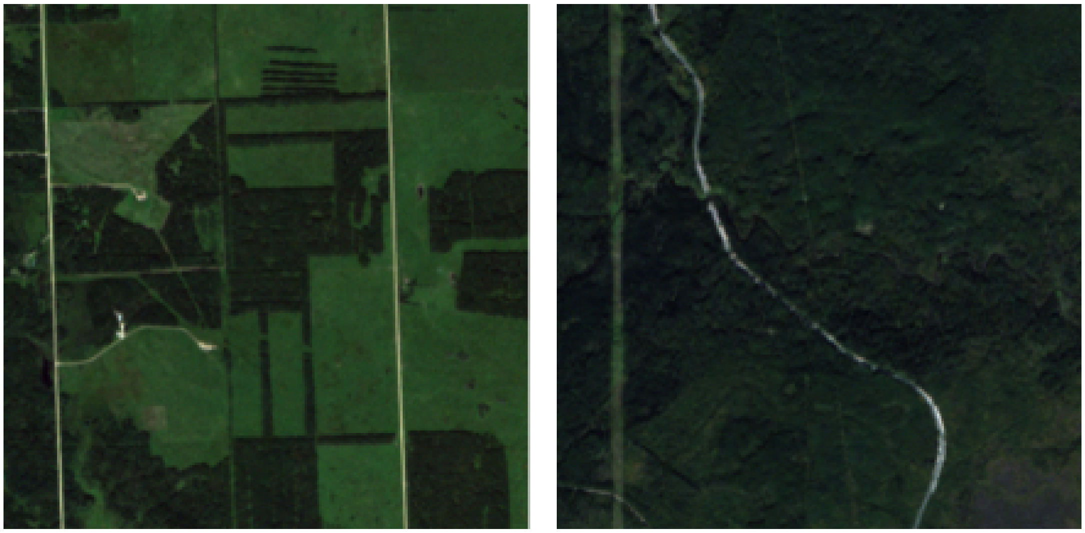

Figure 3.9: Typical Farm (Left) and Forest (Right) Images in the Dataset

Forested background images in the dataset typically are noisier and less organized compared to farmland background images in the study area dataset. The objective of configuring the data in terms of background landscape is to determine whether useful generalizations can be made regardless of background landscape on extracting road and well-pad land disturbances. Table 3.2 highlights the three experiments to determine changes in model performance given different backgrounds.

Table 3.2: Background Landscape Experiments

| # | Background | Objects of Interest | Spectral Range |
|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB |
| 4 | Farmland and Forest | Roads and Well-Pads | RGB |
| 5 | Forest | Roads and Well-Pads | RGB |

### Spectral Range

Given 5 available spectral bands in RapidEye imagery, different 3-banded spectral configurations are possible. As Zhang et al. (2017) has shown the effectiveness of utilizing spectral bands beyond RBG, this study explores the impact on extraction performance given different spectral band configurations. The objective of configuring the data in terms of spectral range is to determine the performance of the model utilizing an NDVI spectral range. Table 3.3 highlights the five experiments to determine changes in model performance given various RGB and NDVI spectral bands configurations.

Experiments 7 and 8 in Table 3.3 test the efficacy of the clipped NDVI scheme in extracting land disturbances. As shown by Equation (3.2), NDVI is calculated using NIR and red spectral bands. Since s6 in Table 3.3 contains both red, red-edge and NIR bands necessary to calculate NDVI, it forms a baseline to the proposed clipped NDVI scheme as tested in experiments 7 and 8.

Table 3.3: Spectral Range Experiments

| # | Background | Objects of Interest | Spectral Range |
|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB |
| 5 | Forest | Roads and Well-Pads | RGB |
| 6 | Farmland | Roads and Well-Pads | Red-Red-edge-NIR |
| 7 | Farmland | Roads and Well-Pads | Clipped NDVI |
| 8 | Forest | Roads and Well-Pads | Clipped NDVI |

## 3.4 Results and Discussion

Table 3.4 contains the best test set results for all experiments. The results will be discussed respectively to the three main axes of configuration: the objects of interest, the background landscape, and the spectral range. Figure 3.10 shows some example predictions made by the model in these experiments.

### 3.4.1 Objects of Interest

Table 3.5 highlights the relevant experiments exploring how configuring an object of interest affects model performance.

Comparing Experiment 2 and 3 from Table 3.5, the DeepLabV3+ model performs road extraction with higher AA and mIoU scores compared to well-pad extraction. Through combining road and well-pad extraction into one integrated task, results for Experiment 1 show performance like Experiment 2 with higher AA but lower mIoU scores.

Taking into consideration the class distribution of road and well-pad pixels (Table 3.6), the performance increase in integrating road and well-pad extraction tasks can be properly measured. Through separately completing road and well-pad extraction tasks, weighted AA of 74.68%, and weighted mIoU of 69.57% can be achieved overall by combining results

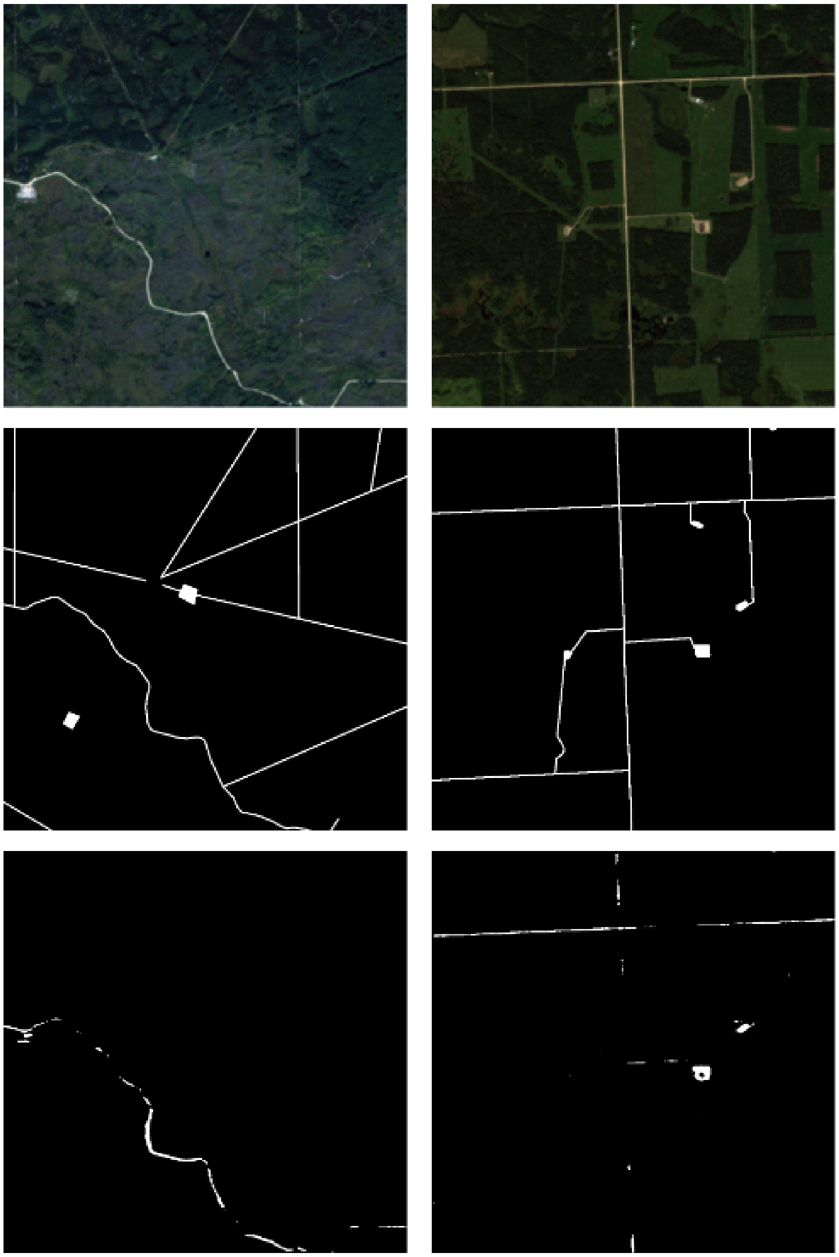

Figure 3.10: Example image (top), label (middle), and predictions (bottom) from DeepLabV3+.

Table 3.4: Results for All Experiments

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.7555 | 70.2978 | 97.699/42.896 |
| 2 | Farmland | Roads | RGB | 76.4662 | 70.9378 | 98.061/43.814 |
| 3 | Farmland | Well-Pads | RGB | 64.1556 | 61.5204 | 99.798/23.240 |
| 4 | Farmland and Forest | Roads and Well-Pads | RGB | 63.9187 | 60.5215 | 96.928/24.514 |
| 5 | Forest | Roads and Well-Pads | RGB | 58.8028 | 56.4271 | 96.590/16.263 |
| 6 | Farmland | Roads and Well-Pads | Red-Red-edge-NIR | 74.5332 | 68.4289 | 97.552/39.305 |
| 7 | Farmland | Roads and Well-Pads | Clipped NDVI | 76.0448 | 69.4143 | 97.599/41.229 |
| 8 | Forest | Roads and Well-Pads | Clipped NDVI | 58.2180 | 55.9014 | 96.561/15.241 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Table 3.5: Results for Objects of Interest Experiments

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.75 | 70.29 | 97.69/42.89 |
| 2 | Farmland | Roads | RGB | 76.46 | 70.93 | 98.06/43.81 |
| 3 | Farmland | Well-Pads | RGB | 64.15 | 61.52 | 99.79/23.24 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Table 3.6: Road and Well-pad Class Distribution

| Class | # of Pixels | % of Total |
|---|---|---|
| Roads | 681598 | 14.5 |
| Well-Pads | 4017264 | 85.5 |

of experiments 2 and 3. Through integrating road and well-pad extraction tasks, AA and mIoU performance are increased by 2.7% and 1% respectively compared to separately completing both tasks. With this performance improvement, it can be determined that model performance is improved by employing an integrated extraction approach combining both road and well-pads.

### 3.4.2 Background Landscape

Table 3.7 highlights the relevant experiments exploring how configuring background landscape affects model performance.

Table 3.7: Results for Background Landscape Experiments

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.75 | 70.29 | 97.69/42.89 |
| 4 | Farmland and Forest | Roads and Well-Pads | RGB | 63.91 | 60.52 | 96.92/24.51 |
| 5 | Forest | Roads and Well-Pads | RGB | 58.80 | 56.42 | 96.59/16.26 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Comparing experiments 1 and 5 in Table 3.7, DeepLabV3+ performs better road and well-pad extraction with higher AA and mIoU scores in imagery with farmland background compared to forest. Through combining extraction for both farmland and forest background, the model has AA and mIoU performance slightly worse than the average of completing them separately.

Taking into consideration that the dataset consists of 200 farmland and 400 forest background images, a better measure of Experiment performance is set by using weighted averages. Through taking the weighted average between experiments 1 and 5, an AA score of 64.78% and mIoU score of 61.04% is achieved by completing land disturbance extraction separately. Completing extraction separately for farmland and forest background improves AA performance by 1.3% and mIoU performance by 0.8% compared to integrating extraction for both background landscapes. With this performance increase, it can be determined that no useful generalizations can be made in an integrated extraction approach combining farmland and forest background landscapes.

### 3.4.3 Spectral Range

Table 3.8 highlights the relevant experiments exploring how configuring the spectral range affects model performance.

