UNIVERSITY OF CAPE TOWN

DEPARTMENT OF ELECTRICAL ENGINEERING

Reducing the Burden of Aerial Image Labelling

Through Human-in-the-Loop Machine Learning Methods

Author:
Muhammed T. Razzak

Supervisor:
Prof. Fred Nicolls

A dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Science in Electrical Engineering

July 2020

Abstract

This dissertation presents an introduction to human-in-the-loop deep learning methods for remote sensing applications. It is motivated by the need to decrease the time spent by volunteers on semantic segmentation of remote sensing imagery. We look at two human-in-the-loop approaches of speeding up the labelling of the remote sensing data: interactive segmentation and active learning. We develop these methods specifically in response to the needs of the disaster relief organisations who require accurately labelled maps of disaster-stricken regions quickly, in order to respond to the needs of the affected communities.

To begin, we survey the current approaches used within the field. We analyse the shortcomings of these models which include outputs ill-suited for uploading to mapping databases, and an inability to label new regions well, when the new regions differ from the regions trained on. The methods developed then look at addressing these shortcomings.

We first develop an interactive segmentation algorithm. Interactive segmentation aims to segment objects with a supervisory signal from a user to assist the model. Work within interactive segmentation has focused largely on segmenting one or few objects within an image. We make a few adaptions to allow an existing method to scale to remote sensing applications where there are tens of objects within a single image that needs to be segmented. We show a quantitative improvements of up to 18% in mean intersection over union, as well as qualitative improvements. The algorithm works well when labelling new regions, and the qualitative improvements show outputs more suitable for uploading to mapping databases.

We then investigate active learning in the context of remote sensing. Active learning looks at reducing the number of labelled samples required by a model to achieve an acceptable performance level. Within the context of deep learning, the utility of the various active learning strategies developed is uncertain, with conflicting results within the literature. We evaluate and compare a variety of sample acquisition strategies on the semantic segmentation tasks in scenarios relevant to disaster relief mapping. Our results show that all active learning strategies evaluated provide minimal performance increases over a simple random sample acquisition strategy. However, we present analysis of the results illustrating how the various strategies work and intuition of when certain active learning strategies might be preferred. This analysis could be used to inform future research.

We conclude by providing examples of the synergies of these two approaches, and indicate how this work, on reducing the burden of aerial image labelling for the disaster relief mapping community, can be further extended.

Acknowledgements

Firstly, thank you to Professor Fred Nicolls for being an amazing supervisor and mentor. His advice, flexibility and support made this dissertation possible.

Secondly, I would like to thank the members of the Mila lab. Dr Kris Sankaran and Professor Yoshua Bengio for advising on much of the project. I benefited enormously from their knowledge and insights. And all members and colleagues of Mila and UCT Digital Image Processing Lab, who provided a welcoming and fun research environment.

Thirdly, this dissertation was completed during a period of significant upheaval in my life. I am incredibly grateful to my amazing medical team, Dr David Epstein and Sr Karin Davidson.

Lastly, all my family and friends. Without their support, this undoubtedly would not be possible.

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This dissertation has been submitted to the Turnitin module and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signature: .....

Muhammed Taufeeq Razzak

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Remote sensing has experienced a wave of new interest in recent years, given the accessibility of data and the additional computational power to analyse the data. Remote sensing and aerial imagery originated from World War 1, when photos taken from aeroplanes were used for reconnaissance [Mnih, 2013]. Remote sensing data sources now include satellite imagery, multi-spectral satellite imagery and drone imagery. Remote sensing has found widespread use across industry, government and academia, where it is used for, among other applications, urban planning, crop management, disaster relief and climate modelling.

Much of the labelling work, from which utility arises, is done by human annotators through thousands of hours of careful annotation. For disaster relief applications, thousands of volunteers across the globe collaboratively produce maps from remote sensing imagery, annotating roads, building and many other map features. These enormous labelling efforts are required before disaster relief organisations, like Medicins sans Frontiers and the American Red Cross, are able to go into various regions to deliver much-needed aid. Human volunteers take time to map regions and the number of volunteer hours are excessive. Reducing the amount of volunteer effort required is of high priority to these disaster relief organisations, as is increasing the speed of service delivery.

In this study, we develop and examine two proposals for reducing the labelling cost: (1) human-in-the-loop labelling through interactive segmentation and (2) human-in-the-loop labelling through active learning. This dissertation focuses on two publicly-available datasets on the land-cover mapping and building footprint segmentation, both formulated as semantic segmentation problems.

1.1 Motivation and Background

The motivation for this project was largely driven by a collaboration between the American Red Cross, Missing Maps, Intel and Mila. These disaster relief organisations require accurate maps of disaster-stricken regions, in order to prepare and respond. Map features such as roads, buildings, bridges and water are critical for, among other reasons, allocating resources. Current approaches for obtaining maps require time-intensive manual labelling crowd-sourced from teams of human volunteers [Dittus et al., 2017].

This collaboration largely focused on exploring how machine learning could assist the labelling workflow. After in-depth analysis of the workflow, discussions with various stakeholders in different roles and a first attempt at the problem we found a number of constraints in deploying machine learning systems to this application. During the course of trying to address this, a few research questions arose. We believe these questions, while more engineering in nature, are useful to the broader community. Thus, this dissertation assists in answering these questions, the overarching one being how to reduce the amount of time humans spend labelling aerial imagery.

Beyond the applications of mapping data for disaster relief, mapping data is used in an extraordinary array of applications that is increasing each day. The methods developed could be useful and applicable to any of the new downstream applications.

1.2 Problem Definition

The overarching problem this dissertation looks at solving is reducing the amount of human effort required to produce accurate maps from remote sensing imagery. Currently, when a model is trained on one region and is tested or used on another region, its performance degrades and often substantially. Thus machine learning models are not often used due to issues with accuracy. If the models could adapt based on a few examples labelled quickly by volunteers, then that would reduce the burden on the volunteer community significantly.

We break this down into two problems that we approach to assist with the above:

1. Reducing the time taken to label an image;
2. Reducing the number of images that are required to be labelled to obtain an accurate model.

We suggest two possible solutions:

1. Interactive segmentation: helping humans produce segmentation maps with less effort from a human, through interacting with an algorithm.
2. Active learning: choosing training samples to label that would maximally benefit a model to improve its performance.

1.3 Scope and Limitations

This work is restricted to interactive segmentation and active learning in the semantic segmentation task for disaster relief. There is a large body of literature associated with remote sensing, labelling and generalisation, but we limit the scope of the dissertation to those two tasks.

The majority of the experimentation was conducted between February and October of 2019, with additional experiments conducted in 2020. Unfortunately, due to numerous factors, the write-up of this work was delayed multiple times until mid 2020. Thus there are limited references to work after 2019.

1.4 Contributions

This dissertation provides the following contributions:

1. Provides a framework and constraints for the problem of aerial image labelling within the context of disaster relief mapping.
2. Design of an interactive corrections algorithm for human-in-the-loop interactive segmentation for land-cover mapping and building segmentation.
3. Present a detailed comparison of some active learning algorithms on the semantic segmentation problem. We use two datasets that can serve as an additional test case for future work in active learning.

1.5 Outline

The rest of the dissertation is structured as follows:

1. Chapter 2 starts by presenting background around disaster relief mapping and remote sensing more generally. Following which, a pointed summary of semantic segmentation using fully convolutional networks is presented. The chapter then moves onto background on a few unsupervised representational learning methods and methods for visualising high dimensional data. The chapter ends by introducing the reader to the relevant literature in building footprint segmentation and land cover mapping. This is quite broad and non-specific, as specific mentions to the literature that are related to the specific work done is provided in the latter chapters.
2. Chapter 3 introduces the datasets that are used through the dissertation, and provides an analysis of them. The datasets are the Inria Aerial Image Labelling dataset (a building footprint segmentation dataset) and the Chesapeake Land Cover dataset (a land cover mapping dataset).
3. Chapter 4 looks at interactive segmentation and compares an iterative corrections algorithm developed, with two baselines. The iterative corrections algorithm outperforms the baselines on both datasets and works well in a variety of scenarios. It provides an up to 18% increase in mean intersection over union, the primary performance metric for semantic segmentation.
4. Chapter 5 looks at active learning, and compares a variety of sample acquisition strategies in contexts relevant to disaster relief mapping. The results show that all active learning strategies evaluated provide minimal performance increases over a simple random sample acquisition strategy. An analysis of the results does, however, provide some understanding of how the various strategies work and intuition of when certain active learning strategies might be preferred.
5. Chapter 6 concludes by providing some examples of the synergies of these approaches, and indicates how this work can be extended.

Chapter 2

Background

This chapter provides a brief introduction to disaster relief mapping and remote sensing, along with methods that automate the mapping process or assist in automating the mapping process.

Section 2.1 provides an introduction to disaster relief mapping and its importance. Analysis of the process of how maps are generated for disaster relief is presented. This analysis provides insight into how machine learning can assist in the disaster relief mapping process. Specifically, the analysis shows that given the need for validation, interactive segmentation and active learning are promising solutions.

Section 2.2 provides an introduction to remote sensing. This includes aspects such as spatial resolution, spectral resolution and other issues that arise from remote sensing data capture.

Section 2.3 provides an introduction to deep learning models for semantic segmentation. In particular, it focuses on the development of fully convolutional networks for semantic segmentation and the important components of training such networks.

Section 2.4 provides an introduction to three unsupervised deep learning methods: autoencoders, variational autoencoders and Tile2Vec. Autoencoders and variational autoencoders are widely used unsupervised learning techniques to learn compact representations of data, while Tile2Vec is an unsupervised learning technique to learn compact representations specifically for remote sensing imagery.

Section 2.5 presents background on two data visualisation methods: Principal Component Analysis and t-SNE. t-SNE is commonly used for visualising high dimensional data and is used within this dissertation.

Sections 2.6 and 2.7 present overarching literature of machine learning techniques applied to building footprint segmentation and land cover mapping. More literature specifically related to interactive segmentation and active learning are presented in their respective chapters.

2.1 Introduction to Disaster Relief Mapping

The world is constantly facing disasters— from natural ones like hurricanes and earthquakes to man-made ones like war zones— and people suffer in the aftermath of those disasters. Disaster relief organisations, like the Red Cross and Médecins Sans Frontières, step in to assist those affected by disasters. These organisations' assistance includes providing access to clean water, sanitation, food and pharmaceuticals and medical attention.

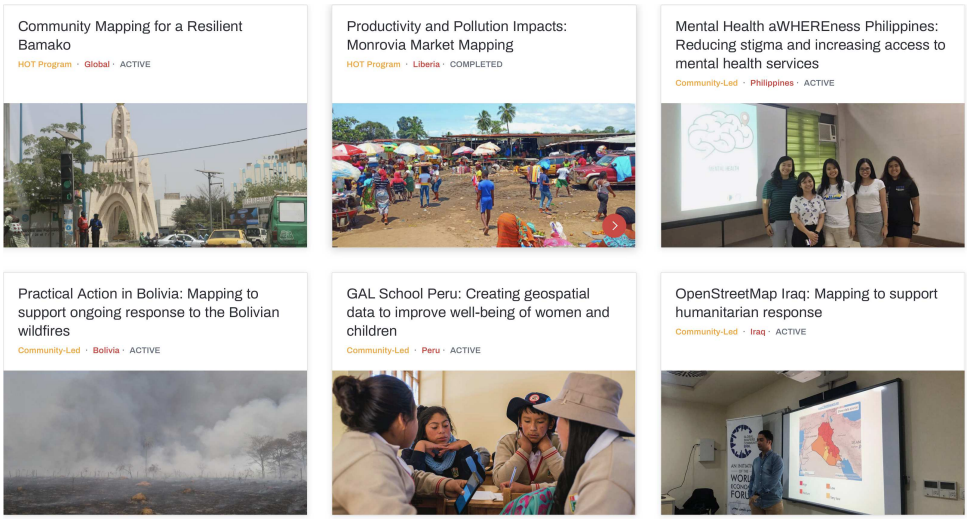


Figure 2.1: Examples of Humanitarian OpenStreetMap Team (HOT) Projects in early 2020 [Humanitarian OpenStreetMap Team, 2020].

A critical aspect of their missions is planning, hence the requirement for maps. To be able to respond quickly and appropriately to the disasters, they need accurate mapping data to reach affected communities. For many of the disaster areas there is no mapping data available or it is outdated. In many scenarios the maps have changed post-disaster (e.g. collapsed buildings and flooding) [Dittus et al., 2017].

Volunteer-driven communities like the Humanitarian OpenStreetMap Team (HOT) [Humanitarian OpenStreetMap Team, 2020] work together with the disaster relief organisations to rapidly map disaster-stricken areas. The data is made available via OpenStreetMaps (OSM) [OpenStreetMap Contributors, 2019].

HOT aims to create maps for disaster-stricken areas with the specific needs of disaster relief organisations in mind. Responses that HOT has assisted with include Cyclone Idai in 2019 and mapping for COVID19 in Botswana. HOT sets up these projects in consultation with their partner organisations. They map certain features (e.g. buildings, bridges and roads) in a particular region to assist with a particular disaster response in mind. For example, the Red Cross might require routes of entry to an area and so request roads to be mapped.

Mapping is difficult and time-consuming; the disaster relief areas are huge. As a consequence, even the large community of volunteers in HOT struggle to keep up with the demand. This significantly affects all disaster relief organisations' ability to service the communities affected by disasters [Dittus et al., 2017].

2.1.1 Process

The process for mapping disaster-stricken areas can be broken down into four steps: (1) project creation, (2) sub-project-creation and prioritisation, (3) mapping and (4) validation.

Step 1: Project Creation

The first step following a disaster is the creation of a project. This is done by the leadership of HOT and their partner disaster relief organisations. Together they defined regions of interest that need to be mapped. These are huge regions encompassing in excess of 100 square km^2. Often not much is known about these regions.

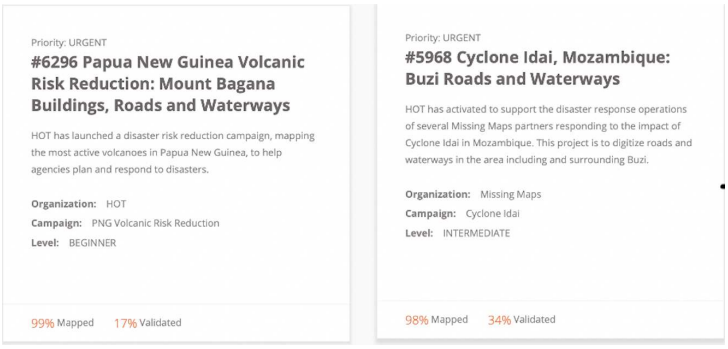


Figure 2.2: Sub-Projects and Priority [OpenStreetMap Contributors, 2019].

Step 2: Sub-Project Creation and Prioritisation

The next step is to divide the project into smaller sub-projects where necessary. For example, it could be defined by a region/city or it could be defined by the type of annotation required (e.g. road, building or water). Once this has been completed, the regions in each sub-project are set-up in a grid. Through the HOT tasking manager, the projects and sub-projects are distributed among mappers across the world, without introducing submission conflicts. Projects tend to focus on mapping specific features (e.g. buildings) in a specific region (e.g. area surrounding a village in Darfur). These regions are then divided into a set of tiles to further divide the work.

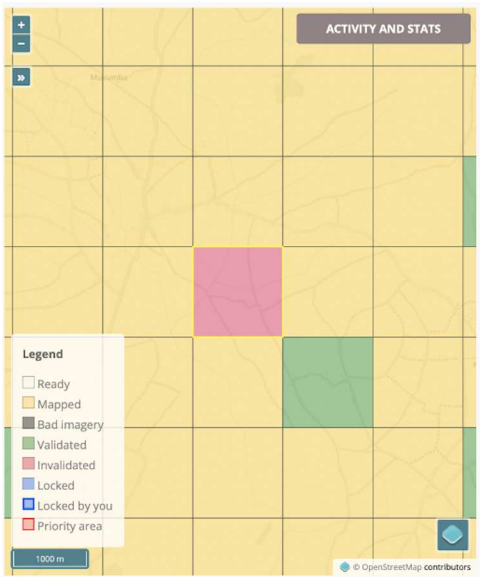


Figure 2.3: Sub-Projects and Priority [OpenStreetMap Contributors, 2019].

It is important to note that the grid of tiles are of a regular size. However, the tiles do not take a uniform amount of time to map. For example, if one were to map buildings there might be a tile with one 1 building while another tile has 1000 buildings. This load-balancing problem is well-known and some efforts have been made to address it using machine learning, which we detail in section 2.1.2.

Step 3: Mapping

The mapping then takes place. Volunteers map the required objects on OSM using map editing tools like iD Editor.

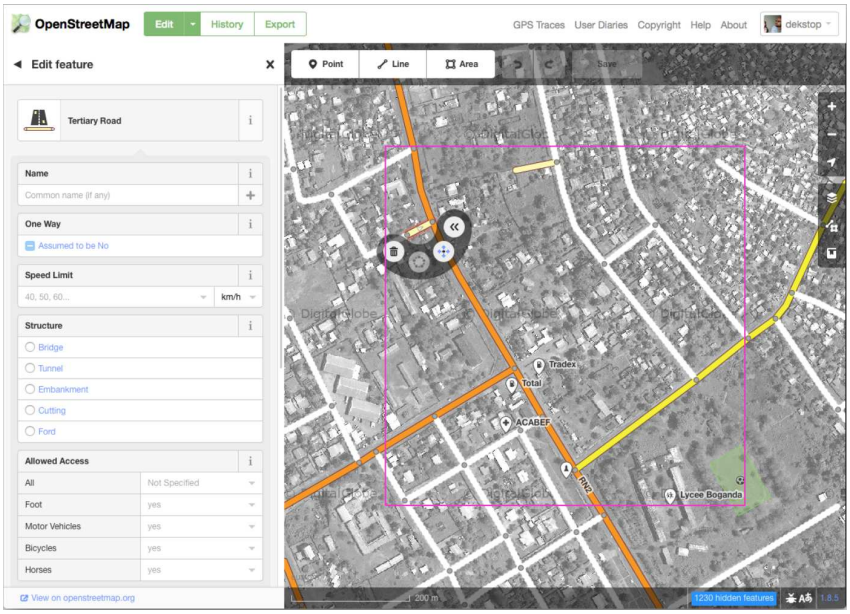


Figure 2.4: iD Editor [OpenStreetMap Contributors, 2019, Dittus et al., 2017].

In iD Editor, a mapper maps by drawing shapes (e.g. polygons, circles and lines) around the objects of interest. Once drawn mappers label the shapes appropriately (e.g. building or road)

Step 4: Validation

Lastly, there is a peer review process in place to validate the data. Only experienced mappers may review submissions. Beyond accepting or rejecting a submission, validators may also correct submissions and communicate with volunteers who made the submissions.

2.1.2 Efforts to Integrate Machine Learning in the Pipeline

There have been numerous attempts to integrate machine learning into the process to reduce the required effort, but applying models directly to the regions and uploading does not work. Machine learning models often get annotations wrong, and are very susceptible to errors when moving to different regions. Thus, direct uploads to mapping databases from machine learning models is not allowed— volunteers have to at the very least validate the output of the machine learning models. However, as noted there have been some recent efforts to integrate machine learning into the pipeline, and some of them have been successful.

Load Balancing & Prioritisation

The most successful and widespread use of machine learning in the disaster relief mapping pipeline was the effort in early 2019 to use machine learning models to generate a first pass. Multiple models are used to generate building or road predictions. These predictions are then used to provide estimates of project and sub-project complexity. The predictions are also used to coordinate the prioritisation of tiles to be mapped, when assigning tiles in the tasking manager to volunteers [Humanitarian OpenStreetMap Team, 2018].

Another way it is used is to provide an estimate of how much of a patch has been mapped. As it has been stated, even broken up tiles are still vast. They are often done in pieces. Machine learning can be used to estimate how much of tile has been mapped [Humanitarian OpenStreetMap Team, 2018].

Road Mapping

MapWithAI [MapWithAI Team, 2020] is a recent project born out of Facebook in late 2019, which released an interactive mapping tool for roads. They train models for different regions of the world (specific to each region), with the models producing a first-draft of a road polygon. Through their adapted iD Editor, RapiD, the user is then able to move nodes on the roads around to correct output. This is an interactive segmentation process, albeit without further machine learning adaptation after the first draft. Notably, every single road from the AI output will be added only after human validation. Mappers must manually select a road to OSM from the RapiD layer so every road will receive human attention. This is a requirement from HOT and OSM to ensure accuracy.

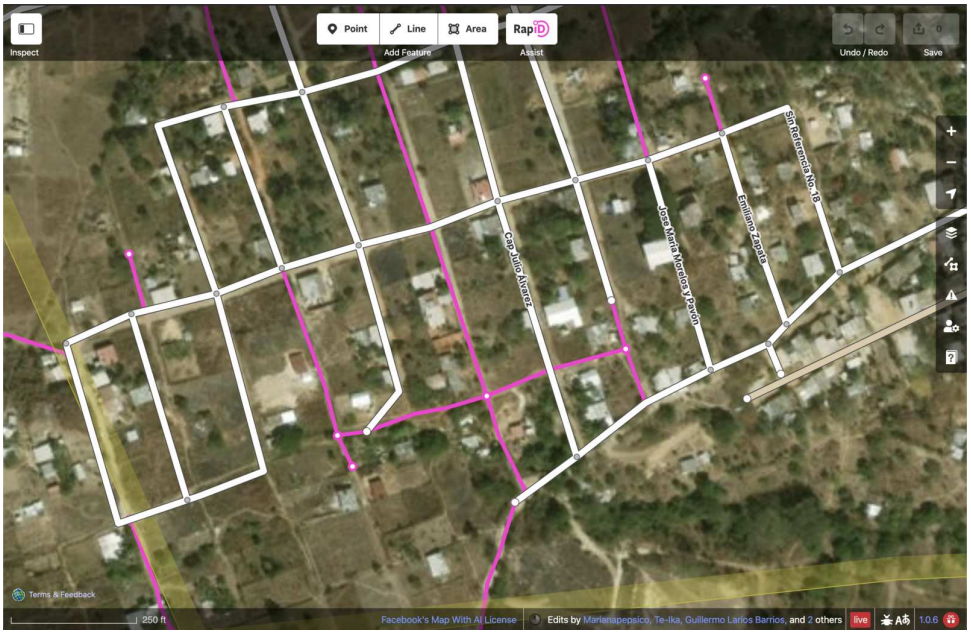


Figure 2.5: RapiD Map Editor from Facebook and MapwithAI [MapWithAI Team, 2020].

MapWithAI has been one of the biggest successful deployments of machine learning for disaster relief mapping.

Buildings with Bing Maps

In a trial in 2019, Microsoft through Bing Maps generated around 18 million building footprints in Uganda and Tanzania using recent deep learning models [Bing Maps Team, 2019]. They did this through a two-step process with semantic segmentation followed by polygonisation. Microsoft applied ResNet34-based RefineNet to produce building footprint segmentation masks. They then applied a noise reduction process on the masks produced, and finally applied a polygonisation algorithm to create building footprints for upload to OpenStreetMaps (and other databases).

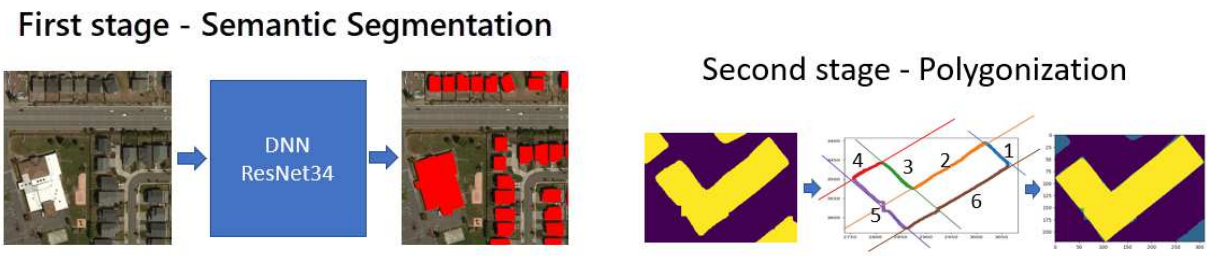


Figure 2.6: Microsoft's Bing Building Footprints Labelling Process in Uganda and Tanzania [Bing Maps Team, 2019].

They noted that these African regions presented additional challenges for them: the landscapes look different to United States or Canada, contain unique settlements and exhibit poor image quality and a paucity of training data in certain sub-regions. Subsequent targeted labelling efforts across the regions improved the model significantly [Bing Maps Team, 2019].

2.1.3 Limitations Holding Adoption of Machine Learning into the Pipeline

The primary reason there is reluctance to use machine learning in the mapping pipeline is accuracy. Models moving from region to region degrade in accuracy as the distribution changes. In addition, some of the outputs from segmentation models are poor.

Roads, which are one of the easier objects to segment as there is more regularity across regions, still require a model to be adapted to a new region to have anything close to acceptable accuracy. This is demonstrated by Facebook's MapWithAI, which only has the RapiD's AI first draft in regions in which it has a model trained for, and only with roads.

There are two interesting gaps in which machine learning could further assist:

1. The community will continue to use volunteers to ensure the maps are accurate and valid. Methods that integrate validators into the machine learning mapping process, like MapWithAI has, will have the most success. Furthermore, methods that make optimal use of the validators' time will be appreciated. Interactive segmentation does this. Developing an interactive segmentation model that deals directly with the problems of annotation at scale will be particularly helpful.
2. As we move to new regions, we would like to train a model with as few labelled samples as possible to achieve an acceptable accuracy. Active learning could be considered as a solution to this problem. If an efficient active learning solution to this problem can be found then less volunteer time would be required to label maps, as the models will be accurate enough to label the maps themselves.

2.2 Introduction to Remote Sensing

Remote sensing data has become ubiquitous and informative. It is used in a wide variety of applications ranging across geological sciences, climate change analysis, agricultural monitoring and disaster relief planning. This section aims to introduce to the reader to the remote sensing data acquisition methods relevant to this dissertation.

Many space agencies and other vendors of space imagery have adopted free or cost-effective access to remote sensing data. The European Space Agency has a free and open-data access policy for the data acquired from its Sentinel satellites.

The United States Department of Agriculture has a National Aerial Imagery Program (NAIP) [United States Department of Agriculture] that provides aerial imagery of the USA and makes it available to the wider public within a year of acquisition. The United States Geological Survey has provided open access to images from the Landsat Satellites for decades.

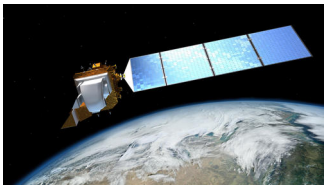


Figure 2.7: Landsat-8 Satellite [USGS].

For the purposes of this dissertation it is useful to know of the two primary considerations that affect the image quality with regards to the sensors' capabilities themselves: spatial resolution and spectral resolution.

Remote sensing imagery is acquired by sensors that record data from energy interacting with the Earth's surface. Remote sensing sensors are installed on satellites or aircrafts, and measure the electromagnetic radiation from reflection, emission, and emission reflection [University of Minnesota].

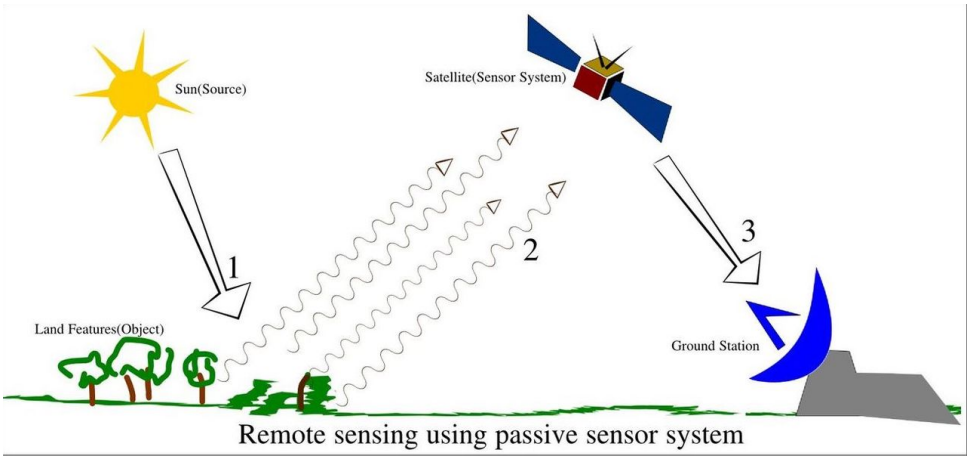


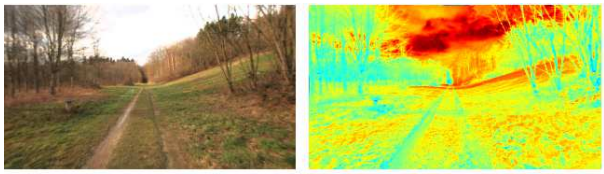
Figure 2.8: Remote Sensing with Passive Sensors Illustration [Wikimedia Commons].

For remote sensing of earth mapping data, we use passive sensors. These sensors use solar radiation to illuminate the Earth's surface and record the electromagnetic waves from the surface in the visible and near-infrared bands. This is illustrated in figure 2.8. More recently, hyper-spectral imaging sensors are being employed. These sensors measure more of the electromagnetic spectrum.

2.2.1 Spectral Resolution

The datasets used have images acquired from multi-spectral sensors (either RGB or RGB and Near Infrared (NIR)). Data from hyper-spectral sources are available and, as with algorithms using deep learning, more data allows for better features to be generated thereby producing better accuracy. Considering the datasets used, this discussion is restricted to RGB and NIR.

The visible spectrum consists of red (0.6 – 0.7 μm), green (0.5 – 0.6 μm) and blue (0.4 – 0.5 μm). The other band of interest is near-infrared (0.7 – 1.2 μm). Near-infrared is particularly useful at distinguishing green vegetation and water. Other infrared bands are also useful for a variety of vegetation applications like crop health. Figure 2.9 shows what can seen in the different bands [University of Minnesota].



(a) RGB **(b)** NIR

Figure 2.9: Illustration of what is seen through RGB and Near Infrared Sensors [Valada et al., 2020].

Water is absorbed by infrared wavelengths and is thus easily distinguishable. In addition, different plants also show variation in this band. The red band is important as it is the band in which chlorophyll absorbs energy. It can thus easily distinguish between vegetation and non-vegetation.

2.2.2 Spatial Resolution

Spatial resolution for remote sensing refers to the extent of the ground covered by a single pixel that is recorded in an image. High-resolution images, which are used in the datasets we use, have an extent of less than 1 meter per dimension of a pixel. Note that spatial resolution is given in meters.

2.2.3 Other Satellite Issues

Sensors often record data that affects the ability to use the imagery for our applications. Examples include atmospheric conditions, illumination geometry and time of day. Changes to these result in inconsistent data.

One significant issue is the nadir angle, which refers to the angle at which the image was captured with respect to the ground. A nadir angle of 90^o is generally what is required, but this often is not what is obtained. A process known as ortho-rectification attempts to correct for the nadir angle and terrain displacement.

Another issue when capturing aerial imagery are clouds and haze. The clouds and haze limit the utility of the data significantly. While methods are being developed to artificially remove clouds or allow models to work with clouds included, this is largely still primarily a research problem. Currently, the extent covered by clouds is simply recaptured without clouds.

2.2.4 Published Datasets versus Typical Workflow

Datasets published for the advancement of algorithms are relatively clean. However, errors in the data and ground truth labels are very common. Still, the cleanness of published datasets differ significantly from what is seen in industrial applications. When working with the American Red Cross, we often found tiles that exhibited cloud

occlusions, off-nadir captured images and other problems. This required going back go the image vendors for new images. Machine learning algorithms that can be robust to less than perfect data is important for industrial applications, and interactive segmentation is an example of a potential method in this respect.

2.3 Semantic Segmentation with Fully Convolution Networks

Semantic segmentation is the task of assigning each pixel in an image a semantically meaningful class. This has been a long-standing research problem within the computer vision community and is a difficult task. Deep learning provided a significant stride forward in the field, as with most computer vision tasks.

