Using Satellite Images and Computer Vision to Study the Effects of Spatial Apartheid in South Africa

Raesetje Bonjo Sefala

844165

A Dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of

Master of Science

Supervisors: Dr. Richard Klein, Dr. Timnit Gebru and Ms. Nyalleng Moorosi

20 April 2020

Acknowledgements

This work is inspired by the different communities in South Africa which were affected negatively by Spatial Apartheid and continue to experience the effects today. We would like to thank everyone who was involved in helping us get context on this topic, those who provided datasets and domain expert knowledge and their time. We would also like to thank the different audiences from around the world through talks and poster sessions on this work who provided insightful feedback and discussions which helped shape the direction of this work. Finally we would like to thank:

- CSIR and Eskom for the building dataset covering the entire country between 2006 and 2017.
- SANSA for providing the satellite images.
- Statistics South Africa for the Enumerator Area Dataset.
- DST-CSIR for the Masters Scholarship award.
- Google for the research award and the compute credits.
- The Deep learning Indaba and NVidia for the NVidia Titan V prize we won for best poster presentation at the 2018 Deep Learning Indaba summer school.

This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number: 118075) which provided computational and storage equipment.

Abstract

Removing many of the legacies of Apartheid, a former policy of political and economic discrimination against non-European groups in South Africa, is a primary concern for the country. Aerial images of residential areas show the clear legacy of spatial apartheid, with completely segregated neighbourhoods of townships next to gated wealthy neighbourhoods, a phenomena which has largely remained unaffected by the ending of apartheid. This research uses computer vision to analyse $698,544$ satellite images of 9 provinces in South Africa, taking the first steps toward examining the evolution of spatial apartheid. To achieve this goal, we first introduce a new dataset consisting of polygons demarcating land use, geographically labelled coordinates of all buildings in South Africa, and high resolution satellite imagery covering the entire country from 2006-2017. Using this dataset, we trained a UNet based semantic segmentation model to detect and classify clusters of buildings for 12 types of classes: Township, Suburb, Industrial area, Commercial land, Informal area, Farm, Collective living Quarters, Village, Smallholdings and Background. We classify these neighbourhoods with an accuracy of 57.45% and a Cohen's Kappa value of 0.4326, giving us the potential to investigate areas affected by the Group Areas Act which enforced spatial apartheid/segregation.

Key words: Spatial Apartheid; South Africa; semantic segmentation; Neighbourhoods Classification; UNet.

Table of contents

List of Figures

List of Tables

List of abbreviations and/or acronyms

EA	Enumeration Area
SANSA	South African National Space Agency
Stats SA	Statistics South Africa
CSIR	Council for Scientific and Industrial Research

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

Analysing large numbers of time-lapse satellite images presents the opportunity to study cities using computer vision and create tools that allow governments to examine the effects of their policies. Like any technology, these tools can either be used to further marginalize those who are already disenfranchised [Benjamin, 2019], or enact policies that aim to reverse the effects of laws that have enforced discrimination. This work takes a step towards the latter goal by starting to analyze the effects of spatial apartheid in South Africa using satellite imagery taken between 2006 and 2017.

Apartheid is a former South African policy of segregation, and political and economic discrimination against non-European groups in the country [Worden, 1994]. The Group Areas Act passed in 1950 [Christopher, 2001] forcefully relocated Black, Coloured and Indian people out of urban areas and into townships where they were allocated uniformly sized small plots of land on the outskirts of cities and towns. While Apartheid legislation was repealed on June 17 1991, its effects are still alive today [Noble and Wright, 2013]. For example, Figure 1.1 shows aerial images taken by photographer Johnny Miller, depicting completely segregated neighbourhoods of townships next to gated wealthy neighbourhoods that have largely remained unaffected by the end of apartheid [Miller, 2016 accessed January 7, 2020]. Although this effect is immediately obvious to any human looking at these photos, people cannot quickly analyse large numbers of such images to gain insights.

This research uses computer vision to understand the relationship between the spatial and socioeconomic makeup of neighbourhoods in South Africa. The goal is to answer the questions: Can we automatically identify clusters of townships and wealthy neighbourhoods using computer vision? Can we measure the sizes of these clusters and how they are changing over time? Can we map/label neighbourhoods in South Africa as they grow/shrink over time, especially townships?