Table 3.8: Results for Spectral Range Experiments

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.75 | 70.29 | 97.69/42.89 |
| 5 | Forest | Roads and Well-Pads | RGB | 58.80 | 56.42 | 96.59/16.26 |
| 6 | Farmland | Roads and Well-Pads | Red-Red-edge-NIR | 74.53 | 68.42 | 97.55/39.30 |
| 7 | Farmland | Roads and Well-Pads | Clipped NDVI | 76.04 | 69.41 | 97.59/41.22 |
| 8 | Forest | Roads and Well-Pads | Clipped NDVI | 58.21 | 55.90 | 96.56/15.24 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Comparing experiments 1 and 5 to experiments 7 and 8 respectively, performance utilizing RGB spectral bands has higher AA and mIoU scores compared to performance with the clipped NDVI spectral bands. Comparing experiments 6 and 7, utilizing the clipped NDVI spectral bands improves model performance over utilizing a raw formatted NDVI. Looking at the results of Table 3.8, it can be determined that RGB is the optimal spectral range for DeepLabV3+ performance in land disturbance extraction.

## 3.5 Chapter Summary

With more oil and gas resources developed globally, there is a need to monitor, manage, and track land disturbances such as roads and well-pads because of development. With increasing accessibility, high-resolution satellite imagery is becoming the best data source to identify and track oil and gas infrastructure. DeepLabV3+ is a powerful, robust, and cost-effective DL method utilizing high-resolution satellite imagery to extract oil and gas development land disturbances. Through this study, an effective methodology for road and well-pad extraction employing DL is demonstrated through the results of 8 experiments.

Several important conclusions can be drawn from this study. First, integrating road and well-pad extraction as one task produces better performance results compared to separately extracting roads and well-pads. By combining road with well-pad extraction, overall model confusion is lowered. Second, land disturbance extraction tasks perform better by separating the data by background landscape farmland and forest into two separate tasks. Model generalizations from integrating farmland and forest background landscapes into one task do not outperform the additional noise and confusion introduced into the data and model. Third, using an RGB spectral range outperforms a raw formatted NDVI and clipped NDVI spectral range. Additional experiments can be conducted to further explore other spectral ranges. In conclusion, our results show the efficacy and effective methodology of using DL to extract oil and gas development land disturbances from satellite imagery.

# Chapter 4

# Maintaining Accurate Maps of Rural Land Disturbances: A Deep-Learning Automatic Change Detection Algorithm

## 4.1 Introduction

Upkeeping accurate and up-to-date maps can be an expensive manual process. Accurate and up-to-date road maps are an important dataset and resource for a wide range of planning, conservation, emergency response, and research activities (Barrington-Leigh and Millard-Ball, 2017). Although OpenStreetMap and crowdsourcing paved road maps have steadily improved, maps in rural, forested regions with unpaved roads are still lacking (Barrington-Leigh and Millard-Ball, 2017; Prendes et al., 2019; Workman et al., 2016). Oil and gas development spanning large geographical areas with numerous unpaved roads and well-pads is difficult to accurately map. Land disturbances from oil and gas infrastructure can have immense impacts on the local community, environment, and wildlife. To successfully assess impacts for eventual reclamation, the monitoring and tracking of oil and gas land disturbances are critical (Antoniuk et al., 2009; Jordaan et al., 2009; Zhang et al., 2017; Rokosh et al., 2012; ERCB, 2012).

What makes tracking, monitoring, and mapping land disturbances from oil and gas development difficult is volatility (Laurence and Balmford, 2013). As oil and gas resources are further exploited, new infrastructure such as roads and well-pads are rapidly developed (Laurence and Balmford, 2013). Depleted oil and gas resource land disturbances can be either reclaimed or abandoned (Kearney et al., 2020). Combined with spanning large geographical regions, rapidly developed, abandoned, and reclaimed land disturbances from oil and gas development is a challenge to map accurately (Kearney et al., 2020).

An increasingly popular approach is to leverage DL and high-resolution satellite imagery for object extraction. With many state-of-the-art performance results, CNNs have been shown to efficiently extract objects from satellite imagery (He et al., 2020; Bastani et al., 2018; Máttyus et al., 2017; Sun et al., 2017; Chen et al., 2018). CNN learns to complete computer vision tasks by training on many examples. Through processing examples with a variety of conditions, CNNs can make generalizations across the examples about the respective task. Through the generalizations learned by CNNs, it can make predictions on new examples. Crucial to this process is the assumption that the examples provided to the CNN model are correct.

Assuming no errors exist in the data, CNNs can learn correct generalizations and make correct predictions. In the case of extracting volatile oil and gas land disturbances, the assumptions that no errors exist in the data cannot be made (Figure 4.1). Even in the case of the urban datasets OpenStreetMap and TorontoCity, recent works report a 14% discrepancy between the two (Máttyus et al., 2017). As oil and gas infrastructure is rapidly changing, the data collected to create training examples can contain errors that quickly erode and limit model performance. Learning from errors or bad examples, CNNs learn to make some incorrect generalizations that sway it to make erroneous predictions. Manual correcting of these errors and bad examples can be a costly, time-consuming, and subjective process. An alternative to manual relabelling would be to allow DL frameworks to account for and correct errors and bad examples automatically.

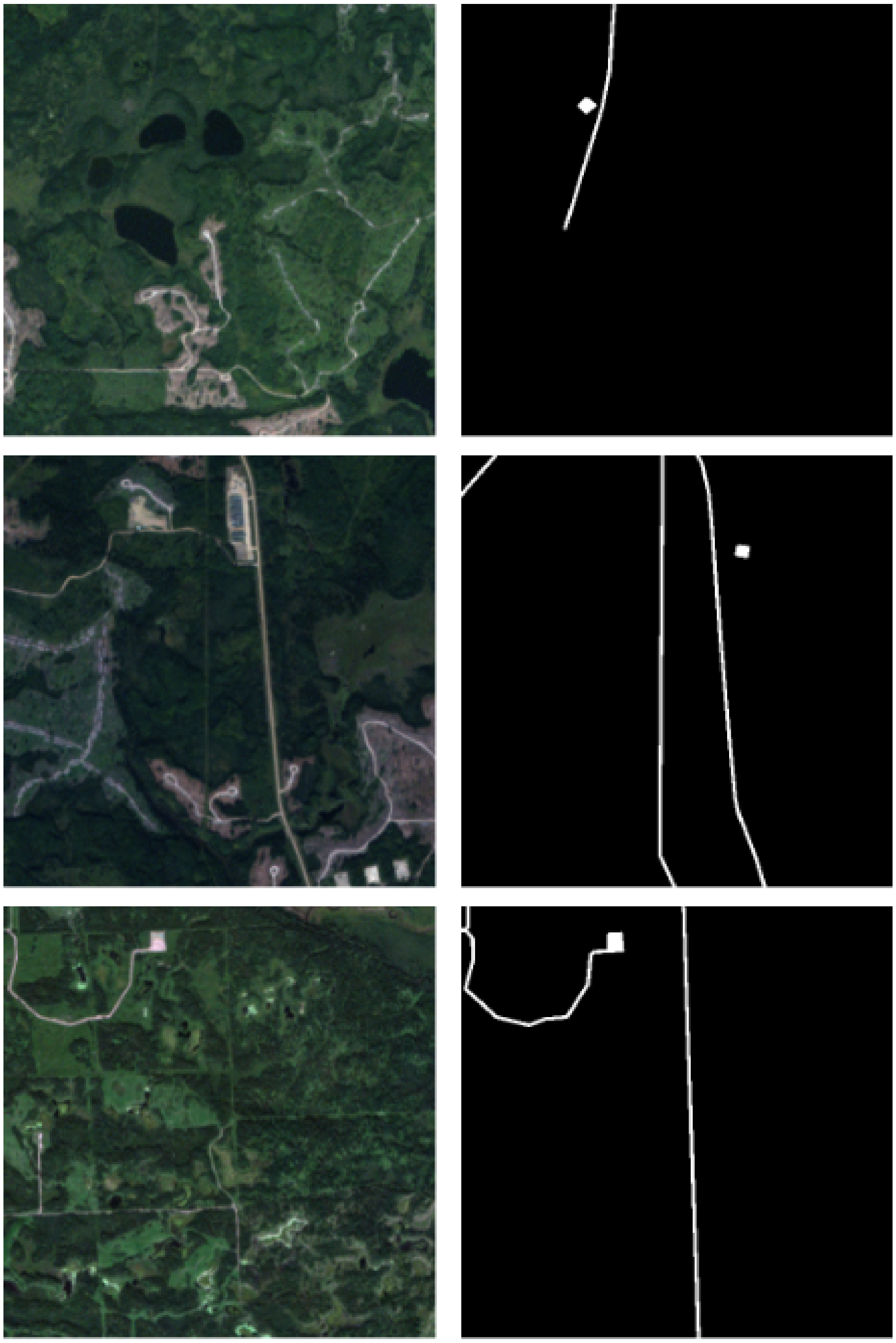

Figure 4.1: Example of Errors in Ground Truth

Instead of the assumption that no errors exist in the data, the paradigm should be relaxed to account for some errors in the data instead. Given the assumption that training examples contain some errors, the expectation on the CNN model is that it will make some correct but also some incorrect generalizations. Under this relaxed paradigm, error-correcting algorithms can be created to leverage the correct generalizations to fix bad examples. By correcting bad examples during training, the erroneous generalizations learned by the model can be reduced and fewer erroneous predictions are made. Using error-correcting algorithms for oil and gas land disturbance extraction can increase the accuracy of the CNN predictions.

The objective of this study is to propose an automatic error-correcting algorithm (AEC) integrated into DL that corrects errors in the data during training for road and well-pad extraction. This study shows that the AEC improves and speeds up performance over a baseline model not utilizing the algorithm. This study explores new hyperparameters introduced by the AEC algorithm and their effects on model performance. DeepLabV3+ is used as the baseline CNN model for this study for its state-of-the-art performance in various other segmentation tasks (Chen et al., 2018). The Alberta oil sands are used as the study area given its diverse set of background landscapes and land disturbances. High-resolution RapidEye RGB imagery is used. The structure of the paper is as follows: in Section 4.2, descriptions of the study area data, and preprocessing is provided. In Section 4.3, a description of the DL model, metrics, the AEC algorithm and its parameters, and Experiment details are provided. Section 4.4 provides the results and discussions on the performance of the model with the AEC algorithm, and different hyperparameter settings of the AEC algorithm. Section 4.5 provides conclusions and further discussions.

## 4.2 Data and Preprocessing

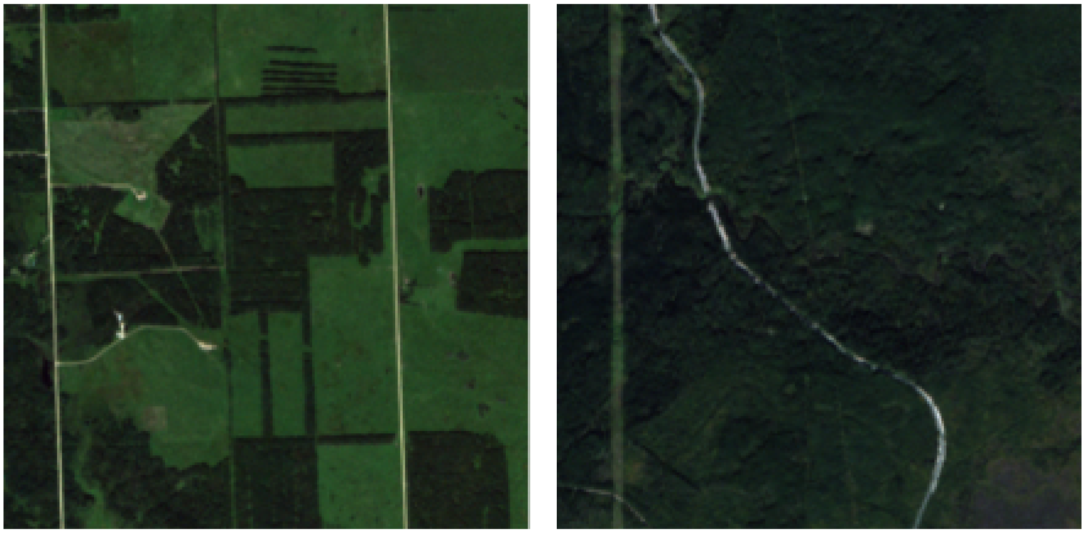

Figure 4.2: Typical Farm (Left) and Forest (Right) Images in the Dataset

The dataset used consists of 600 (500x500px) patches generated from RapidEye satellite imagery with a spatial resolution of 5m and RGB spectral bands. The dataset is divided into two subsets depending on background landscape (Figure 4.2): 400 forest background patches, and 200 farmland backgrounds patches. Patches have pixel values normalized between 0 and 1 using equation (4.1).

$$Norm(p_i, img = \{p_1, p_2, ..., p_i, ..., p_n\}) = \frac{p_i}{max(img)} \tag{4.1}$$

Each patch has a combined road and well-pad label. Each label image corresponds to a dataset image and is 500x500px and black and white. Figure 4.3 shows some examples of label images. The two dataset subsets are further divided into training and validation sets by using a 10-fold algorithm. As the patches are sequentially numbered respective to their geographic positioning, a 10-fold division of training and testing sets ensures a relatively unbiased uniform distribution for both sets. Of 600 patches, 60 patches are used for testing and 540 patches are used for training.