Initially, a patch-based approach using ConvNets was used [Ning et al., 2005, Farabet et al., 2013], followed by the introduction of fully convolutional methods [Long et al., 2015b]. Fully convolutional methods are still the best-performing models in the field, with recent innovations including skip connections, atrous convolutions and conditional random field post-processing [Chen et al., 2018a]. In subsection 2.3.1, we discuss these model architectures.

Other components of the neural network approach are discussed in subsections 2.3.2 (loss functions), 2.3.3 (optimisers) and 2.3.4 (weight initialisation). In subsection 2.3.5, we discuss the evaluation metrics used to measure the performance of semantic segmentation algorithms.

2.3.1 Model Architectures

Fully convolutional model architectures used for semantic segmentation generally consist of two components: an encoder and a decoder. The encoder downsamples the image while encoding high-dimensional feature representations of the data; the decoder then upsamples the image from the high-dimensional features to generate semantically meaningful class output per pixel.

This subsection provides a history of notable fully convolutional network model (FCN) architectures. The section includes the FCN introduced by Long et al. [2015b], U-Net, SegNet and a short summary of more recent model architectures.

FCN

The most effective basic deep learning based approach to semantic segmentation was developed by Long et al. [2015a]. They adapted an existing CNN architectures for pixel-wise semantic segmentation by removing the fully connected layers of the network. The spatial resolution at the end of this CNN (effectively the encoder) is significantly reduced due to strides of the convolution and pooling. Their initial FCN had a single upsampling layer at the end in order to produce segmentation maps to match the spatial resolution of the input image and ground-truths provided. This process provided very coarse segmentation outputs. To increase the spatial resolution they introduced skip connections from earlier blocks in the encoder with upsampling, which results in finer segmentation outputs. These architectures along with their resulting outputs are illustrated in figure 2.10.

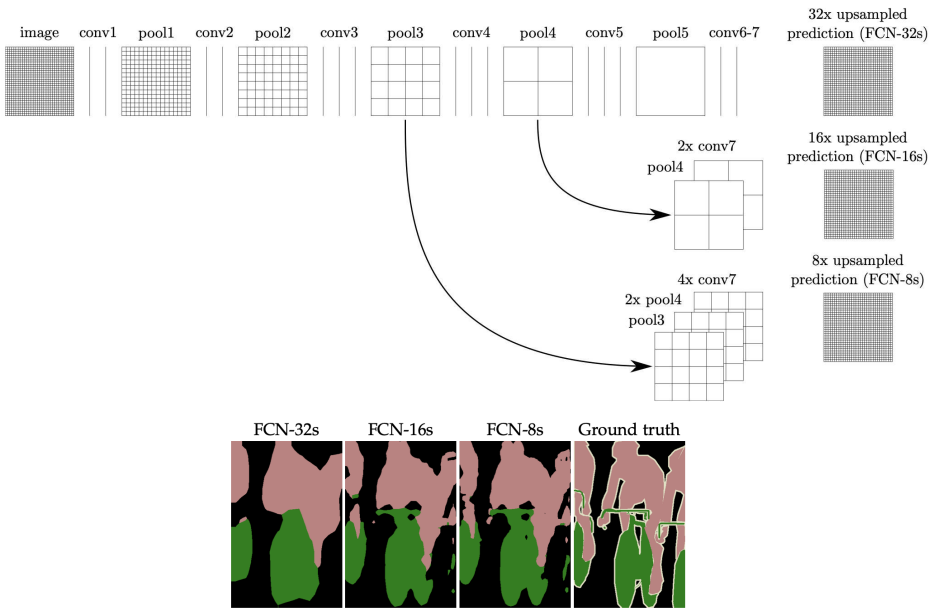


Figure 2.10: The FCN architectures and example outputs presented by Long et al. [2015a].

ResNet18-Based FCN

A popular semantic segmentation approach that works well is simply replacing the encoder in the FCN architecture with a more powerful encoder. ResNet18, 35, 50, 101 and 152 [He et al., 2016a] are all extensively used within the field. We chose to use ResNet18 as our backbone as it provides a balance between computational requirements and performance.

For a discussion on the ResNet backbone we refer the reader to the original ResNet paper by He et al. [2016a]. The paper provides a detailed discussion around the residual blocks. In summary, the residual blocks consist of convolutional layers, batch normalisation and rectified linear units along with, and most importantly, skip connections. The skip connections allow deeper networks to be trained, which results in increased accuracy. For information around the basic elements (like convolutional layers, batch normalisation and rectified linear units), we refer the reader to Goodfellow et al. [2016], which provides an excellent introduction.

A diagram of our ResNet18-based FCN architecture is shown in 2.11. In the ResNet backbone, the numbers next to the residual blocks indicates the number of filters in the convolutional layers. Once the image passes through the ResNet backbone, upsampling and transposed convolution layers are used similar to the original FCN implementation. We use this architecture, following Singh et al. [2019] who used it for land cover mapping.

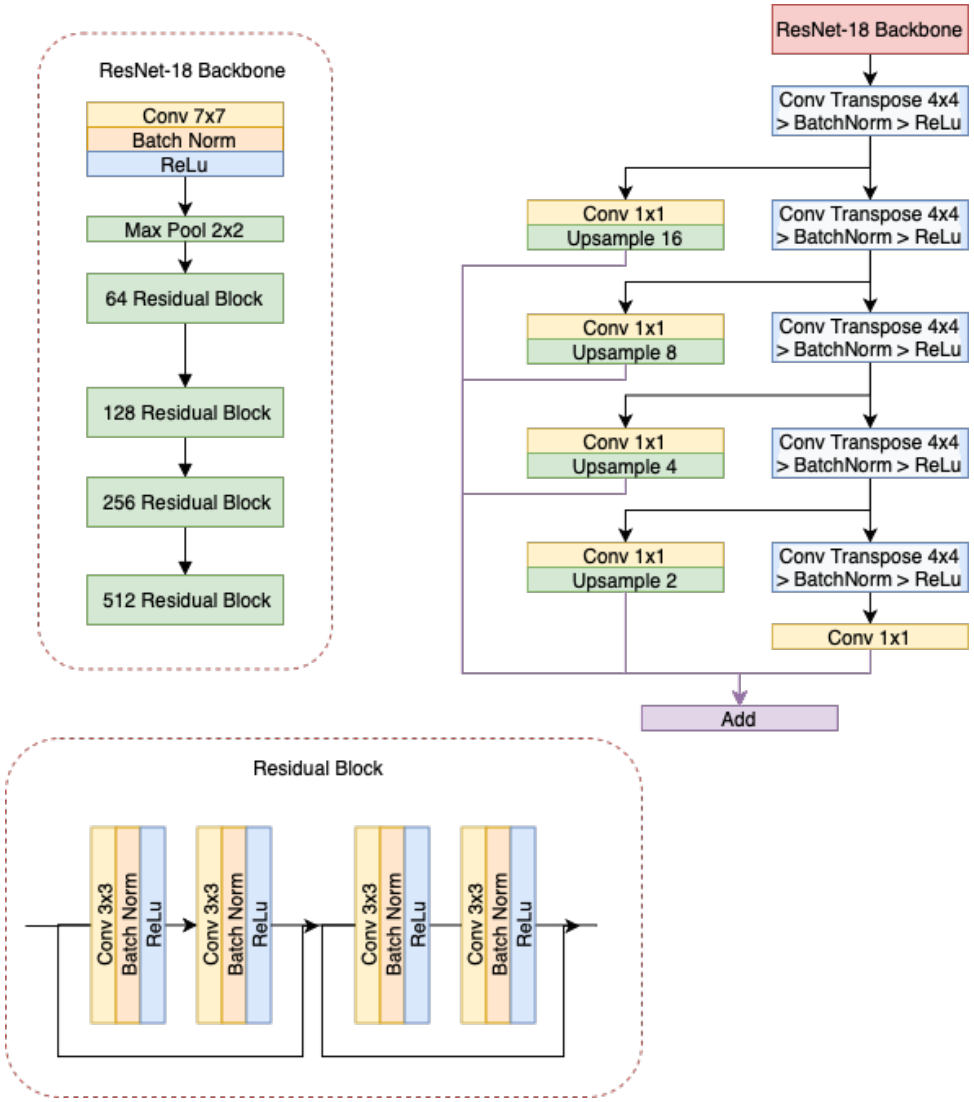


Figure 2.11: ResNet18-based FCN architecture.

U-Net

U-Net [Ronneberger et al., 2015] is a fully convolutional encoder-decoder network with skip connections. It was designed for use with medical images. The encoder reduces the spatial resolution but increases the feature richness. The decoder creates a high-resolution segmentation map from the low resolution, rich feature set created by the encoder. Regular skip connections (implemented as concatenation) were used to connect the encoder layers that precede the max pooling layers to the decoder. The architecture is illustrated in figure 2.12. U-Net, as opposed to the FCN architectures, uses longer skip connections. These skip connections retain more high resolution components thus allowing for better segmentation around the edges.

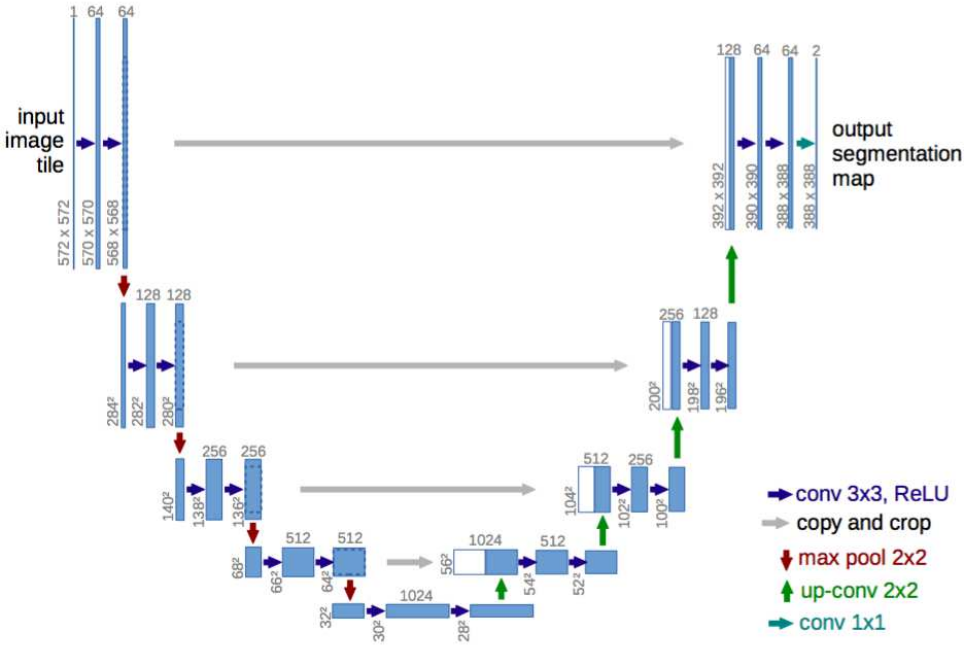


Figure 2.12: U-Net Architecture [Ronneberger et al., 2015].

One can make the model simpler and reduce memory and compute requirements by reducing the number of filters. We make use of such a simplification, in chapter 4, by dividing the number of filters in each section by 4 (i.e. 64 filters becomes 16 filters, and 512 filters becomes 128).

SegNet

SegNet [Badrinarayanan et al., 2017] introduced a fully convolutional encoder-decoder architecture for segmentation. Regular skip connections were introduced from the encoder component to the decoder component to aid the decoder in determining the fine-grain segmentation maps. However, these skip connections do not copy the feature maps from the encoder like U-Net. Instead, they copy the pooling indices from the decoder (from downsampling), and use these pooling indices for upsampling. This reduces the computational burden.

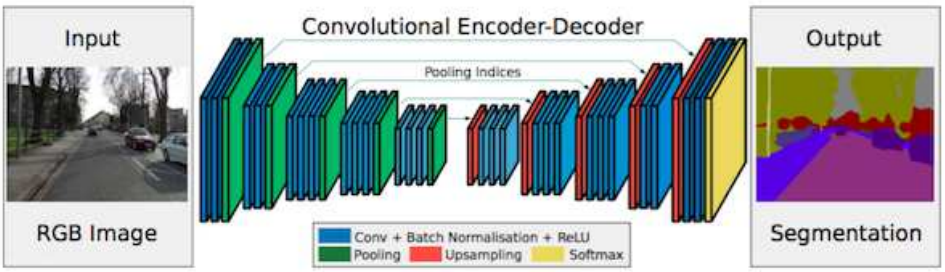


Figure 2.13: SegNet Architecture [Badrinarayanan et al., 2017].

Further Progress in the Field

FC-DenseNet [Jegou et al., 2017] took the ideas of these papers further by creating a deeper encoder, with a corresponding decoder. FC-DenseNet also made use of more skip connections.

DeepLab [Chen et al., 2017] improved fine-grain segmentation by using atrous convolution and fully connected conditional random fields. The atrous convolutions and CRFs help particularly with boundaries and edges in segmentation.

DeepLabv3 [Chen et al., 2018a] builds on DeepLab by specifically addressing the problem of segmenting objects at multiple scales. It uses an Atrous Spatial Pyramid Pooling module with image-level features for encoding global context. It also utilises multiple atrous blocks in parallel and cascade. All of this boosts performance.

Gated Shape CNNs for semantic segmentation [Takikawa et al., 2019] proposes using two-stream CNN architecture, in which shape information is processed in a separate stream. This emphasises more accurate boundaries in the segmentation and is further enforced by the proposed gated convolution layer.

For aerial imagery, where scale is relatively similar across images, ResNet-based FCNs and U-Net remain the most used networks and are the baselines in the field. Gated shape CNNs will undoubtedly work well for building footprint segmentation, but we view the methods we investigate as complementary to the improvements made in these recent advancements (i.e. one can simply substitute the better network in for some performance).

2.3.2 Loss Functions

The choice of loss function plays an important role in ensuring that the model converges and we get the best performance from the model. For semantic segmentation tasks there are two common loss functions used: cross entropy and soft-IoU.

Firstly, it is important to note that the final layer of our models always is the softmax activation function, which normalises the output values across the class dimension to between 0 and 1. Thus these values can be interpreted as probabilities.

Cross Entropy Loss

Let X be the set of all pixels in an image and C be the number of classes. The output of the softmax function for a pixel is a vector p^x, and p^x_c can be interpreted as the probability that pixel x is of a class c. The target is one hot encoded; in other words y^x_c is 1 if pixel x is of class c, otherwise y^x_c is 0. Thus, categorical cross entropy is calculated as follows:

$$L_{CCE} = -\sum_{x \in X} \sum_{c \in C} y^x_c \log(p^x_c). \tag{2.1}$$

Minimising cross entropy is equivalent to maximising the likelihood, and equivalent to minimising the Kullback–Leibler divergence. Cross entropy has a number of qualities that make it very popular to use, the primary one being that its gradient is easy to calculate and it propagates well through a network [Goodfellow et al., 2016].

Soft-IoU Loss

However, for class-imbalanced problems such as in segmentation, a soft-IoU based loss function is often used. Intersection-over-Union or IoU is also the main measure of performance for semantic segmentation. Using set notation, IoU is given as follows for a single class:

$$IoU = \frac{A \cap B}{A \cup B} \tag{2.2}$$

where A is the predicted mask of a class and the B is the ground-truth mask of a class. We provide further explanation of IoU as a performance metric, in subsection 2.3.5.

This IoU calculation is not differentiable. However, it is approximated to what is termed a soft-IoU loss:

$$L_{IoU} = 1 - IoU = 1 - \frac{\sum_{x \in X} \sum_{c \in C} (y_c^x p_c^x)}{\sum_{x \in X} \sum_{c \in C} (y_c^x + p_c^x - y_c^x p_c^x) + \epsilon}. \tag{2.3}$$

2.3.3 Optimisers

This subsection briefly introduces the optmisers used in the dissertation. We assume the reader is familiar with stochastic gradient descent (SGD), but will briefly mention the update process so we can compare it to newer techniques:

1. Evaluate the gradient of the loss function with respect to the weights at that point
2. Update the weights of the network by the calculated gradient multiplied by the learning rate.

This is done for each mini-batch. The non-vanilla versions of gradient descent simply change the way the second step— the parameter updates— is done [Goodfellow et al., 2016].

Adam [Kingma and Ba, 2015], a variant of gradient descent has become increasingly popular due its ease of use. Adam uses per-parameter adaptive learning rates. This method need less learning rate tuning due to its adaptive quality and does it for each parameter— which is intuitively a better way to do the updates. Adam computes an exponential moving average of the first moment (using the gradient) and the second moment (using the gradient squared) for each parameter. Adam attaches two hyper-parameters to these moving averages to control the decay: β_1 and β_2. In addition, Adam includes a bias correction mechanism to correct for a zero initialisation.

2.3.4 Weight Initialisation

Good weight initialisation is critical for getting a network to converge, and to converge to a high degree of accuracy. If the weights are too small then the gradient will vanish and if the weights are too large the gradient will explode. He et al. [2016b] developed an initialisation that particularly considers the rectified linear unit activation function that is commonly used in most networks,

$$Var(W) = \frac{2}{n_{in}}, \tag{2.4}$$

where $Var(W)$ is the variance of the weights and n_{in} is the number of input units. This is known as the He initialisation and is now the default on most deep learning frameworks. This initialisation allowed deeper networks to converge, and for most networks to converge more quickly.

2.3.5 Evaluation Metrics

There are two primary evaluation metrics for semantic segmentation. One is simply the accuracy and the second is the IoU.

Accuracy

Accuracy is simply the percentage of pixels in the image that were correctly classified. The pixel accuracy is reported for each class separately as well as globally across all classes. This metric often provides misleading results as it does not account for the significant class imbalance that is present in most imagery for segmentation tasks.

Accuracy can be calculated as follows using confusion matrix notation:

$$Accuracy = \frac{TP}{TP + TN + FP + FN}, \tag{2.5}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

Intersection over Union

Intersection over Union (IoU) or Jaccard Index is measure of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. This can again be evaluated per class and over all classes (mean IoU or mIoU). The mIoU is commonly reported as the value to measure against. It takes into account the class imbalance that is present and indicates how well the model performs across all classes.

IoU can be calculated as follows using either set or confusion matrix notation:

$$IoU = \frac{A \cap B}{A \cup B} = \frac{TP}{TP + FP + FN} \tag{2.6}$$

where A is the predicted mask of a class and the B is the ground-truth mask of a class.

The mIoU is simply the mean of the IoUs for each class.

$$mIoU = \frac{1}{C} \sum_{c}^{C} \frac{A_c \cap B_c}{A_c \cup B_c} \tag{2.7}$$

2.4 Unsupervised Representational Learning

We provide a brief overview of three methods of learning in an unsupervised manner for various purposes: an autoencoder, which we use as a pre-training mechanism in an active learning scenario; a variational autoencoder, which is used in an active learning acquisition strategy; and Tile2Vec, an unsupervised learning strategy which we also use for visualisation and active learning.

Beyond these methods, there have been a number of unsupervised learning approaches for aerial imagery. Singh et al. [2019] demonstrated the use of a standard autoencoder, context prediction & context encoders, Splitbrain autoencoders and their semantic in-painting coach network method on aerial imagery. They found that while their semantic in-painting coach network method worked best, the autoencoder method worked almost as well. We chose to use an autoencoder for our experiments due to its simplicity.

2.4.1 Autoencoder

An autoencoder is a neural network that is designed construct a compact representation of the input data. To learn this representation it has an encoder that compresses the input into a compact latent representation and a decoder that must learn to reconstruct the input from the compact latent representation [Hinton and Salakhutdinov, 2006, Weng, 2018].

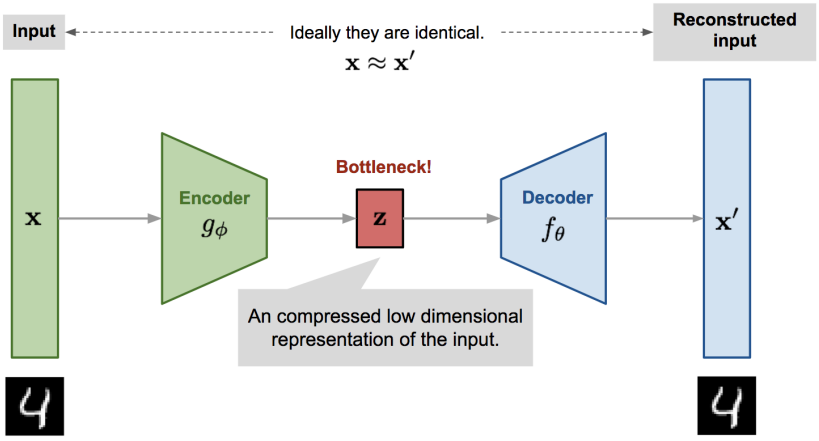


Figure 2.14: Diagram of an Autoencoder [Weng, 2018].

The model contains an encoder function g_ϕ, and a decoder function f_θ. A low-dimensional representation for input is learnt, z.

We learn the parameters of the encoder and decoder together, to output the reconstructed input. The autoencoder is trained using a simple mean square error loss (MSE) between the reconstructed input and the original input:

$$L_{MSE}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2 \tag{2.8}$$

The autoencoder was originally designed for data compression, learning good latent representations and reconstructing data. It has become common to use autoencoder trained networks for other downstream tasks. As we can see, the basic autoencoder structure is not dissimilar to FCNs for semantic segmentation. In fact the same ResNet18-based FCN, described earlier, can do auto-encoding. Singh et al. [2019] used an autoencoder as baseline for their self-supervised pre-training algorithm for aerial imagery. The same is done in our active learning chapter.

2.4.2 Variational Autoencoders (VAEs)

VAEs [Kingma and Welling, 2014] are similar to traditional autoencoders in some respects, but are very different in others. VAEs are generative models which aim to simulate how data is generated in the real world (as per the distribution of the training data provided). Instead of encoding the input to single latent variable, they encode it as a latent distribution (typically a Gaussian).

For a thorough explanation we direct the reader to the tutorial by Weng [2018]. The graphical model of a VAE and a VAE model with a multivariate Gaussian as the latent distribution is visualised in figure 2.15.

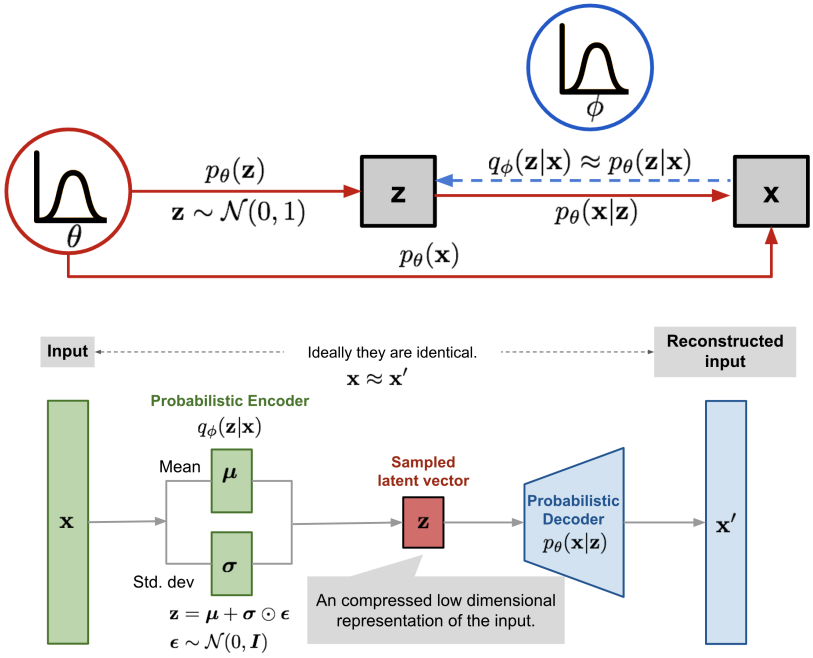


Figure 2.15: Graphical Model & Diagram of VAE with a multivariate Gaussian as the latent distribution [Weng, 2018].

We briefly summmarise from Weng [2018], to define some of the nomenclature:

1. a latent distribution is defined as p_θ parameterized by θ.
2. The conditional probability $p_\theta(\mathbf{x}|\mathbf{z})$ defines a generative model and is also known as probabilistic decoder.
3. The approximation function $q_\phi(\mathbf{z}|\mathbf{x})$ is the probabilistic encoder.
4. To obtain the parameters of the probabilistic encoder and decoder, we optimise the loss function given by:

$$L_{\text{VAE}}(\theta, \phi) = -\log p_\theta(\mathbf{x}) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) = -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})), \tag{2.9}$$

   where D_{KL} is the Kullbeck-Leibler divergence between the distributions.
5. A reparameterization trick is used to make the sampling of $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ deterministic and allow for the backpropogation of gradients [Kingma and Welling, 2013].

2.4.3 Tile2Vec

Tile2Vec [Jean et al., 2019] is another method of learning representations. It does this by using the distributional hypothesis: image patches that are close spatially should have similar representations while distant patches should have different representations. The distributional hypothesis has had an enormous impact on natural language processing with most state-of-the-art systems now using this idea for self-supervised pre-training on large corpus to learn better representations. In this case, we use Tile2Vec primarily as a method of visualisation as we found the distributional hypothesis had some unwanted effects with remote sensing data. We summarise the Tile2Vec method as proposed by Jean et al. [2019] in this subsection.

Unsupervised triplet loss

To learn a compact representation, z, Jean et al. [2019] train a CNN on a triplets of tiles. A triplet consists of an anchor tile t_a, a neighbour tile t_n, and a distant tile t_d. They then minimise the Euclidean distance between the feature embeddings, generated by the CNN, of the t_a and t_n, while maximising the distance between t_a and t_d.

Thus the triplet loss is given as:

$$L(t_a, t_n, t_d) = [||f_\theta(t_a) - f_\theta(t_n)||_2 - ||f_\theta(t_a) - f_\theta(t_d)||_2 + m]_+ \tag{2.10}$$

A rectifier with margin m was introduced into the loss function to prevent the distant tile from being pushed too far away in the embedding space. If, in the embedding space, the distance to the distant tile exceeds the distance to the neighbour tile plus the margin, then the loss is set to 0. This margin parameter is set to 10 in their implementation.

Jean et al. [2019] also found that by penalising the embeddings Euclidean norms, the network produces better representations and embeddings. Thus the objective function is given by:

$$\min_\theta \sum_{i=1}^{N} \left[L(t_a^{(i)}, t_n^{(i)}, t_d^{(i)}) + \lambda \left(||z_a^{(i)}||_2 + ||z_n^{(i)}||_2 + ||z_d^{(i)}||_2 \right) \right], \tag{2.11}$$

where λ is a parameter which determines the strength of the regularisation, and $z_a^{(i)} = f_\theta(t_a^{(i)})$, $z_n^{(i)} = f_\theta(t_n^{(i)})$ and $z_d^{(i)} = f_\theta(t_d^{(i)})$.

Triplet sampling

Jean et al. [2019] suggest using smaller tiles to improve representation and so suggest a sampling procedure with the following parameters:

- **Tile size:** In their implementation they chose a tile size of 100×100 pixels on United States Department of Agriculture (USDA) National Agriculture Imagery Program (NAIP) data [United States Department of Agriculture]. We follow that.
- **Neighbour tile:** In their implementation they suggest selecting a tile whose centre is within 100 pixels of the anchor tile.
- **Distant tile:** The distant tile can either be from the same initial image (as long as it is outside the neighbouring 100 pixel radius of the original tile) or from a different tile. This choice is random.

For further details on the sampling process we refer the reader to Jean et al. [2019].

Further Implementation Details

We direct the reader to Jean et al. [2019] as we used the provided hyper-parameters and their provided implementation. Among them:

- We use the same network backbone to generate the compact representations: ResNet18.
- We use the same optimiser: Adam with a learning rate of 0.001.

2.5 High Dimensional Data Visualisation

Humans can only see in three dimensions, and so high dimensional data can only become accessible to us visually if we reduce the dimensionality. To visualise high dimensional data, such as images or features from neural networks, we can use a variety of dimensionality reduction techniques. By reducing the dimensionality to two or three dimensions, we can visualise the data which can in turn help us explore the data, inform the algorithms used and analyse results. The underlying assumption of these techniques is that there are dimensions of the data that of the greater importance and those dimensions can be use to project onto. We provide a brief overview of two popular dimensionality reduction techniques, Principle Component Analysis and t-Distributed Stochastic Neighbourhood Embeddings. Over the course of this dissertation, we use t-SNE to visualise the datasets and analyse how certain methods perform in relation to the t-SNE visualisations generated.

2.5.1 Principal Component Analysis

Principal Component Analysis [Jolliffe, 1986] is a dimensionality reduction technique which works works by projecting the data or features of the data onto a set of orthogonal basis vectors, in which the data is the most uncorrelated.

The principal components are the dimensions that account for the greatest variance in the data: the first principal component accounts for the most variance, with the second principal component orthogonal to the first and accounting for second greatest variance.

PCA is a linear dimensionality reduction technique. The problem is that most datasets and features generated from datasets are highly non-linear, and so PCA fails to visualise the data appropriately. Ideally, we want the lower dimensional space to preserve the relationships among the data points in the original high-dimensional space in order to visualise the data more appropriately (similar data points should be close together, and dissimilar should be further away). Unfortunately, linear reduction techniques like PCA fail to do this.

2.5.2 t-SNE

t-Distributed Stochastic Neighbourhood Embeddings is a nonlinear dimensionality reduction technique developed by Van Der Maaten and Hinton [2008] to visualise high-dimensional data. t-SNE aims to obtain a good low-dimensional representation for visualisation by considering the probability that one point is the neighbour of all other points.

Algorithm

We summarise how the t-SNE algorithm works from Van Der Maaten and Hinton [2008], Nakul Verma et al. [2018].

The first step of the t-SNE algorithm requires one to compute the conditional probability that each of the high-dimensional points is similar to every other point:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}, \tag{2.12}$$

where σ_i is the variance of the Gaussian kernel centred around x_i. The variance is defined so that the perplexity of the conditional distribution equals a perplexity defined by the user.

t-SNE then computes probabilities p_{ij} that are proportional to the similarity of objects $\mathbf{x}_i$ and $\mathbf{x}_j$, as follows:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{2.13}$$

where N is the total number of data points.

A low-dimensional map (typically 2 or 3 dimensions) that reflects the similarities p_{ij} as well as possible is then defined. Probabilities q_{ij} are calculated which captures the similarity between two data points on the low-dimensional map $\mathbf{y}_i$ and $\mathbf{y}_j$:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq i}(1 + \|\mathbf{y}_i - \mathbf{y}_k\|^2)^{-1}}. \tag{2.14}$$

However, instead of Gaussian, a heavy-tailed student t-distribution is used to measure similarities between low-dimensional points. The heavy tail student-t kernel allow dissimilar input objects x_i and x_j to be modelled by low-dimensional counterparts $\mathbf{y}_i$ and $\mathbf{y}_j$ that appear far apart on the map.

The coordinates of the low-dimensional embedded points are then determined by minimising the KL divergence of the distribution P from the distribution Q, using gradient descent:

$$D_{\mathrm{KL}}\left(P \parallel Q\right) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.15}$$

Figure 2.16 compares the visualisations produced by PCA and t-SNE on the MNIST dataset (on the image data itself). t-SNE is able to generate clearer and more interpretable visualisations, compared to PCA.

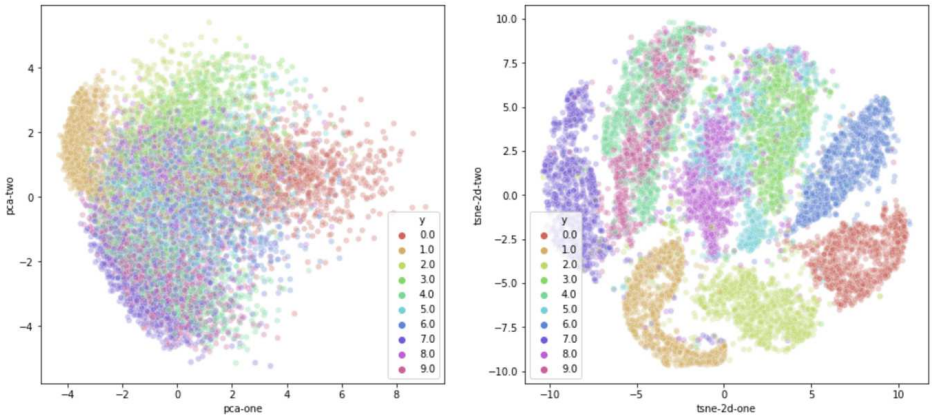


Figure 2.16: MNIST Dataset Visualised using PCA and t-SNE [Derkson, 2016].