This work takes the first step towards these goals by using semantic segmentation methods to distinguish between clusters of buildings making up 12 neighbourhood classes in South Africa. These classes are: Township, Suburb, Industrial area, Commercial land, Informal area, Farm, Collective living Quarters, Village, Smallholdings and Background in the 9 South African provinces. As seen in Figure 1.1, townships

(a) (b)

Figure 1.1: Aerial images showing some of the legacy of spatial apartheid in Cape Town, South Africa depicting completely segregated neighbourhoods of townships next to gated wealthy neighbourhoods that have largely remained unaffected by the ending of apartheid [Miller, 2016 accessed January 7, 2020].

and suburbs can have distinct visual characteristics: e.g. suburbs are usually more sparsely populated and green, townships are densely populated but with a grid-like structure, while informal settlements are densely populated without a consistent structure. These visual differences allow us to train a model distinguishing between different types of neighbourhoods. Although the majority of works in computer vision focus on algorithmic development, the most difficult and time-consuming step in projects such as this is procuring and labelling the necessary datasets for the task. We performed an iterative process that uses the insights gained from baseline models to understand the shortcomings of the datasets, adding new elements to the data as necessary. The final dataset consists of geo-referenced satellite images covering the entire country of South Africa and a corresponding mask of where all the neighbourhoods are labelled according to their type. This mask was built from a combination of geo-referenced polygons called Enumerator Areas (EAs) subdividing land-use as specified by the government, and data points of locations of all buildings in South Africa. To summarize our contributions:

1. We introduce the first spectral dataset for the study of spatial apartheid in South Africa, consisting of satellite images depicting the entire country from 2006 to 2017 and a corresponding ground truth dataset depicting the neighbourhoods in the country in 2011. The ground truth was built from geo-referenced polygons labelled by the type of land-use designated by the South African government and the location of all 12,515,874 identified building data points in South Africa in 2011.

2. Using the building dataset and the satellite imagery dataset, we introduce an image dataset with a ground truth mask locating the developed parts of the land in South Africa between 2006 and 2017, allowing researchers to detect the growth/shrinkage of built-up land in different years throughout the country.

3. Using our dataset, we train a baseline U-Net semantic segmentation model [Ronneberger et al., 2015] to perform the first analysis of South African spatial apartheid from satellite images, achieving 57.45% classification accuracy across 12 classes (Township, Suburb, Industrial area, Commercial land, Informal area, Farm, Collective living Quarters, Village, Smallholdings and Background).

The rest of this dissertation is structured as follows. Chapter 2 discusses background knowledge required and the related works in the next section, Chapter 3 introduces our visual dataset consisting of multiple sources (satellite images, land use polygons, and building points). We describe the pipeline used to create the ground truth dataset in Chapter 4, as well as various challenges unique to this task. Chapter 5 discusses experimental results and future work, and we conclude with Chapter 6.

CHAPTER 2

BACKGROUND AND RELATED WORKS

2.1. BACKGROUND

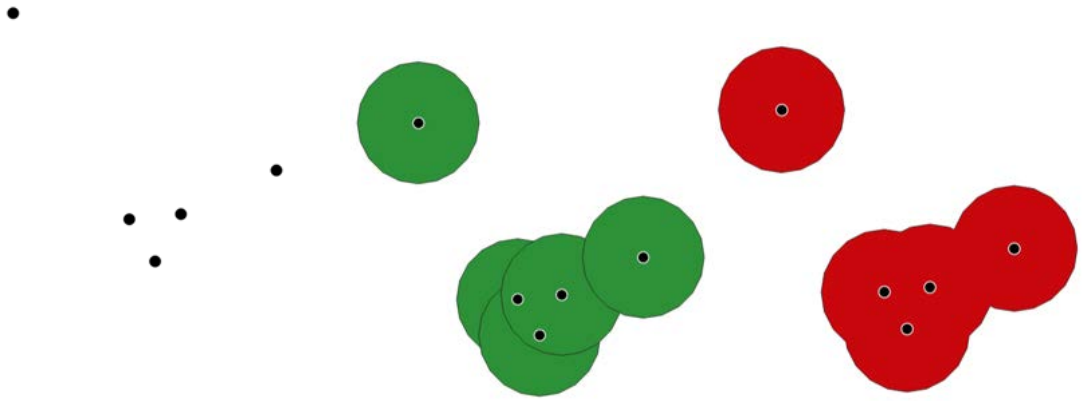
This chapter provides relevant background information pertaining to our dissertation. We first describe the spatial data manipulation techniques used in this work and briefly discuss common computer vision tasks. We then discuss semantic segmentation techniques in detail (including the architectures we use), and end the section with a detailed discussion of related work.