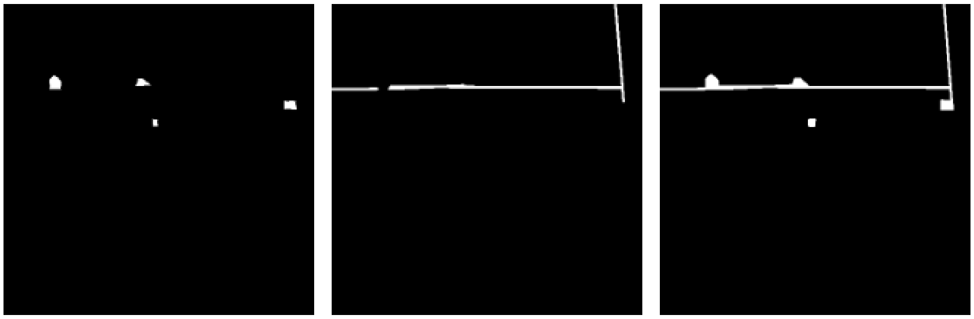

Figure 4.3: Well-Pad Label (Left), Road Label (Center), and Combined Label (Right)

## 4.3 Methods

### 4.3.1 DeepLabV3+

Refer to Section 3.3.1

### 4.3.2 Evaluation Metrics

Refer to Section 3.3.2

### 4.3.3 Automatic Error-Correcting Algorithm (AEC)

Ground truth labels can contain errors that slow learning and limit model performance. Obtaining high-quality ground truth labels is a non-trivial task. Recent work (Máttyus et al., 2017) has shown that even between two high-quality datasets OpenStreetMap (OSM) and TorontoCity a 14% discrepancy exists in ground truth labels. Errors exist in our ground truth data (Figure 4.1) and correcting these errors can increase the performance of our model. Even if only training set errors are fixed, performance in the validation dataset still improves as the model learns more efficiently. Manually correcting ground truth labels can be a very time-consuming and costly task. Automating Error-Correcting (EC) is an important process in this task and other studies.

The main idea of the AEC algorithm is to allow ground-truth corrections to be made in proportion to how accurately the model performs. Assuming $m$ number of images in the validation set, let $V = (Y^*, Y)$ where $Y^* = \{Y_1^*, ..., Y_m^*\}$ are model predictions from the validation dataset and $Y = \{Y_1, ..., Y_m\}$ are the validation dataset ground truths. Assuming $n$ number of images in the training set, let $y^* = \{y_1^*, ..., y_n^*\}$ be the model predictions on the training dataset and $y = \{y_1, ..., y_n\}$ be the training dataset ground truths. Let $p^*$ and $p$ be corresponding pixels in corresponding images in $y^*$ and $y$. Let $class(a)$ be a function that returns the class of pixel $a$. Let $cIoU(c, V)$ return the validation set cIoU of class c. Let the probability $P(p|y^*, V)$ be the EC probability that pixel $p$ in $y$ will be updated to match pixel $p^*$ in $y^*$. Equations (4.2) and (4.3) describe the AEC algorithm.

$$P(p|y^*, V) = Min(cIoU(class(p^*), V)^{f(p,V)}, 1) \tag{4.2}$$

$$\begin{aligned} &f(p,V) = \alpha - \beta * cIoU(class(p^*), V) \\ &\alpha, \beta > 0, \alpha \neq 0 \end{aligned} \tag{4.3}$$

This AEC algorithm will only update pixels that differ between $y^*$ and $y$ proportional to the respective validation set cIoU and function $f(p,V)$. When the model accuracy increases and the cIoUs increase in the validation set, the probability for updates to be made in the training set also increases. The AEC algorithm can be used with varying additional restrictions during training. Other types of $f(p,V)$ functions can also be utilized and alter the behaviour of the AEC algorithm but will not be explored in this study.

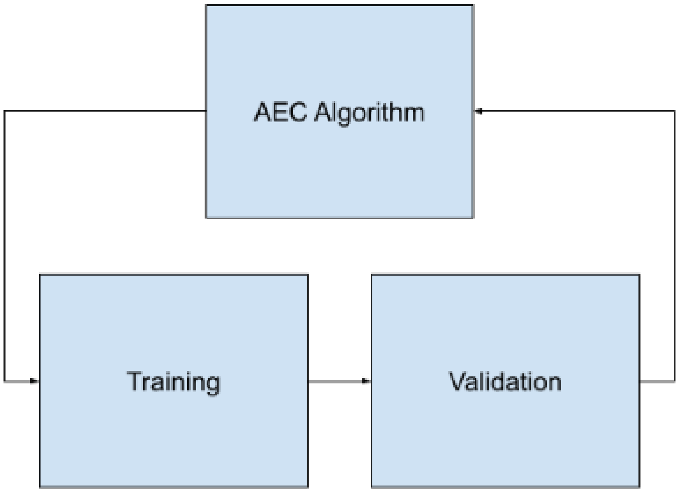

Figure 4.4: Implementation of AEC Algorithm

Figure 4.4 shows how the DL finite-state machine works with the AEC algorithm. The AEC algorithm takes the validation performance of the model to determine the EC probability. The AEC algorithm does not make updates to the validation set such that the validation set remains independent. As no updates are made to the validation set, any validation set performance improvements over control experiments is attributed to the

AEC algorithm. Algorithm 1 outlines the psuedocode for the AEC algorithm.

**Algorithm 1** AEC Algorithm

```
for image $im_i^*$ in $y^*$ do
    for pixel $p^*$ in $im_i^*$ do
        $c \leftarrow class(p^*)$
        $cIoU \leftarrow cIoU(c, V)$
        $probability \leftarrow Min(cIoU^{\alpha - \beta * cIoU}, 1)$
        if $Random(probability) == True$ then
            $p \leftarrow p^*$
        end if
    end for
end for
```

### 4.3.4 AEC Hyperparameters

The objective of this study is to propose an automatic error-correcting algorithm (AEC) integrated into DL that corrects errors in the data during training for road and well-pad extraction. This study also explores new hyperparameters introduced by the AEC algorithm and their effects on model performance. In the following sections, experiments for a baseline AEC algorithm, and on AEC hyperparameters are described. All AEC experiments are run for 10k iterations.

**AEC Baseline**

Table 4.1 lists the experiments with DeepLabV3+ with a baseline version of the AEC algorithm with only $\alpha$ changing. The parameters $\beta$, threshold, remove, and interval are explained and tested in later sections and experiments. Table 4.2 shows the DeepLabV3+ baseline results without AEC integrated to compare with Table 4.1 experiments. Figure 4.5 shows the AEC update probabilities for experiments in Table 4.1. The values of $\alpha$ are somewhat arbitrarily picked in this experiment. The $\alpha$ values of 4 and 5 are selected as they allow somewhat non-trivial EC probability given the peak cIoU of around 42% in Experiment 1 in Table 4.2.

Table 4.1: Baseline AEC Algorithm Experiments

| # | Background | $\alpha$ | $\beta$ | Threshold | Remove | Interval |
|---|---|---|---|---|---|---|
| 9 | Forest | 2 | 0 | 0 | F | 200 |
| 10 | Farmland | 2 | 0 | 0 | F | 200 |
| 11 | Farmland | 4 | 0 | 0 | F | 200 |
| 12 | Farmland | 5 | 0 | 0 | F | 200 |

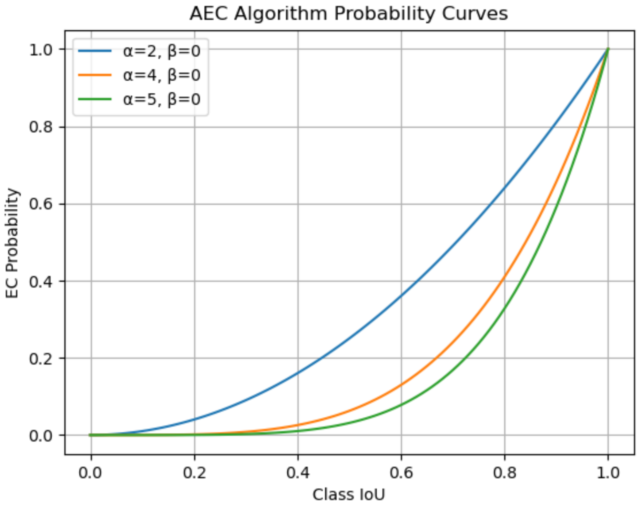

Figure 4.5: Graph of EC Probability for Experiments

### Remove Parameters

The remove parameter represents the type of updates the AEC algorithm is programmed to do. The baseline AEC algorithm is programmed to only add pixels. The remove parameter allows the AEC algorithm to also remove pixels. Table 4.3 lists the relevant experiments for

Table 4.2: Baseline Control Experiments and Results

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.75 | 70.29 | 97.69/42.89 |
| 5 | Forest | Roads and Well-Pads | RGB | 58.80 | 56.42 | 96.59/16.26 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

the remove hyperparameter. Note that the remove parameter does not allow the removal of pixels belonging to the original ground truth. This parameter only allows the removal of previously added pixels by the AEC algorithm. Figure 4.5 shows the AEC update probabilities for experiments in Table 4.3.

Table 4.3: Experiments on the AEC Algorithm for Remove Parameter

| # | Background | $\alpha$ | $\beta$ | Threshold | Remove | Interval |
|---|---|---|---|---|---|---|
| 12 | Farmland | 5 | 0 | 0 | F | 200 |
| 13 | Farmland | 5 | 0 | 0 | T | 200 |

### Interval Parameter

The interval parameter sets the number of training iterations between update iterations by the AEC algorithm. Given a known total number of model iterations, the interval parameter can be used to control the total number of AEC updates over the course of training (equation Equation (4.4)).

$$NumUpdates(intervalParameter, trainingIterations) = \frac{trainingIterations}{intervalParameter} \tag{4.4}$$

One interesting interval parameter Experimentto conduct is to only allow the AEC algorithm to make updates when model performance peaks. Only allowing AEC updates to be made on peak model performances is described as ”On Best” in the interval parameter as seen in Table 4.4. Experiment 13 in Table 4.4 establishes a baseline for experiments 14 and 15. Experiment 14 in Table 4.4 establishes a baseline to compare with Experiment 15. Experiment 15 differs from Experiment 14 in Table 4.4 as it uses the final updated labels of Experiment 14 as its ground truth. Figure 4.5 shows the AEC update probabilities for experiments in Table 4.4.