Notes for Interpretation

t-SNE comes with a number of provisos when interpreting the visualisations produced [Wattenberg et al., 2016].

Firstly, changing the perplexity parameter can provide very different visualisations. Sometimes, if an inappropriate perplexity parameter is chosen, random noise will form clusters or data that should form clusters will appear as noise in the t-SNE visualisation. Thus, trying multiple different perplexity values is common practice in order to obtain an appropriate visualisation. Figure 2.17 illustrates this.

Secondly, t-SNE focuses on the local structure of the data, so the global structure is only sometimes preserved. "If two points are close in the original space, there is a strong attractive force between the the points in the embedding. Conversely, if the two points are far apart in the original space, the algorithm is relatively free to place these points

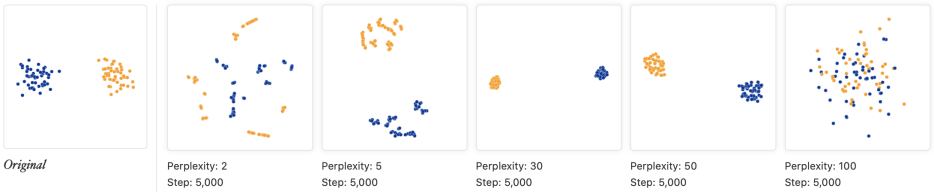


Figure 2.17: t-SNE visualisations of the same data with different perplexity values [Wattenberg et al., 2016].

around." [Karpathy, 2014]

Additionally, distance and so density of the data have very little meaning too, as can be seen in figure 2.18. Thus, points in the t-SNE embedding space carry no inherent meaning, but t-SNE remains a very useful tool for visualisation when used correctly.

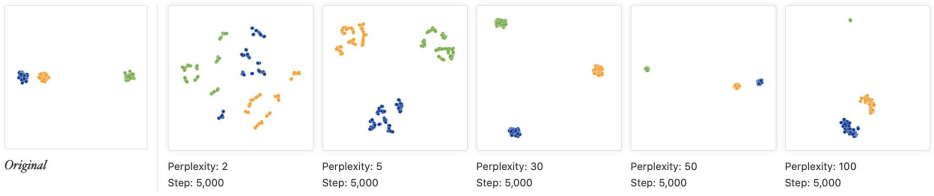


Figure 2.18: t-SNE visualisations of the same data illustrating that distance and density of points carry no meaning [Wattenberg et al., 2016].

2.6 Building Footprint Segmentation

Prior to the success of the deep learning building footprint segmentation was based on traditional computer vision methods. These methods used features such as edges, textures, spectral properties and shadows [Shrivastava and Rai, 2015] as inputs to machine learning techniques. These algorithms include the Hough transform (which produces line-segments of proposed buildings) [Guducu, 2008], watershed segmentation [Shrivastava and Rai, 2015], segmentation based on principal component analysis [Aytekin et al., 2009], and Huang and Zhang [2012] use of a morphological shadow index to identify buildings. Chen et al. [2018b] used edges and shadows as features, and tried AdaBoost, random forests and support vector machines to classify and identify buildings [Li et al., 2018]. There is a significant body of literature in this area and we have not covered all the work. Maxwell et al. [2018] provides a comprehensive review of non-deep learning methods for remote sensing generally, including features and classifiers used. We focus on the work using deep convolutional neural networks as they are far superior and have been so for over the past 5 years.

Deep Learning Based Methods

Since deep learning's success in other computer vision tasks, deep convolutional neural network semantic segmentation algorithms have been applied to building extraction with great success. Initially patch-based segmentation algorithms were employed, using a CNN based classifier to output the building segmentation map. Mnih [2013] proposed this approach by training a CNN to classify pixels in the image by using small patches of the original image, with or without overlap from the larger original image.

More recently, the state-of-the-art for building footprint segmentation has converged to a common algorithm: identify instances and extract polygons. The identify instance step has been approached as instance segmentation task

[Zhao et al., 2018], in which instances are directly obtained from an instance segmentation network, Alternatively and more commonly it is first approached as semantic segmentation task [Maggiori et al., 2017a], following which instances are derived using the connected components of the mask predicted by a semantic segmentation model. A polygon is then extracted for each building instance. Among the semantic segmentation innovations that have been applied to building image segmentation include residual-based networks [Liu et al., 2019], denser networks with attention [Yang et al., 2018], conditional random fields [Shrestha and Vanneschi, 2018], multiple data sources [Ji et al., 2018, Li et al., 2019], and the use of LIDAR [Lai et al., 2019].

Improvements have been made to general segmentation models that are more specific to buildings. They often incorporate priors on the geometry of the building footprints, whereas previous efforts tended to result in unrefined boundaries. Some algorithms incorporate a loss to improve the boundary with some success [Bokhovkin and Burnaev, 2019]. Algorithms like Deep Structured Active Contours (DSAC) [Marcos et al., 2018a] and Deep Active Ray Network (DARNet) [Cheng et al., 2019], as shown in figure 2.19, incorporate reasoning about the geometry of the buildings to create more refined and accurate boundaries compared to the rather inaccurate boundaries obtained via direct polygonisation of semantic and instance segmentation methods.

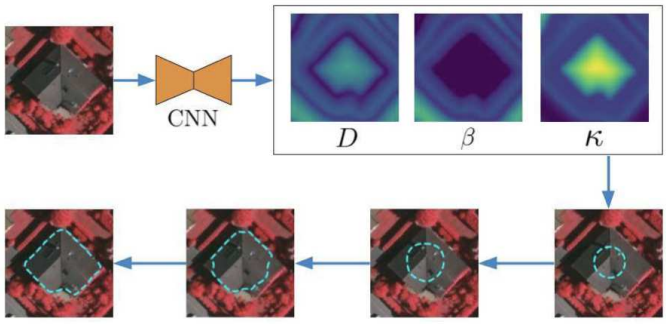


Figure 2.19: Deep Active Ray Network (DARNet) for building segmentation [Cheng et al., 2019].

In the context of this dissertation it is important to note that these improvements are somewhat independent to the work done here in interactive segmentation and active learning. They are complementary, and one can relatively easily incorporate these methods with our proposed approaches.

Datasets

Beyond the algorithms, some of the associated problems of early methods were dataset related: small datasets [Mnih, 2013], inaccessible datasets due to rights restrictions on expensive satellite imagery and lack of high-resolution aerial imagery. Many papers used a single image covering a relatively small ground area in a single region. Since 2010, multiple datasets have been introduced including Massachusetts Buildings Dataset [Mnih, 2013], Vaihingen and Potsdam Dataset [International Society For Photogrammetry And Remote Sensing, 2013], and the Inria Building Footprint Segmentation Dataset [Maggiori et al., 2017b]. The availability of satellite imagery, along with their utility in a variety of applications, has subsequently made this a more popular research area.

Generalisation and Out of Distribution Adaptation

A jump from evaluating on a single region and images with hand-crafted features to training deep neural networks with large datasets has occurred over the past decade. As such, we have seen the generalisation of models improve over time, due to both proliferation of large datasets and better algorithms.

However, one of the main limitations at present is the ability of the deep learning models to generalise to data that was not within the distribution of the training data [Goodfellow et al., 2016]. This limitation is an active research problem both within the machine learning and remote sensing communities. The Inria Aerial Imagery dataset [Maggiori et al., 2017b] was specifically created to evaluate a model's ability to generalise building segmentation from one city to another, taking into account intra-class variability across the cities. Among approaches to improve generalisation are domain adaptation [Benjdira et al., 2019], incremental learning [Tasar et al., 2019] and adversarial learning [Shi et al., 2019]. These approaches, while taking steps to improve performance out of distribution, still largely fail to achieve sufficient accuracy for industry usage.

2.7 Land Cover Mapping

The work in land cover mapping is not too different from building footprint segmentation. However, as the output is not required to be as refined, prior to deep learning this was often pursued as a classification task where the task is to classify patches of aerial images. With more data and more accurate algorithms there is an increased focus on viewing the task as a semantic segmentation problem with pixel-wise segmentation for more accurate maps.

Prior to deep learning the features used in land cover classification included textures and shape, but algorithms also made prominent use of the spectral properties of the various land cover labels [Herold et al., 2002]. Similar to building footprint segmentation, these hand-crafted features were used with various classifiers like AdaBoost [Isaac et al., 2017] and random forests [Hayes et al., 2014]. A hurdle commonly faced by these algorithms is generalising within the same region, due to variation between objects that are of the same class (e.g. farms, forests and fields looking different in different areas). These hand-crafted features with non-deep classifiers are unable to learn high-level representations that generalise across regions. It should be noted that multi-spectral imagery did assist with classifying vegetation more accurately [Govender et al., 2007, 2008].

Land cover mapping has two approaches: (1) patch-based classification and (2) pixel-based segmentation. Both remain well-utilised approaches, despite the significant advancement in pixel-based semantic segmentation.

Deep Learning Based Approaches

In early patch-based approaches, overlapping patches were passed through CNN to be classified. Earliest among these works was Mnih [2013], who proposed a patch-based CNN for feature learning of aerial imagery. Längkvist et al. [2016], Paisitkriangkrai et al. [2016] used a CNN with overlapping patches for land cover classification, outperforming the existing classification approaches. The disadvantages of a patch-wise procedure are redundant computation and less fine-grained output.

As with all semantic segmentation tasks in computer vision, the focus has shifted from patch-based CNNs towards semantic segmentation on a pixel-wise level. Fully convolutional networks, as proposed by [Long et al., 2015b], have been broadly applied to land cover mapping [Volpi and Tuia, 2018]. The initial FCN-based methods struggled with fine-grained outputs for the reasoning provided in the section of semantic segmentation (pooling). Following the broader vision community, the use of atrous convolution and skip connections is now commonplace. The baseline architecture for land cover mapping however remains ResNet18-based FCN [Singh et al., 2019], although U-Net [Ronneberger et al., 2015] is quite popular too. An augmented UNet often with the VGG-19 backend as an encoder, which was termed TernausNet and its successor TernausNetv2 [Iglovikov et al., 2018], which uses a ResNet backend as the encoder are also popular with the Kaggle community. Pashaei et al. [2020] provide a thorough review of deep learning architectures for land cover mapping.

Beyond the architectures, approaches to improve segmentation and/or reduce the supervision required include: weakly-supervised approaches using geo-tagged labels [Wang et al., 2020], unsupervised approaches like Tile2Vec

[Jean et al., 2019] and pre-training in a variety of manners [Singh et al., 2019].

Despite these significant advancements in performance, there remains hesitation around using these methods in industrial and critical applications, due to accuracy issues caused by an inability to generalise beyond the regions trained upon.

Datasets

Similar to building footprint extraction, more and larger datasets have become available in recent years. In addition to more recent datasets being larger, they have also been of a higher spatial resolution, spectral resolution and of generally better quality. The UC Merced Dataset [Yang and Newsam, 2010] covers 7km^2 with a spatial resolution of 0.3m and and the spectral resolution is RGB. The DeepGlobe Land Cover Classification Challenge dataset [Demir et al., 2018] consists of satellite imagery focusing on rural areas covering a total area 1 717km^2, with spatial resolution of 0.5m and a spectral resolution of RGB. The Chesapeake Land Cover dataset [Robinson et al., 2019a] is one of the largest publicly available datasets covering 160 000km^2, with spatial resolution of 1m and spectral resolution of RGB and near infrared. We introduce this dataset further in section 3.1. However, one of the key aims of the dataset was to further research in model generalisation and adaption to new regions.

Generalisation and Out of Distribution Adaptation

As with deep learning more generally, and building footprint segmentation, Robinson et al. [2019a] shows that a network trained on the large Chesapeake Dataset does not perform well in other regions of the USA. Large errors limit their applicability and deployment of machine learning models in a variety of remote sensing applications [Robinson et al., 2019b]. As with building footprint segmentation, many of the same approaches have been applied to land cover mapping: domain adaptation [Benjdira et al., 2019, Fang et al., 2019], incremental learning [Tasar et al., 2019] and adversarial [Bejiga et al., 2019].

2.8 Conclusion

In this chapter, we introduced disaster relief mapping as a problem, analysed the process– highlighting the constraints– and discussed how machine learning can assist in reducing the burden of the mapping on volunteers. Remote sensing more generally was discussed along with the various properties of remote sensing imagery, such as spectral resolution and the relationship to mapping various objects.

Literature and background were presented on a variety of machine learning techniques relevant to mapping. Semantic segmentation more generally was presented, followed by three popular methods of unsupervised representational learning. Background on two high dimensional data visualisation techniques was presented.

Lastly, overarching literature of machine learning techniques applied to building footprint segmentation and land cover mapping was presented.

Chapter 3

Datasets

This chapter introduces the datasets used in this dissertation.

Section 3.1 introduces the Chesapeake Land Cover Dataset. It is a land cover mapping dataset, designed to assess the abilities of semantic segmentation models to generalise across regions. In addition, the section provides some analysis of the dataset and the sub-regions within it. This is used to help inform the chosen experiments and interpret the results. In subsequent chapters this dataset is also referred to as the Land Cover Mapping dataset or LCM dataset.

Section 3.2 introduces the Inria Aerial Image Labelling Dataset. It is a building footprint segmentation dataset, designed to assess the abilities of models to generalise across cities. Like section 3.1, this section also provides analysis of the dataset. This dataset is also referred to as the Inria dataset in subsequent chapters.

3.1 Chesapeake Land Cover

The Chesapeake Land Cover dataset [Robinson et al., 2019a] is a land cover mapping dataset covering the Chesapeake Bay area in the United States of America. The organisation of this dataset was designed to accelerate machine learning research into land cover mapping and allow researchers to easily test questions related to the problem of geographic generalisation, i.e. how to train machine learning models that can be applied over wider areas.

This dataset contains data and labels from multiple sources of varying quality and resolution. We, however, only use a subset that includes high-resolution aerial imagery from USDA NAIP and high-resolution land cover labels from the Chesapeake Conservancy [Chesapeake Conservancy, 2017].

Figure 3.1: Example 1 km^2 image patches from the Chesapeake Land Cover Dataset [Robinson et al., 2019a]. Top row: NAIP imagery from 2012, NAIP imagery from 2015, ground truth land cover. Bottom row: Landsat leaf-on imagery, Landsat leaf-off imagery, NLCD land cover.

The NAIP imagery consists of RGB and near-infrared bands at 1m spatial resolution. High-resolution land cover labels from the Chesapeake Conservancy were aligned to those images. The Chesapeake Conservancy spent over 10 months and $1.3 million creating a consistent six-class land cover dataset covering the Chesapeake area. The accuracy of the labels was reported to be around 90%-95% [Robinson et al., 2019b]. While the purpose of the mapping effort by the Chesapeake Conservancy was to create land cover data to be used in conservation efforts, the same data can be used to train machine learning models that can be applied over even wider areas. The Chesapeake Conservancy created six classes of land-cover:

1. water,
2. tree canopy/forest,
3. low vegetation/field,
4. barren land,
5. impervious (other),
6. impervious (road).

Robinson et al. [2019a], when introducing the dataset, reduced this to a four class problem by combining barren land, impervious (other) and impervious (road). The impervious land cover classification refers to artificial structures such as roads, parking lots and buildings.

The dataset consists of a total of 732 tiles from states within the Chesapeake Bay area acquired during 2013 and 2014. The states are Delaware, New York , Maryland, Pennsylvania, West Virginia and Virginia. There are 100 training tiles (except for Delaware, which has only has 82 training tiles) and 5 validation tiles.

This dataset had a subset which included 500 generated patches from 25 training tiles and all validation tiles. Here, a patch is defined as a random 240×240 meter crop from the tile's extent. This results in 12500 training patches and 2500 validation patches per state, for a total of 75000 training patches and 15000 validation patches. We used

this subset in our experiments.

Thus the data used per patch is as follows:

1. the four-channel aerial image, with red, green, blue and near-infrared bands.
2. A ground truth land-cover labels consisting of four classes: water, forest, field and impervious.

Why this dataset?

This dataset is the largest publicly-available high-resolution, multi-spectral land-cover mapping dataset, covering 160000km^2. It was specifically designed to test generalisation and it covers a large area. It is therefore appropriate to evaluate models and their abilities across this difficult and diverse dataset.

3.1.1 Dataset Analysis

In this sub-section we briefly analyse the dataset, looking at the proportions of the different classes per state and the band content per class in each state. The analysis is useful in understanding the difficulty of generalisation across these states. It provides some quantitative proof, beyond the easily seen qualitative proof, that a class can appear very different in different states, making generalisation difficult. Furthermore, the proportions of different classes in each state also affects model performance with respect to the optimisation and generalisation across states.

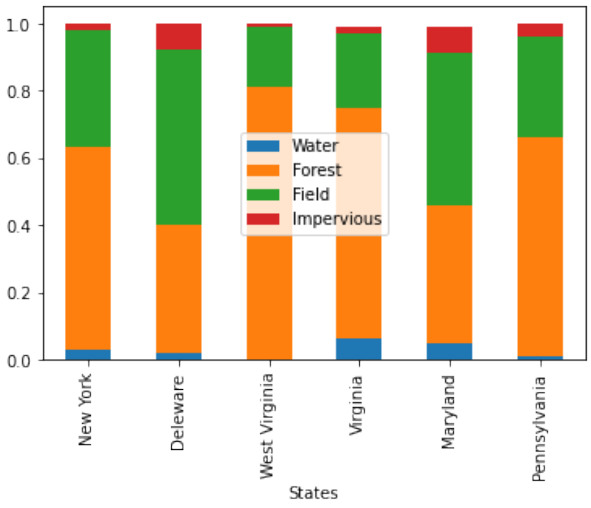


Figure 3.2: Proportion of Labels per State.

Figure 3.2 shows the land cover label proportions across each state in the Chesapeake dataset. We can see across all these states that the dataset is very imbalanced. The proportion of areas that contain water or are impervious are tiny compared to the areas consisting of forest and field. In fact, water and impervious make up only 3% and and 4% of the dataset, respectively, while forest and field make up 59% and 34% of the dataset, respectively.

Delaware and Maryland have the most impervious land cover with 8.0% and 8.4% respectively, while Maryland and New York have the most land cover labelled water with 4.9% and 3.3% respectively. West Virginia is the most class imbalanced dataset. Water and impervious land cover consists of merely 0.3% and 1.2% of the state, respectively. Maryland is clearly the most balanced of the six states.

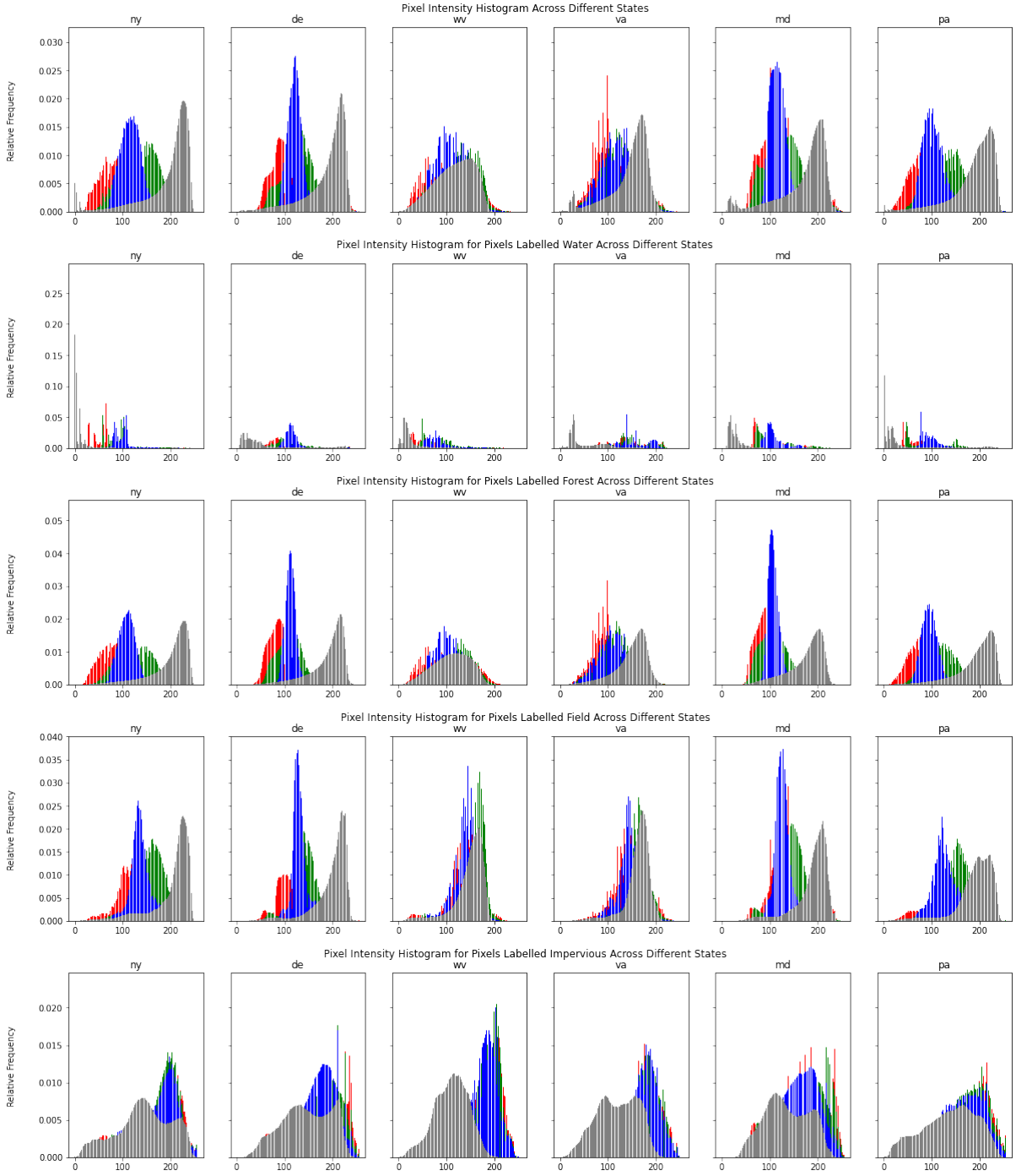


Figure 3.3: Pixel histograms across states for overall and for each label. *ny, de, wv, va, md, pa* indicate New York, Delaware, West Virginia, Maryland and Pennsylvania respectively.

Figure 3.3 shows the pixel histograms across each state overall and for each land cover label. In the histogram, red, green and blue are indicated by their respective colours, while near infrared is indicated by grey. The vertical axis indicates the relative frequency of a particular intensity value within the image. The images are 8-bit, and so the maximum intensity value is 255.

From the pixel histograms in figure 3.3 alone, one can see that Maryland and Delaware share similar image characteristics. West Virginia and Virginia also show similar characteristics. One could hypothesise that generalising

across these states could be easier, although that may not always be the case. In particular, in our experience, we have often seen that roads or houses look very different within the same region.

In section 2.2, we noted that the red band is used to distinguish vegetation and we can see the importance of the red band when comparing the relative frequency for the forest and field land cover labels, as opposed to the impervious and water labels at a particular intensity. The water labels are also seen to have a higher relative frequency in the near infrared band near an intensity of 0. This indicates that water absorbs much of the near infrared radiation.

Lastly, we look at a t-SNE plot of the dataset— a popular method for high-dimensional data visualisation [Van Der Maaten and Hinton, 2008]. The features for this t-SNE plot were generated using Tile2Vec. Figure 3.4 shows the t-SNE plots for the Chesapeake dataset. It shows that the states are relatively easily distinguishable from each other. Clearly, Tile2Vec is picking up on some difference between the data in the different states. It is finding features to differentiate between the regions rather than finding features to differentiate between different classes more generally as the Tile2Vec method intended. However, it is useful to note that t-SNE plots also confirms what we saw from the pixel intensity histograms: West Virginia and Virginia look similar, as do Maryland and Delaware. The imagery of these similar looking regions were likely captured during a similar period, with similar sensors along with similar visual make up.

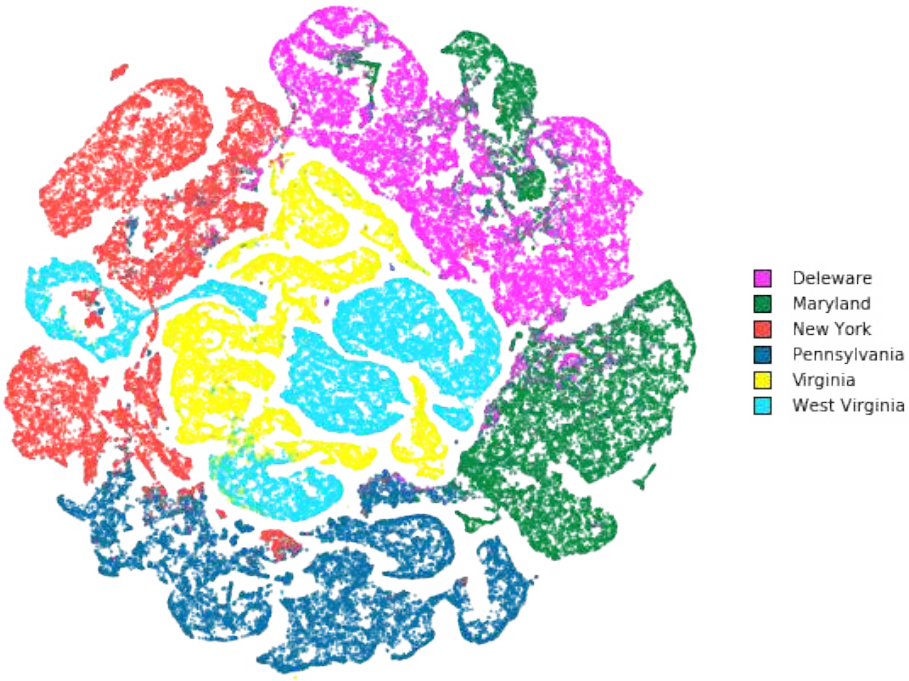


Figure 3.4: t-SNE embedding diagram for the Chesapeake Land Cover Dataset.

3.2 Inria Aerial Image Labelling Dataset

The Inria Aerial Image Dataset [Maggiori et al., 2017c] is a building footprint segmentation dataset that was constructed by combining publicly available imagery and building footprints. The dataset covers 5 cities: 3 in the United States (Austin, Texas; Chicago, Illinois and Kisap County, Washington) and 2 in Austria (Vienna and West-Tyrol). It covers a further 5 cities for evaluation purposes, in a online challenge. The ground truths for evaluation cities are not available we do not use these cities and make no further reference to in this dissertation.

Each city contains 36 tiles of 1500×1500 pixels at a 0.3 meter resolution, for a total coverage per city of 81km^2. It consists of ground truth data for two semantic classes: building and not building. The imagery cover dissimilar urban regions, ranging from densely populated cities like Chicago to more sparsely populated regions like Tyrol in Austria.

The original US imagery was obtained from USGS through the National Map service. It was obtained at a spatial resolution of 0.15m or 0.3m, and a spectral resolution of RGB or RGB and near-infrared. The Tyrol and Vienna imagery was obtained from the local government authorities. This imagery was obtained at a spatial resolution of 0.1m or 0.2m, and a spectral resolution of RGB. The data was re-sampled with the common factors: a spatial resolution of 0.3m and a spectral resolution of RGB.

To enable us to use the data, we chose to create patches of 250×250 pixels for each tile. We created a total of 72000 patches, which enables full coverage of the area. Following common practice use of this dataset, the first 5 tiles from each city are the validation set and the rest are training.

Figure 3.5: Examples from Inria Aerial Image Labelling Dataset. **Top row:** Imagery. **Bottom row:** Reference ground truths.

Figure 3.5 provides a few example images from the dataset. The first three images are from cities in the test set and are not used in the dissertation. Also note the difference in image quality and general appearance of the model across the cities.

3.2.1 Dataset Analysis

In this sub-section we briefly analyse the dataset, looking at the frequency of the different classes per city and the band content per class in each city. This is useful in understanding the difficulty of generalisation across these cities.

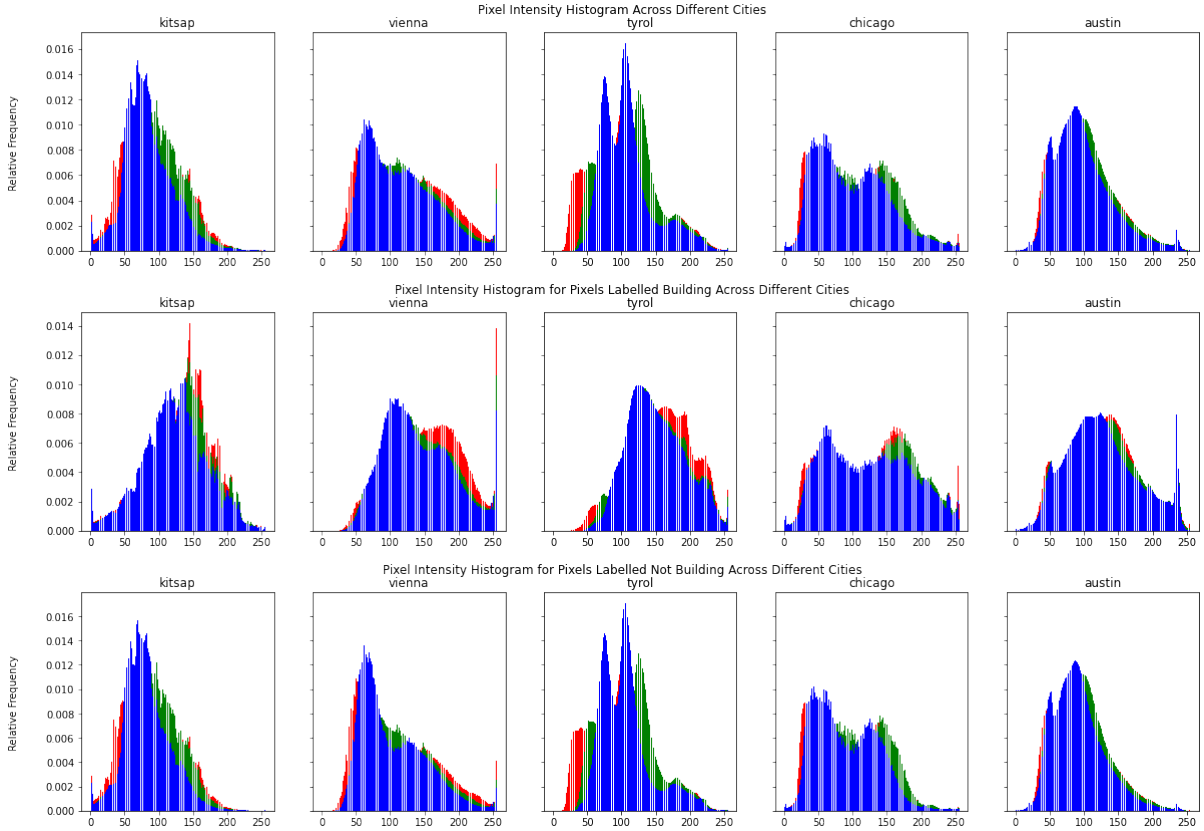


Figure 3.6: Pixel histograms across each city, overall and for each label.

Figure 3.6 shows the pixel histograms across each city overall and for the two labels *Not Building* and *Building*. Red, green and blue are indicated by their respective colours. Relative frequency of a particular intensity value within the image is reflected on the vertical axis. The images are 8-bit, and so the maximum intensity value is 255.

From the pixel histograms presented in figure 3.6, we also see that the dominant label is the *Not Building* label, as the overall pixel histograms reflect the pixel histogram of that label. We also see that pixel histograms for the two Austrian cities look similar, as do two of the American cities, Chicago and Austin. The other American city, Kitsap County, does however appear very different.