2.1.1. Spatial data

This section describes spatial data manipulation techniques that are used throughout this work.

A **shapefile** is a vector data (points, lines or polygons) storage format used to store geographical datasets together with other attributes like class type, geometry coordinates, province etc. Given vector data, the **buffer** algorithm computes a buffer zone around the features in an input layer, using a fixed or dynamic distance. The distance is measured in terms of decimal degrees, meters or feet. The buffer zone is always represented by a polygon. Given point data for example, a buffer zone will be a square or circular shaped polygon with a radius of the specified distance parameter. This distance is a hyperparameter which is chosen by the user according to their needs. If a government is trying to determine where to build a new neighbourhood next to a river, for example, they may create a buffer zone around the river so they can determine a consistent/fixed distance between the river and the neighbourhood for safety purposes. An example of a buffer is illustrated using Figure 2.1 where the point data in Figure 2.1a have been buffed into circular polygons by a fixed distance/radius of 0.0007 decimal degrees. This radius typically covers a standard South African suburban house together with the yard. In some cases, buffered zones may overlap with each other as illustrated in Figure 2.1b. Depending on the end goal, boundaries of buffer zones may be **dissolved** so that there are no overlapping areas between adjacent buffer zones as illustrated in Figure 2.1c or they may be left intact instead. **Spatial Intersection** is a spatial overlay technique between two or more vector data points where the output layer is just the overlapping (intersecting) parts between all the input vector data as illustrated in Figure 2.2. The resulting output attribute table is the union of the input attribute tables joined by geographical

coordinates. All of these are techniques which are commonly used to manipulate data for Geographic Information Systems (GIS).



(a) (b) (c)

Figure 2.1: Illustration of buffing point data into circular polygons (figure 2.1a to figure 2.1b) and then dissolving overlapping buffer zones (figure 2.1c).

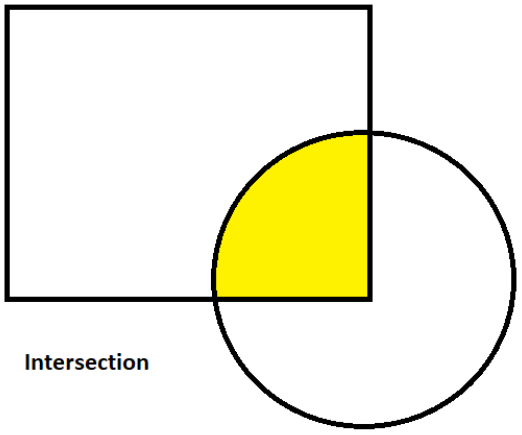
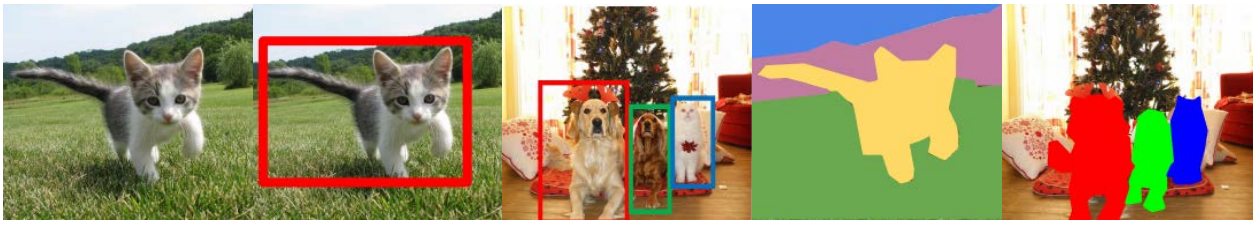


Figure 2.2: Illustration of spatial intersection: given two vector data points, polygons in this example. The output of applying the spatial intersection algorithm will be just the yellow polygon.

2.1.2. Common computer vision tasks

In the past decade, we have seen a drastic improvement in the accuracy of computer vision techniques. The most basic computer vision task assigns one label to an entire image. This method is referred to as **classification** and is illustrated in Figure 2.3a. Classification summarizes the entire image with one label

like "Cat" in this case. On the other hand, tasks such as object **localization** specify the exact location of the object in the image, usually with a bounding box as illustrated in Figure 2.3b. When there are multiple objects in the image view, the task of **object detection** assigns a corresponding bounding box to each object of interest in the image. These bounding boxes often capture some background pixels in these images which could be undesirable depending on the task. In these cases, the **semantic segmentation** technique as illustrated in Figure 2.3d could be more suitable. This technique is used in tasks ranging from land-use classification [Demir et al., 2018] to medical imaging [Ronneberger et al., 2015]. Instead of using bounding boxes, each pixel in the image is associated with a specific class. If there are many objects of the same class in a single image, it can be difficult to know the number of instances of that object in the image view. The task of distinguishing between objects of the same class as in Figure 2.3e is called **instance segmentation**. In this work, we use semantic segmentation techniques to segment South African neighbourhoods.