Table 4.4: Experiments on the AEC Algorithm for Interval Parameter

| # | Background | $\alpha$ | $\beta$ | Threshold | Remove | Interval |
|---|---|---|---|---|---|---|
| 13 | Farmland | 5 | 0 | 0 | T | 200 |
| 14 | Farmland | 5 | 0 | 0 | T | On Best |
| 15* | Farmland | 5 | 0 | 0 | T | On Best |

*Experiment 15 uses the final updated ground-truth from Experiment 14.

### Threshold Parameter

The threshold parameter controls the minimal iteration value when AEC updates start being made. Model performance increases with time as model iterations increase. Using the threshold parameter, it is possible to set a fuzzy lower bound on model performance before pixel updates by the AEC algorithm are allowed to be made. Table 4.5 lists the experiments made by thresholding updates. The threshold value of 3k is chosen in Table 4.5 as the baseline experiments 1 and 5 in Table 4.2 roughly achieves 95% of peak performance by 3k iteration . Figure 4.5 shows the AEC update probabilities for experiments in Table 4.5.

Table 4.5: Experiments on the AEC Algorithm for Threshold Parameter

| # | Background | $\alpha$ | $\beta$ | Threshold | Remove | Interval |
|---|---|---|---|---|---|---|
| 11 | Farmland | 4 | 0 | 0 | F | 200 |
| 12 | Farmland | 5 | 0 | 0 | F | 200 |
| 16 | Farmland | 4 | 0 | 3000 | F | 200 |
| 17 | Farmland | 5 | 0 | 3000 | F | 200 |

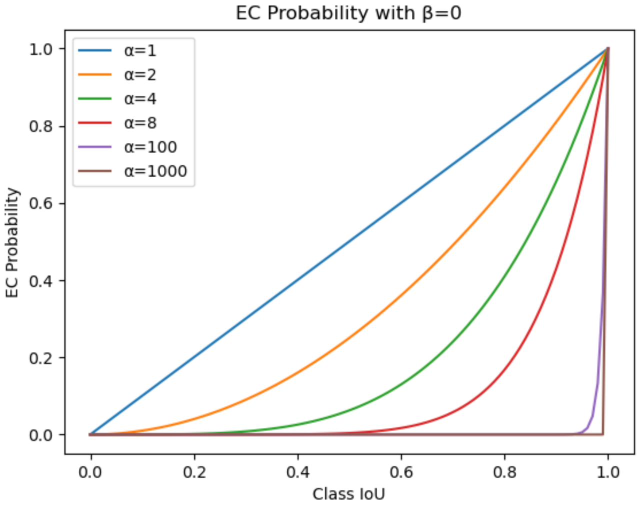

Figure 4.6: Graph of EC Probability with $\beta = 0$ and various $\alpha$

### Parameters $\alpha$ and $\beta$

Deep-learning performance, in general, is asymptotic and follows a logistical trend, the function $P(p|y^*, V)$ is designed to keep this in mind. The constants $\alpha$, and $\beta$ are important hyperparameters that dictate how the AEC algorithm behaves during training. Setting $\beta$ to be 0, the AEC algorithm is dictated in relation to a power of $\alpha$ (Figure 4.6). As $\alpha$ approaches infinity given $\beta$ is 0, the behaviour of the AEC algorithm approaches the behaviour of the piecewise function equation (4.5).

$$f(x) = \begin{cases} 0 & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x = 1 \end{cases} \tag{4.5}$$

In the case where $\alpha$ approaches infinity given $\beta$ is 0, this would be the idealistic behaviour of the AEC algorithm assuming the deep-learning model performance asymptotically approaches 100%. In this case, the model only makes ground truth updates when it is extremely confident that they are correct. However, the behaviour of allowing updates to be made when the model is already confident is a catch-22. The catch-22 is that if the model is already confident, the number of pixels that are errors must be small. Another key aspect is that practically speaking, most if not all complex deep-learning models for segmentation tasks asymptotically do not approach 100% cIoU performances. A non-zero $\beta$ hyper-parameter addresses the practical asymptotic performance of deep-learning models. Utilizing different values of $\beta$, EC probability can be set to match the asymptotic behaviour of the deep-learning model. By setting different values of $\alpha$ and $\beta$, it is possible to directly set the EC probability given arbitrary cIoU.

There are two cases that greatly change how the AEC algorithm behaves: $\beta > \alpha$, and $\beta \leq \alpha$. In the case where $\beta \leq \alpha$ (Figure 4.7), it is clear to see that as the confidence of the

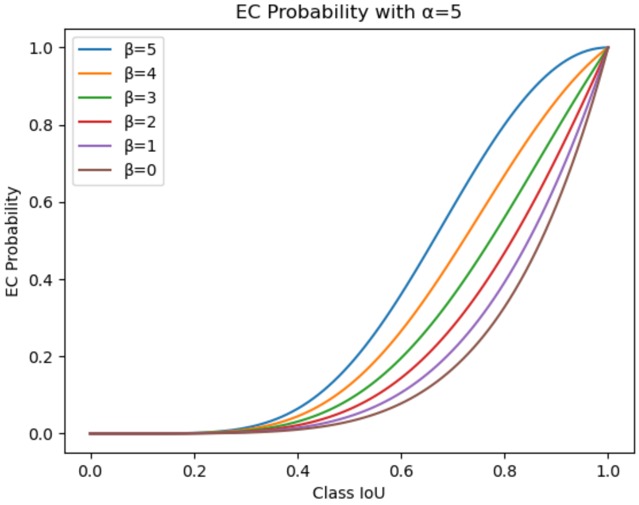

Figure 4.7: Graph of EC Probability with $\alpha = 6$ and $\beta \leq \alpha$

model increases to 100%, the behaviour of $P(p|y^*, V)$ changes from $IoU(class(p^*), V)^\alpha$ to $IoU(class(p^*), V)^{\alpha-\beta}$. Setting $\beta \leq \alpha$, $\beta \neq 0$ provides greater control over the EC behaviour then simply setting $\alpha$,with $\beta = 0$ as it mitigates the piecewise-like behaviour with large values of $\alpha$ (Figure 4.8).

In the case of $\beta > \alpha$, the EC probability can be set to any value at an arbitrary cIoU value. Given that any value to the power of 0 is equal to 1, the EC probability becomes 100% when the cIoU equals $\alpha/\beta$ for arbitrary values of $\alpha$ and $\beta$ in the AEC algorithm. Given a cIoU value equal to $\alpha_1/\beta_1$ given $\alpha_1$,and $\beta_1$, setting $\alpha = \alpha_1$, and $\beta = \beta_1$ in $f(p, V)$ will set EC probability at cIoU value $\alpha_1/\beta_1$ to be equal to 100% (Figure 4.9). If instead it is desired for EC probability to be $\lambda = (\alpha_1/\beta_1)^c$ given a constant $c$ at cIoU value equal to $\alpha_1/\beta_1$, setting $\alpha = \alpha_1 + c$, and $\beta = \beta_1$ will produce the desired result (Figure 4.10). In addition to setting a desired EC probability given a cIoU value, different multiples of the $\alpha$ and $\beta$ values can alter the slope of the graph (Figure 4.11). With the $\beta > \alpha$ case, full

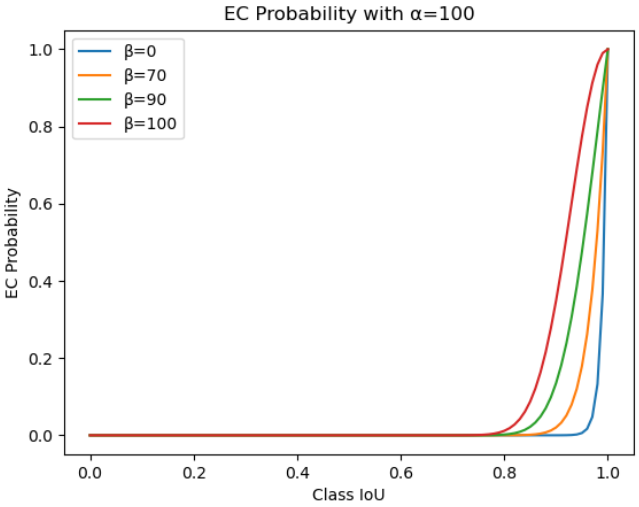

Figure 4.8: Graph of EC Probability with $\alpha = 100$ and various $\beta$ values

control over the behaviour of the AEC algorithm can be achieved.

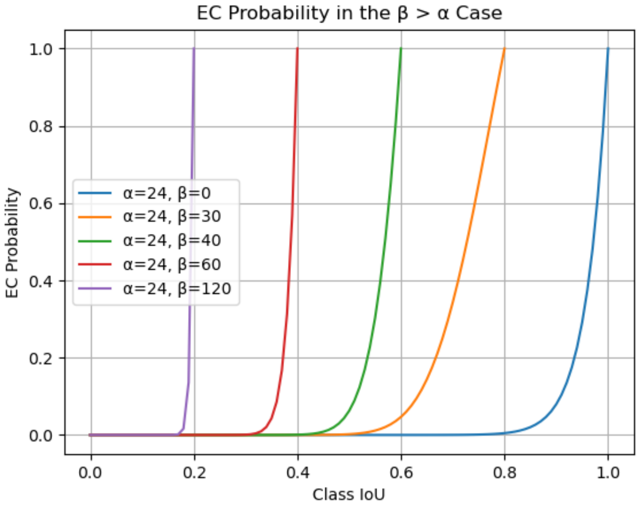

Figure 4.9: Graph of EC Probability in the $\beta > \alpha$ case

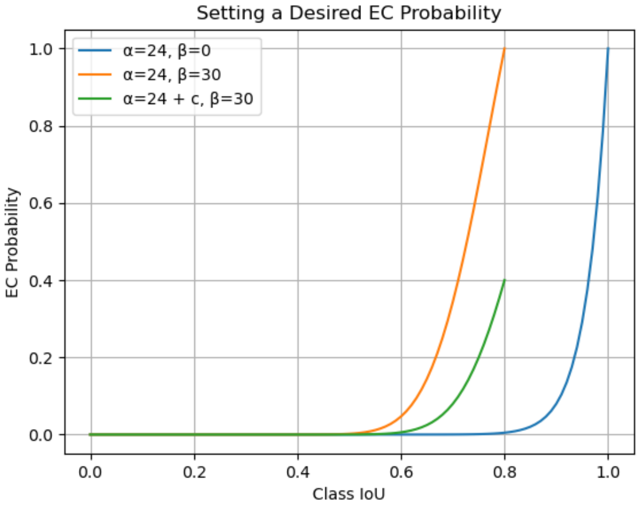

Figure 4.10: Setting a desired EC probability. In this case $c = \frac{log(0.4)}{log(0.8)}$ will set the EC probability to be 40% at cIoU value 80%

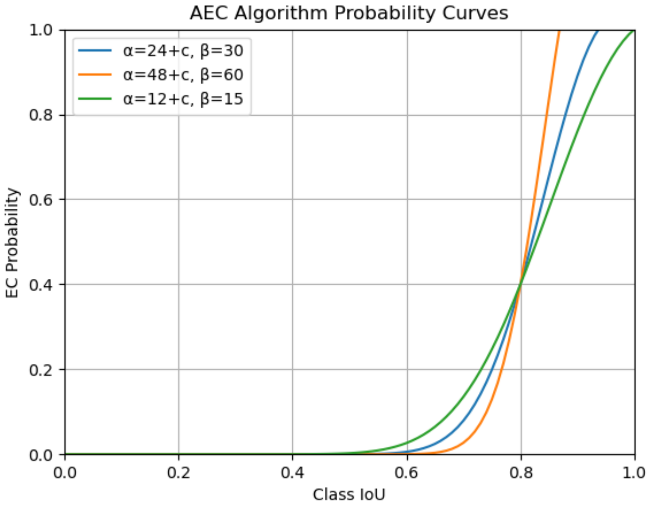

Figure 4.11: Changing Slope of EC Probability. In this case $c = \frac{log(0.4)}{log(0.8)}$ will set the EC probability to be 40% at cIoU value 80%

Table 4.6: Experiments on the AEC Algorithm with $\alpha$ and $\beta$

| # | Background | $\alpha$ | $\beta$ | Threshold | Remove | Interval |
|---|---|---|---|---|---|---|
| 16 | Farmland | 4 | 0 | 3000 | F | 200 |
| 17 | Farmland | 5 | 0 | 3000 | F | 200 |
| 18 | Farmland | 5 | 3 | 3000 | F | 200 |
| 19 | Farmland | 7 | 5 | 3000 | F | 200 |

Table 4.6 lists the experiments testing different values of $\alpha$ and $\beta$. Experiments 16 and 17 provide a baseline to compare with results of experiments 18 and 19. As there are an infinite number of possibilities for values of $\alpha$ and $\beta$, a deep exploration of $\alpha$ and $\beta$ is beyond the scope of this study. The values of $\alpha$ and $\beta$ for experiments 18 and 19 are picked for the similarity in the EC curve in experiments 16 and 17 but with some additional variations. Figure 4.12 shows the AEC update probabilities for experiments in Table 4.6.