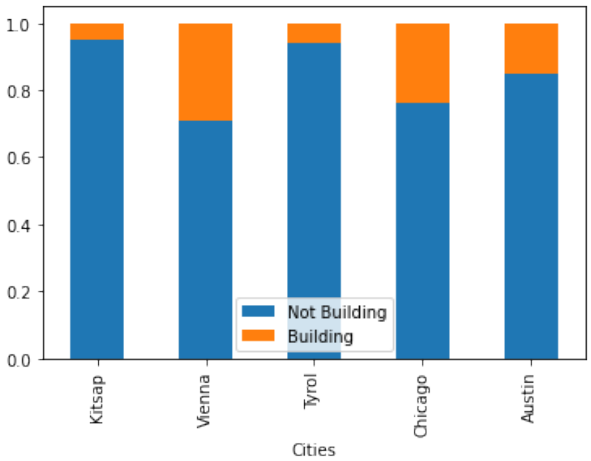


Figure 3.7: Proportion of labels per city.

Figure 3.7 displays the proportions of each label in each city. Across all cities, the number of *Not Buildings* pixels far outnumber the *Building* pixels, which makes sense. However, the proportion varies from city to city. More urban cities like Chicago have more buildings, while more rural cities like Tyrol-West have fewer. This is reflected in the proportions.

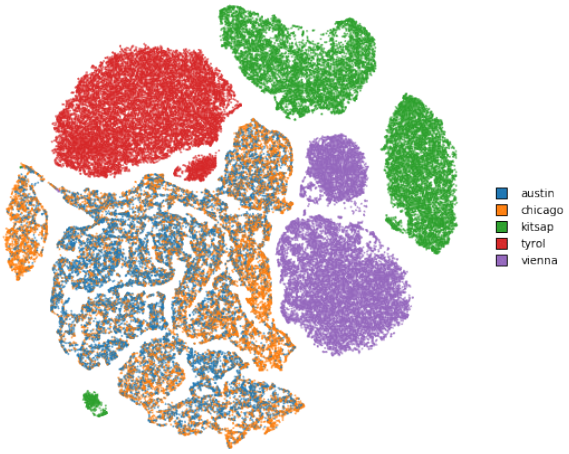


Figure 3.8: t-SNE embedding diagram for the Inria Aerial Image Labelling Dataset.

Figure 3.8 shows the t-SNE embedding diagram for the dataset. Again, the t-SNE plots indicates that Tile2Vec seems to be 'cheating' when generating features. The visualisation of Tile2Vec's features show that the technique is not learning features that we consider useful for differentiating between buildings and not buildings, but rather it is learning to differentiate between cities. The t-SNE plot suggests that Chicago and Austin appear very similar. Certainly, they are both urban American cities and their imagery comes from the same source (USGS). Kitsap County also comes from America but does not appear similar according to the Tile2Vec features and t-SNE. It should be noted Kitsap County is more rural than both the other American cities. The European cities were each collected from two different sources. Tile2Vec is likely picking up on the difference in data acquisition again. Much of this reflects what was seen in the pixel histograms in figure 3.6.

3.3 Conclusion

This chapter introduced and presented analysis of the two datasets used in this dissertation. The analysis highlighted the differences between regions in the datasets, the class imbalance and potential difficulties in generalising across the regions. The pixel histograms and t-SNE visualisation lend further credence to the differences between regions and the difficulty machine learning models have in generalising across regions.

Lastly, table 3.1 presents the summary of the data used:

Table 3.1: Summary of datasets.

Dataset	Regions Covered	Spectral Resolution	Spatial Resolution	Training Samples per Region	Validation Samples per Region	Sample Size (pixels)
Chesapeake Land Cover	6	RGB+NIR	1m	12500	2500	240×240
Inria Aerial Imagery	5	RGB	0.3m	12400	2000	250×250

Chapter 4

Interactive Segmentation

This chapter presents work on interactive segmentation. In the traditional interactive segmentation paradigm, the annotator is asked to segment a single object and a single class at a time. To segment an object in an image, the annotator is required to indicate which object needs to be segmented. A machine learning model then uses this information along with the image data to produce a segmentation mask. Some interactive segmentation algorithms allow for multiple rounds of user interactions to improve the segmentation mask.

The work on interactive segmentation in this chapter differs from the traditional paradigm, in that we do not aim to segment a single object at a time. In building segmentation, we look at images that contain hundreds of buildings all of which need to be segmented at the same time. In land cover mapping, we look at segmenting the entire image into four classes simultaneously.

Section 4.1 presents literature related to interactive segmentation for land cover mapping and building footprint segmentation. This section also highlights how the work presented in this chapter differs from previous work in the field.

Section 4.2 presents models that are tested on the two tasks. For the establishment of baselines, a standard ResNet18-Based FCN and a ResNet18-Based FCN that takes in supervisory signals in addition to the image data are used. Section 4.2.3 presents the proposed model for building footprint segmentation and land cover mapping. In the proposed model, the human annotator iteratively corrects the semantic segmentation masks provided by the machine learning model. The corrections are provided by clicks on the largest incorrect regions, with multiple rounds of corrections.

Section 4.3 presents the details on the experiments conducted. This section also presents the questions we attempt to answer and how the experiments assist in answering these questions. These questions include whether the proposed model works, the performance improvement it provides and whether it works in different regions to those it was originally trained on.

Section 4.4 presents and discusses the results of the experiments that were detailed in section 4.3. The results show that the proposed algorithm provides a significant performance increase over the baselines, in both quantitative and qualitative respects. In a quantitative respect, mean intersection over union score increases up to 18%. In the qualitative respect, the proposed algorithm removes incorrect segmentation masks, adds missing segmentation masks and makes the segmentation masks more regular. Thus, the proposed approach provides results more suitable for submission to mapping databases like OpenStreetMaps. The results also indicate that the algorithm works well when extended to regions not in the training data. In fact, it provides a significant improvement in performance, showing that some weak additional supervision can improve out-of-distribution model performance significantly.

Overall, this chapter proposes and investigates the utility of an iterative corrections algorithm for interactive

semantic segmentation of buildings and land cover classes from aerial imagery. For faster disaster relief mapping this algorithm does not automate but rather assists volunteers in mapping, allowing for implementation in the short-term.

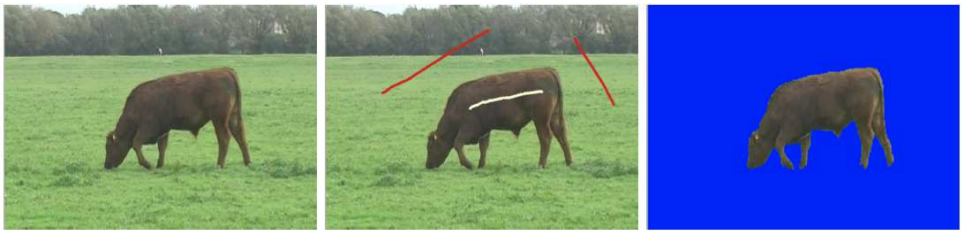
4.1 Related Work

Segmenting images into semantically meaningful components is a core computer vision task with many downstream uses. However, it is also difficult, and while the advent of deep learning has brought substantial improvements it is a computer vision task that remains challenging. Given the difficulty and the ability of algorithms to perform the task, many interactive segmentation algorithms have been proposed and are in widespread use in industry. Notable examples can be found within Microsoft's Office suite and variety of Adobe's software applications.

Active Contours and Intelligent Scissors were amongst the earliest works in the field and worked on the basis of finding edges. The field then moved towards using more features within the image to guide the interactive segmentation and graphical models forming the basis of these algorithms. The seminal pieces of literature in the area are the works of Boykov and Jolly [2002], which uses graph cuts to segment objects in images based on labelled pixels provided by the annotator, and Rother et al. [2004], which segments from bounding boxes by iteratively updating the model (known as GrabCut).

Subsection 4.1.1 presents this early work on interactive segmentation, with a focus on the interactions required by the annotator. Subsection 4.1.2 presents initial approaches that incorporate deep learning, while subsection 4.1.3 presents the more recent interactive segmentation approaches. Subsections 4.1.4 and 4.1.5 present some of the work that has been on done on weakly-supervised semantic segmentation and interactive segmentation specifically in remote sensing applications. Finally, subsection 4.1.6 details how the building footprint segmentation and land cover mapping differ from previous interactive segmentation set-ups.

4.1.1 Pre-Deep Learning



(a) Original image (b) User Annotation (c) Output segmentation

Figure 4.1: Graph Cuts Example [Gulshan, 2012].

The most widely used interactive segmentation algorithms in industry are briefly discussed. The focus here is not to discuss and explain the algorithm, although we briefly do, but rather to show the evolution on the interaction required by an annotator.

In segmentation using graph cuts, each of the pixels in the image is a vertex in a graph. An annotator draws or scribbles a few lines indicating the foreground object and the background. This information along with other pixel information is used to compute edge weights in a graph, which represents the probability between a vertex (pixel) and the vertices corresponding to its neighbourhood pixels belonging to the same class. Once the graph is defined, the segmentation of the the foreground object is obtained through a min-cut optimisation algorithm run on the graph. Figure 4.1 shows an image, an example of annotations required by the user and corresponding segmentation

mask output obtained from the graph cut algorithm.

GrabCut used a similar algorithm to Boykov and Jolly [2002], but instead of requiring multiple interactions to define the foreground/background it only required a bounding box around the object to be segmented. The annotator first draws a rectangle around the foreground object. A Gaussian Mixture Model is used to model the foreground and background. A graph is built from this pixel distribution. Then a min-cut algorithm is used to segment the graph. The algorithm segments iteratively to obtain the best result. In some cases the segmentation will not be good enough for the annotator. The annotator is then allowed to touch up the segmentation and the optimisation is rerun. Figure 4.2 shows initial bounding box around the person and soccer ball to be segmented, along with additional touch-ups provided by the user. The image to the right is the corresponding output from the GrabCut algorithm [OpenCV].

Figure 4.2: An Example of the GrabCut Algorithm [OpenCV].

4.1.2 Initial Deep Learning Approaches

More recent methods make use of fully convolutional networks, which has resulted in far superior performance compared to previous approaches. iFCN [Xu et al., 2016] was the first work in the line of deep-learning based interactive segmentation. It makes use of positive and negative clicks, in the spirit of Boykov and Jolly [2002], to guide the segmentation. They create a Euclidean distance mask of each of the positive and negative clicks and concatenate them with the original image, thereby acting as additional channels for the FCN. Finally, the output of the FCN is refined using graph-cut optimisation, which marginally increases the performance over the FCN output. Figure 4.3 provides an illustration of the iFCN algorithm. It shows the input image, positive and negative clicks, the Euclidean distance masks and the corresponding output after these inputs are fed through a fully convolutional network.

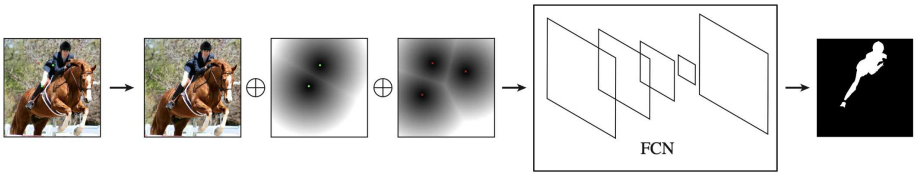


Figure 4.3: Illustration of iFCN Algorithm by Xu et al. [2016].

The iFCN deep learning pipeline was extended to GrabCut, with Deep GrabCut [Xu et al., 2017]. Similar to GrabCut, Deep GrabCut uses a bounding box as the required annotator interaction. This interaction is transformed into a distance transform and fed as an additional channel to the FCN. Xu et al. [2017] then also use conditional random fields (CRFs) to refine the segmentation. This post-processing step is common to all segmentation algorithms in the FCN-era to improve the segmentation result, although it only provides a marginal improvement. Figure 4.4 illustrates the Deep GrabCut algorithm similar to the previous illustration of the iFCN algorithm in figure 4.3.

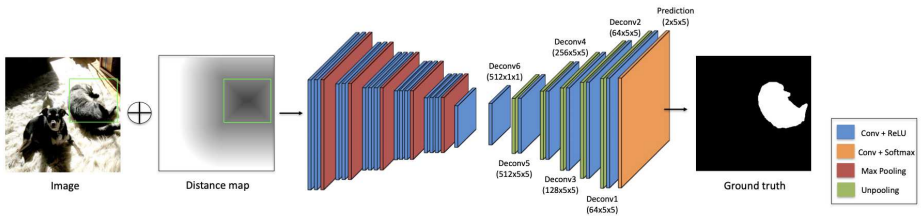


Figure 4.4: Illustration of Deep GrabCut Algorithm by Xu et al. [2017].

4.1.3 More Recent Deep-Learning Based Approaches

The more recent state-of-the-art algorithms make changes to either (1) the annotations process, (2) the backbone FCN or (3) the output processing. The backbone FCN and output processing are fairly uninteresting for our purposes. As with the general trend in computer vision, deep and denser networks perform better, while CRFs smooth the annotations at significant computational cost for limited performance gain.

In terms of the annotation process, Maninis et al. [2018] used four extreme points as the supervisory signal and showed significant improvements over previous methods. The extreme points were placed at the top-right, top-left, bottom-right and bottom-left pixels of the object selected. Figure 4.5 shows the segmentations masks produced by the DEXTR algorithm along with the four extreme points.

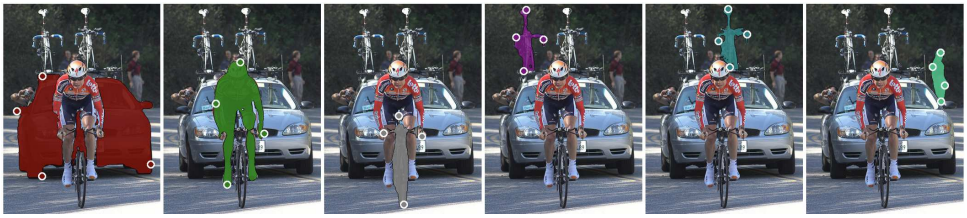


Figure 4.5: Examples of Segmentation Masks produced by DEXTR [Maninis et al., 2018].

Maninis et al. [2018] encoded their points using Gaussian Kernels, instead of the Euclidean distance mask as previously used in iFCN and Deep GrabCut.

Corrections-Based Approaches

The current state-of-the-art has moved beyond just providing an initial supervisory signal towards a corrective approach, with one [Mahadevan et al., 2018] or more models [Benenson et al., 2019] used in the interactive segmentation algorithm.

Mahadevan et al. [2018] used an iterative corrections algorithm, ITIS, to significantly improve the state-of-the-art interactive segmentation results. They required a few initial clicks to define the foreground and background. This was all fed into a FCN. Mahadevan et al. [2018] also trained this corrections model in an iterative manner. They provided multiple corrections in multiple rounds, while doing backpropagation and taking a stochastic gradient descent step on each round.

Benenson et al. [2019] followed a similar approach, but employed the use of two models instead of a single one. The first model would take in a bounding box input from the annotator and the image, while the second would take in the image data, the output for first model and corrections provided by the user. Benenson et al. [2019] also provided comprehensive analysis of the different components of the deep interactive segmentation models (including different click encoding methods). This work, like all the work presented thus far, only segments a single object in an image. This work by Benenson et al. [2019] was done in parallel to some of our work.

Polygon-RNN Segmentation

An alternative approach to the FCN-based approach was developed in Polygon-RNN [Castrejon et al., 2017] and further developed by Acuna et al. [2018]. These methods predict a polygon outlining of the object to be segmented. The polygon can then up edited by the annotator, and the model will reformulate the polygon based on the annotator's input. This work is particularly useful in a variety of applications, because the corrections from the annotator were explicitly ingrained into the model output. Figure 4.6 illustrates the Polygon-RNN+ algorithm.

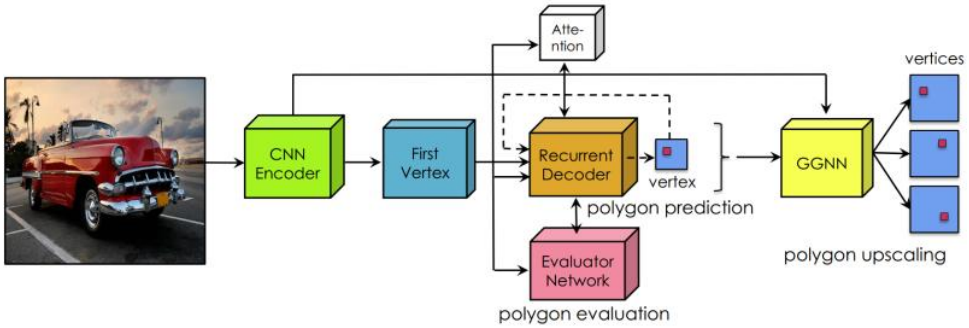


Figure 4.6: Polygon RNN+ by Acuna et al. [2018].

Full Image Interactive Segmentation

All these methods perform single or few object segmentation— this is the traditional set-up. However, we study multi-object segmentation on the scale of tens to hundreds of objects per image and multi-class interactive segmentation. Of the recent deep learning methods, we only found the work of Andriluka et al. [2018], Agustsson et al. [2019] to address full image interactive segmentation that vaguely represents our scenario. However, their methods focus on panoptic segmentation, which considers instance and semantic segmentation simultaneously, along with ideas of depth and overlap. These methods make use of Mask-RCNN as a backend, and use ideas from previously presented papers like extreme points to annotate natural images. We are only studying semantic segmentation applied to remote sensing imagery.

4.1.4 Weakly-Supervised Semantic Segmentation

An alternative to interactive segmentation is weakly-supervised semantic segmentation, with both approaches attempting to reduce the time spent labelling images. A number of weakly-supervised semantic segmentation for natural images has been proposed: image-level [Bearman et al., 2016], point-level [Bearman et al., 2016] and scribble labels [Lin et al., 2016]. Of particular relevance, Wu et al. [2018] used scribble supervision for building extraction, and most recently Wang et al. [2020] used point supervision for land cover mapping.

The aim of weakly-supervised semantic segmentation methods is to train models with less fine-grained labels required (which are time-consuming to create), while the aim of our interactive segmentation approach is to improve the segmentation quality such that it can be used for disaster relief applications. The samples produced could also be used for training models for further use within the region. That distinction is important.

4.1.5 Interactive Segmentation applied to Building Detection and Land Cover Mapping

To the best of our knowledge, interactive segmentation has not been well explored in remote sensing applications. GrabCut has been applied to remote sensing images with some success [Yang et al., 2017a], and Acuna et al. [2018]

do show their Polygon-RNN algorithm working on a small 64 image Rooftop dataset.

4.1.6 Unexplored Area

In the traditional interactive segmentation set-up, one segments a single object (in the foreground) from the background. Note the examples in figure 4.7 from one of the state of the art methods, DEXTR:

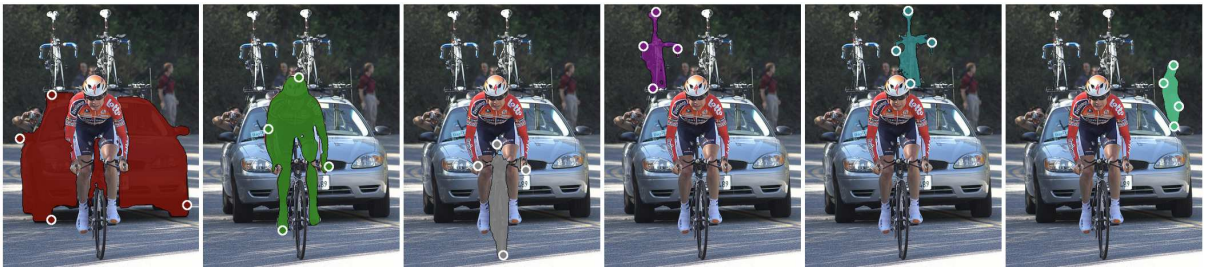


Figure 4.7: Examples of Traditional Interactive Segmentation [Maninis et al., 2018].

Like DEXTR, the majority of interactive segmentation methods require annotations for each object, while segmenting one object at a time. For land-cover mapping and building detection, the problems we consider differ as follows:

1. **Building Detection:** we are looking at many buildings and are asked to segment them all simultaneously. There are many buildings in each image and we would like to obtain all of them with limited interaction. This can be addressed as either a binary semantic segmentation problem or an instance segmentation problem.
2. **Land Cover Mapping:** we would like to map the entire image into four land cover classes, again simultaneously. Each pixel in the image should have a semantic label. This is a semantic segmentation problem.

We choose to address both problems as semantic segmentation tasks.

4.2 Algorithms & Models

This section presents our proposed interactive segmentation algorithm, along with two baselines.

Subsection 4.2.1 presents the baseline fully convolutional network. This is a standard semantic segmentation network. Subsection 4.2.2 presents the simple adaptation made to the baseline FCN to incorporate some user interaction. Subsection 4.2.3 presents our proposed iterative corrections interactive segmentation model.

A conscious decision was made against testing certain algorithms that are considered state-of-the-art, such as Polygon-RNN++ [Acuna et al., 2018] and DEXTR [Maninis et al., 2018], or baselines in the field [Rother et al., 2004, Xu et al., 2017] as they are ill-suited to this task. These algorithms require bounding boxes to be drawn around the objects and/or multiple interactions per object. For large scale annotation of buildings (and other map features), this sort of interactivity does not provide a significant reduction in annotation time, as it requires numerous initial interactions for each individual object.

Furthermore, in our experiments we use relatively simple network architectures, but more performant ones can easily be substituted such as FC-DenseNet [Jegou et al., 2017]. Additionally, algorithms like DSAC [Marcos et al., 2018b] which introduce a shape prior for building segmentation to improve the regularity of the segmentation, could easily be integrated into the proposed algorithm to provide even better segmentation results.

4.2.1 Baseline FCNs

The first baseline is a fully convolutional network for semantic segmentation. This compares whether the proposed algorithm improves performance over the standard semantic segmentation output.

Network Architecture

As our network architecture we propose to use the ResNet18-based FCN as the baseline for both land cover mapping and building footprint segmentation, as previously explained in section 2.3.1. This follows previous work in land cover mapping. [Singh et al., 2019]. While U-Net tends to be more popular with the building footprint segmentation community, due to the model providing better delineation of boundaries, reviewing a single model on the two different datasets makes analysis easier. Furthermore, initial tests found little difference in the relative performance of the baseline to the iterative interactive segmentation algorithm proposed.

Figure 4.8 shows the diagram of the ResNet18-based FCN network architecture. For an explanation of the architecture, please refer to section 2.3.1.

For the baseline, the input is the image data. For the Chesepeake dataset the FCN has four input channels (RGB+NIR), while for the Inria dataset the FCN has three input channels (RGB).

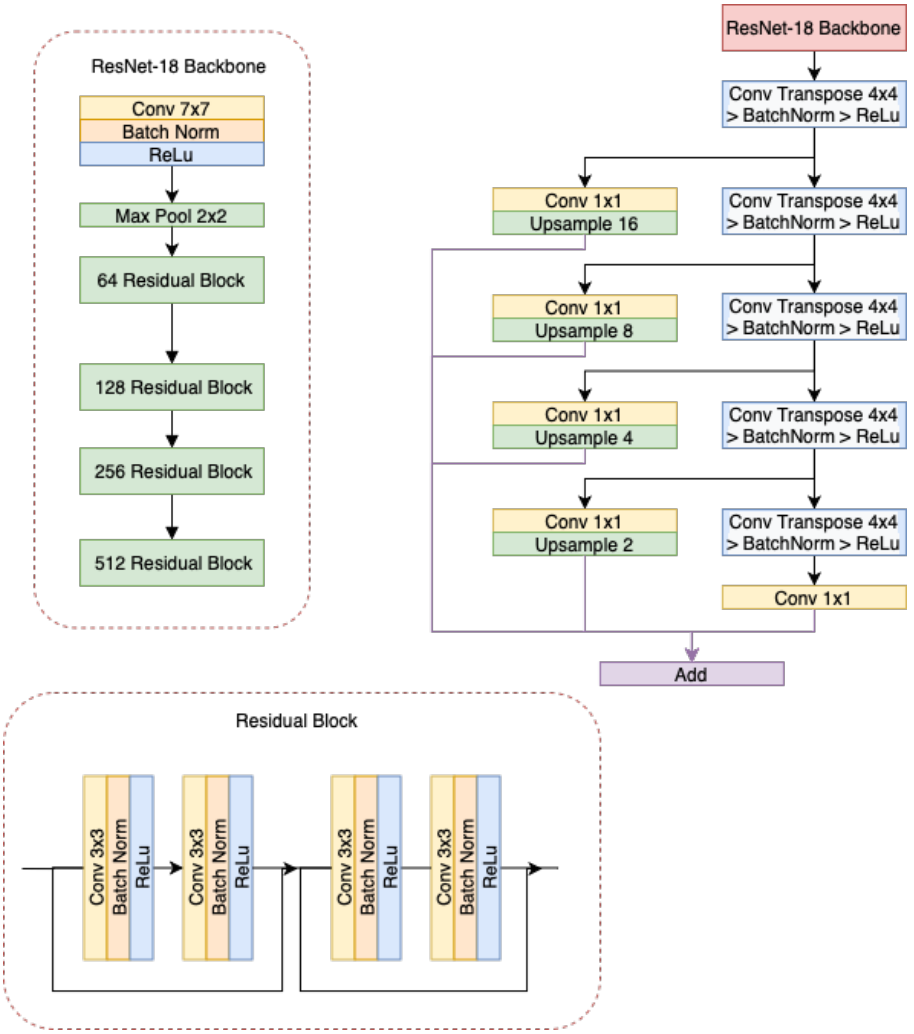


Figure 4.8: ResNet-18 Backbone and the derived FCN network.

4.2.2 Multi-class Multi-object iFCN

As a second baseline, we adapt the existing iFCN framework for multiple classes and multiple objects. We do that by generating as many channels as there are classes, instead of two channels for positive and negative clicks. We concatenate these additional channels to the image data and use the concatenated data as the input to the model. To train the model we use simulated clicks, based on the strategies we detail below.

Network Architecture

We use the same network architecture, the ResNet18-based FCN. The only change is the number of input channels. For the Inria dataset the number of input channels becomes 5 (RGB and 2 classes), while for the LCM dataset the input channels rises to 8 (RGB+NIR and 4 classes).

Click Simulation Strategy

To simulate clicks we derive two strategies:

1. LCM dataset: due to the irregular shapes of the classes, it is difficult to do anything other than select random points, in which we reveal the class.
2. Inria dataset: For the Inria dataset we could choose to select points to reveal in a smarter way, for example with points at centroids of buildings. To remain consistent we use random points as with the LCM dataset.

Click Encoding

The literature on interactive segmentation differs regarding how we should encode the clicks. The original iFCN paper, along with Deep GrabCut, used a Euclidean distance mask. Others like Maninis et al. [2018], have used Gaussian kernels over each click. We study these three options:

1. Binary 3×3 pixel disk: encode the clicks as a 3x3 disk. The disk is placed in the appropriate channel for the class the user selected.
2. Gaussian kernel: this is implemented by placing the points clicks with a value of 1, followed by convolving the map of points with an isotropic 2D Gaussian kernel with $\sigma = 2$:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

   This is done on a per channel basis.
3. Euclidean distance mask: we create a mask with the same 2D dimensions of the original image, and with as many channels as there are outputs classes. The clicks are encoded in the appropriate channel representing the class. For each channel, the pixel value of the mask is p_{xy}, where x and y represent the co-ordinates of the pixel. Let $p_{ij} \in A$, where i and j are the co-ordinates of the user provided clicks and A is the set of all user provided click co-ordinates in that channel. Then we can define p_{xy} as follows:

$$p_{xy} = min_{\forall p_{ij} \in A} \sqrt{(x-i)^2 + (y-j)^2}.$$

   Further, to aid training the inputs are normalised.

4.2.3 Proposed Algorithm: Iterative Corrections

This subsection presents the proposed iterative corrections algorithms for interactive semantic segmentation of remote sensing imagery. The proposed algorithm is guided by the following principles:

1. fewer manual annotations (clicks) are better,
2. corrections should take less time than clicking on each object for manual segmentation.

Overall Algorithm

As opposed to most of the previous work, we do not have the user provide any initial interaction — no bounding box, extreme points or initial clicks. A base network provides the initial segmentation mask, and a second network incorporates the corrections. The corrections come from a user, who marks the incorrect regions with points indicating the correct class. This is not dissimilar from what Benenson et al. [2019] propose, the difference being that their base network takes in a bounding box input to signal the object being selected. To incorporate the output from the base network and the clicks from the user, we concatenate the inputs.

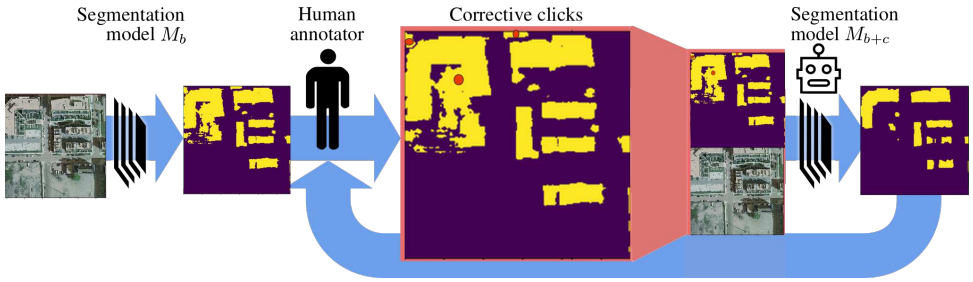


Figure 4.9: The proposed algorithm: M_b indicates the base network and M_{b+c} indicates the correction network. This examples shows 3 negative clicks being used, but this could be any combination of positive and negative clicks. Adapted from Benenson et al. [2019].

Inria: For the Inria dataset, the base model, M_b, takes in 3 channels, RGB, and outputs 2 channels for the 2 classes, *building* and *not building*. The corrections models, M_{b+c}, takes in 7 channels: red, green, blue, *building* prediction, *not building* prediction, *building* correction clicks and *not building* correction clicks. The corrections model also outputs 2 channels for the 2 classes.

LCM: For the LCM dataset, the base model, M_b, takes in 4 channels, RGB and Near Infrared, and outputs 4 channels for the 4 classes. The corrections models, M_{b+c}, takes in 12 channels: RGB+IR, 4 output classes and 4 correction click channels (one for each class).

Network Architectures

As with the baseline, we used the ResNet-18 based FCN as M_b, as illustrated in figure 4.8.

For M_{b+c}, we use a simplified U-Net network (with far fewer filters), as it is very lightweight in terms of computational budget compared to ResNet-18 based FCN. This allows both models to fit on the GPU for training. Note that the representational power of the simplified U-Net is comparatively limited. This is by design so we can test if it is adapting the previous model's output rather than just learning a better representation. This is useful as it allows for faster throughput and training due to the simpler network, and can all be done on a single GPU with less memory making it more accessible. Figure 4.10 illustrates the architecture of the simplified U-Net network.

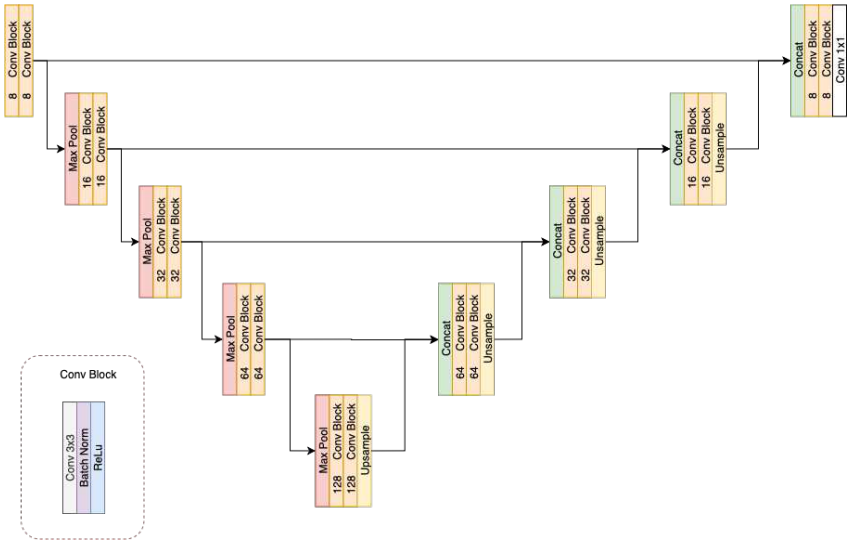


Figure 4.10: Simplified U-Net architecture used for the corrections model, M_{b+c}.