(a) Classification (b) Localization (c) Object Detection (d) Semantic segmentation (e) Instance Segmentation

Figure 2.3: Images representing various computer vision tasks [Fei-Fei Li, 2017 accessed April 19, 2020]. Figure 2.3d labels for: Yellow pixels- Cat, Green pixels- Grass, Purple pixels- Tree and Blue pixels- Sky. Figure 2.3e labels for: Red pixels- Dog number 1, Green pixels- Dog number 2, and Blue pixels- Cat number 1.

2.1.3. Semantic segmentation

Semantic segmentation is a more fine-grained technique for classifying regions of an image in comparison to bounding boxes which capture objects as well as some part of their background. Instead of classifying an entire image, the model learns to classify each pixel in the image into specified classes. These models require as input, images, and as labels ground truth images in which all the pixels are labelled using specific colours. Each colour represents a certain class as illustrated in Figure 2.4. The model then learns to classify each pixel in the image, collectively classifying objects in the image. One advantage of this technique is that it captures irregular object boundaries present in objects like cars, dams, people,

etc.

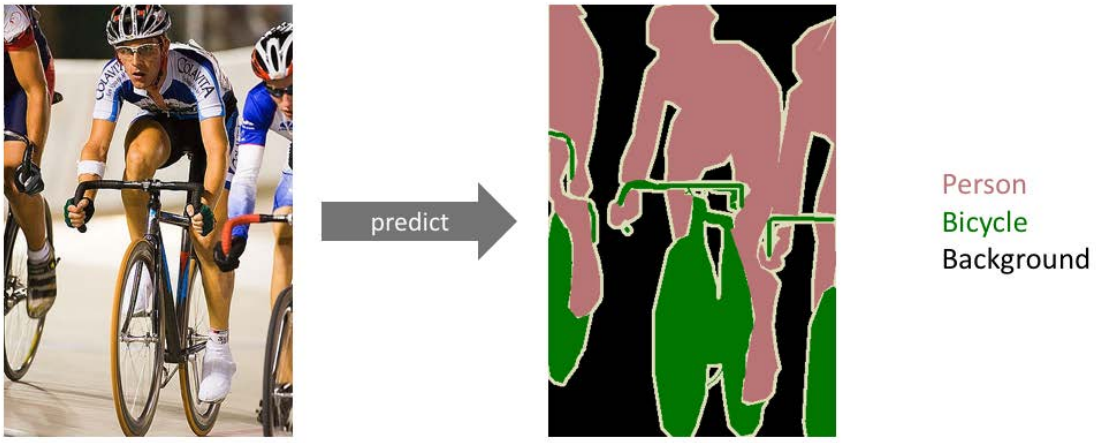


Figure 2.4: Input image on the left and ground truth image in which all the pixels are labelled using specific colours [Everingham et al., 2010].

There are several architectures to perform image segmentation. Early models used techniques such as Markov chain Monte Carlo [Tu and Zhu, 2002] for finding edges and distinguishing pixels by texture. Various state-of-the-art models in 2018 [Rota Bulò et al., 2017; Chen et al., 2018] used a Convolutional Neural Network (CNN) architecture [LeCun et al., 2015] with an encoder-decoder setup where a pre-trained classification model such as ResNet [Szegedy et al., 2017] is used as an encoder, and a decoder projects the features learnt by the encoder onto the original/desired pixel resolution as illustrated in Figure 2.5.

2.1.3.1. Fully Convolutional Networks(FCNs)

Convolutional neural networks (CNNs) consist of two main parts: the first part extracts features from an input image, and the second part consists of fully connected layers which require the image inputs to have fixed-sizes. Contrary to CNNs, fully convolutional networks (FCNs) just have convolutional and pooling layers with no fully connected layers. This allows FCNs to make predictions on input images of any size [Long et al., 2015]. When FCNs were first introduced by Long et al. [2015], they extended CNNs by converting the fully connected layers into convolutions. [Long et al., 2015] considered pre-trained architectures such as GoogLeNet [Szegedy et al., 2015] and VGG nets [Simonyan and Zisserman, 2014].