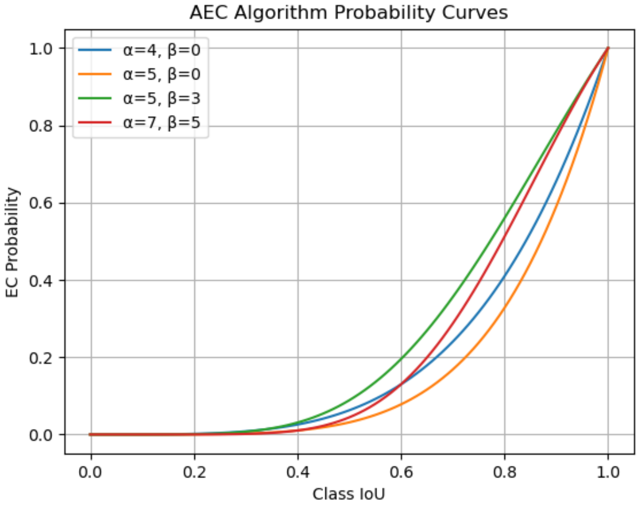

Figure 4.12: Graph of EC Probability for Experiments
erodes

By looking at model performance without the AEC algorithm, a strategy to pick good alpha and beta can be formed. Given the asymptotic peak model performance, the goal of the AEC algorithm is to improve model performance past this peak. One strategy is to set $\alpha$ and $\beta$ to allow for subjectively non-trivial EC probability near the peak cIoU performance without AEC of the model. This strategy allows the model to begin making non-trivial EC corrections once it nears its natural peak performance without AEC. Through this strategy, noise introduced through EC is reduced, EC is matched with model performance, and good $\alpha$ and $\beta$ can be selected. The $\alpha$ and $\beta$ selected in experiments in Table 4.6 are selected using this strategy while considering the peak cIoU performance of roughly 42% of Experiment 1 in Table 4.2.

## 4.4 Results and Discussion

Table 4.7 contains the best validation set results for all experiments. The results will be discussed in the sections below separately respective to the baseline model and each hyperparameter.

Table 4.7: Results for Experiments on the AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (F/T)(%) |
|---|---|---|---|---|---|---|---|---|
| 9* | 2 | 0 | 0 | F | 200 | 67.94 | 59.71 | 95.50/23.91 |
| 10 | 2 | 0 | 0 | F | 200 | 83.35 | 67.87 | 96.57/39.18 |
| 11 | 4 | 0 | 0 | F | 200 | 81.79 | 69.87 | 97.18/42.55 |
| 12 | 5 | 0 | 0 | F | 200 | 83.17 | 70.62 | 97.23/44.01 |
| 13 | 5 | 0 | 0 | T | 200 | 83.11 | 70.68 | 97.25/44.11 |
| 14 | 5 | 0 | 0 | T | On Best | 84.62 | 70.52 | 97.09/43.94 |
| 15** | 5 | 0 | 0 | T | On Best | 87.80 | 67.21 | 95.95/38.47 |
| 16 | 4 | 0 | 3000 | F | 200 | 81.57 | 69.92 | 97.21/42.63 |
| 17 | 5 | 0 | 3000 | F | 200 | 81.28 | 70.39 | 97.34/43.45 |
| 18 | 5 | 3 | 3000 | F | 200 | 83.38 | 69.79 | 97.03/42.54 |
| 19 | 7 | 5 | 3000 | F | 200 | 81.62 | 70.27 | 97.28/43.26 |

* This Experiment has a background forest. All other experiments have farm backgrounds
**Experiment 15 uses the final updated ground-truth from Experiment 14.

### 4.4.1 AEC Baseline

Table 4.8 highlights the results of the AEC baseline experiments. Table 4.9 shows the DeepLabV3+ baseline results without AEC integrated to compare with Table 4.8 Experiment results.

Table 4.8: Results for Experiments on the Baseline AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (B/O)**(%) |
|---|---|---|---|---|---|---|---|---|
| 9* | 2 | 0 | 0 | F | 200 | 67.94 | 59.71 | 95.50/23.91 |
| 10 | 2 | 0 | 0 | F | 200 | 83.35 | 67.87 | 96.57/39.18 |
| 11 | 4 | 0 | 0 | F | 200 | 81.79 | 69.87 | 97.18/42.55 |
| 12 | 5 | 0 | 0 | F | 200 | 83.17 | 70.62 | 97.23/44.01 |

* This Experiment has background forest, all other experiments have farm backgrounds
** B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Table 4.9: Baseline Control Experiments and Results

| # | Background | Objects of Interest | Spectral Range | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|
| 1 | Farmland | Roads and Well-Pads | RGB | 76.75 | 70.29 | 97.69/42.89 |
| 5 | Forest | Roads and Well-Pads | RGB | 58.80 | 56.42 | 96.59/16.26 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

Comparing Experiments 9 and 5, the AEC algorithm shows considerable performance improvements over the baseline control experiment. When it comes to extracting land disturbances from forested areas, using the AEC algorithm improves the AA and mIoU score by 15.4% and 5.8% respectively over the baseline.

Comparing Experiments 1 to 10, 11, and 12, analysis can be provided into how the AEC affects extracting land disturbances from farmland. Comparing Experiments 1 and

12, improvements are shown in land disturbance extraction in farmland with an increase of 8.3% and 0.5% in AA and mIoU scores respectively.

These results may be reflected in the fact that farmland areas have higher accuracy ground truth maps of land disturbances when compared to forested areas. If ground truth maps are more accurate in farmland compared to forest imagery, the performance increase of the AEC algorithm on land disturbance extraction will be smaller for farmland compared to forest imagery. Figure 4.14 shows two examples of the best error-corrected training labels for farmland and forest imagery.

Looking at Figure 4.14, noise and artifacts introduced by the AEC algorithm are visible. Additional heuristics or deep-learning models can be applied to each AEC update to reduce noise and artifacts based on meta properties. Through enforcing meta properties like straightness and connectedness through heuristics or other models, better labels can be produced by the AEC algorithms with less noise. Additional discussions in reducing noise and artifacts are provided in Section 5.2.3. Figure 4.13 provides an overview of an updated finite-state-machine with additional heuristics and deep-learning models to reduce label noise.

In addition to improving performance results, AEC also speeds up model convergence but with a caveat. Looking at Figure 4.15 and Figure 4.16, model performance in forest imagery with the AEC algorithm seems to improve linearly in some sections. Compared to the generally logarithmic growth of the baseline control model, implementing AEC seems to speed up learning to a linear pace for land disturbance extraction.

On the other hand, looking at Figure 4.17 and Figure 4.18 it is clear that AEC can derode not improve performance in some cases. Although $\alpha = 5$ has mIoU performance improvements over the control baseline in the farmland land disturbance extraction task,

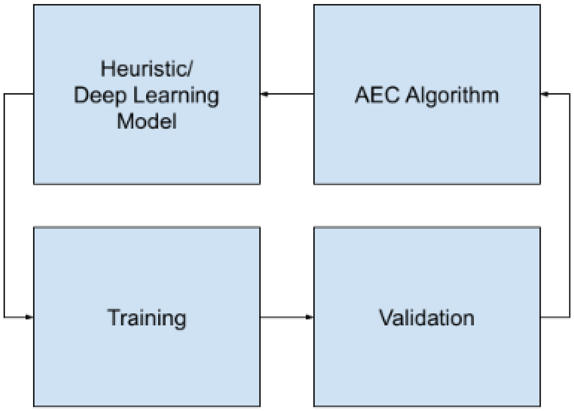

Figure 4.13: Additional Heuristics or DL Models in the Workflow

$\alpha = 2$ has divergent behaviour.

The caveat is that performance with AEC can quickly erode with a $\alpha$ that is too small. As the main assumption of the AEC algorithm is that correct generalizations can be leveraged to fix incorrect training data. This assumption may not be true for some $\alpha$ values that are too small. If the $\alpha$ allows AEC to make substantial updates before the model can establish correct generalizations, AEC will not be able to fix incorrect training data. This case can be seen in $\alpha = 2$ in Figure 4.17 and Figure 4.18. Increasing the value of $\alpha$ seems to stabilize performance from divergent to convergent. From Figure 4.5 it is observed that $\alpha$ and EC probability have negative correlation. As $\alpha$ increases, the probability of EC at a given mIoU value decreases. Looking at Figure 4.17 and Figure 4.18, as $\alpha$ increases the graph of convergence approaches the baseline control graph. Fewer changes are made to the training data as $\alpha$ increases such that the graph of convergence approaches the baseline control as $\alpha$ increases.