Simulated Click Strategy

The simulated clicks are placed at the largest error regions in that order. For example, if we have 3 rounds of corrections we place the clicks at the 3 largest errors in that round. The process is described more concretely below:

1. Compare the output from M_b to the ground truth.
2. Calculate the largest error regions in terms of area in each class using connected components.
3. Select the largest *X* number error regions and place the 'clicks' at the centroid of each region in the appropriate channel to indicate the correct class.
4. Concatenate the inputs of the initial data, the output of the base model, M_b, and the corrections as inputs to M_{b+c}.
5. For rounds following, instead compare the outputs of M_{b+c} to the ground truth and follow the procedure above.

Click Encoding

A binary 3×3 pixel disk is placed on the centre of each click, in the relevant channel. This was based upon the experiments done on the click encodings with iFCN.

Training Strategy

The base model, M_b, is trained prior to the training M_{b+c}. We then adopt the iterative training strategy initially proposed in Mahadevan et al. [2018] and further established in Benenson et al. [2019] to train the correction model where corrections are iteratively added. This strategy involves the use of multiple rounds of corrections for each image, with the error backpropogated each round. We explore the design space regarding the number of clicks and the number of rounds of corrections in the experiments.

Both models are trained with the same training data.

The training strategy is described concretely below:

1. Compare the output from M_b to the ground truth and obtain the corrections using the simulated click strategy.

2. Concatenate the inputs of the initial data, the output of the base model and the corrections as inputs to M_{b+c}.
3. Compare the output of M_{b+c} to the ground truth, using the loss function. Use gradient descent and backpropagation to update the weights of model M_{b+c}
4. For rounds following, compare the outputs of M_{b+c} to the ground truth and follow the procedure above. The number rounds and number of corrections provided per round are provided in section 4.3.

4.3 Experimental Design

This section presents further details on how the models were trained. It expands on questions we attempt to answer and details experiments we have conducted in order to answer them.

Subsections 4.3.1, 4.3.2 and 4.3.3 further describe the training of the baseline FCN, iFCN and the proposed algorithm.

Subsection 4.3.4 describes an experiment on iFCN to determine the best click encoding method. Subsections 4.3.5 and 4.3.6 describe the experiment around the evaluation of the proposed approaches when the regions are the training data and when the regions are not. Lastly, subsection 4.3.7 describes the experiment on how to structure the corrections in our proposed approach.

4.3.1 Training of Baseline FCN

We train and test on the two datasets separately. We train a model for the Inria dataset and train another model for the LCM dataset. To train the baseline model the following hyper-parameters were used:

- **Initialisation:** The model is initialised using the He initialisation.
- **Model:** ResNet18-Based FCN.
- **Cities or Regions:** All except Kitsap County in Inria dataset and New York on LCM dataset.
- **Input Channels:** 3 on the Inria dataset and 4 on the LCM dataset.
- **Loss Function:** Soft IoU Loss.
- **Optimiser:** Adam with learning rate of 0.001.
- **Epochs:** Train for 30 epochs or to convergence.

4.3.2 Training of Proposed iFCN Model

Like the baseline FCN model, we train and test on the two datasets separately. We train a model for the Inria dataset and train another model for the LCM dataset. The only difference in the training of the two models is the number of input channels and the simulated clicks in those input channels.

Nevertheless, the hyper-parameters for the experiment are presented below:

- **Initialisation:** The models are initialised using the He initialisation.
- **Base Model:** ResNet18-Based FCN.
- **Training Regions:** All except Kitsap County in Inria dataset and New York on LCM dataset.
- **Number of Random Clicks Provided:** 9 for the Inria dataset, and 15 for the LCM dataset.
- **Input Channels:** 5 on the Inria dataset and 8 on the LCM dataset.

- **Loss Function:** Soft IoU.
- **Optimiser:** Adam with learning rate of 0.001.
- **Epochs:** Train for 30 epochs or to convergence.

4.3.3 Training of Proposed Algorithm

We train and test on the two datasets separately. A model is trained for the Inria dataset and another for the LCM dataset. We use the model from the training of the baseline FCN as the base model.

The hyper-parameters for experiment are listed below:

- **Initialisation:** The models are initialised using the He initialisation.
- **Base Network Architecture (M_b):** ResNet18-Based FCN (from the baseline FCN training).
- **Corrections Network Architecture (M_{b+c}):** Simplified U-Net.
- **Training Cities:** All except Kitsap County in Inria dataset and New York on LCM dataset.
- **Number of Correction Rounds:** 3 for Inria dataset, and 5 for LCM dataset.
- **Number of Corrections Per Round:** 3.
- **Input Channels for M_{b+c}:** 7 on the Inria dataset and 12 on the LCM dataset.
- **Loss Function:** Soft IoU.
- **Optimiser:** Adam with learning rate of 0.001.
- **Epochs:** Train for 30 epochs or to convergence.

4.3.4 Click Encoding

As mentioned previously, the literature differs on how to encode the user interactions. The previously described approaches— binary disk, Gaussian kernel and Euclidean distance mask— are compared using iFCN on the Inria dataset, evaluating on the validation set.

We use the hyper-parameters listed in subsection 4.3.2 changing only the click encoding method. We looked at the performance of the various strategies on Kitsap County in the Inria dataset, which was out-of-distribution of the training data.

4.3.5 Overall Performance on Regions in Training Data

We compare the performance of the algorithms on validation data from regions in the building footprint segmentation and land cover mapping datasets. The proposed approaches have not been tested in these scenarios before.

We look at the performance in terms of mean intersection over union, but also in terms of accuracy for specific classes in the case of the land cover mapping dataset.

4.3.6 Performance on Regions not in Training Data

As noted previously, model performance degrades significantly when moving to different cities and regions. An important potential use case is moving the models to regions which are different from the regions we trained on. This is typically where volunteers might need to label. If this improves performance out-of-distribution, it would be of great assistance to the volunteers who map.

To simulate this, the models were tested on the cities or regions they were not trained on — Kitsap County and New York respectively. We evaluate the performance increases over the baselines.

4.3.7 Structure of Iterative Corrections

Finally, we look at an important design choice: How to structure the corrections? While this would be very dependent on the various base models' performance and datasets used, we provide some analysis to help guide the practitioner on how to structure the corrections into rounds. We do this by examining by how to structure the corrections on New York in the LCM dataset.

4.4 Results & Discussion

In this section, the results of the experiments and questions posed in the previous section are presented and discussed.

Subsection 4.4.1 presents the results of the experiment comparing the click encoding methods. The results showed that a simple binary disk works best, and this encoding method was used in all subsequent experiments.

Subsection 4.4.2 presents the results comparing the proposed iterative corrections algorithm with the two baselines. The performance of the proposed algorithm is compared on regions and cities included in the training data.

Subsection 4.4.3 presents the results on regions and cities not included in the training data. The results show a substantial increase in the performance, particularly where the base model performed badly. As this is the case with imagery from regions the models were not trained on, the corrections algorithms performed particularly well in those cities or regions not included in the training data. Given the results in subsections 4.4.2 and 4.4.3, subsection 4.4.4 presents results that provide intuition on how to structure the corrections in rounds.

Finally, subsection 4.4.5 presents the analysis of qualitative results obtained from the proposed iterative interactions corrections model. This subsection presents analysis of how the model performs when presented with an initial segmentation with a large number of mistakes, how it performs when the initial segmentation is already well-described, and how the models translates to larger patches. It also analyses failure cases and the limitations of the proposed model.

4.4.1 Click Encoding

One of the first questions we posed, given the lack of clarity in existing literature on what to use (with different methods doing different encodings without comparison), was how to encode the clicks. We performed the analysis on Kitsap County in the Inria dataset (out-of-distribution), averaged over 3 runs. The results can be seen in table 4.1:

Table 4.1: Click encoding comparison.

	Binary mask	**Gaussian kernel**	**Distance mask**
Building IoU	$0,39$	$0,38$	$0,35$

(Note: we present the Building IoU rather than mean IoU as a greater differentiation can be seen. In results presented later we use mean IoU.)

From the results, it is apparent that the type of post-processing done has negligible effect. The performance of the binary disk and Gaussian kernel methods are roughly similar. The network likely generates its own representation

which it finds more useful.

What was surprising was the under-performance of the the most commonly used encodings (Euclidean distance masks) compared to binary disks. It seems like an unnecessary processing step was used, which is perhaps an artefact of computer vision techniques pre-deep learning. The reason for the under-performance, as compared to a binary disk, and why the the distance mask was likely used in the initial paper was because it was typically segmenting a single object at a time. When having multiple objects and multiple classes, it becomes difficult to interpret exactly where the click occurred as compared to the binary disk and Gaussian kernel.

Based on this result we used a 3×3 pixel binary encoding for all further experiments.

4.4.2 In-Distribution Performance

Tables 4.2 and 4.3 present the performance results across the different regions within the two datasets. Quite broadly, we saw that within the distribution of the data trained on, the interactive clicks provided a significant boost over the baseline and iFCN. iFCN by itself provided around a 5% boost over the baseline. Clearly, additional supervision is helpful and that is what we expected. However, where that supervision comes from is important, as the interactive method provides a 10% boost over the baseline. This is a significant improvement over the baseline.

Table 4.2: mIoU Comparison on the Inria Dataset.

	Vienna	**Tyrol-West**	**Chicago**	**Austin**
Baseline	0,782	0,780	0,725	0,760
iFCN	0,803	0,801	0,744	0,771
Interactive	**0,861**	**0,861**	**0,808**	**0,865**

Table 4.3: mIoU Comparison on LCM Dataset.

	Deleware	**West Virginia**	**Virginia**	**Pennsylvania**	**Maryland**
Baseline	0,816	0,691	0,794	0,743	0,804
iFCN	0,826	0,704	0,806	0,768	0,811
Interactive	**0,878**	**0,818**	**0,869**	**0,819**	**0,859**

The iterative corrections algorithm seems to help most where the there is particularly bad performance in the base model output, while it is less useful where the errors are small. West Virginia and Chicago are good examples of this. In table 4.3 we can see that the performance increase is 18% on West Virginia.

However, even small mIoU improvements are very useful. In order for the results to be used broadly, for example uploaded to OpenStreetMaps, the segmentation maps need to comprehensively and appropriately cover the area. This method seems to enable that as verified in section 4.4.5.

Effect on Individual Classes

Generally, we can see that the dataset class imbalance has an effect on the performance of the model when looking at the baseline. We show this in table 4.4, noting the accuracy of a class with respect to the proportion of the dataset that class consists of.

Table 4.4: Baseline model's per class accuracy and dataset proportions.

	Water		Forest		Field		Impervious	
	Accuracy	**% of Dataset**	**Accuracy**	**% of Dataset**	**Accuracy**	**% of Dataset**	**Accuracy**	**% of Dataset**
Deleware	81%	1,7%	92%	37,9%	96%	52,4%	83%	8,0%
West Virginia	**54%**	**0,3%**	98%	80,6%	84%	18,0%	**72%**	**1,2%**
Virginia	94%	6,0%	96%	69,5%	87%	22,2%	72%	2,3%
Pennsylvania	75%	1,0%	95%	64,6%	90%	30,1%	**70%**	**4,3%**
Maryland	85%	4,9%	92%	41,2%	89%	45,5%	90%	8,4%

The model struggles with classifying water pixels in West Virginia, where the proportion of water is low (0.3%). Virginia, with a higher proportion of water pixels (6.0%), is able to classify with 94% accuracy. A similar result can be seen with pixels classified as impervious in West Virginia compared to Maryland. We can infer that dataset proportion is likely one reason for the poor performance. Other reasons could be related to the model's ability to generalise to intra-class variation.

Table 4.5: Corrections model's per class accuracy and dataset proportions.

	Water		Forest		Field		Impervious	
	Accuracy	**% of Dataset**	**Accuracy**	**% of Dataset**	**Accuracy**	**% of Dataset**	**Accuracy**	**% of Dataset**
Deleware	90%	1,7%	97%	37,9%	97%	52,4%	90%	8,0%
West Virginia	**81%**	**0,3%**	99%	80,6%	93%	18,0%	**81%**	**1,2%**
Virginia	96%	6,0%	98%	69,5%	94%	22,2%	80%	2,3%
Pennsylvania	86%	1,0%	97%	64,6%	92%	30,1%	**82%**	**4,3%**
Maryland	91%	4,9%	96%	41,2%	92%	45,5%	92%	8,4%

Table 4.5 when compared to table 4.4 shows that the interactive algorithm is clearly able to assist in the model detecting the minority classes. The accuracy of West Virginia jumps from 54% to 81% on the water class, and from 72% to 81% on the impervious class.

This is likely occurring due to the base model detecting the class initially. The provided corrections would enable the corrections model to detect the class and make the appropriate changes.

4.4.3 Out-of-Distribution Performance

The algorithm works in-distribution, but we also want to see if works out-of-distribution. This simulates the scenario of moving to different regions or cities. Interestingly, comparing tables 4.3 and 4.6, we see that the out-of-distribution performance increase, from the interactive corrections algorithm over the baseline, exceeds that of the in-distribution performance increase.

Table 4.6: Comparison of methods in regions not in training data.

	Baseline	**iFCN**	**Interactive**
Kitsap County	0,664	0,684	0,749
New York	0,793	0,795	0,867

The performance in New York seems to be in line with the rest of the results. This was because the baseline model seemed to be able to generalise well from the rest of the dataset. On Kitsap County, however, we see a clear degradation of the base model performance and the interactive corrections algorithm helps significantly.

4.4.4 How to structure corrections

Understanding that the iterative corrections strategy works, the following question arises: how should one structure the corrections? In other words, how many rounds of corrections should one have and how many corrections in each round?

We examine this on New York in the LCM dataset and display the results in figure 4.11.

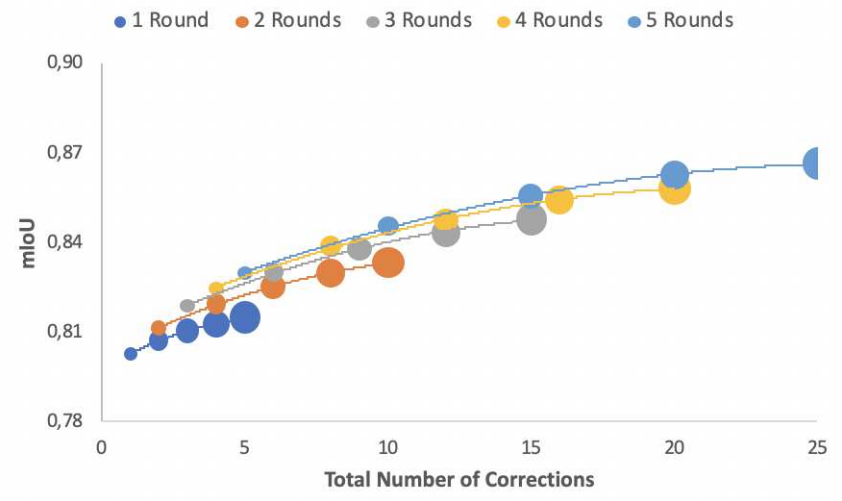


Figure 4.11: A graph showing the relationship between the total number of corrections, the number of rounds and the mIoU in New York in the LCM dataset. Bubble size indicates number of corrections in a round.

The results in figure 4.11 show diminishing returns on the number of clicks. The large errors (in both spatial area and IoU) would be reduced first and then afterwards it would be fine-tuning the smaller errors. This makes sense in the context of the model missing houses, but once corrected it is able to detect the appropriate class and correct the entire segmentation mask.

It is also seen that having a number of rounds of corrections provides better performance than simply having more clicks in a single round. Again, this shows that providing points where the model fails works best. More points in a single round are helpful, but additional points provide less information and mostly assist with edges.

4.4.5 Qualitative Analysis

In this subsection, we extract some results from the models to analyse the utility of the models in various scenarios and provide some qualitative analysis. We highlight a number of scenarios where it seems to work well and also highlight a few failure cases.

Does it work with different size patches?

While we trained on smaller patches of 240×240 pixels; the model and the iterative corrections strategy performs well when we apply it to an image of 576×576 pixels— over 5 times the size of the images we trained on. We see this qualitatively on the Inria dataset in figure 4.12.

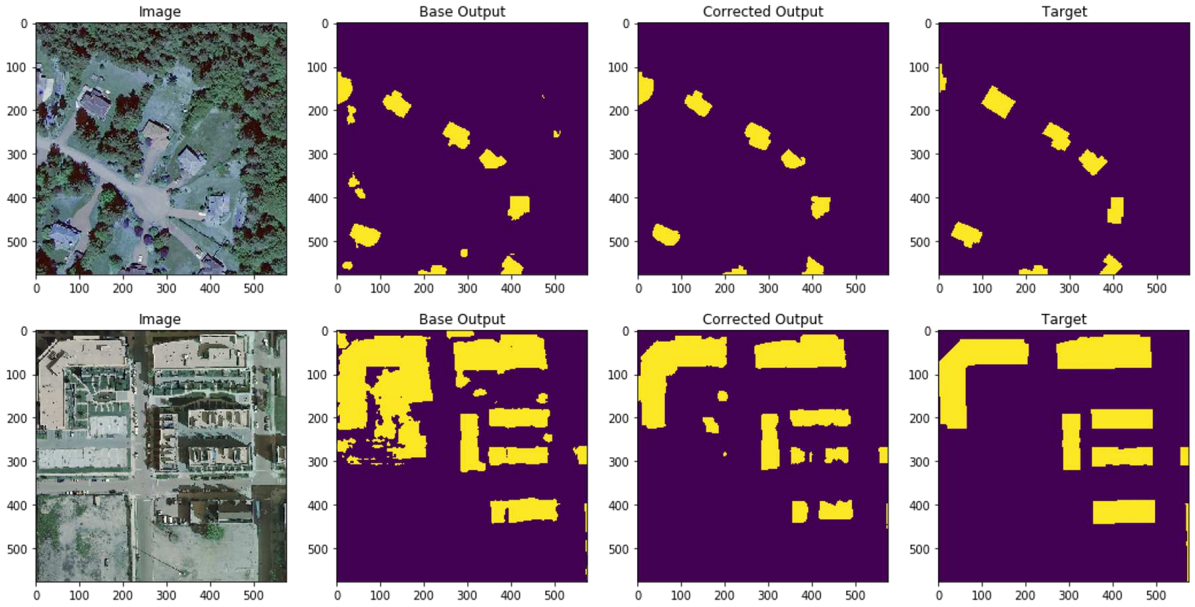


Figure 4.12: Using Model on Bigger Patches on Inria Aerial Image Dataset.

This means the model can be applied to larger patches at a time, as is often required. However, note that this was on data with the same spatial resolution and the models we trained were not trained to be robust to multiple resolutions.

Adaptation beyond the area corrected

The first notable qualitative result is that the iterative corrections model is able to adapt to the entire image, beyond where the initial correction is provided, as seen in figure 4.13. In the examples below we see that the model classifies part of the image as a field. With a single round of corrections, which do not cover all the blobs incorrectly classified, the model corrects the entire image.

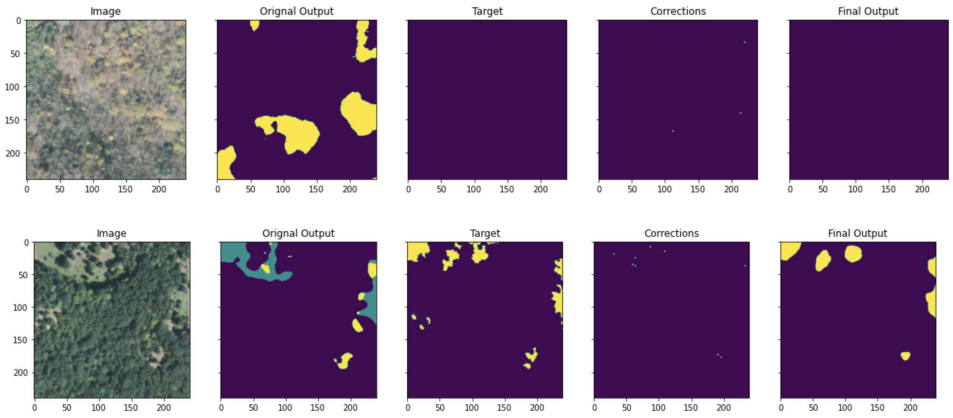


Figure 4.13: Interactive corrections adapting beyond the nearby region area of the corrections.

Multi-class scenario: LCM dataset

In the multi-class scenario in land cover mapping, the algorithm is able to work and improve on the original output from the base model. It is able to change the class when it is incorrect and this results in changes to the entire

image.

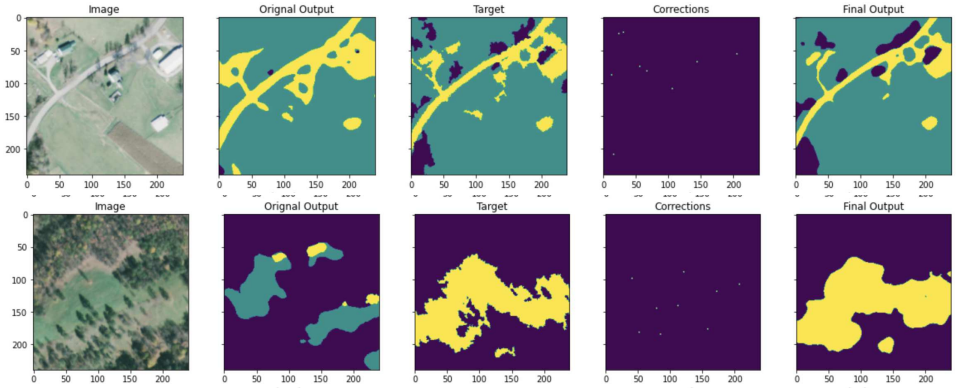


Figure 4.14: Interactive Corrections working in Multi-Class Scenario.

Refinement of segmentation masks

The model is also capable of refinement of the edges and mistakes in individual components, as demonstrated in the top image in figure 4.15. However, it is not as good at this for buildings. Adding to the model ideas like DARNet [Cheng et al., 2019], DSAC [Marcos et al., 2018b] or a loss function prioritising boundaries could further improve the results and the level of refinement the model is able to achieve. In figure 4.15 we see better edges and segmentations from the original output, so the network is taking into account the corrections provided at the edge of the buildings or land cover class.

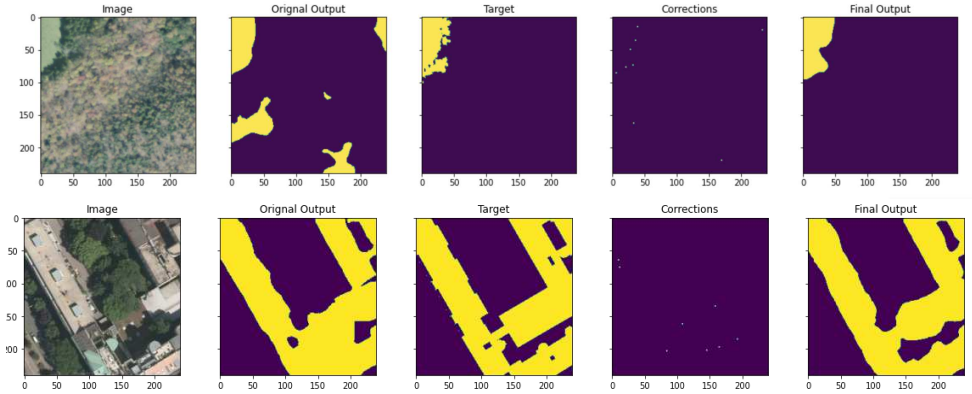


Figure 4.15: Interactive corrections doing refinements.

In the first example in figure 4.16 we see the corrections algorithm making the buildings more accurate, and in the second example we buildings being split up to match the target (and the user input).

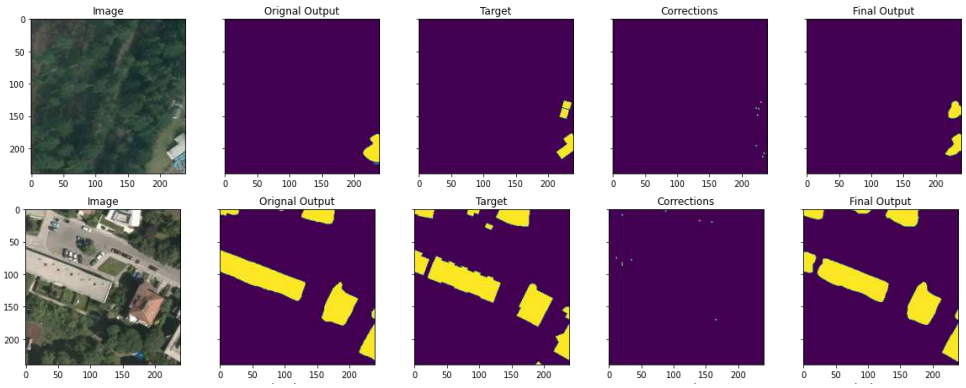


Figure 4.16: Interactive corrections doing refinements.

Failures

It is useful to note at this point that both the datasets, Inria and Chesapeake, are fairly noisy despite the quality assurance process instituted. The Chesapeake dataset stated that it has an estimated accuracy of 90%-95%. One of the failures of this model is that the priors are quite strong. For example, in this first failure example shown in figure 4.17, the target is clearly incorrect. However, despite the corrections provided the model is not able to adapt because it quite clearly sees a delineation between forest and field, when the target is just field.

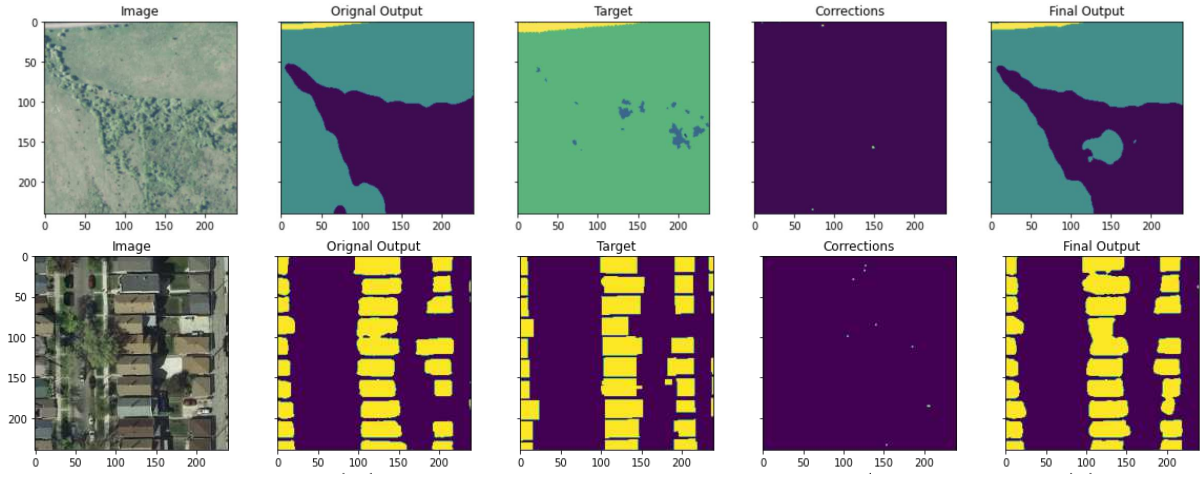


Figure 4.17: Example of failures on the Chesapeake (top) and Inria (bottom) dataset.

The second failure, shown in figure 4.17 is probably more representative of what might happen in failure cases on the Inria dataset. Buildings that were separate (and rightfully so), were joined. One might speculate that a better base/corrections model might assist, but that is unclear from the experiments conducted. This error would cause volunteer mappers to spend time splitting the buildings up.

4.5 Conclusion

In this section, the utility of an iterative corrections algorithm for interactive semantic segmentation of buildings and land cover classes from aerial imagery was investigated. For faster disaster relief mapping, this complements

existing efforts rather than attempting to automate them. We showed that the proposed algorithm achieved an improvement of up to 18% in mIoU and was robust, working both in-distribution and out-of-distribution. This means it can be used to annotate images in new regions for uploading to mapping databases. The annotated images can also be used to train more accurate machine learning models. The algorithm can thus be easily packaged and extended for use within the mapping process. This would reduce the time annotators spend mapping and increase the speed at which disaster relief organisations can respond to disasters.

The biggest limitation of the work is that the models need to be pre-trained on the categories of the classes to be used. Work such a graph cuts, despite their limitations, were able to label a single object at a time and that object could be anything — not necessarily defined as an object in the initial training of the model(s).

Further work that could be pursued to improve this paradigm include incremental learning, where the base and corrections models would adapt in perpetuity given the corrections. This is a very difficult deep learning problem that remains largely unsolved in the broader context, as these models require large datasets and are not amenable to training on single images.

Chapter 5

Active Learning

Our aim is to decrease the amount of time volunteers are required to spend labelling remote sensing data to obtain the required mapping features. If we are able to develop a model that can be trained from fewer labelled samples then this would be ideal. Active learning aims to do this by selecting what should be training data from the enormous pool of unlabelled data.

We consider a pool-based active learning scenario, where the pool of unlabelled samples remains fixed throughout the process. There are other active learning scenarios, such as stream-based and membership query synthesis.

For pool-based active learning, the active learning loop is visualised in figure 5.1.

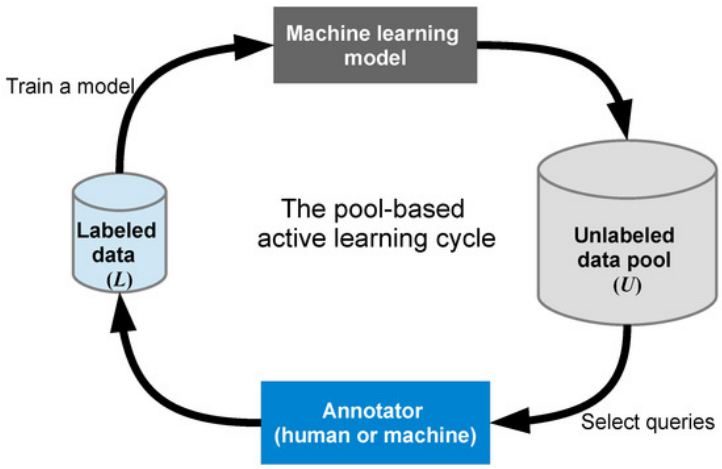


Figure 5.1: Pool based active learning loop [Yang et al., 2017a].

Consider the following: you have a large pool of unlabelled data, X_U. In an active learning loop, one samples data from the unlabelled pool using a sample acquisition strategy, until a budget, b, is reached. The selected samples are annotated by an oracle (a human or machine annotator who provides the ground truth) and added to the the labelled pool, (X_L, Y_L). The machine learning model is then trained on the entire labelled data pool.

After training the model, you continue this process to expand the labelled dataset using the sample acquisition strategy. The idea is that the sample acquisition strategy should select the most informative samples for a model to train on— such that one is able to arrive at a high model accuracy with fewer labelled samples. This has been a longstanding problem within pattern recognition.

Active learning worked relatively well prior to the advent of deep learning methods and there were theoretical

guarantees for when it might or might not work. However, with deep learning models, traditional sample acquisition strategies have been shown to have limited effect. Deep learning models require massive amounts of labelled data to be trained effectively, and labelled data is expensive. Thus active learning has become a vibrant research topic within the machine learning and related communities.

Some early work in the area has shown promise on toy problems (e.g. MNIST [LeCun et al., 1998]) and natural images (e.g. CIFAR [Krizhevsky and Hinton, 2009], ImageNet [Deng et al., 2009] for classification and BDD [Yu et al., 2018] and CityScapes [Cordts et al., 2016] for semantic segmentation), from the uncertainty-based approaches to diversity-based approaches and their combination [Sinha et al., 2019]. Despite the interest in active learning for deep learning, there have not been any studies on applying such approaches to remote sensing in either a land cover mapping or building footprint segmentation context. Furthermore, these methods have not been evaluated in more realistic scenarios, such as when transferring a model.