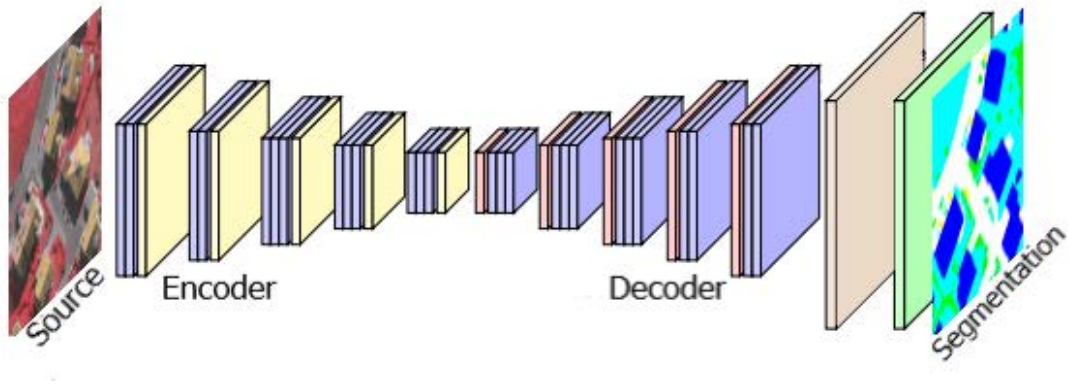


Figure 2.5: Typical semantic segmentation architecture[Audebert et al., 2016]. It takes as input labelled image pixels then follows an encoder-decoder type of model which learns to classify individual pixels which collectively define objects.

They removed the final (output) layer, converted all fully connected layers into convolutions and then appended 11n channels (n = number of classes) of convolution layers to predict scores for each class. In the decoder part of the model, they upsampled the outputs from the encoder into pixel-dense outputs. Results from this model are usually low in resolution because of the downsampling and upsampling procedures. Architectures such as DeepLabv3+ [Chen et al., 2018] try to correct this problem by using upsampled filters (atrous/dilated convolution) with multiple sampling rates at different fields of view as part of the encoder and use a decoder to help sharpen results along the edges. This technique allows the model to capture detailed object boundaries and image context at multiple scales.

2.1.3.2. UNet

One of the most common FCN architectures used to perform semantic segmentation in satellite images is the U-Net semantic segmentation architecture. This model was originally introduced in 2015 by Ronneberger et al. [2015] to segment biomedical images. A U-Net model follows an encoder-decoder architecture with skip connections between the encode and the decode layers. The encoder consists of convolutional and max-pooling layers which are meant to capture the context of the image and the decoder uses upsampling/transpose convolutional layers to enable precise remapping of pixels back to the original size of the input image. The skip connections allow the decoder to retrieve information from prior layers in the encoder rather than solely depending on the low-resolution bottleneck (the output of the encoder). Figure 2.6 below illustrates this architecture.

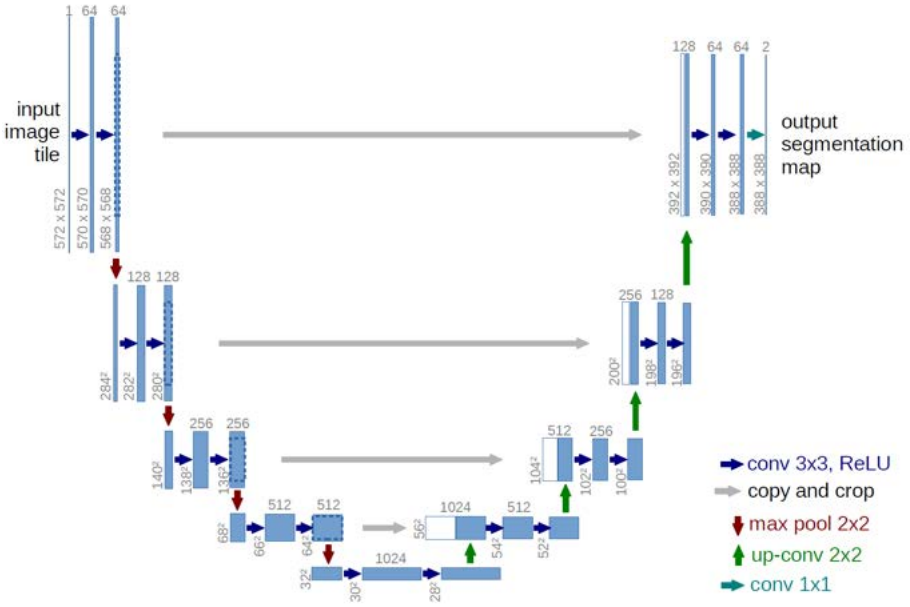


Figure 2.6: The U-net architecture. The encoder is on the left side of the U shape which consists of the convolutions and pooling layers, the decoder part which consists of the transpose convolutions is on the right side of the U and the middle arrows going across represent the skip connections [Ronneberger et al., 2015].