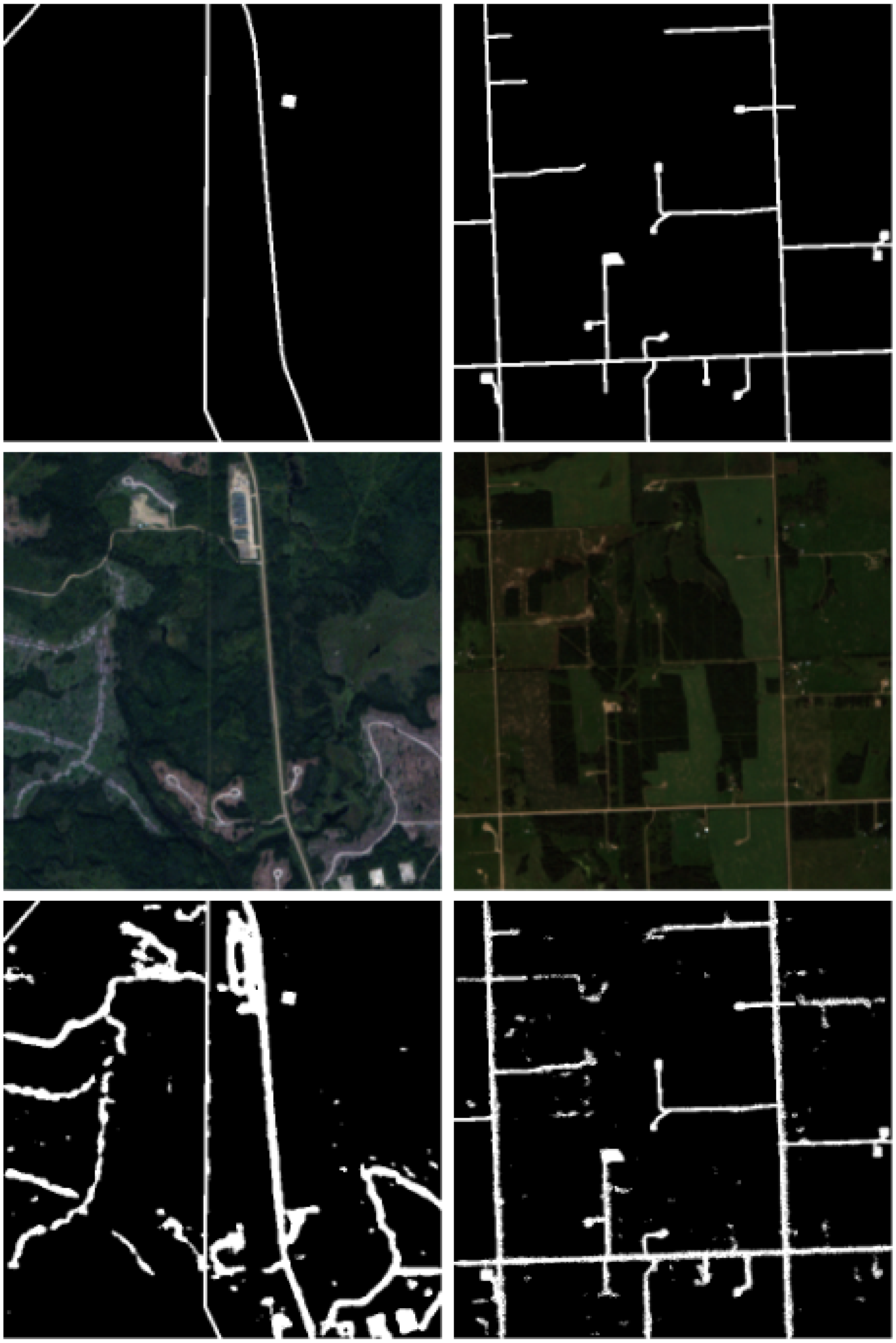

Figure 4.14: Example of Error Correction: Forest (left), and Farmland (right)

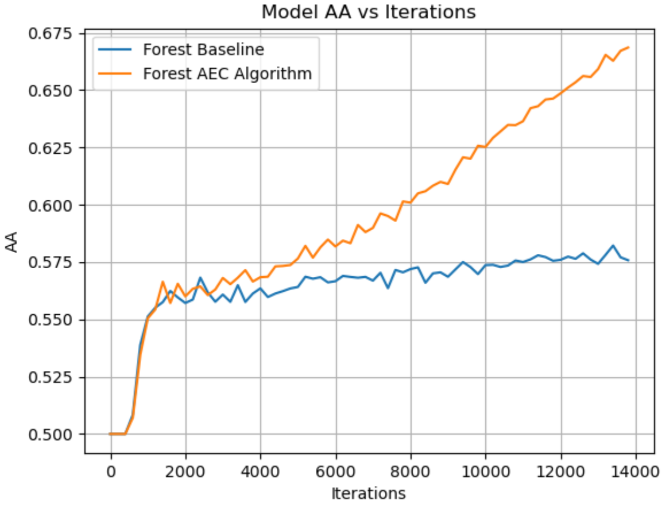

Figure 4.15: Graph of AA vs Iteration for Forest Control and AEC Baseline

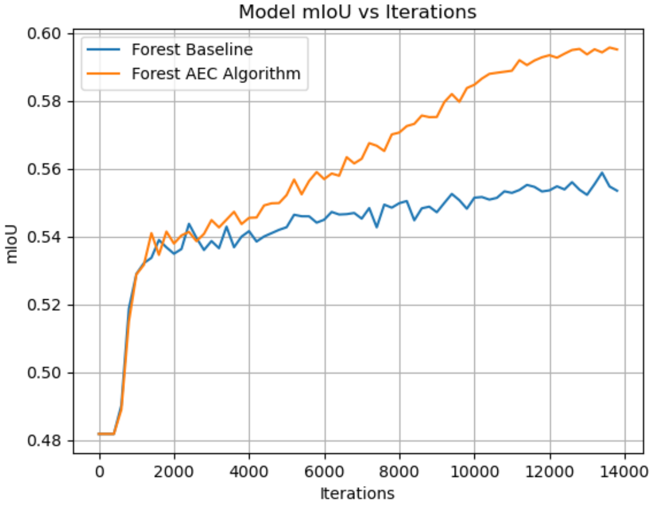

Figure 4.16: Graph of mIoU vs Iteration for Forest Control and AEC Baseline

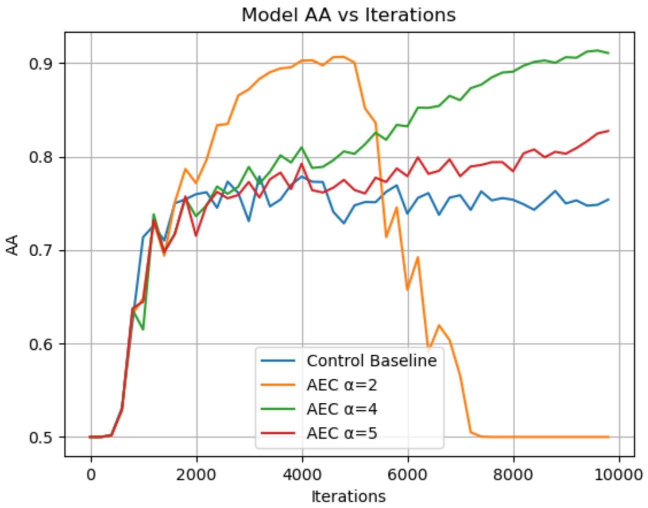

Figure 4.17: Graph of AA vs Iteration for Farm Control and AEC Baseline

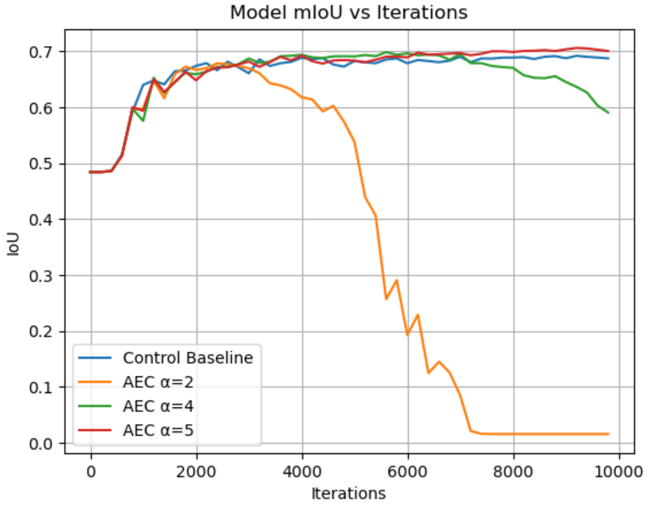

Figure 4.18: Graph of mIoU vs Iteration for Farm Control and AEC Baseline

### 4.4.2 Remove Parameter

Table 4.10 highlights results of the experiments on the remove parameter. Figure 4.19 shows the AA and mIoU graphs for the two experiments. Looking at the results of Experiments 12 and 13 in Table 4.10 and Figure 4.19, no substantial changes in performance are made in allowing the model to remove pixels.

As no substantial changes in performance are found in allowing AEC to remove pixels, this means that there are not many transitory artifacts being added to labels by the AEC algorithm. Since most updates by the AEC algorithm are non-transitory, the AEC algorithm is correcting errors in the training set successfully.

The remove parameter can be useful in cases where there are many transitory artifacts predicted by the model that disappear with convergence. Transitory artifacts could be the result of the model and data itself or also due to the hyperparameters of the AEC algorithm. In our experiments in Table 4.10, no real effect is made by allowing the AEC algorithm to also remove pixels.

Table 4.10: Results for Experiments on the Remove Parameter in the AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|---|---|
| 12 | 5 | 0 | 0 | F | 200 | 83.17 | 70.62 | 97.23/44.01 |
| 13 | 5 | 0 | 0 | T | 200 | 83.11 | 70.68 | 97.25/44.11 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

### 4.4.3 Interval Parameter

Table 4.11 highlights results of the experiments on the interval parameter. Figure 4.20 shows the AA and mIoU graphs for the experiments. Comparing Experiments 14 and

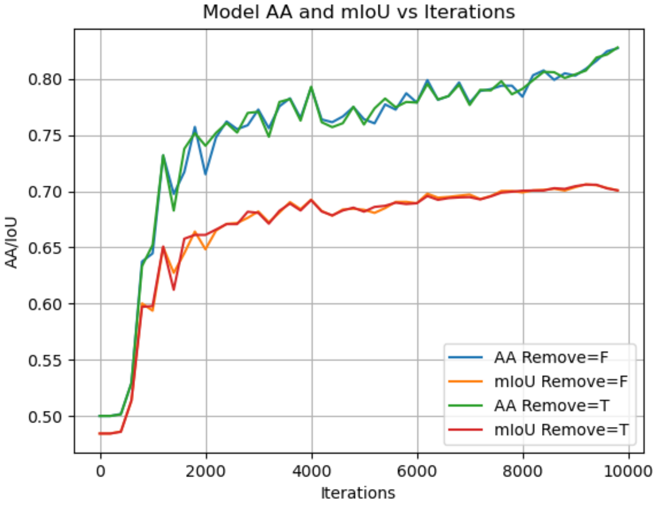

Figure 4.19: Graph of mIoU vs Iteration for Remove Parameter

15 with Experiment 13, the ”On Best” Experimentrequires fewer updates while achieving similar performance to baseline. Experiments 14 and 15 make respectively 18 and 9 number of updates compared to 50 updates made in Experiment 13.

Experiment 14 has higher AA and similar mIoU performance when compared to Experiment 13. As Experiment 14 makes only 36% of the number of updates Experiment 13 makes, this means that the AEC algorithm can improve performance by strategically selecting when updates are made. Comparing Experiments 15 with 14, mIoU scores quickly erode which can mean that the selected $\alpha$ value is too small. A $\alpha$ value that is too small can cause transitory artifacts to be updated to labels which can lower performance.