In this chapter we attempt to apply a variety of active learning methods to land-cover mapping and building footprint segmentation. We evaluate and compare the performance of a number of sample acquisition strategies in the remote sensing datasets. We also examine the performance of the active learning strategies with two different pre-training approaches to see if strategies work any better.

Section 5.1 presents literature on active learning with a focus on deep learning. It also touches on active learning for remote sensing and semantic segmentation.

Section 5.2 presents the sample acquisition strategies used in our experiments, with section 5.3 explaining the modifications if any when applied to semantic segmentation of remote sensing data.

Section 5.4 presents the experimental design used to answer some of the questions listed above. It includes details on how certain techniques were implemented and how the protocols for active learning were changed to answer particular questions.

Finally, section 5.5 presents the results of the experiments and a discussion of the results. We found that while none of the algorithms worked universally better than the others, the results provided some intuition on when a particular strategy might work better. It was also noted that there was significant variability in the results and that random sample acquisition strategy often performs on par with the sample acquisition strategies we compared.

We conclude with a summary, along with recommendations and potential avenues for future research.

5.1 Related Work

Active learning sample acquisition strategies can largely be grouped into two classes of methods: uncertainty-based methods and diversity-based methods. A number of strategies then use a combination of the above or learn a new strategy entirely. Further, it should be noted that active learning is traditionally pursued in a classification context, with a few modifications made for a segmentation task.

Subsection 5.1.1 presents an overview of active learning prior to deep learning. This subsection groups work by the classes described above: uncertainty-based, diversity-based and other methods.

Subsection 5.1.2 presents an overview of active learning methods that incorporate deep learning. This subsection also groups work by the classes described above: uncertainty-based, diversity-based and other methods.

Subsection 5.1.3 presents work in deep active learning applied to semantic segmentation specifically.

Subsection 5.1.4 presents an overview in active learning applied to remote sensing applications. Lastly, subsection 5.1.5 provides an overview of active learning when some sort of pre-training has previously been done.

5.1.1 Prior to Deep Learning

A comprehensive literature review of active learning methods prior to the advent of deep learning is provided by Settles [2014]. We provide a more basic overview, noting some of the important methods. In comparison to methods that incorporate deep learning, these methods typically have stronger empirical results and theoretical justification.

Uncertainty

The most common class of sample acquisition strategies is uncertainty-based. In this class of methods, the active learner will request samples the model is uncertain about. For classification and segmentation tasks, entropy [Shannon, 1948], max-margin [Scheffer and Wrobel, 2001] and least confidence are among the most common heuristics. Query-by-committee [Freund et al., 1997] is a strategy in which a committee (or ensemble) of models are trained, and the samples that the committee most disagree on are selected. Strategies along these lines typically make use of the the previously listed uncertainty heuristics as methods to compute the disagreement. Related to this, Houlsby et al. [2011] proposed Bayesian Active Learning by Disagreement (BALD) which uses mutual information between the model predictions and the model parameters as a heuristic to compute the disagreement for Gaussian processes.

Diversity Based

Diversity-based sampling is concerned with selecting the fewest samples that would most represent the underlying dataset. One of the challenges of this approach is that it requires a good representation of the underlying dataset to best select samples. It does, however, allow for leveraging the unlabelled samples to help build a representation of the dataset. Examples of this approach include Xu et al. [2009] who used clustering to assist active learning for support vector machines, and Settles and Craven [2008] who used density-weighting.

Other Approaches

A number of other approaches have also been proposed that fall outside the above two categories. Hybrid approaches which combine the two have been proposed. Other methods include Kapoor et al. [2007] who introduce a strategy that aims to maximise the expected model improvement for Gaussian processes, and Roy and McCallum [2001] who propose an error reduction strategy for a naive-Bayes classifier.

5.1.2 With Deep Learning

Unfortunately, many of the ideas and results have not translated well to deep learning models. Among the issues is that deep learning models are trained on batches of data. This requires designing active learning sampling strategies which can query several samples at once. In addition, most of the previous uncertainty sampling methods have been specifically shown to perform poorly with deep learning models due to poor calibration.

Uncertainty-Based

As mentioned previously, the traditional uncertainty-based strategies, such as maximum entropy, confidence and min-max margins, have been investigated in deep learning in a variety of contexts including classification [Sener and Savarese, 2018] and segmentation [Sinha et al., 2019, Siddiqui et al., 2019]. Of particular note, Sener and Savarese [2018] showed that these sample acquisition strategies do not translate to CNNs and struggle with non-toy

datasets. More generally, Gal and Ghahramani [2016], Kendall and Gal [2017], Lakshminarayanan et al. [2017] showed that the softmax outputs of the neural networks are often poor estimates of uncertainty. This cripples uncertainty-based acquisition strategies that leverage heuristics like entropy.

Amongst the notable work in using uncertainty for active learning with deep learning, was Gal et al. [2017]. They use the the result of Gal and Ghahramani [2016], which approximates the posterior distribution over the weights in the network using Monte Carlo dropout (MC-dropout), to provide better calibration. They then use BALD as the active learning sample acquisition strategy. This approach showed improvements on MNIST and diabetic retinopathy datasets, which are relatively small datasets. Lakshminarayanan et al. [2017] extend Monte Carlo dropout by using a ensemble of neural networks, which produces better calibrated results. Beluch et al. [2018] experiment with various uncertainty heuristics to compute disagreement with ensembles for active learning. The traditional heuristics like entropy have also been tried using the MC-dropout approximation [Siddiqui et al., 2019].

Other methods have been proposed using committees or ensembles to represent uncertainty and in turn use that for active learning [Kuo et al., 2018, Yang et al., 2017b].

Diversity-Based

Among the earlier methods that showed promise within the context of deep learning models were diversity-based methods from Dutt Jain and Grauman [2016], Sener and Savarese [2018]. However, their effectiveness seem to wane in the face of increasing task and dataset complexity [Sinha et al., 2019].

Core-sets [Sener and Savarese, 2018] aims to minimise the Euclidian distance between samples in the labelled pool and unlabelled pool in some feature space. This approach was amongst the first methods that seemed to have significant improvement in large scale image classification tasks. However, for high-dimensional data and datasets with many classes using a distance-based approach becomes impractical and ineffective [Sinha et al., 2019]. Other diversity-based methods seek to model the unlabelled sample space more directly, leveraging the large set of unlabelled samples. A more recent approach is variational adversarial active learning [Sinha et al., 2019]. In VAAL, they train a VAE to learn a latent space and a discriminator to classify samples as labelled or unlabelled, from the latent space generated by the VAE.

Other approaches

As with the pre-deep learning approaches, a number of methods combine the above approaches. Wang et al. [2017] introduced a weakly-supervised algorithm that incorporates pseudo-labels into a semi-labelled pool that the model trains on. These samples are obtained from the unlabelled pool when the model has high confidence on the sample. Kirsch et al. [2019] showed an improvement on BALD for classification on MNIST, by combining BALD with diversity criteria, but their method becomes extremely computationally expensive with large images and large datasets. Of the newer and more different algorithms, Casanova et al. [2020] used deep reinforcement learning (with the DQN network) for active learning. While novel and interesting, it failed to provide significant improvement over baselines.

5.1.3 Active learning for Semantic Segmentation

Active learning for semantic segmentation has been mostly investigated in the context of natural images and medical images. The topic within the medical imaging community is particularly popular given the significant cost of collecting data and labels. Medical images require specialist doctors, whose time is valuable and it is therefore costly to provide the segmentation masks.

Among the approaches taken is Kuo et al. [2018] who propose a strategy using uncertainty from an ensemble of models, and suggestive annotation [Yang et al., 2017b] which does the same but incorporates core-sets for choosing representative data points. Gorriz et al. [2017] use Monte Carlo dropout instead of ensembles. Sinha et al. [2019] showed their VAAL strategy to work on semantic segmentation, along with core-sets and max-entropy strategies. Other strategies focus on obtaining smaller parts of the image like super-pixels or patches for labelling, instead of full images, and use similar uncertainty heuristics as above [Siddiqui et al., 2019, Mackowiak et al., 2018]. These approaches also incorporate an idea of cost associated with labelling certain components of an image into their criterion of choosing what patches of each sample to label.

5.1.4 Active Learning for Remote Sensing Applications

There is a significant body of literature for active learning for remote sensing in both the pre and post deep learning era [Geib et al., 2018, Rajan et al., 2006, Wan et al., 2015, Persello and Bruzzone, 2014, Liu et al., 2017, Patra and Bruzzone, 2014, Tuia et al., 2011a]. The work is largely derivative of work in the broader machine learning and computer vision communities, with ideas of uncertainty, diversity and a combination of the two. The work consists almost exclusively on classification as a task. Tuia et al. [2011b] provides a comprehensive survey of active learning for remote sensing classification prior to deep learning. We do not delve deeply in the these topics as we focus here on semantic segmentation. Of the more interesting active learning papers with respect to deep learning, was the the work by Lu et al. [2017], who incorporated spatial information. They use specific knowledge of remote sensing to improve their classification output. Robinson et al. [2019b] present some of the first work using deep learning and active learning on remote sensing imagery. They look at a rather limited number of active learning approaches within a weak-supervision context and compare it to human chosen samples, finding that humans choose better samples. They did however use very simplistic active learning heuristics.

5.1.5 Active Learning after Pre-Training

Active learning for adaptation to new regions, after pre-training, has not been a widely-considered scenario. Wang et al. [2017], Beluch et al. [2018] both used pre-trained ImageNet weights for all their experiments. Huang and Chen [2016] looked at active learning for domain adaptation; with access to the data both in the source and target domain. Muthakana and Singh [2019] did, in parallel to the experiments we conducted, some experiments on transfer learning with active learning using ImageNet and CIFAR pre-trained weights. There have also been a number of papers that do semi-supervised active learning, in which models make use of unlabelled data and labelled data [Wang et al., 2017, Berardo et al., 2015, Rhee et al., 2017, Gao et al., 2020]. Prior to deep learning, Bruzzone and Fernàndez Prieto [1999] used an uncertainty-based approach with a support vector machine model, and considered the remote-sensing scenario of moving to a new region. The work of Bruzzone and Fernàndez Prieto [1999] reinforces the utility of our formulation of the problem.

5.2 Acquisition Strategies

We investigate a number of acquisition strategies proposed. These methods are either standard baselines in the field or proposed state-of-the-art algorithms. We test a number of uncertainty-based approaches (softmax max-entropy, softmax max-entropy with MC-dropout and BALD with MC-dropout), two diversity-based approaches (core-sets and VAAL) along with a random baseline approach. The strategies and how they are adapted for semantic segmentation, if any adaptation is made, are explained in this section.

5.2.1 Random

Using the random acquisition strategy, in each active learning loop the samples acquired for labelling are randomly selected samples from the unlabelled pool. We implement this using Numpy's [Van Der Walt et al., 2011] random sampler.

5.2.2 Softmax Max-Entropy

Shannon entropy [Shannon, 1948] is a long-standing measure of uncertainty. It measures the amount of information present in a prediction, taking into account the probabilities of the prediction for each class. Entropy is given by the following equation:

$$\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] = -\sum_{c}(p_c)\log(p_c), \tag{5.1}$$

where c is the number of classes and p_c is the softmax probability that the pixel segmented is of class c.

As we are doing semantic segmentation, this will output a single entropy value for each pixel. In order to decide whether we should sample the image or not, we sum up the entropy values for each pixel up to get the entropy value for the whole image. We then select samples that produce the greatest entropy.

5.2.3 Softmax Max-Entropy with MC-Dropout

It has been well established that the softmax probabilities output by neural networks are not well calibrated, and do not represent uncertainty well [Gal and Ghahramani, 2016]. Dropout is a regularisation technique used in training deep neural networks, where random units of the network are dropped during training (different units for each mini-batch). When applying a model at test time no units are dropped. However, [Gal and Ghahramani, 2016] shows that sampling with dropout at test time approximates a Bernoulli distribution over the model's weights. This technique is known as Monte Carlo dropout or MC-dropout. We provide further explanation of MC-dropout and its implementation in section 5.4.2.

For softmax max-entropy with MC-dropout, we average the results of softmax outputs from the MC-dropout passes to obtain better uncertainty estimates. We then use these to obtain the the entropy per pixel, which we sum up and use to select the batch of images to be labelled. Softmax entropy with MC-dropout can thus be given by

$$\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] \approx -\sum_{c}(\frac{1}{T}\sum_{t}p_c^t)\log(\frac{1}{T}\sum_{t}p_c^t), \tag{5.2}$$

where c is the number of classes, T is the total number of MC-dropout passes and p_c^t is the softmax probability that a pixel is of class c in the Monte-Carlo pass t.

5.2.4 Bayesian Active Learning by Disagreement using MC-Dropout

BALD [Houlsby et al., 2011] is an acquisition strategy that estimates the mutual information between the model predictions and the model parameters [Kirsch et al., 2019]. It was introduced prior to deep learning, but Gal et al. [2017] adapted it for convolutional neural networks. The equation for calculating BALD is given as

$$\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_{\text{train}}] = \mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\big[\mathbb{H}[y|\mathbf{x}, \boldsymbol{\omega}]\big], \tag{5.3}$$

and can be approximated in deep learning models with Monte Carlo Dropout as follows:

$$\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_{\text{train}}] \approx -\sum_c (\frac{1}{T}\sum_t p_c^t) \log(\frac{1}{T}\sum_t p_c^t) + \frac{1}{T}\sum_{t,c} p_c^t \log(p_c^t), \tag{5.4}$$

where c is the number of classes, T is the total number of MC-dropout passes and p_c^t is the softmax probability that a pixel is of class c in the Monte-Carlo pass t.

Pixels that maximise BALD are pixels where the entropy is high, while the expectation of the entropy is low. This occurs when the model has many ways to explain the data, which means that the MC-dropout passes are disagreeing among themselves [Kirsch et al., 2019].

5.2.5 Core-Sets

Core-Sets [Sener and Savarese, 2018] is a diversity-based approach. This strategy aims to select a subset of points, the budget b, such they provide the smallest δ to cover the set of all points in some feature space. This is visualised in figure 5.2.

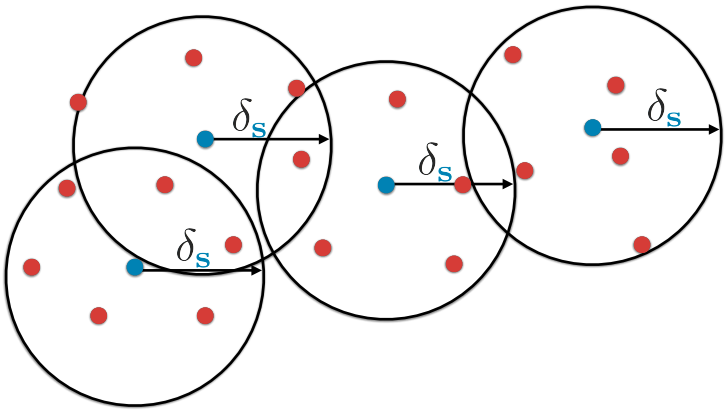


Figure 5.2: Core-sets visualisation from Sener and Savarese [2018].

Effectively, the core-sets acquisition strategy minimises the maximum distance between samples in the unlabelled pool and samples in the labelled pool. The simple k-centre greedy strategy, shown in algorithm 1, is used to obtain select samples in each active learning loop.

Algorithm 1 k-center-greedy [Sener and Savarese, 2018]

Input: data $\mathbf{x}_U$, existing pool $\mathbf{s}^0$ and a budget b
Initialise $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{L \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_U, \mathbf{x}_L)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

For the distance metric, Sener and Savarese [2018] use the Euclidean distance between the features. In their experiments on natural images, they used the output activation features of the VGG network that they trained on as the features for the samples. One could use pre-trained features as we do in this case. We experimented with two features: one generated by a VAE and the other generated by Tile2Vec.

5.2.6 Variational Adversarial Active Learning

In variational adversarial active learning or VAAL, Sinha et al. [2019] train a variational autoencoder to learn a latent space and a discriminator to classify samples as labelled or unlabelled. A mini-max game is then played between the VAE and the discriminator such that the VAE aims to fool the discriminator into classifying all data points from the labelled pool, while the discriminator learns the latent space generated by the VAE to discriminate between labelled and unlabelled data. In the acquisition phase, the discriminator is applied to the latent embeddings of the unlabelled pool to decide which samples to acquire.

They train the VAE with latent dimension of between 32 and 128, depending on the size of the images. VAAL uses transductive learning on unlabelled data to learn features that would otherwise be missing if just trained on the labelled pool. The loss function of the VAE with transductive learning is formulated as follows:

$$\begin{aligned} L_{\text{VAE}}^{trd} = \quad & \mathbb{E}[\log p_\theta(x_L|z_L)] - \mathrm{D}_{\text{KL}}(q_\phi(z_L|x_L)||p(z)) \\ & + \mathbb{E}[\log p_\theta(x_U|z_U)] - \mathrm{D}_{\text{KL}}(q_\phi(z_U|x_U)||p(z)), \end{aligned} \tag{5.5}$$

where q_ϕ and p_θ are the encoder and decoder, respectively. The prior, $p(z)$, is a unit Gaussian.

Through transductive learning, the representation learnt by the VAE includes latent features associated with both labelled and unlabelled data. The discriminator then learns to distinguish between labelled and unlabelled data from the the latent features. The VAE learns in an adversarial fashion to improve the representations and emphasise important features that distinguish between the pools of data. To do this an adversarial loss is formulated as follows:

$$L_{\text{VAE}}^{adv} = -\mathbb{E}[\log(D(q_\phi(z_L|x_L)))] - \mathbb{E}[\log(D(q_\phi(z_U|x_U)))], \tag{5.6}$$

where D is the discriminator.

Thus, the full loss function for the VAE in VAAL is given by:

$$L_{\text{VAE}} = \lambda_1 L_{\text{VAE}}^{trd} + \lambda_2 L_{\text{VAE}}^{adv}, \tag{5.7}$$

where λ_1 and λ_2 are hyper-parameters that determine the effect of each loss component. The reparameterisation trick is used to calculate the gradients [Kingma and Welling, 2013].

The loss function of the discriminator is simply binary cross entropy, distinguishing between unlabelled and labelled data:

$$L_D = -\mathbb{E}[\log(D(q_\phi(z_L|x_L)))] - \mathbb{E}[\log(1 - D(q_\phi(z_U|x_U)))]. \tag{5.8}$$

Lastly, in addition to training the VAE and discriminator, we must simultaneously train the semantic segmentation model, which VAAL refers to as the task model. We show the VAAL algorithm as provided by Sinha et al. [2019] in algorithm 2:

Algorithm 2 Variational adversarial active learning [Sinha et al., 2019]

Input: labelled pool (X_L, Y_L), unlabelled pool (X_U), initialised models for θ_T, θ_{VAE}, and θ_D

Input: Hyper-parameters: epochs, λ_1, λ_2, α_1, α_2, α_3

for $e = 1$ to epochs **do**
  sample $(x_L, y_L) \sim (X_L, Y_L)$
  sample $x_U \sim X_U$
  Compute L_{VAE}^{trd} using Eq. 5.5
  Compute L_{VAE}^{adv} using Eq. 5.6
  $L_{\text{VAE}} \leftarrow \lambda_1 L_{\text{VAE}}^{trd} + \lambda_2 L_{\text{VAE}}^{adv}$
  Update VAE by descending stochastic gradients:
  $\theta'_{VAE} \leftarrow \theta_{VAE} - \alpha_1 \nabla L_{\text{VAE}}$
  $L_{\text{D}} \leftarrow L_{\text{BCE}}(q_\phi(z_L|x_L)), \mathbb{1}) + L_{\text{BCE}}(q_\phi(z_U|x_U)), \mathbb{0})$
  Compute L_{D} using Eq. 5.8
  Update D by descending its stochastic gradient:
  $\theta'_D \leftarrow \theta_D - \alpha_2 \nabla L_{\text{D}}$
  Train and update T:
  $\theta'_T \leftarrow \theta_T - \alpha_3 \nabla L_{\text{T}}$
end for
return Trained $\theta_T, \theta_{VAE}, \theta_D$

Acquisition strategy

Once the VAE and discriminator have been trained, they are run on the unlabelled pool. VAAL uses the discriminator's softmax output as the probabilities, ranking the samples from the highest probability of being unlabelled to the lowest. It then selects the samples that are most likely to be unlabelled, until the budget has been reached.

Chosen Hyper-parameters

Following the suggestions provided by Sinha et al. [2019], we set the parameters to the following:

- **Models**: The VAE is the Wasserstein autoencoder [Tolstikhin et al., 2018] and the discriminator is a 3-layer MLP. This is per the original implementation. The task model is the ResNet18-based FCN. The VAE and discriminator architectures are shown in figure 5.3.
- **Optimiser**: we use Adam as per the original paper.
- **Learning rates**: α_1 (VAE), α_2 (discriminator) and α_3 (task module— in this case ResNet18-based FCN) are set to 0.0005, 0.0005 and 0.001 respectively.
- **Regularisation weights**: λ_1 is set to 1 and λ_2 is set to 20.
- The latent dimension of the VAE, z, is set to 64.

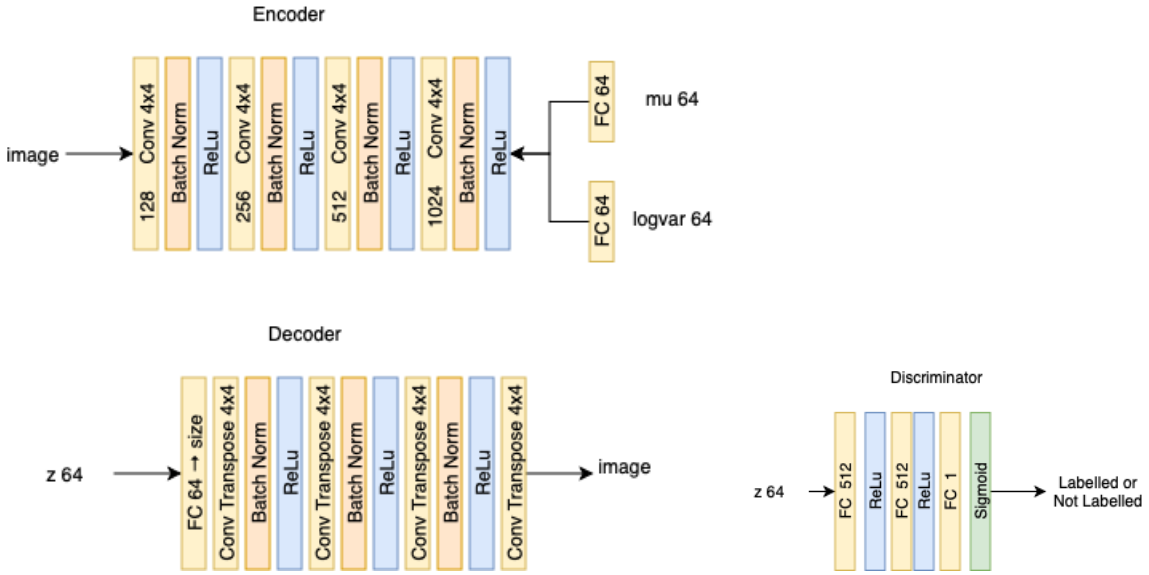


Figure 5.3: Wasserstein autoencoder architecture and the discriminator used in the VAAL implementation.

5.3 Acquisition Strategies Aggregation

The uncertainty-based heuristics described provide uncertainty on a per-pixel level. In order to decide on the overall uncertainty of an image, one needs to aggregate the per-pixel uncertainty to a per-image uncertainty, which can then be used to sort and decide which images to label. We do this aggregation by summing the uncertainty of the pixels.

The diversity-based approaches described above provide an overall score per image, which can then be used to decide which images to label. They do not require any aggregation.

Many active learning algorithms for semantic segmentation, instead of acquiring labels of whole images, ask for labels for smaller areas of the images. For the disaster relief and land cover mapping datasets we have, the patches are already fairly small in comparison to overall sizes of aerial images, and in addition one needs some context to do the labelling. Further, in general [Casanova et al., 2020, Siddiqui et al., 2019] it seems that while smaller patches (or superpixels) in natural images improve absolute performance; the relative performance is largely similar. A final consideration, which was an important consideration when conducting these experiments, is the computational burden of using smaller regions. One is then required to get the scores (from the uncertainty or diversity heuristic) for each region, often on a sliding window basis, and aggregate per region. This adds significant computational cost to all methods.

For these reasons we choose to use the patches when comparing these heuristics.

5.4 Experimental Design

In this section, we provide further details on implementation and explain some of the questions we try to answer with our experiments.

Subsections 5.4.1, 5.4.2 and 5.4.3 provide further details on the implementation of how the models are trained, how uncertainty is calculated in the FCNs and how we generate features used for the core-sets acquisition strategy.

Subsections 5.4.4, 5.4.5 and 5.4.6 explain some of the questions we would like to answer, and further explain the experimental procedure for answering those questions. These questions include how active learning performs on

the remote sensing datasets, and how pre-training affects the active learning strategies.

An important note on the experiments: while many were conducted on both datasets, some were only conducted on a single dataset. The reason for this is that these experiments require significant computational resources, which we had limited access to. Regardless, the results show that the analysis is often transferable across the datasets.

5.4.1 Basic Experiment Implementation

The basic implementation for all experiments and acquisition strategies is listed below:

- **Model:** For all the experiments, on both the Inria and LCM datasets, we used the ResNet18-based FCN previously described.
- **Optimiser:** We used Adam with a maximum learning rate of 0.001.
- **Initialisation:** This depended on the experiment run.
- **Batch size:** we use a batch size of 32. This size fits on the GPU memory, while also being small enough that we can have multiple batches for each budget (e.g. 1% of the training data for a region on the Inria dataset is 144 patches, which makes 4 batches). We shuffle the batches for each epoch.
- **Loss function:** Soft-IoU loss.
- **Epochs:** We ran the experiment for 100 epochs or to convergence when using smaller budgets (5% or less of the training data), or 30 epochs with larger budgets (more than 5% of the training data).
- **Runs:** In order to account for the stochasticity, we do 2 or 3 runs of each experiment and report the mean (and standard deviation where appropriate).

These experiments were implemented using PyTorch [Paszke et al., 2019] and Python. Where parameters are not specified the defaults were used.

5.4.2 Calculating Uncertainty

As mentioned in the related work, calculating uncertainty with deep neural networks requires more than just the softmax outputs. These outputs do not represent well-calibrated measurements of uncertainty. Dropout [Srivastava et al., 2014] is a regularisation technique where during training random layers or units in the neural network are dropped (i.e. weights connecting to the layer or unit are set to 0). At test time, however, no units are dropped. However, Gal and Ghahramani [2016] show that using dropout at test time approximates a Bernoulli distribution over the model's weights. This is achieved by sampling predictions from the model with dropout enabled. This technique is known as Monte Carlo dropout (MC-dropout). This technique can also be considered an implicit ensemble [Lakshminarayanan et al., 2017].

Kendall and Gal [2017] adapt SegNet [Badrinarayanan et al., 2015], to create a Bayesian formulation that allows for some modelling of uncertainty. They train SegNet with dropout layers and sample the posterior distribution over the weights at test time using MC-dropout.

In their design of Bayesian SegNet, they note that a fully Bayesian network should be trained with dropout after every convolutional layer, but found the regularising effect to be too strong which caused undesired effects (slow learning and less accuracy). They found the best configuration had dropout layers inserted in deepest halves of the encoder and decoder. Figure 5.4 shows this configuration.

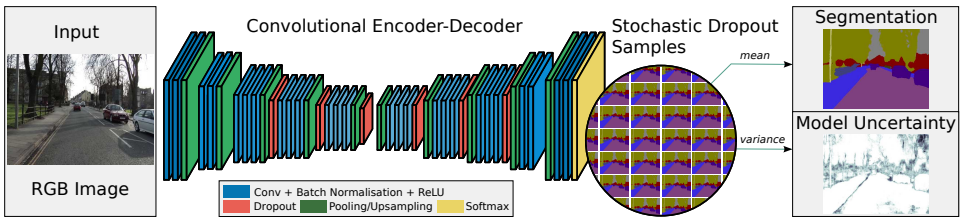


Figure 5.4: Bayesian SegNet architecture [Kendall and Gal, 2017].

At sample acquisition time, the unlabelled data pool is passed through model, with random dropout enable. The resulting predictions combined are better calibrated in terms of representing uncertainty. We use these outputs to calculate the various uncertainty heuristics.

We follow the design of SegNet to create a Bayesian ResNet18-based FCN, as illustrated in figure 5.5. We set the dropout probability to 0.1 (following their work) and use 5 Monte Carlo passes, as Lakshminarayanan et al. [2017] found 5 passes to be sufficient to obtain well calibrated results.

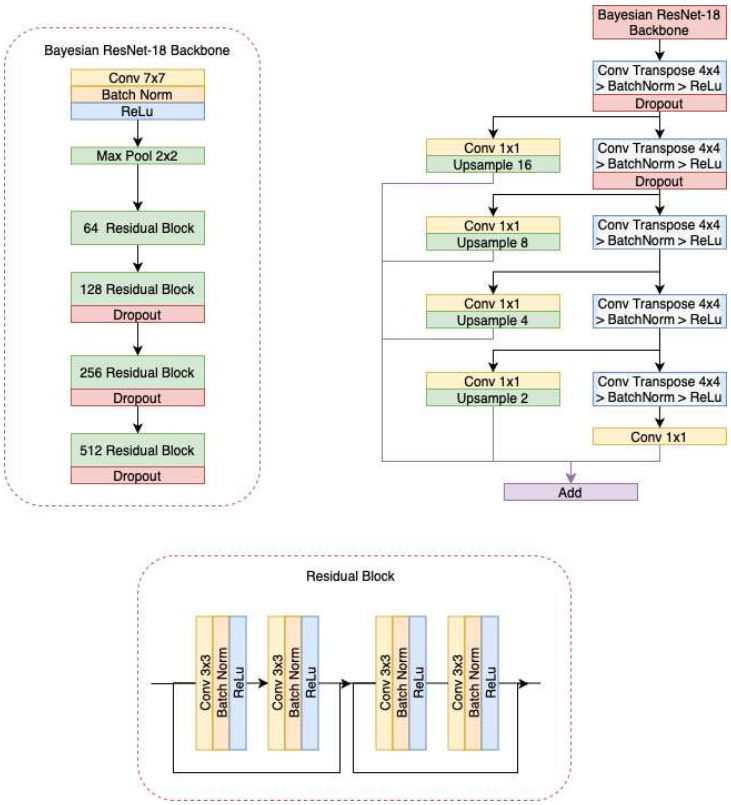


Figure 5.5: Bayesian ResNet18-based FCN.

5.4.3 Features for core-sets

In the original core-sets method, the authors used the activations of the last layer of their VGG classification network as features. In a fully convolutional segmentation network we cannot easily produce such compact features. Instead we look at two approaches of generating the features: Tile2Vec and VAE.

Both of these are unsupervised methods of obtaining representations. In Tile2Vec, the distributional hypothesis is used to train the network and the VAE is trained as an autoencoder to reconstruct the output. Either way both methods have been shown to be effective methods of representation learning.

We examine both and train them in the following fashions.

VAE

Two models are trained. One for the Inria dataset and one for the LCM dataset.

- **Model:** We use the Wasserstein auto-encoder [Tolstikhin et al., 2018] architecture with a latent dimension, z, of 64. This was done for simplicity as it matches the VAE used by VAAL.
- **Initialisation:** The model is initialised using the He initialisation.
- **Loss function:** Mean squared error for reconstruction loss and KL Divergence for the loss of the latent distribution, per Kingma and Welling [2014]:

$$L_{recon}^{(i)} = -\sum_{k=1}^{K}(y_k^{(i)} - x_k^{(i)})^2,$$

$$L_{latent}^{(i)} = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^{(i)})^2 - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2),$$

$$L = L_{latent}^{(i)} + L_{recon}^{(i)}.$$

- **Dataset:** Full Inria or LCM dataset.
- **Epochs:** For each active learning loop we trained for 30 epochs or until convergence. We found that the VAEs converged well in under 30 epochs.

Tile2Vec

We trained our Tile2Vec model largely as per the specification of the original paper.