U-Net based architectures have won several semantic segmentation challenges since its introduction in 2015 [Kaggle, 2019; Iglovikov et al., 2017] and its efficiency allows us to train and evaluate models quickly. This was particularly important while trying to understand the nature of our dataset. As we discuss in Chapter 4, we constructed our dataset using an iterative process where we first train a model which allows us to see shortcomings in our data while examining the results on held-out data, then augment/alter our dataset as necessary, and repeat the process.

2.2. RELATED WORKS

Although classifying neighbourhoods from aerial images is not a new problem, it is still very challenging to collect a ground truth dataset to train and validate supervised machine learning techniques for this purpose. First, labelling each pixel by the neighbourhood it represents makes data sets very expensive to create. This type of label is needed because of the irregularly shaped boundaries of neighbourhoods (e.g. a suburb is not bounded by a square shape for example). Another problem is that over time neighbourhoods change as buildings are built or destroyed, resulting in previous datasets becoming quickly inaccurate and thus outdated. Although satellite images are becoming cheaper and easier to access, manual labour to update ground truth labels is expensive and challenging to manage, which is why researchers usually turn to use other proxies to approximate labels of neighbourhood classes on satellite/aerial images [Xie et al., 2016; Jean et al., 2016].

2.2.1. Using proxies to label neighbourhoods

Researchers like Blumenstock et al. [2015] have used mobile phone data as a proxy to label neighbourhoods according to their poverty rates in Rwanda, and Jean et al. [2016] used survey data from the World Bank's Living Standards Measurement Study (LSMS) survey and the Demographic and Health surveys as a proxy to estimate consumption expenditure and asset wealth per neighbourhood from Nigeria, Tanzania, Uganda, Malawi, and Rwanda. Given the fact that it is very difficult to obtain mobile phone data (let alone having it in the same resolution and spatial coverage) for specific regions, [Maluleke et al., 2018] focused on one province in South Africa and hand-labelled neighbourhoods using their roof types. They used a household level poverty estimation survey from National Income Dynamic Survey (NIDS) to predict the level of poverty in informal settlements. However, survey data is expensive to create and when survey datasets are available, they are not large enough to train deep learning models from scratch [Jean et al., 2016]. Works such as those by Xie et al. [2016] used night light data overlaid on daytime satellite images to create pixel-level labels designating wealthy and non-wealthy neighbourhoods. Although this experiment had a large ground truth dataset, using satellite night lights as a proxy for labelling neighbourhoods by wealth does not accurately represent densely populated neighbourhoods such as informal settlements or even townships in the South African context. Neighbourhoods can have a high density of light mainly due to housing density while not necessarily representing great wealth.

In an effort to create a dataset localizing buildings in South Africa, Mudau et al. [2020] uses computer vision techniques such as texture analysis to distinguish between pixels of built-up and non-built-up land on satellite imagery. The authors then use methods such as Canny edge detection [Canny, 1986] and examining the Soil Adjusted Vegetation Index (SAVI) to segment out individual buildings. This methodology fails to accurately detect buildings in high-density areas (e.g. townships) because it is difficult to examine soil contents or building edges when buildings are located very close to each other. As part of our dataset creation pipeline, we instead use the Building dataset [Mudau, 2010] which has human-labelled localizations of buildings ensuring that we have accurate ground truth data.

Research by Hofmann et al. [2001]; Ikokou and Smit [2019]; Ngcofe et al. [2017]; Hurskainen and Pellikka [2004]; Ojala et al. [1994]; Guo et al. [2010] use rule-based techniques in conjunction with handcrafted features such as Local Binary Patterns (LBP) [Guo et al., 2010] to label images for land-use related object classification tasks. For instance, [Ikokou and Smit, 2019] uses domain knowledge to determine that any building that is of size 20m x 100m and further than 100m from the main road must be a commercial building. The authors then use software applications such as eCognition [Karakış et al., 2006] to create ground truth datasets using this rule-based segmentation system. While handcrafted features with hand-tuned parameters can potentially be used to create small scale datasets, they are not feasible for large scale ones [Mboga et al., 2017]. This is because rigid rule-based measurements such as building dimensions assume that class characteristics are the same from place to place. For instance, [Ikokou and Smit, 2019] assumes that the size of a commercial building is always 20m x 100m. The advantage of feature learning approaches, and why they are almost unilaterally used in modern computer vision systems, is that the necessary features to distinguish between different classes are learned from the training data rather than prior assumptions.