### 4.4.4 Threshold Parameter

Table 4.12 highlights the results of the experiments on the threshold parameter. Figure 4.21 shows the AA and mIoU graphs for the experiments. With thresholding, Experiments 16

Table 4.11: Results for Experiments on the Interval Parameter in the AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (B/O)**(%) |
|---|---|---|---|---|---|---|---|---|
| 13 | 5 | 0 | 0 | T | 200 | 83.11 | 70.68 | 97.25/44.11 |
| 14 | 5 | 0 | 0 | T | On Best | 84.62 | 70.52 | 97.09/43.94 |
| 15* | 5 | 0 | 0 | T | On Best | 88.04 | 67.56 | 96.03/39.08 |

*Experiment 15 uses the final updated ground-truth from Experiment 14. **B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

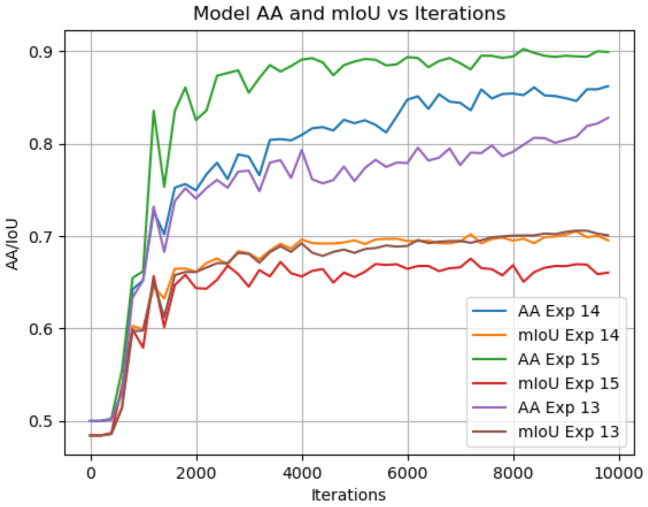

Figure 4.20: Graph of mIoU vs Iteration for Interval Parameter

and 17 make 35 total AEC updates compared to 50 updates in Experiments 11 and 12. Comparing performance, no substantial performance increase is gained by thresholding when AEC updates can occur. One interesting observation in Figure 4.21 can be made by comparing mIoU graphs between Experiments 16 and 11. The graph for mIoU erodes quicker for Experiments 11 than for 16. Noting that Experiment 16 has thresholding while Experiment 11 does not, fewer transitory artifacts may be added by the AEC algorithm in Experiment 16 compared to 11. Fewer transitory artifacts mean that performance becomes stable and is less likely to diverge which is reflective when comparing mIoU graphs Experiments 11 and 16.

Table 4.12: Results for Experiments on the Threshold Parameter in the AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|---|---|
| 11 | 4 | 0 | 0 | F | 200 | 81.79 | 69.87 | 97.18/42.55 |
| 12 | 5 | 0 | 0 | F | 200 | 83.17 | 70.62 | 97.23/44.01 |
| 16 | 4 | 0 | 3000 | F | 200 | 81.57 | 69.92 | 97.21/42.63 |
| 17 | 5 | 0 | 3000 | F | 200 | 81.28 | 70.39 | 97.34/43.45 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

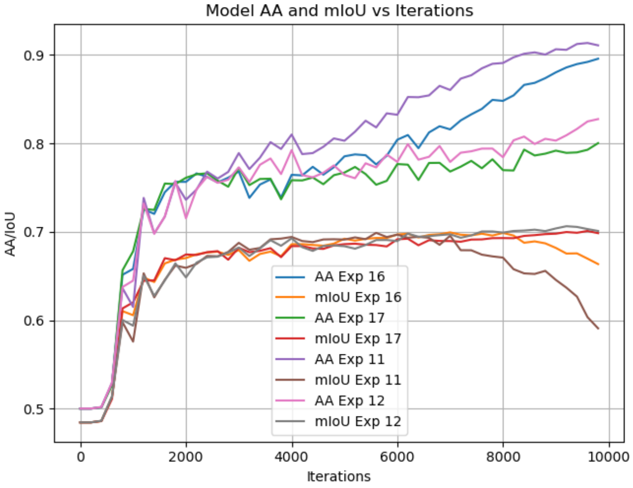

Figure 4.21: Graph of mIoU vs Iteration for Threshold Parameter

### 4.4.5 Parameters $\alpha$ and $\beta$

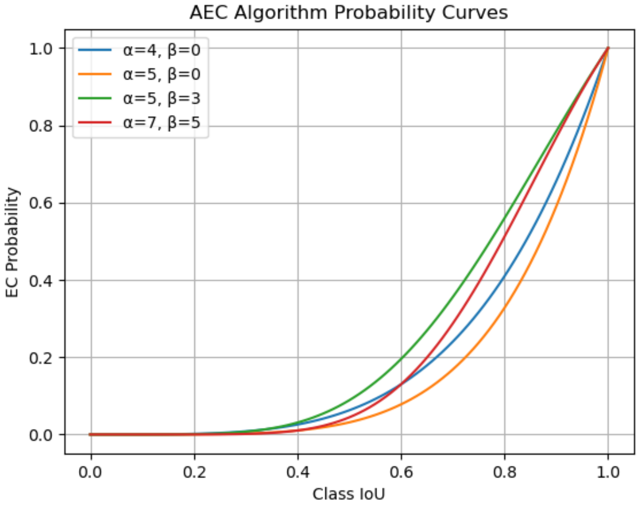

Figure 4.22: Graph of EC Probability for Experiments

Table 4.13 highlights results of the experiments on different values of $\alpha$ and $\beta$. Figure 4.23 shows the AA and mIoU graphs for the experiments. Looking at Table 4.13, mIoU performance ranks the experiments in order Experiment 17, 19, 16, 18 from best to worst. As cIoU for the object of interest peaks at 45% for all experiments in Table 4.13, the EC probability of the AEC algorithm is relevant up to the 45% cIoU in Figure 4.22.

As the graphs in Figure 4.22 are relevant up to 45% cIoU, ranking them in terms of EC probability at 45% cIoU yields Experiment 17, 19, 16, 18 from lowest to highest. Both EC probability at 45% and mIoU performance rankings match up. Looking further at experiments in Table 4.13, experiments 17/19 have a large performance gap to experiments 16/18. Small performance gaps exist between Experiments 17 and 19 and between Experiments 16 and 18. Comparing these performance gaps to Figure 4.22 and EC probability, all these gaps are mirrored. In Figure 4.22 Experiments 17 and 19 have a large EC probability gap to Experiments 16 and 18. Similarly, Figure 4.22 shows a smaller EC probability gap between Experiments 17 and 19 and between Experiments 16 and 18. Through these experiments, a strong correlation can be formed between AEC EC probability and model performance.

Table 4.13: Results for Experiments on $\alpha$ and $\beta$ Parameters of the AEC Algorithm

| # | $\alpha$ | $\beta$ | Thresh | Remove | Interval | AA(%) | mIoU(%) | cIoU (B/O)*(%) |
|---|---|---|---|---|---|---|---|---|
| 16 | 4 | 0 | 3000 | F | 200 | 81.57 | 69.92 | 97.21/42.63 |
| 17 | 5 | 0 | 3000 | F | 200 | 81.28 | 70.39 | 97.34/43.45 |
| 18 | 5 | 3 | 3000 | F | 200 | 83.38 | 69.79 | 97.03/42.54 |
| 19 | 7 | 5 | 3000 | F | 200 | 81.62 | 70.27 | 97.28/43.26 |

*B/O - B stands for Background class IoU, and O stands for class IoU of the object(s) of interest

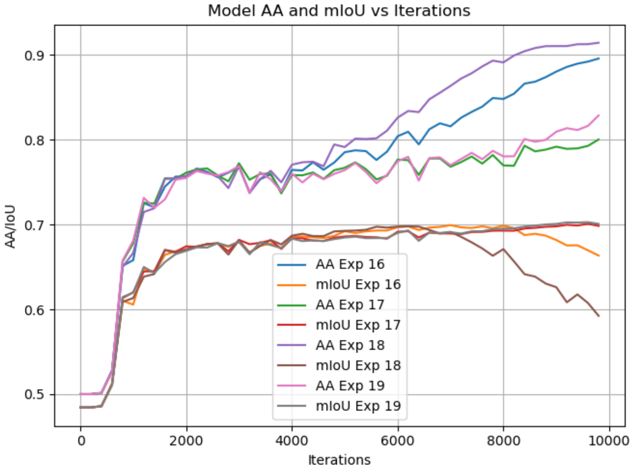

Figure 4.23: Graph of mIoU vs Iteration for Parameters $\alpha$ and $\beta$

## 4.5 Chapter Summary

Upkeeping accurate and up-to-date maps can be an expensive manual process. Relying on crowdsourcing, OpenStreetMaps and others have been successful in creating maps for urban and interurban paved roads (Barrington-Leigh and Millard-Ball, 2017). Even so, urban and interurban maps are subjective and contain errors with recent work showing 14% discrepancy between OpenStreetMaps and TorontoCity datasets (Máttyus et al., 2017). Beyond the need for urban mapping, there is a necessity for rural unpaved road-mapping given resource exploitation and development. With increasing development, creating maps of land disturbance and infrastructure for monitoring, tracking, assessment, and eventual reclamation becomes essential (Antoniuk et al., 2009; Jordaan et al., 2009; Zhang et al., 2017; Rokosh et al., 2012; ERCB, 2012). Compared to urban areas, maps of rural areas are error-prone as resource development can be volatile and span large geographical areas (Laurence and Balmford, 2013). The Alberta oil sands study area is a rural region critically in need of infrastructure mapping.

Combined with high-resolution satellite imagery, DeepLabV3+ is a powerful CNN method to extract oil and gas development land disturbances. On the other hand, given volatile or error-prone data, CNNs can learn to make incorrect generalizations that sway them to make erroneous predictions. Through this study, an automatic error-correcting algorithm (AEC) is proposed that corrects errors during training for road and well-pad extraction from the study area. Experiments in this study show that the proposed AEC algorithm improves and speeds up performance over a baseline model not utilizing the algorithm. Furthermore, experiments in this study explore hyperparameters introduced by the AEC algorithm and how they affect performance.

Several important conclusions can be drawn on AEC hyperparameters from this study. First, $\alpha$ and $\beta$ are important hyperparameters that control the performance of a model utilizing the AEC algorithm. Carefully selecting $\alpha$ and $\beta$ will stabilize and speed up the performance of a model using AEC and accurately correct errors. Second, with careful selection of $\alpha$ and $\beta$, the AEC algorithm will not add substantial transitory artifacts. As there are few transitory artifacts with careful selection of $\alpha$ and $\beta$, allowing the AEC algorithm to remove added pixels will not greatly improve performance. Third, the AEC algorithm needs only to make a few updates to labels to improve performance. Fourth, thresholding when the AEC algorithm runs can reduce the number of transitory artifacts without altering $\alpha$ and $\beta$ and stabilize training. In conclusion, the study results show the efficacy and effectiveness of the proposed AEC algorithm in extracting land disturbances in the study area.

# Chapter 5

# Conclusions and Recommendation for Future Research

## 5.1 Thesis Conclusions

With increasing mining development, land-disturbance extraction and mapping are critical for tracking, managing, and assessing environmental impacts. With the Alberta oil sands being the third-largest oil reserve in the world, it is crucial to create strategies and algorithms to map oil sands development. Manual mapping is an expensive and time-consuming process, requiring periodical updates. Given that oil sands development spans large geographical areas, manual labelling and extraction of oil sands road and well-pad land disturbances become uneconomical.

With the advent of high-resolution satellite imagery, an alternative solution to the manual mapping of land extraction with ML is possible. DL, a subset of ML, provides algorithmic solutions through CNNs to solve complex computer vision tasks. Two problems need to be solved for efficient extraction of land disturbances in the oil sands study area. The first problem is that creating accurate land disturbance maps, in general, is a challenging DL computer vision problem. Through this study, an effective methodology for road and well-pad extraction utilizing CNNs is proposed addressing the first problem. Three main axes of configuration for the methodology are explored: the objects of interest, the background image, and spectral information. Several important conclusions are drawn from addressing this first problem. First, integrating road and well-pad extraction as one task improves model performance over completing them as two separate tasks. Second, land disturbance extraction should be separated between farmland and forest backgrounds. Combining forest and farmland datasets erodes land disturbance extraction performance. Third, RGB spectral bands outperform using raw NDVI or a clipped NDVI spectral range.