- **Model:** ResNet18-based CNN adapted for CIFAR10. Fully convolution, but generates a latent dimension of 512. We note that this is larger than the latent dimension of the VAE, but note that the objective when training is different.
- **Initialisation:** The model is initialised using the He initialisation.
- **Loss function:** Augmented triplet loss, as provided in the background section.
- **Dataset:** Full Inria or LCM dataset. We created Tile2Vec samples using their Tile2Vec sampling strategy (for each set, sample a patch at random, sample a nearby neighbour and sample a patch that is far away (either from the same tile or from a random other tile).
- **Epochs:** For each active learning loop we trained for 30 epochs.

5.4.4 Overall Performance when Training from Scratch

The first question we want to answer is how do these methods fare on the remote sensing datasets. These datasets differ significantly from traditional classification datasets (MNIST, ImageNetand diabetic retinopathy) and natural image segmentation datasets (BDK and CitiScapes). The images in remote sensing not only appear very different, including different bands as in the case the LCM dataset, but they are highly imbalanced and contain similar data.

Our hypothesis is that diversity-based methods like core-sets and VAAL might work better than uncertainty-based methods, as they would try to include data that looks different to current samples and thus improve the

representations. This would the make the models' representations more robust to intra-class differences and inter-class similarities.

To test this we looked at a region in each dataset and tested how all the algorithms worked on them. We looked at Kitsap County in the Inria dataset and West Virginia in the Chesapeake dataset.

Here we consider the standard active learning regime, training a model from scratch for each budget. After each round the model selects a number of new samples (i.e. the budget), b, to add to the labelled pool, X_L, Y_L.

For these experiments, we used a VAE pre-trained on all the data within the dataset to generate features for the core-sets strategy.

The protocol was as follows:

- **Initialisation:** The model is initialised using the He initialisation.
- **Initial budget::** 1% of the data.
- **Maximum labelled pool size:** 5% of the data.
- **Budget:** 1% of the training data for each active learning iteration. Note that specific sizes in relation to each dataset used:
    1. Kitsap County in the Inria Dataset: the dataset size of each city is 14400 samples. Thus the budget is 144.
    2. West Virginia in the LCM Dataset. The dataset size of each region is 15000 samples. Thus the budget is 150.
- **Initial samples:** As we are training from scratch, we used a random selection of samples for the initial selection. However, as we used a fixed pre-trained VAE to generate features for the core-sets acquisition strategy, we can use core-sets acquisition strategy to obtain the initial samples for that strategy.

We also compared the strategies on Vienna and New York, but did not compare the VAAL strategy. This was due to the computational burden that VAAL imposes, and based on the initial results, comparing VAAL on these regions was unlikely to provide additional insight.

5.4.5 Effect of Pre-Training

A real-world scenario that the remote sensing community faces is transferring between regions. This has largely been ignored in active learning literature.

In many cases one has access to a model trained on a different region, with significant amounts of data. How do active learning strategies weigh-up in this context? How many samples can we use to have the model work in the new region?

In another case, we might have access to significant amounts of unlabelled data from a range of regions, including the one we are interested in transferring to. In this case, what is the benefit in pre-training using some sort of self-supervised method and then using active learning?

We would like to evaluate how well these pre-training methods work with active learning. Does it make active learning work better? Does it favour a particular strategy? We also compare the two pre-training methods, looking at the difference in absolute performance. Certainly, if unsupervised pre-training works well and reduces the need for labelled samples then it would be of great assistance.

Note that VAAL cannot be easily applied in these scenarios, as the VAE requires access to all the data that the model has been previously trained on. Having access to all the labelled and unlabelled data is core to training the VAE in an adversarial fashion, and to transductive learning. This makes the VAAL method not easily amenable to

pre-training scenario.

Training from Unsupervised Pre-Training

Here we consider training a model, that was pre-trained using the auto-encoding task. In these experiments we used the same segmentation network, but pre-train the network on the full unlabelled training dataset.

The protocol for the unsupervised pre-training is as follows:

- **Initialisation:** The model is initialised using the He initialisation.
- **Model:** We use exactly the same segmentation model as used previously: ResNet18-based FCN.
- **Task:** Autoencoding. The network structure for FCNs for segmentation are not dissimilar from convolutional autoencoders: they both consist of an encoder and decoder. The encoder creates a compact representation and the decoder decodes the representation to recreate the original image. Therefore, the input and the target are the same: the image.
- **Loss function:** Mean square error.
- **Epochs:** We train for 30 epochs or to convergence.

Pre-Trained on Different Regions

Here, we consider the active learning scenario when a model is pre-trained on a region, and then apply active learning to train on a new region. This is also known as transfer learning. We are not so much concerned with analysing the transfer learning process, but rather whether it has an effect on the active learning process. Note that in this context none of the methods require a cold-start, i.e. the initial samples drawn can be drawn using the acquisition strategy described as the model is already trained to give reasonable outputs for the task.

The protocol for the supervised pre-training is as follows:

- **Initialisation:** The model is initialised using the He initialisation.
- **Network:** Same segmentation model as used previously: ResNet18-based FCN.
- **Region:** We pre-train on every state excluding the region to be adapted to, which is West Virginia.
- **Learning Rate:** We reduce the learning rate of the optimiser to 0.0005, as we are fine-tuning. This is common practice in transfer learning [Liu et al., 2020].
- **Task:** Semantic segmentation.
- **Loss function:** Soft-IoU loss.
- **Epochs:** We train for 30 epochs or to convergence.

Active Learning Protocol

The active learning protocol then follows from the standard active learning paradigm, previously described in section 5.4.4. However, instead of using random weight initialisation we use the weights learnt from one of the pre-training methods. In terms of the experiment itself we only conduct it on the LCM dataset, where we look at West Virginia.

The protocol was as follows:

- **Initialisation:** Pre-trained weights.
- **Initial budget::** 1% of the data.

- **Maximum labelled pool size:** 5% of the data.
- **Budget:** 1% of the training data for each active learning. iteration. The dataset (West Virginia) size is 15000 samples. Thus the budget is 150.
- **Initial samples:** Initial samples are selected at random for unsupervised pre-train weights. For supervised pre-training, we can use the models to obtain the initial samples rather than choosing samples at random. Core-sets remains the same as the learning from scratch as we used a fixed pre-trained VAE to generate features for core-sets acquisition strategy.
- **Learning rate:** We decrease the learning rate to 0.0005, as this is a fine-tuning scenario.

5.4.6 Effect of Changing Representation for Diversity-Based Methods

So far we tested core-sets with the VAE representation. What if we used the Tile2Vec representation instead? Will it make a difference? We study this on Kitsap County in the Inria dataset.

We generate the features as described in section 5.4.3 and see if there is a meaningful difference in performance. As noted in the beginning of this dissertation, Tile2Vec seemed to be cheating with its representation, picking up on the different data sources rather than similarities between actual useful features that we would see as humans (such as the presence of water, forest, etc.). It is unlikely that a VAE has this problem, however, it is unknown how useful the representation generated will be.

5.5 Results & Discussion

In this section we present the results of the experiments on active learning in a cohesive manner, showing the links between experiments and the conclusions that can be drawn from the results.

Subsection 5.5.1 presents the results of the experiments comparing the active learning strategies in West Virginia and Kitsap County. It notes that it is quite difficult to outperform the random acquisition strategy. However, there are some strategies that work better than others on certain regions in the datasets. Further analysis is provided in subsection 5.5.3.

Subsection 5.5.2 analyses the results of the effect of the different pre-training strategies. While we found that both pre-training strategies help substantially with improving the absolute performance of the model, we found little evidence to suggest that pre-training assists one method over another. These results did have less variability than when training from scratch, indicating that assessing active learning strategies with pre-trained weights could be helpful in reducing the conflicting results in the literature. The results show that the simple unsupervised pre-training strategy proposed could reach the performance of a supervised pre-training strategy with 5% of the data.

Subsection 5.5.3 looks at the masks produced by entropy and BALD functions to understand where the uncertainty arises from, whether MC-dropout assists with uncertainty and why uncertainty-based strategies might assist with certain datasets and not others.

Subsection 5.5.4 presents the results of our experiment comparing the two different types of representation learning methods for the core-sets strategy. It also provides some analysis of why the diversity methods often struggle in active learning — including representations of the data used to obtain that diversity.

Subsection 5.5.5 presents analysis and commentary on the computational expense of the various methods.

Subsection 5.5.6 provides commentary on why the development of context specific active learning algorithms might be useful direction to pursue.

Finally, we conclude the active learning chapter by assessing the potential of active learning in disaster relief mapping and related scenarios for aerial imagery.

5.5.1 Overall Performance

Our hypothesis states that diversity-based strategies (VAAL and core-sets) would outperform uncertainty based methods due to diversity assisting the network to learn better representations to segment the imagery. The results do not confirm this hypothesis. The results reflect what can be seen when traversing the literature [Sinha et al., 2019, Siddiqui et al., 2019, Casanova et al., 2020]: there is no clear best acquisition strategy. Based on our 3 runs for each experiment, we found little that separates the strategies when looking at the standard deviation.

Figure 5.6 indicates that in West Virginia, while diversity-based strategies perform better on average, the standard deviation is within the range of other strategies. If we look at max-entropy with MC-dropout, we see a drop in the mean at 2% of the West Virginia subset; but the standard deviation increases significantly. Clearly there was a bad result that caused the drop in the mean, but there were also some results more on par with the mean of the other methods.

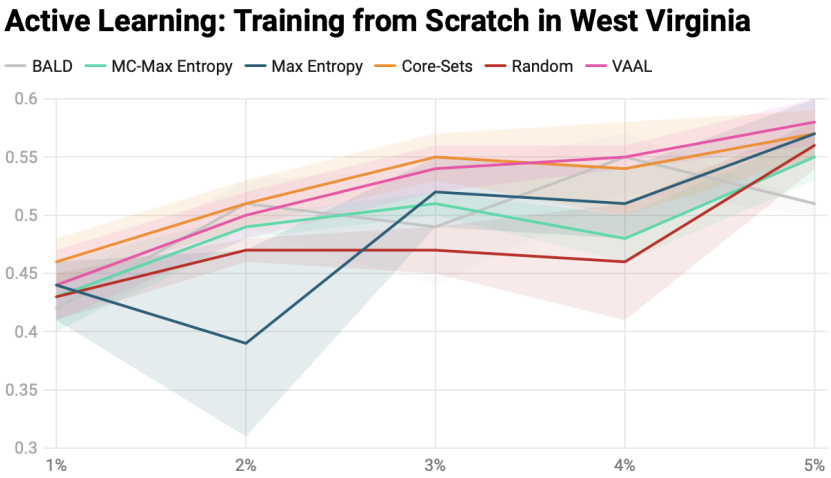


Figure 5.6: Mean IoU versus proportion (%) of West Virginia subset of the Chesapeake Dataset. We report the mean and the standard deviation of 3 runs.

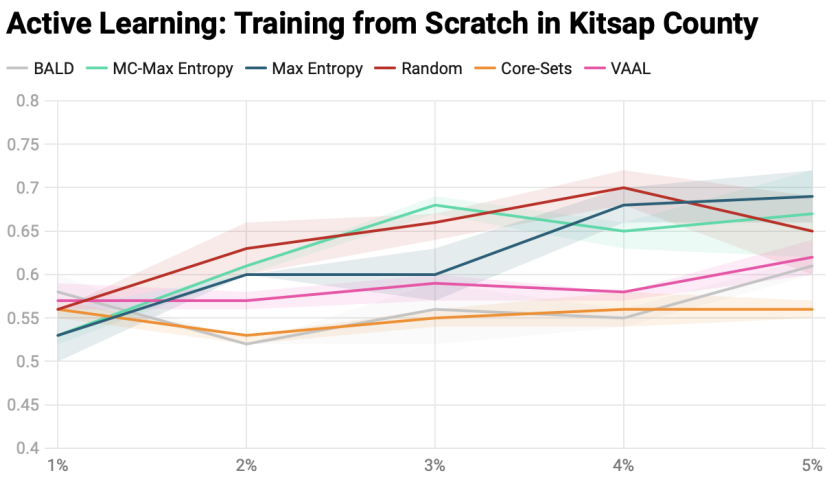


Figure 5.7: Mean IoU versus proportion (%) of Kitsap subset of the Inria Dataset. We report the mean and the standard deviation of 3 runs.

In figure 5.7, which displays the results of the experiment on Kitsap County, the results are somewhat clearer: the uncertainty-based strategies seemed to perform better. However, in both datasets the random acquisition strategy seems competitive with all other methods. There seems is to be no clear best strategy overall. We can however analyse the results, as in certain situations some of these strategies perform better. This analysis can inform further research.

We also note the significant variance seen in the results. This can, to some extent, be attributed to the random weight initialisation. Lakshminarayanan et al. [2017] described, when introducing their deep ensembles method for uncertainty estimation, that random weight initialisation alone resulted in enough diversity for better calibration.

Based on running these experiments we found VAAL to be extremely computationally expensive with training times upwards of 3 times the training times of the other methods. This is due to the need to train the segmentation model, the VAE and a discriminator all while using unlabelled data throughout the process. Due to the computational intensiveness of the method and computational constraints we faced, we decided to refrain from using VAAL in further experiments.

We did compare the other sampling strategies on New York and Vienna, with the results shown in figure 5.8. The results of these regions bear resemblance to the other region evaluated in their respective datasets. On the Inria dataset, uncertainty-based strategies seem to perform better than diversity-based strategy. In both datasets on all regions tested, random performs well compared to all of the strategies.

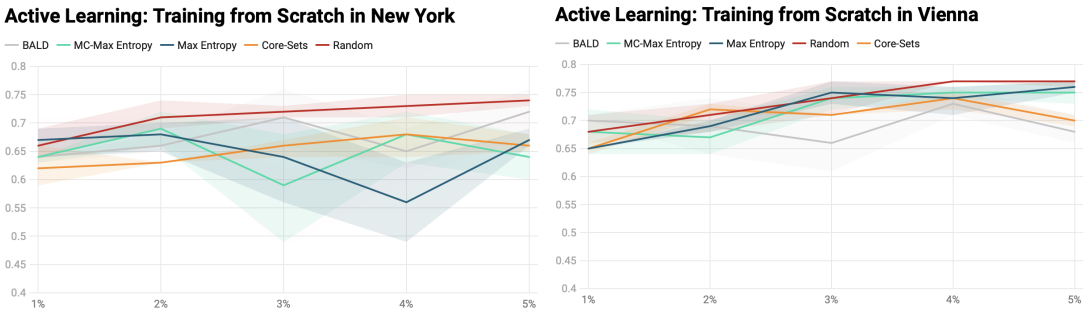


Figure 5.8: Performance of Active Learning Sample Acquisition Strategies in New York and Vienna respectively.

5.5.2 Effect of Pre-Training

This subsection presents the results when using the different pre-training methods: (1) random or learning from scratch, (2) unsupervised pre-training and (3) pre-training in a different region.

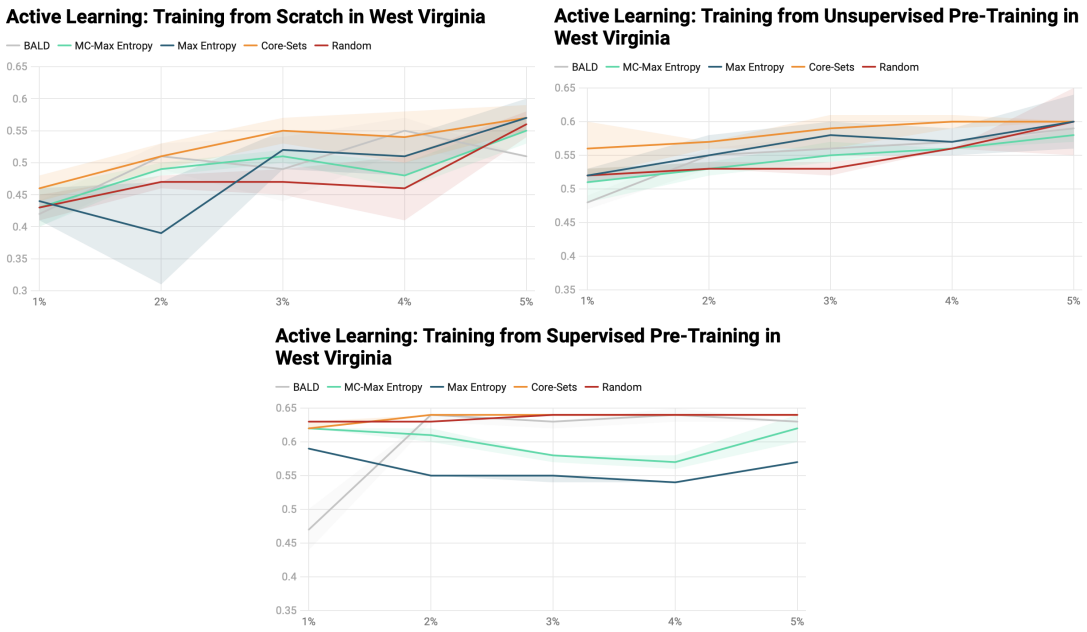


Figure 5.9: Comparison of pre-training methods on West Virginia.

Figure 5.9 compares the performance of the different pre-training methods. Clearly, pre-training increases absolute performance. Pre-training is well known to improve performance for transfer learning [Erhan et al., 2010, Liu et al., 2020]. We have not added new information here, however the results are useful. The results shown an increase of up to 47% in mIoU, with this increase in performance seen particularly with smaller budgets of labelled data. Interestingly, the results show that unsupervised pre-training performance is almost on par with the performance of supervised pre-training, when a budget of around 5% of the data within the West Virginia region (750 samples). The unsupervised pre-training method performed at the same level as training from scratch, with just 3% of the data, while the the training from scratch method required 5% of the data. This further points towards the importance of learning good representations.

One of the more important questions posed was whether pre-training would aid active learning strategies. These results suggest that pre-training does not aid active learning in any meaningful way nor does it favour an active learning strategy. If anything the results indicate that pre-training reduces the utility of any active learning, as the mIoU of the various active learning strategies when using pre-training have a smaller range. This further points towards to the importance of self-supervised or semi-supervised methods, as more significant gains in performance are likely to come from these approaches rather than active learning.

On a different note, the model trained using supervised pre-training weights did not improve beyond 0.65. We note the performance we obtained in the previous section for West Virginia on a model trained on all the data was 0.691. While close, one would expect the model to continue to improve but it does not. This suggests investigating how to adjust the protocol for training after pre-training given more samples.

Another interesting phenomenon that can be noted is that using pre-trained weights results in less variability in the results. This gives more credence to the results of Lakshminarayanan et al. [2017] who found that Deep Ensembles with random initialisation resulted in enough diversity for better calibration. Erhan et al. [2010] found empirical evidence suggesting that unsupervised pre-training acts as a regulariser and smooths the loss landscape for later

training. Liu et al. [2020] visualised this with pre-training on ImageNet datasets. As mentioned previously, there are conflicting results in the active learning literature surrounding the best acquisition strategy. One approach could be to devise a protocol for assessing active learning acquisition strategies with pre-trained weights. This would reduce the variability and so likely the conflicting results often seen in the literature.

5.5.3 Uncertainty-Based Sampling

The performance of uncertainty-based methods worked better on the Inria dataset, while diversity methods under performed. The opposite occurred in the LCM dataset. To better understand why uncertainty-based sampling works well in some areas but not in others, we study the outputs of the various uncertainty functions (entropy, MC-entropy and BALD).

What causes the uncertainty?

Firstly, an important piece of analysis is where the uncertainty comes from (which is then aggregated to get an score for an image). There seem to be a few primary causes of the uncertainty: (1) minority classes and (2) edges.

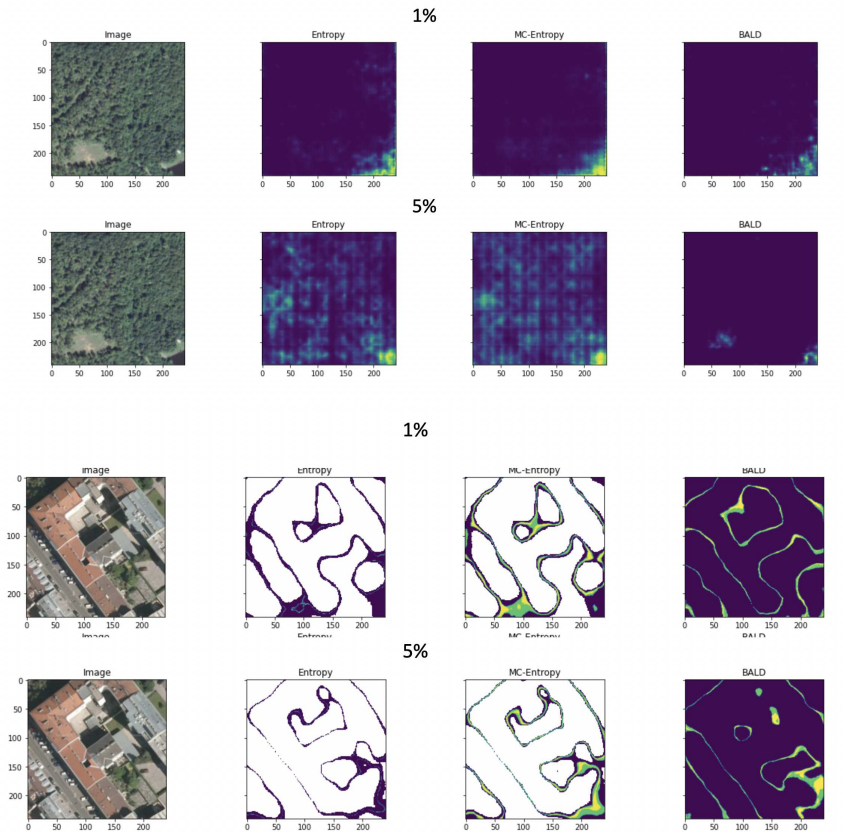


Figure 5.10: Example of uncertainty maps, showing the cause of high uncertainty.

This makes sense. Firstly, edges are harder to classify, particularly given the level of granularity that the FCN provides. Secondly, while minority classes are not seen as often, and have significant intra-class variability, which makes the minority classes difficult to classify. In figure 5.10 the uncertainty in the images according the heuristics can be seen.

We also obtained some examples which were strange, like those in figure 5.11. In figure 5.11 the uncertainty was high even for patches of contiguous classes. This was less present in BALD, but present in both entropy visualisations.

Furthermore, in both figures 5.10 and 5.11, we saw models become less certain when the amount of data increased from 1% to 5% on the LCM dataset, even for contiguous patches of field and forest area where the segmentation should be easy. This is not as evident with the BALD function, as it is with the entropy function. Visually, we can see that BALD, as compared to the entropy heuristics, seems to be better at displaying uncertainty in a manner that is easy to interpret.

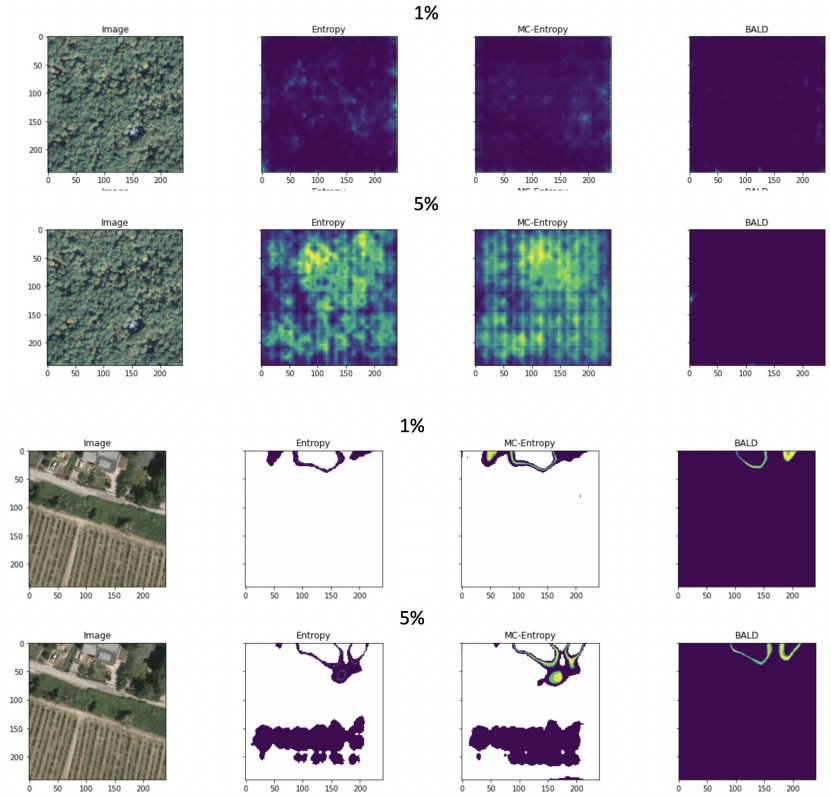


Figure 5.11: Patches with Contiguous Classes: Sample from LCM dataset at the top, and Inria at the bottom.

MC-dropout

It is important to note is that MC-dropout provides better uncertainty estimates and these make the model perform better for the most part. This can be seen visually in many of the figures displaying the entropy maps. This is also shown quantitatively in figure 5.12. Figure 5.12 shows the average performance increase in mIoU when using MC-dropout, across all regions used in this chapter.

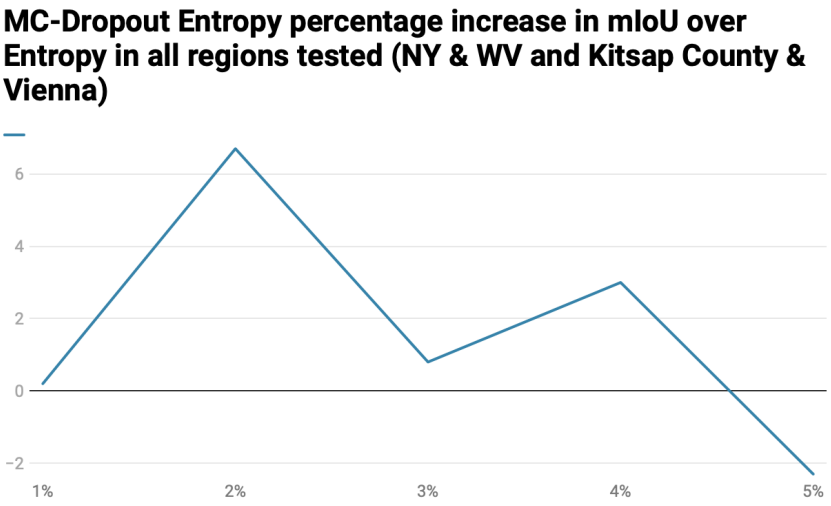


Figure 5.12: Comparison of entropy with and without MC-dropout.

This confirms two things: (1) MC-dropout does provide better uncertainty estimates than a single pass and (2) uncertainty as heuristic many not be a bad idea if the correct heuristic can be found. Better calibration of uncertainty is an important general direction of research for deep learning as it has implications for a variety of applications including active learning.

Why Uncertainty Works Best for Inria

It appears that the uncertainty-based acquisition strategies work better on the Inria dataset. As mentioned previously, uncertainty is often found at the edges. On the Inria dataset, the figures showed that the models were quite certain at knowing what is not a building and what is a building, except at the edges. From this it would seem that the uncertainty-based heuristics favour images with many buildings having many edges. These are a minority class within the dataset, so the active learning seems to be able to do some dataset balancing.

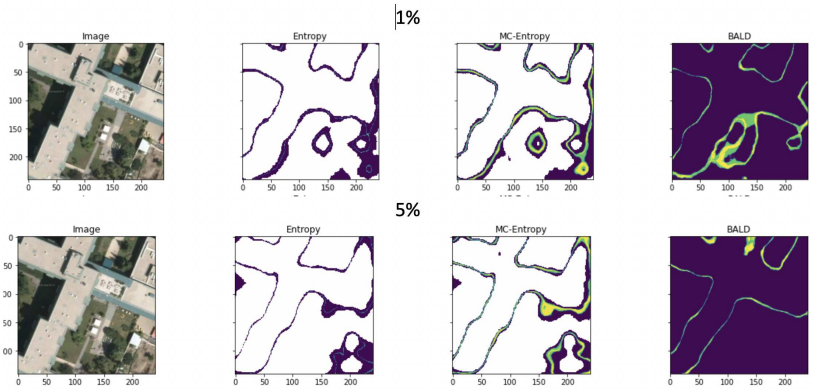


Figure 5.13: Uncertainty at the edges. Samples from the Inria Dataset.

On the other hand, the LCM entropy maps do not illustrate more uncertainty around minority classes and do not have as many edges in images with minority classes. This means there is no implicit dataset balancing occurring when applying uncertainty-based heuristics to the LCM dataset.

5.5.4 Diversity-Based Sampling

The performance of diversity-based sampling, if it is indeed useful, is heavily dependent on how good the features it uses to create that diversity are. VAAL uses a VAE and for a fair comparison we trained a VAE to learn a representation of the data for the core-sets sampling strategy.

We conducted an experiment, changing the method from which the features were generated. We conducted this on Kitsap County and Vienna, respectively. We compared the results of VAE and Tile2Vec (T2V) embeddings with the core-sets strategy and found somewhat mixed results.

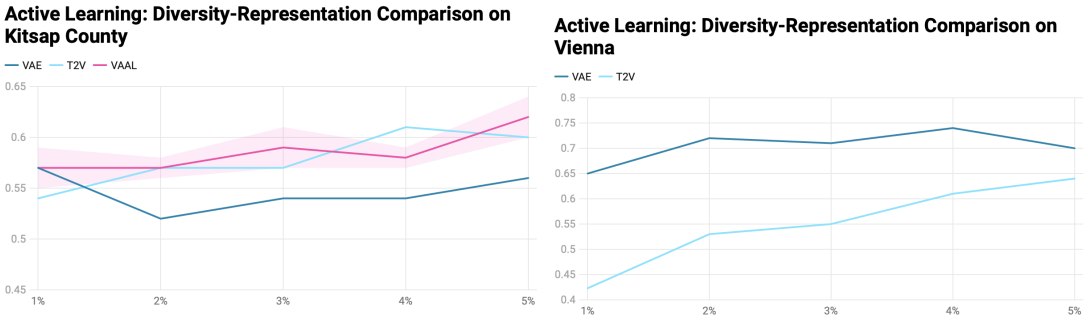
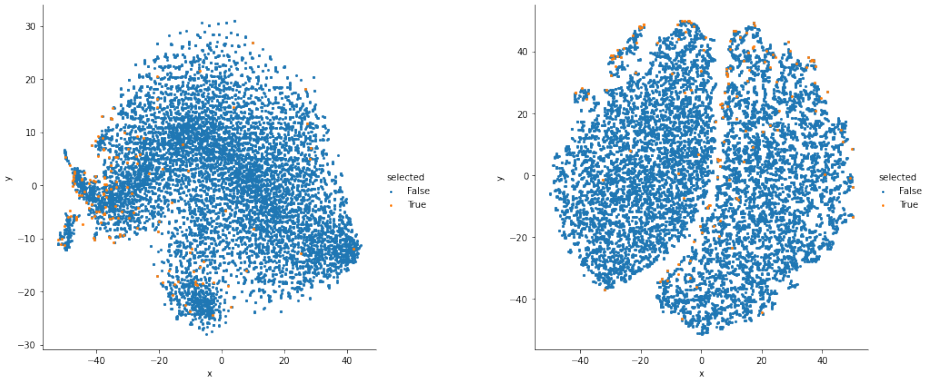


Figure 5.14: Comparison of diversity-based methods on Kitsap and Vienna.

Based on these results, it is difficult to suggest which method to use. Tile2Vec has inherent problems with it, in terms of it learning to generate features that inform one about the region it was collected rather than general features like forest, fields and water. On the other hand, the VAE generates no discernible clusters around the features. The mixed performance shown in figure 5.14 does not provide any insight into the utility of one method over the other. Either way the t-SNE visualisation and the results more generally suggest that the representations learnt are not useful for active learning.