Hence, while works before the advent of AlexNet [Krizhevsky et al., 2017] used handcrafted features such as LBPs to extract features for land-use related tasks [van den Bergh, 2011; Mdakane and Van den Bergh, 2012; Mdakane, 2014; Ella et al., 2008], the computer vision field as a whole now uses neural network-based feature extraction methods as they have been shown to be superior in every vision task [Krizhevsky et al., 2017]. For example, [Mboga et al., 2017] compared a neighbourhood classification pipeline for Dar es Salaam, Tanzania using LBPs followed by support vector machines (SVMs) [Suykens and Vandewalle, 1999], to using only CNNs (which perform both feature extraction and classification). The CNN based methodology received the highest accuracy on this task. Since our goal is to label images for the entire

country of South Africa, we do not rule-based methods that assume the same structure for different geographical regions. For instance, materials and methods used in KZN for construction in hilly tropical conditions are different from those used in the Cape Flats [Kemper et al., 2015]. We thus need to use a methodology which allows us to capture these variations at scale.

2.2.2. Commonly used datasets for land cover classification

Detecting and classifying neighbourhoods in images falls under a broader task called land cover classification. Common tasks in this field include segmenting residential areas, roads, water bodies, industrial land, forests and agricultural lands from satellite/aerial imagery. While there are several freely available datasets for this task such as DeepGlobe [Demir et al., 2018], UC Merced [Yang and Newsam, 2010a], ISPRS Vaihingen and Potsdam [for Photogrammetry and Sensing, 2020 accessed February 12, 2020], Dstl Kaggle [Science and Laboratory, 2017 accessed February 12, 2020], etc., most of these have been created for the developed world. The Deepglobe dataset was first introduced at the Conference on Computer Vision and Pattern Recognition (CVPR) in 2018 as a satellite imagery classification challenge with 3 tracks: Road Extraction, Building Detection and Land cover classification. The land cover classification dataset consisted of satellite imagery and masks outlining: urban land, agricultural land, rangeland, forest, water, barren land and unknown. This dataset covered both urban and rural land sampled from Thailand, Indonesia, and India. Although this dataset was created using images from countries which might share a lot of visual characteristics with South Africa, it does not differentiate between the sub-classes of neighbourhoods within the urban land class. The UC Merced dataset, on the other hand, tries to classify neighbourhoods in terms of: dense residential, medium residential, sparse residential and just buildings as part of 18 other classes which describe non-buildings in greater detail (agricultural, beach, chaparral, forest, freeway, golf course, baseball diamond, aeroplane, harbour, intersection, mobile home park, overpass, parking lot, river, runway, storage tanks, and tennis court) [Yang and Newsam, 2010a]. Although these classes are more detailed and more closely resemble the task at hand, they cover urban areas for cities in the United States of America which might not necessarily share similar visual characteristics with those in South Africa, e.g. slums in the United States of America might look different to those in South Africa [Kuffer et al., 2016].

Publicly available datasets denoting land-use in the African continent usually only have 2 classes distinguishing between informal settlements and everything else. Furthermore, these are usually small datasets

typically cover a single city [Mboga et al., 2017; Hurskainen and Pellikka, 2004; Persello and Stein, 2017]. On the other hand, datasets from other developing countries such as India and Indonesia usually only cover random segments of various cities. And these cities' neighbourhoods have very different characteristics to those in South Africa. For instance, middle-income residential neighbourhoods and informal settlements can have high rise buildings. This characteristic makes neighbourhoods associated with different income levels in other developing countries very different from those in South Africa [Kuffer et al., 2016; Ansari et al., 2020; Stark, 2018; Panboonyuen et al., 2019].

As mentioned before, datasets available for land use related tasks usually consist of images covering developed countries such as the US, Germany, the United Kingdom, etc. [Yang and Newsam, 2010a; for Photogrammetry and Sensing, 2020 accessed February 12, 2020; Science and Laboratory, 2017 accessed February 12, 2020] with different visual characteristics to those in many African countries including South Africa. Besides the difference in visual characteristics of neighbourhood types in developed versus developing countries, another limitation of these datasets is that they do not usually cover large geographical spaces (such as an entire country) mostly due to cost constraints.