The second problem arises from the fact that CNN relies on error-free data and that rural area maps can be prone to error given a lack of routine mapping. Through this study, a change detection algorithm (AEC) integrated into DL is proposed and tested that automatically fixes errors in the data. Experiments in this study show that the proposed AEC algorithm improves AA and mIoU by $8.3\% - 15.4\%$ and $0.5\% - 5.8\%$ respectively and speeds up performance over baseline models. Experiments in this study also explore hyperparameters of AEC and how they affect performance. Several important conclusions are drawn from addressing this second problem. First alpha and beta are important AEC hyperparameters that influence the performance of a model using the AEC algorithm. Second, with careful selection of alpha and beta, the AEC algorithm will not add substantial transitory artifacts. Third, the AEC algorithm only needs a few update iterations to labels to improve performance. Fourth, a threshold before the AEC algorithm runs can reduce the number of transitory artifacts without altering alpha or beta.

Through addressing the two problems of land disturbance extraction using DL, a powerful and adaptable alternative to manual mapping is proposed. Although further research can be conducted, the proposed methodology with AEC in this study can be used to effectively extract and update land disturbance maps for oil and gas development in the Alberta oil sands study area. The proposed methodology and AEC algorithm in this study is a powerful tool for tracking, managing, and assessing the environmental impacts of oil and gas development. The methodology explored in this study can have applications widespread in computer vision.

## 5.2 Recommendations for Future Research

Given the limitations of this study, there are a few future research recommendations. In this section, future recommendations are listed along with a short discussion.

### 5.2.1 Larger Dataset and K-Fold Cross Validation

For a more comprehensive study, a larger dataset can be collected, labelled, and utilized in the same experiments to reverify results. A greater number of experiments can also be conducted covering different methodology combinations. Due to time and resource constraints, only a select few exploratory experiments are conducted in this study.

Utilizing cross-validation will produce results with lower bias. Cross-validation will reduce sampling bias as the training and validation dataset are shuffled. Using cross-validation, a fairer result and performance are generated as all cross-validation results are averaged.

### 5.2.2 Channels and Hyperparameters

As RapidEye data consists of 5 spectral bands, there is merit to adapt the model to accept 5 channels as input. Additional spectral information with RGB can provide the model with more discriminatory information that can produce better extraction performance. Principal component analysis can also be conducted to determine the best spectral data to utilize. Exploring other more powerful backbones, compared to MobileNet, for

DeepLabV3+ like ResNet can improve performance. DeepLabV3+ hyperparameters can also be further optimized for this research.

Some strategies can be explored to improve the classification of extracted land disturbances. Adding additional classes in extraction can be used to classify roads apart from well-pads while also maintaining an integrated approach.

### 5.2.3 Additional Heuristics and Models to Improve Updates

Additional heuristics can be applied to AEC updates to improve update labels. Given meta-qualities of roads and well-pads such as connectedness or shape, a heuristic enforcing these meta-qualities can be used to improve AEC updates. Heuristics can be used to potentially remove transitory artifacts introduced by the AEC algorithm. Through improving AEC updates, performance may be further improved.

Building onto the idea of adding heuristics, another deep learning model can be used that operates in place of the heuristic. The deep learning model can be used to accomplish a meta-learning task in that it learns what labels should look like. This data used to train this meta-model is generated from dataset labels with different filters and transformations applied. Hypothetically, AEC can be applied to this meta-learning model based on the validation training results to improve performance. Through using this meta-learning framework, meta-properties such as the connectedness of roads and shapes of well-pads can be learned. This meta-learning approach is like specialized graph-based approaches for road extraction in that meta-properties are learned but can also be further generalized to other computer vision tasks. This meta-learning framework is a new approach to deep learning through leveraging error-correcting and meta-properties.

### 5.2.4 Other Computer Vision Tasks

A DL framework that allows for and corrects errors and bad examples in the data can be beneficial for many other computer vision tasks. Often, data used for DL is not perfect. Beyond just time and monetary constraints to perfection, perfection itself is often subjective in nature. As perfect data is often unattainable, DL frameworks should leverage imperfect data more efficiently and effectively. With the AEC algorithm, the dataset is improved quantitatively through a learning process with respect to the model performance. Broadly speaking, the AEC algorithm, starting with an imperfect dataset, can improve the quality of the data through training and achieve better performance. In this regard, the AEC algorithm can be applied to any computer vision task with imperfect data.

# References

Akosa, J., 2017. Predictive accuracy : A misleading performance measure for highly imbalanced data.

Antoniuk, T., Manuel, K., Sutherland, M., Bowen, J.T., 2009. In situ oil sands footprint monitoring project.

Ball, J.E., Anderson, D.T., Chan SR, C.S., 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community 11.

Barrington-Leigh, C., Millard-Ball, A., 2017. The world's user-generated road map is more than 80% complete. PLOS ONE 12, e0180698. doi:10.1371/journal.pone.0180698.

Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., DeWitt, D., 2018. Roadtracer: Automatic extraction of road networks from aerial images. arXiv:1802.03680.

Batra, A., Singh, S., Pang, G., Basu, S., Jawahar, C., Paluri, M., 2019a. Improved road connectivity by joint learning of orientation and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Batra, A., Singh, S., Pang, G., Basu, S., Jawahar, C., Paluri, M., 2019b. Improved road connectivity by joint learning of orientation and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934.

Cai, Z., Vasconcelos, N., 2017. Cascade r-cnn: Delving into high quality object detection. arXiv:1712.00726.

Chaurasia, A., Culurciello, E., 2017. Linknet: Exploiting encoder representations for efficient semantic segmentation. 2017 IEEE Visual Communications and Image Processing (VCIP) URL: http://dx.doi.org/10.1109/VCIP.2017.8305148, doi:10.1109/vcip.2017.8305148.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv:1802.02611.

Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Liu, M.Y., Torralba, A., Fidler, S., 2019. Neural turtle graphics for modeling city road layouts. arXiv:1910.02055.

ERCB, 2012. Regulating unconventional oils & gas in alberta. a discussion paper .

Erzurumlu, S.S., Erzurumlu, Y.O., 2015. Sustainable mining development with community using design thinking and multi-criteria decision analysis 46.

Girard, N., Charpiat, G., Tarabalka, Y., 2019. Noisy supervision for correcting misaligned cadaster maps without perfect ground truth data. arXiv:1903.06529.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2018. Mask r-cnn. arXiv:1703.06870.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. arXiv:1512.03385.

He, S., Bastani, F., Jagwani, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Elshrif, M.M., Madden, S., Sadeghi, A., 2020. Sat2graph: Road graph extraction through graph-tensor encoding. arXiv:2007.09547.

Jiao, X., Zhang, Y., Guindon, B., 2015. Synergistic use of radarsat-2 ultra fine and fine quad-pol data to map oilsands infrastructure land: Object-based approach. International Journal of Applied Earth Observation and Geoinformation 38. doi:10.1016/j.jag.2015.01.007.

Jordaan, S., Keith, D., Stelfox, B., 2009. Quantifying land use of oil sands production: A life cycle perspective. Environ. Res. Lett 4, 24004–15. doi:10.1088/1748-9326/4/2/024004.

Kailkhura, B., Gallagher, B., Kim, S., Hiszpanski, A., Han, T.Y.J., 2019. Reliable and explainable machine learning methods for accelerated material discovery. arXiv:1901.02717.

Katal, A., Wazid, M., Goudar, R.H., 2013. Big data: Issues, challenges, tools and good practices. 2013 Sixth International Conference on Contemporary Computing (IC3) , 404–409.

Kearney, S.P., Coops, N.C., Sethi, S., Stenhouse, G.B., 2020. Maintaining accurate, current, rural road network data: An extraction and updating routine using rapideye, participatory gis and deep learning. International Journal of Applied Earth Observation and Geoinformation 87, 102031. URL: https://www.sciencedirect.com/science/article/pii/S0303243419309882, doi:https://doi.org/10.1016/j.jag.2019.102031.

Laurence, W.F., Balmford, A., 2013. A global map for road building. URL: https://doi.org/10.1038/495308a, doi:10.1038/495308a.

Li, Z., Wegner, J.D., Lucchi, A., 2019. Topological map extraction from overhead images. arXiv:1812.01497.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2018. Focal loss for dense object detection. arXiv:1708.02002.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. Lecture Notes in Computer Science , 21–37URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2, doi:10.1007/978-3-319-46448-0_2.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. arXiv:1411.4038.

Máttyus, G., Luo, W., Urtasun, R., 2017. Deeproadmapper: Extracting road topology from aerial images, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3458–3466. doi:10.1109/ICCV.2017.372.

Prendes, C., Bujan, S., Ordonez, C., Canga, E., 2019. Large scale semi-automatic detection of forest roads from low density LiDAR data on steep terrain in Northern Spain. iForest - Biogeosciences and Forestry , 366–374URL: https://iforest.sisef.org/contents/?id=ifor2989-012, doi:10.3832/ifor2989-012, arXiv:https://iforest.sisef.org/pdf/?id=ifor2989-012.

Rahman, M., Wang, Y., 2016. Optimizing intersection-over-union in deep neural networks for image segmentation, pp. 234–244. doi:10.1007/978-3-319-50835-1_22.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. arXiv:1506.02640.

Redmon, J., Farhadi, A., 2016. Yolo9000: Better, faster, stronger. arXiv:1612.08242.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv:1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv:1506.01497.

Rokosh, C., Lyster, S., Anderson, S., Beaton, A., Berhane, H., Brazzoni, T., Chen, D., Cheng, Y., Mack, T., Pana, C., Pawlowicz, J., 2012. Summary of alberta's shale- and siltstone-hosted hydrocarbon resource potential .

Sessions, V., Valtorta, M., 2006. The effects of data quality on machine learning algorithms., pp. 485–498.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Sun, L., Tang, Y., Zhang, L.., 2017. Rural building detection in high-resolution imagery based on a two-stage cnn model 14.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv:1602.07261.

Vargas-Munoz, J.E., Srivastava, S., Tuia, D., Falcao, A.X., 2021. Openstreetmap: Challenges and opportunities in machine learning and remote sensing. IEEE Geoscience and Remote Sensing Magazine 9, 184–199. URL: http://dx.doi.org/10.1109/MGRS.2020.2994107, doi:10.1109/mgrs.2020.2994107.

Wegner, J., Montoya-Zegarra, J.A., Schindler, K., 2015. Road networks as collections of minimum cost paths. ISPRS Journal of Photogrammetry and Remote Sensing 108, 128–137. doi:10.1016/j.isprsjprs.2015.07.002.

Workman, R., Otto, A., Irving, A., 2016. The use of appropriate high-tech solutions for road network and condition analysis, with a focus on satellite imagery, in: Thame.

Yang, Z., Li, J., Zipper, C.E., Shen, Y., Miao, H., Donovan, P.F., 2018. Identification of the disturbance and trajectory types in mining areas using multitemporal remote sensing images. 644.

Yeh, C., Perez, A., Driscoll, A., Azzari, G., Tang, Z., Lobell, D., Ermon, S., Burke, M., 2020. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. Nature Communications 11. doi:10.1038/s41467-020-16185-w.

Zhang, Y., Guindon, B., Lantz, N., Shipman, T., Chao, D., Raymond, D., 2014. Quantification of anthropogenic and natural changes in oil sands mining infrastructure land based on rapideye and spot5. International Journal of Applied Earth Observation and Geoinformation 29, 31–43. URL: https://www.sciencedirect.com/science/article/pii/S0303243413001669, doi:https://doi.org/10.1016/j.jag.2013.11.013.

Zhang, Y., Lantz, N., Guindon, B., Jiao, X., 2017. Spectral-analysis-based extraction of land disturbances arising from oil and gas development in diverse landscapes. Journal of Applied Remote Sensing 11, 015026. doi:10.1117/1.JRS.11.015026.

Zhang, Z., Liu, Q., Wang, Y., 2018. Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters 15, 749–753. URL: http://dx.doi.org/10.1109/LGRS.2018.2802944, doi:10.1109/lgrs.2018.2802944.

Zhou, L., Zhang, C., Wu, M., 2018. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) , 192–1924.