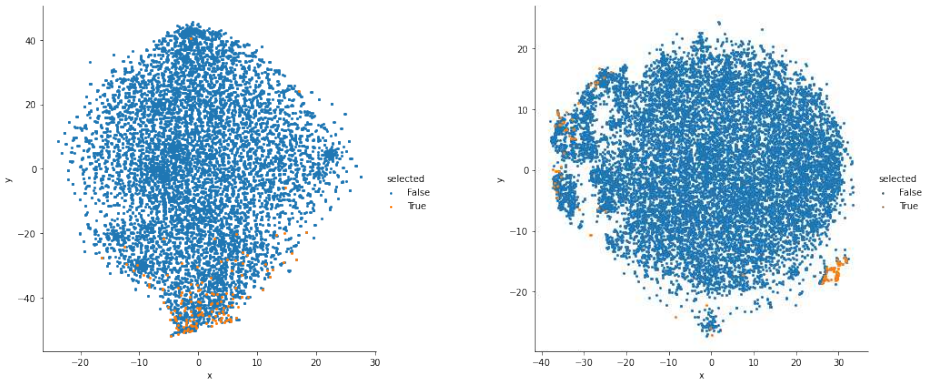
Figure 5.15 and 5.16 shows the t-SNE embeddings of the different methods, for Kitsap County and Vienna respectively, along with the initial samples selected by the core-sets method. In Vienna, there seems to be no discernible clustering with either pre-training method, so the t-SNE visualisation does not assist much and the visualising of the samples selected by core-sets does not mean much.

However, in Kitsap County there are the discernible clusters generated by both the VAE and Tile2Vec. The methods are likely picking up on the difference between the rural and urban areas, as Kitsap County has a significant rural sub-region. In this t-SNE visualisation we see that core-sets is picking diverse points when using both representation learning methods. However, with Tile2Vec core-sets is picking more diverse points and this reflects in a performance increase over the VAE in figure 5.14. It would seem that learning better representations and picking diverse points does aid performance.



(a) VAE Embeddings (b) Tile2Vec Embeddings

Figure 5.15: t-SNE Embeddings of Kitsap County from the features generated by the VAE and Tile2Vec respectively.



(a) VAE Embeddings (b) Tile2Vec Embeddings

Figure 5.16: t-SNE Embeddings of Vienna from the features generated by the VAE and Tile2Vec respectively.

Diversity-based sampling did not perform significantly better than other acquisition strategies. Based on the analysis here, one potential factor is that the representations of the data, used to choose the samples to create the diversity, are not good enough. The representations might not be picking up on important components of the imagery, such as a small patch of water or a small house. Perhaps if the representation could factor in these smaller features of the imagery it would do better. This is all speculation, however, and leaves a lot of room for further work and development of active learning algorithms for aerial imagery.

5.5.5 Computational Expense

Active learning is often done to reduce the cost burden of labelling imagery. Another cost burden is computational cost for training these models. We try to provide some information on this for the active learning practitioner. Unfortunately, this question will be answered primarily qualitatively with some reference to the method, and the literature. This question only became more apparent after conducting the experiments, but we make some comments so as to provide potential practitioners of active learning with an idea of the computational and time burden the various methods have.

Active learning is expensive in terms of computation: one has to train models for many epochs before obtaining a new batch of samples (and evaluating active learning strategies even more so!) and then doing those loops over again.

Training

At training time the uncertainty heuristics and core-sets take the same amount of time to train. VAAL on the other hand takes upwards of 3 times the times it takes to train the other models— this is because you require more GPU memory and CUDA cores to be allocated to VAE and discriminator respectively in addition to the segmentation model. This means one requires smaller batch sizes. In addition, one needs to train both of those using gradient descent and backpropagation.

Acquisition

When evaluating the unlabelled pool, the scale of the unlabelled pool matters:

- For entropy, the strategy requires passing each example through the segmentation model, aggregation across the image and ranking the scores.
- BALD and entropy with MC-dropout requires passing each example through the segmentation model, and doing so multiple times (to obtain Monte Carlo samples to estimate the uncertainty). Thus, these approaches will be slower than the previous approach as it requires multiple Monte Carlo passes.
- Core-Sets requires one to perform the k-centre greedy algorithm from features generated. To generate the features a single pass through the network after each iteration may be required if using features from the trained network, or all the features could be generated prior to the active learning loop. The complexity of the k-centre greedy algorithm is $O(n^2)$, but could end up taking less time than the other methods if the features are low dimensional and generated before the active learning loop.
- VAAL on the other hand simply requires a single pass through the VAE and discriminator (which are comparatively lightweight to the segmentation model). Thus it would be comparatively quicker than any of the other acquisition strategies.

Overall, while VAAL might scale well if the unlabelled pool is very large in comparison to the labelled pool; training VAAL is troublesome and requires significant computational resources. The other methods would not scale as well with larger unlabelled pools, but would be easier to train on the larger labelled pools than VAAL. However, the computational cost is significantly less (using any strategy) than the cost incurred in obtaining human labels. Thus, if an active learning strategy does show clear and consistent benefit, then it would be the preferred strategy for real world applications.

5.5.6 Dataset Specific Active Learning Algorithms

Active learning methods are most often developed on natural images and toy problems. On toy problems like MNIST these methods seem to perform well and exceed the performance of the random acquisition strategy. However, when moving to natural image datasets like ImageNet or CityScapes, the performance gain over the random acquisition strategy becomes small and the results comparing strategies often conflict.

Moving to aerial imagery resulted in negligible performance increases from various acquisition strategies. However, we did find that certain methods worked best on some regions or datasets, and there seems to be various reason why a certain heuristic might work better. For example, Inria worked well with uncertainty because patches with buildings have many edges, which is where the model was uncertain. One possible way forward is to choose an active learning algorithm based on the dataset characteristics. Alternatively, learn an active learning algorithm (for example through reinforcement learning [Casanova et al., 2020]).

5.6 Conclusion

In this chapter we examined various active learning strategies to see if we can reduce the amount of labels instead of randomly selecting samples to train our semantic segmentation model on. We found that some strategies worked better than others in certain situations, like uncertainty-based algorithms on the Inria dataset, and analysed why this was the case. However, a random sample acquisition strategy often performed on par with the proposed acquisition strategies.

This chapter also examined how pre-training affects active learning. While it increases absolute performance, we found little evidence to suggest that pre-training assists one method over another. The results did show less variability, suggesting that the use of pre-trained weights might be useful in reducing the variability of the results seen in the literature.

There are significant opportunities for further research, the most interesting of which would be to develop active learning algorithms specific to aerial imagery with its unique characteristics. Additionally, region/superpixel acquisition strategies on aerial imagery dataset could prove more useful than the patch level strategies we tested. These strategies, however, impose a significant computation burden.

Further to the point of decreasing the amount of labelling that is required to be done by a human, self-supervised (or unsupervised) learning and semi-supervised learning seem to hold better promise for decreasing the number of labels required to achieve higher accuracy. With even a simple unsupervised learning pre-training strategy we found vastly superior performance that active learning methods were not able to compete with. Further development of methods along the lines of Jean et al. [2019], Singh et al. [2019], Wang et al. [2020] are likely to have more of an impact on reducing the labelling burden than active learning methods.

Chapter 6

Conclusion

The objective of this dissertation is to reduce the burden of labelling aerial imagery for disaster relief mapping. We study two approaches for reducing the burden: interactive segmentation and active learning.

The dissertation began by providing background on disaster relief mapping problem faced by disaster relief organisations— the need for accurate foundational maps of disaster-stricken regions and these maps to be provided as soon as possible. The section provided an overview of the disaster relief mapping process and provided context on where machine learning has been used in the process, where machine learning has had success and where it has failed. Using insights developed in this section, we developed a framework on how and when machine learning could be used to assist in the disaster relief mapping process. In particular, we noted that in the interim, where machine learning methods are not sufficiently and consistently accurate, human validation would be required and examples from various regions would be required to train accurate machine learning models. As such, human-in-the-loop methods would be of great utility to these organisations, so as to reduce the human effort required and reduce the time taken to map a region.

In the chapter on interactive segmentation, we introduced a method of speeding up the labelling with a user interactively correcting the outputs of a first-stage model. We show this works on building footprint segmentation and land cover mapping, with a mIoU performance increase of up to 18%. We show the model works especially well out-of-distribution. We also analyse where the model is particularly useful and where it fails. This human-in-the-loop method would be of particular use to disaster relief mapping where validation of the results is required.

We then studied active learning, which looks at selecting samples to label which improve the model more than others, thereby reducing the number of samples to label. Unfortunately, we found there to be mixed results from the various active learning methods with aerial imagery. We tried to provide some intuition on when and why certain strategies might work best. We studied the various approaches to see why they might fail and how we might improve them going forward, and this could be used in further work to guide a method that might work better and more generally. We also examine active learning in the context of some prior training, either from unsupervised pre-training or pre-training on other regions. We find that both pre-training strategies have significant improvements over training from scratch, but find no difference in the efficacy of any active learning method. Based on the overall results of the chapter, we suggest pursuing research in more context specific methods for active learning, but also note that greater performance gains are likely to arise from self and weakly supervised approaches.

In addition to utility of each method on its own for disaster relief mapping, we had hoped to combine these two approaches to reduce the burden of labelling aerial imagery: using active learning to select the samples for the volunteers to label and using interactive segmentation to speed up the labelling of the samples. Figure 6.1 illustrates this process.

Despite the negative results with active learning, we can still use interactive segmentation to reduce the burden of aerial image labelling. We can also use the intuitions developed on when certain active learning strategies might work to guide applications going forward.

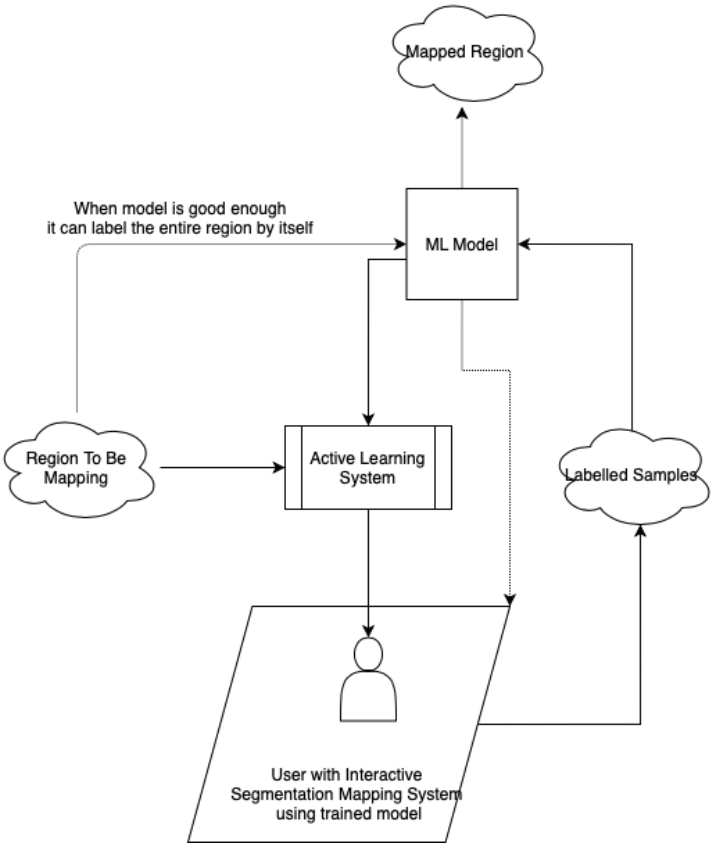


Figure 6.1: Integrating active learning and iterative segmentation model for disaster relief mapping.

Future Work

There are significant opportunities to extend this work to decrease the burden on labelling aerial imagery. Among them:

- For interactive learning, being able to update models online could lead to a substantial decrease in time spent labelling by annotators if the models become more accurate as more of the area is labelled.
- Related to updating models online is incremental learning. A good incremental learning methods would allow neural networks to retain knowledge without loss of previously learnt knowledge or features, which currently does not occur.
- In the realm of active learning, there seems to be significant scope for developing active learning strategies that provide meaningful performance gains and that are robust to different datasets and tasks.
- More generally, the development of methods that are able to adapt to new regions, with fewer labelled examples (such as meta-learning), is an interesting direction for further research.
- Self-supervised learning approaches appear to be the most promising going forward, as we saw across this dissertation the importance of learning good representations of data for downstream tasks. Further development of methods along the lines of Jean et al. [2019], Singh et al. [2019], Wang et al. [2020] are likely to have more of an impact on reducing the labelling burden than many of the other methods listed here.

Bibliography

Humanitarian OpenStreetMap Team. Humanitarian OpenStreetMap Team, 2020. URL https://www.hotosm.org.

OpenStreetMap Contributors. OpenStreetMap (OSM), 2019.

Martin Dittus, Giovanni Quattrone, and Licia Capra. Mass Participation During Emergency Response. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW '17*, pages 1290–1303, New York, New York, USA, 2017. ACM Press. ISBN 9781450343350. doi: 10.1145/2998181.2998216. URL http://dl.acm.org/citation.cfm?doid=2998181.2998216.

MapWithAI Team. MapWithAI, 2020. URL https://mapwith.ai.

Bing Maps Team. Microsoft releases 18M building footprints in Uganda and Tanzania to enable AI Assisted Mapping, 2019. URL https://blogs.bing.com/maps/2019-09/microsoft-releases-18M-building-footprints-in-uganda-and-tanzania-to-enable-ai-assist

USGS. Landsat 8. URL https://www.usgs.gov/land-resources/nli/landsat/landsat-8.

Wikimedia Commons. Remote_Sensing_Illustration. URL https://commons.wikimedia.org/wiki/File:Remote_Sensing_Illustration.jpg.

Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-Supervised Model Adaptation for Multimodal Semantic Segmentation. *International Journal of Computer Vision*, 128(5), 2020. ISSN 15731405. doi: 10.1007/s11263-019-01188-y.

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

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. ISBN 9783319245737. doi: 10.1007/978-3-319-24574-4{_}28.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. ISSN 01628828. doi: 10.1109/TPAMI.2016.2644615.

Lilian Weng. From autoencoder to beta-vae. *lilianweng.github.io/lil-log*, 2018. URL http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html.

Luke Derkson. Visualising high-dimensional datasets using PCA and t-SNE in Python, 2016. URL https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b.

Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL http://distill.pub/2016/misread-tsne.

Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. Darnet: Deep active ray network for building

segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Caleb Robinson, Le Hou, Kolya Malkin, Rachel Soobitsky, Jacob Czawlytko, Bistra DIlkina, and Nebojsa Jojic. Large scale high-resolution land cover mapping with multi-resolution data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019a. ISBN 9781728132938. doi: 10.1109/CVPR.2019.01301.

Varun Gulshan. *From Interactive to Semantic Image Segmentation*. PhD thesis, University of Oxford, 2012.

OpenCV. GrabCut Tutorial. URL https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html.

Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep Interactive Object Selection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 373–381. IEEE, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.47. URL http://arxiv.org/abs/1603.04042http://ieeexplore.ieee.org/document/7780416/.

Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep GrabCut for Object Selection, 2017. URL http://arxiv.org/abs/1707.00243.

K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep Extreme Cut: From Extreme Points to Object Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 616–625. IEEE, 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00071. URL https://ieeexplore.ieee.org/document/8578169/.

David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 859–868. IEEE, 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00096. URL https://ieeexplore.ieee.org/document/8578194/.

Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators, 2019. URL http://arxiv.org/abs/1903.10830.

Y. Yang, H. Li, Y. Han, and F. Yu. Research on Method of Interactive Segmentation Based on Remote Sensing Images. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W7:961–964, 2017a. ISSN 2194-9034. doi: 10.5194/isprs-archives-XLII-2-W7-961-2017. URL https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W7/961/2017/.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

Volodymyr Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013.

Humanitarian OpenStreetMap Team. Humanitarian openstreetmap team | integrating machine learning into the tasking manager: Notes on a direction, 09 2018. URL https://www.hotosm.org/updates/integrating-machine-learning-into-the-tasking-manager/.

United States Department of Agriculture. NAIP Imagery. URL https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/.

University of Minnesota. Introduction to Satellite Imagery. URL https://www.pgc.umn.edu/guides/

commercial-imagery/intro-satellite-imagery/.

Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9), 2005. ISSN 10577149. doi: 10.1109/TIP.2005.852470.

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

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. IEEE, 2015b. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7298965. URL http://ieeexplore.ieee.org/document/7298965/.

Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Rethinking Atrous Convolution for Semantic Image Segmentation Liang-Chieh. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 2018a. ISSN 01628828. doi: 10.1109/TPAMI.2017.2699184.

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

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Suriya Singh, Anil Batra, Guan Pang, Lorenzo Torresani, Saikat Basu, Manohar Paluri, and C. V. Jawahar. Self-supervised feature learning for semantic segmentation of overhead imagery. In *British Machine Vision Conference 2018, BMVC 2018*, 2019.

Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, volume 2017-July, 2017. doi: 10.1109/CVPRW.2017.156.

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

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-SCNN: Gated shape CNNs for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-October, 2019. doi: 10.1109/ICCV.2019.00533.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786), 2006. ISSN 00368075. doi: 10.1126/science.1127647.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 2019. ISSN 2159-5399. doi: 10.1609/aaai.v33i01.33013967.

I. T. Jolliffe. Principal Component Analysis and Factor Analysis. 1986. doi: 10.1007/978-1-4757-1904-8{_}7.

Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2008. ISSN 15324435.

Nakul Verma, Ziyuan Zhong, and Vincent Liu. *t-Distributed Stochastic Neighbor Embedding*. 2018. URL http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne.pdf.

Andrej Karpathy. Visualizing Top Tweeps with t-SNE, in Javascript, 2014. URL http://karpathy.github.io/2014/07/02/visualizing-top-tweeps-with-t-sne-in-Javascript/.

Neeti Shrivastava and Praveen Kumar Rai. Remote-sensing the urban area: Automatic building extraction based on multiresolution segmentation and classification. *Geografia: Malaysian Journal of Society & Space*, 11(2), 2015.

Hasan Volkan Guducu. *BUILDING DETECTION FROM SATELLITE IMAGES USING SHADOW AND COLOR IFORMATION*. PhD thesis, Middle East Technical University, 2008.

Ö Aytekin, A. Erener, I. Ulusoy, and H. S.B. Düzgün. Automatic and unsupervised building extraction in complex urban environments from multi spectral satellite imagery. In *RAST 2009 - Proceedings of 4th International Conference on Recent Advances Space Technologies*, pages 287–291, 2009. ISBN 9781424436286. doi: 10.1109/RAST.2009.5158214.

Xin Huang and Liangpei Zhang. Morphological building/shadow index for building extraction from high-resolution imagery over urban areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(1), 2012. ISSN 19391404. doi: 10.1109/JSTARS.2011.2168195.

Renxi Chen, Xinhui Li, and Jonathan Li. Object-based features for house detection from RGB high-resolution images. *Remote Sensing*, 10(3), 2018b. ISSN 20724292. doi: 10.3390/rs10030451.

Xiang Li, Xiaojing Yao, and Yi Fang. Building-a-nets: Robust building extraction from high-resolution remote sensing images with adversarial networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, (99):1–8, 2018.

Aaron E. Maxwell, Timothy A. Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review, 5 2018. ISSN 13665901.

Kang Zhao, Jungwon Kang, Jaewook Jung, and Gunho Sohn. Building extraction from satellite images using mask R-CNN with building boundary regularization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, volume 2018-June, 2018. doi: 10.1109/CVPRW.2018.00045.

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2): 645–657, 2017a.

Penghua Liu, Xiaoping Liu, Mengxi Liu, Qian Shi, Jinxing Yang, Xiaocong Xu, and Yuanying Zhang. Building footprint extraction from high-resolution images via spatial residual inception convolutional neural network. *Remote Sensing*, 11(7), 2019. ISSN 20724292. doi: 10.3390/rs11070830.

Hsiuhan Lexie Yang, Jiangye Yuan, Dalton Lunga, Melanie Laverdiere, Amy Rose, and Budhendra Bhaduri. Building extraction at scale using convolutional neural network: Mapping of the united states. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(8):2600–2614, 2018.

Sanjeevan Shrestha and Leonardo Vanneschi. Improved fully convolutional network with conditional random fields for building extraction. *Remote Sensing*, 10(7), 2018. ISSN 20724292. doi: 10.3390/rs10071135.

Shunping Ji, Shiqing Wei, and Meng Lu. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on Geoscience and Remote Sensing*, (99):1–13, 2018.

Weijia Li, Conghui He, Jiarui Fang, Juepeng Zheng, Haohuan Fu, and Le Yu. Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source GIS data. *Remote Sensing*, 11(4), 2019. ISSN 20724292. doi: 10.3390/rs11040403.

Xudong Lai, Jingru Yang, Yongxu Li, and Mingwei Wang. A building extraction approach based on the fusion of LiDAR point cloud and elevation map texture features. *Remote Sensing*, 11(14), 2019. ISSN 20724292. doi: 10.3390/rs11141636.

Alexey Bokhovkin and Evgeny Burnaev. Boundary Loss for Remote Sensing Imagery Semantic Segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11555 LNCS, 2019. doi: 10.1007/978-3-030-22808-8{_}38.

Diego Marcos, Michele Volpi, Benjamin Kellenberger, and Devis Tuia. Land cover mapping at very high resolution with rotation equivariant cnns: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:96–107, 2018a.

International Society For Photogrammetry And Remote Sensing. 2d semantic labeling challenge, 2013.

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229. IEEE, 2017b. ISBN 978-1-5090-4951-6. doi: 10.1109/IGARSS.2017.8127684. URL http://ieeexplore.ieee.org/document/8127684/.

Bilel Benjdira, Yakoub Bazi, Anis Koubaa, and Kais Ouni. Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images. *Remote Sensing*, 11(11):1369, Jun 2019. ISSN 2072-4292. doi: 10.3390/rs11111369. URL http://dx.doi.org/10.3390/rs11111369.

Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. Incremental Learning for Semantic Segmentation of Large-Scale Remote Sensing Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12 (9), 2019. ISSN 21511535. doi: 10.1109/JSTARS.2019.2925416.

Yilei Shi, Qingyu Li, and Xiao Xiang Zhu. Building footprint generation using improved generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 16(4):603–607, 2019.

Martin Herold, Meg Gardner, Brian Hadley, and Dar Roberts. The spectral dimension in urban land cover mapping from high-resolution optical remote sensing data. *Symposium A Quarterly Journal In Modern Foreign Literatures*, 6(June), 2002.

Ebenezer Isaac, K. S. Easwarakumar, and Joseph Isaac. Urban landcover classification from multispectral image data using optimized AdaBoosted random forests. *Remote Sensing Letters*, 8(4), 2017. ISSN 21507058. doi: 10.1080/2150704X.2016.1274443.

Matthew M. Hayes, Scott N. Miller, and Melanie A. Murphy. High-resolution landcover classification using random forest. *Remote Sensing Letters*, 5(2), 2014. ISSN 21507058. doi: 10.1080/2150704X.2014.882526.

M. Govender, K. Chetty, and H. Bulcock. A review of hyperspectral remote sensing and its application in vegetation and water resource studies, 2007. ISSN 03784738.

M. Govender, K. Chetty, V. Naiken, and H. Bulcock. A comparison of satellite hyperspectral and multispectral remote sensing imagery for improved classification and mapping of vegetation. *Water SA*, 34(2), 2008. ISSN 18167950. doi: 10.4314/wsa.v34i2.183634.

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

Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, and Anton Van Den Hengel. Semantic Labeling of Aerial

and Satellite Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7), 2016. ISSN 21511535. doi: 10.1109/JSTARS.2016.2582921.

Michele Volpi and Devis Tuia. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. *ISPRS journal of photogrammetry and remote sensing*, 144:48–60, 2018.

Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. TernausNetV2: Fully Convolutional Network for Instance Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 228–2284. IEEE, 2018. ISBN 978-1-5386-6100-0. doi: 10.1109/CVPRW.2018.00042. URL https://ieeexplore.ieee.org/document/8575501/.

Mohammad Pashaei, Hamid Kamangir, Michael J. Starek, and Philippe Tissot. Review and evaluation of deep learning architectures for efficient land cover mapping with UAS hyper-spatial imagery: A case study over a wetland. *Remote Sensing*, 12(6), 2020. ISSN 20724292. doi: 10.3390/rs12060959.

Sherrie Wang, William Chen, Sang Michael Xie, George Azzari, and David B. Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12(2), 2020. ISSN 20724292. doi: 10.3390/rs12020207.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 2010. doi: 10.1145/1869790.1869829.

Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Caleb Robinson, Anthony Ortiz, Kolya Malkin, Blake Elias, Andi Peng, Dan Morris, Bistra Dilkina, and Nebojsa Jojic. Human-machine collaboration for fast land cover mapping, 2019b.

Bo Fang, Rong Kou, Li Pan, and Pengfei Chen. Category-sensitive domain adaptation for land cover mapping in aerial scenes. *Remote Sensing*, 11(22), 2019. ISSN 20724292. doi: 10.3390/rs11222631.

Mesay Belete Bejiga, Farid Melgani, and Pietro Beraldini. Domain adversarial neural networks for large-scale land cover classification. *Remote Sensing*, 11(10), 2019. ISSN 20724292. doi: 10.3390/rs11101153.

Chesapeake Conservancy. Land cover data project. 2017. URL https://chesapeakeconservancy.org/wp-content/uploads/2017/01/LandCover101Guide.pdf,.

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017c.

Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112. IEEE Comput. Soc, 2002. ISBN 0-7695-1143-0. doi: 10.1109/ICCV.2001.937505. URL http://ieeexplore.ieee.org/document/937505/.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut". *ACM Transactions on Graphics*, 23(3):309, 2004. ISSN 07300301. doi: 10.1145/1015706.1015720. URL http://portal.acm.org/citation.cfm?doid=1015706.1015720.

Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively Trained Interactive Segmentation, 2018. URL http://arxiv.org/abs/1805.04398.

Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating Object Instances with a Polygon-RNN. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4485–4493. IEEE,

2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.477. URL http://ieeexplore.ieee.org/document/8099960/.

Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. Fluid annotation. *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*, 2018. doi: 10.1145/3240508.3241916. URL http://dx.doi.org/10.1145/3240508.3241916.

Eirikur Agustsson, Jasper Uijlings, and Vittorio Ferrari. Interactive full image segmentation by considering all regions jointly. 2019. URL https://arxiv.org/abs/1812.01888.

Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. In *European Conference on Computer Vision*, pages 549–565. 2016. ISBN 9783319464770. doi: 10.1007/978-3-319-46478-7{_}34. URL http://link.springer.com/10.1007/978-3-319-46478-7_34.

Di Lin, Jifeng Dai, and Kaiming He. ScribbleSup : Scribble-Supervised Convolutional Networks for Semantic Segmentation The Chinese Univeristy of Hong Kong. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/CVPR.2016.344.

Guangming Wu, Xiaowei Shao, Zhiling Guo, Qi Chen, Wei Yuan, Xiaodan Shi, Yongwei Xu, and Ryosuke Shibasaki. Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. *Remote Sensing*, 10(3):407, 2018.

Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, and Raquel Urtasun. Learning Deep Structured Active Contours End-to-End. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018b. doi: 10.1109/CVPR.2018.00925.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Jia Deng, Wei Dong, Richard Socher, Lie-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Samrath Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-October, 2019. doi: 10.1109/ICCV.2019.00607.

Burr Settles. Active learning literature survey. 2010. *Computer Sciences Technical Report*, 1648, 2014.

Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

Tobias Scheffer and Stefan Wrobel. Active learning of partially hidden Markov models. *PKDD Workshop on Active Learning, Database Sampling, Experimental Design. Views on Instance Selection.Principles and Practice of Knowledge Discovery in Databases*, 2001.

Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee

algorithm. *Machine learning*, 28(2-3):133–168, 1997.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011. URL http://dblp.uni-trier.de/db/journals/corr/corr1112.html#abs-1112-5745.

Zuobing Xu, Christopher Hogan, and Robert Bauer. Greedy is not enough: An efficient batch mode active learning algorithm. In *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, 2009. doi: 10.1109/ICDMW.2009.38.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL*, 2008. doi: 10.3115/1613715.1613855.

Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

Yawar Siddiqui, Julien Valentin, and Matthias Niebner. Viewal: Active learning with viewpoint entropy for semantic segmentation, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

Weicheng Kuo, Christian Häne, Esther Yuh, Pratik Mukherjee, and Jitendra Malik. Cost-sensitive active learning for intracranial hemorrhage detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 715–723. Springer, 2018.

Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 399–407. Springer, 2017b.

Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2864–2873, 2016.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, 2019.

Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgC6TNFvr.

Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma

segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. Cereals - cost-effective region-based active learning for semantic segmentation, 2018.

Christian Geib, Matthias Thoma, and Hannes Taubenböck. Cost-Sensitive Multitask Active Learning for Characterization of Urban Environments with Remote Sensing. *IEEE Geoscience and Remote Sensing Letters*, 15(6), 2018. doi: 10.1109/LGRS.2018.2813436.

Suju Rajan, Joydeep Ghosh, and Melba M. Crawford. An active learning approach to knowledge transfer for hyperspectral data analysis. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2006. doi: 10.1109/IGARSS.2006.143.

Lunjun Wan, Ke Tang, Mingzhi Li, Yanfei Zhong, and A. K. Qin. Collaborative active and semisupervised learning for hyperspectral remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5), 2015. ISSN 01962892. doi: 10.1109/TGRS.2014.2359933.

Claudio Persello and Lorenzo Bruzzone. Active and semisupervised learning for the classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11), 2014. ISSN 01962892. doi: 10.1109/TGRS.2014.2305805.

Peng Liu, Hui Zhang, and Kie B. Eom. Active Deep Learning for Classification of Hyperspectral Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2), 2017. ISSN 21511535. doi: 10.1109/JSTARS.2016.2598859.

Swarnajyoti Patra and Lorenzo Bruzzone. A novel SOM-SVM-based active learning technique for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11), 2014. ISSN 01962892. doi: 10.1109/TGRS.2014.2305516.

D. Tuia, E. Pasolli, and W. J. Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9), 2011a. ISSN 00344257. doi: 10.1016/j.rse.2011.04.022.

Devis Tuia, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Muñoz-Marí. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal on Selected Topics in Signal Processing*, 5(3), 2011b. ISSN 19324553. doi: 10.1109/JSTSP.2011.2139193.

Qikai Lu, Yong Ma, and Gui-Song Xia. Active learning for training sample selection in remote sensing image classification using spatial information. *Remote Sensing Letters*, 8(12):1210–1219, 2017. doi: 10.1080/2150704X.2017.1375610. URL https://doi.org/10.1080/2150704X.2017.1375610.

Sheng Jun Huang and Songcan Chen. Transfer learning with active queries from source domain. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2016-January, 2016.

Hariank Muthakana and Rita Singh. Uncertainty and diversity in deep active image classification. 2019.

Saul Berardo, Eloi Favero, and Nelson Neto. Active learning with clustering and unsupervised feature learning. In Denilson Barbosa and Evangelos Milios, editors, *Advances in Artificial Intelligence*, pages 281–290, Cham, 2015. Springer International Publishing. ISBN 978-3-319-18356-5.

Phill Kyu Rhee, Enkhbayar Erdenee, Shin Dong Kyun, Minhaz Uddin Ahmed, and Songguo Jin. Active and semi-supervised learning for object detection with imperfect data. *Cognitive Systems Research*, 45:109 – 123, 2017. ISSN 1389-0417. doi: https://doi.org/10.1016/j.cogsys.2017.05.006. URL http://www.sciencedirect.com/science/article/pii/S1389041716301127.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan O. Arik, Larry S. Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling budget, 2020. URL https://openreview.net/forum?id=HJl8SkBYPr.

L. Bruzzone and D. Fernàndez Prieto. Incremental-learning neural network for the classification of remote-sensing images. *Pattern Recognition Letters*, 20(11-13), 1999. ISSN 01678655. doi: 10.1016/S0167-8655(99)00091-4.

Stéfan Van Der Walt, S. Chris Colbert, and Gaël Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13(2), 2011. ISSN 15219615. doi: 10.1109/MCSE.2011.37.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito Facebook, A I Research, Zeming Lin, Alban Desmaison, Luca Antiga, Orobix Srl, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems 32*, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Towards understanding the transferability of deep representations, 2020. URL https://openreview.net/forum?id=BylKL1SKvr.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, 2010. ISSN 15324435. doi: 10.1145/1756006.1756025.