Although there are some relevant datasets covering small sections of South Africa, they are either not publicly available, are very outdated, do not have labels which would allow us to study neighbourhood types at a higher granularity than formal versus informal settlements [Kemper et al., 2015; Ella et al., 2008; Luus et al., 2013; Mdakane, 2014; Hofmann et al., 2001], or address entirely separate tasks from ours such as building detection [Chatterjee and Poullis, 2019]. Other datasets, however, have a lot more categories which further break down informal and formal residential neighbourhoods into more fine-grained classes such as subsidised housing, high-rise buildings/flats, mining hostels, traditional settlements, formal townships/suburbs with backyard shacks, and formal townships/suburbs without backyard shacks. Unfortunately, these datasets only cover small regions such as single cities, and the methodologies used to create them are too costly and labour intensive to scale to the entire country [Busgeeth et al., 2008; Ikokou and Smit, 2019; Mdakane and Van den Bergh, 2012].

Thus, it is not surprising that we could not find publicly available datasets that would allow us to examine the effects of spatial apartheid in South Africa at scale. The goal of this work is to create such a dataset so that researchers can use machine learning models to classify neighbourhoods based on their visual characteristics. This could then be used to explore neighbourhood classification on future satellite

images of South Africa.

In some cases, a specific location may be accompanied by localized neighbourhood type labels for a particular year but no satellite images for that year (usually in developing countries). In other cases including ours, there may be an abundance of satellite images across time but labels may only exist for one of these years. An example of this case is work done by [Kemper et al., 2015] which classifies satellite images of South Africa captured in 2012 according to land-use, while the ground truth labels obtained from the South African National Land Cover (NLC) dataset were captured between the years 2000 and 2003. This misalignment between the years when satellite images were taken and land-use labels were obtained, creates an inaccurate dataset because new buildings might have been constructed or demolished between 2003 and 2012. This work is also not appropriate for pixel-level image segmentation tasks because the evaluation dataset consists of vector points instead of image masks. Thus, instead of performing the pixel-wise evaluation of results, the authors investigate satellite image regions of specific sizes in their test set and compare the density of buildings predicted by their model to ground truth labels.

Similarly, works like [Robinson et al., 2019] use a combination of high resolution and low-resolution images from 2011-2016. The associated labels, however, were obtained from a combination of land cover maps created as part of the 2011 National Land Cover database [Homer et al., 2015] and the Chesapeake Bay watershed land cover dataset captured in 2013 and 2014. In this work, they assume that land cover labels are less likely to change over short periods of time and use the same labels as ground truth across all of the years. While this method disregards the fact that a region of land can look very different within a short time span, the authors argue that it helps the model generalize better to image differences across the years. They go on to use the 2011 National Land Cover database [Homer et al., 2015] labels together with the 2013-2014 land cover labels from the Chesapeake Bay to train a machine learning model. This model is then used to infer a land cover map of the entire USA which we have no way of evaluating.

It cost the Chesapeake Conservancy over 10 months and 1.3 million dollars to create a six-class land cover dataset covering the Chesapeake Bay watershed. This watershed is approximately 7 times smaller than the size of South Africa, meaning it will cost a lot more time and money to reproduce this dataset in South Africa. Other researchers are exploring cheaper methods which use fewer resources to train

machine learning algorithms for land cover classification. [Jochem et al., 2018] explored using vector data to create a dataset of Afghanistan neighbourhoods in different provinces. This was done by using point data of all the buildings in a particular study area as input data and training a machine learning model to classify these neighbourhoods into two classes: regular neighbourhoods and irregular neighbourhoods. This work depended largely on point patterns to make classification predictions which worked relatively well for two classes. We want to distinguish between multiple classes of neighbourhoods in which some might have similar grid-like patterns but could be visually different (e.g. townships and suburbs).

2.2.3. Models for land cover classification

It is not only datasets for the task of land cover classification that are challenging to construct. Building models which can detect irregularly shaped neighbourhood boundaries and consistently differentiate between them presents many challenges. Some of the earlier methods for image segmentation include works by Tu and Zhu [2002] where they first hand-craft features: they formulated the problem of segmentation by defining an image as a set of k disjoint regions. They then extracted features using computer vision methods like Canny edge detection [Canny, 1986] and colour clustering. They expressed results from these methods as weighted samples used to encode non-parametric probabilities in various image proposals. These probabilities were then used to design importance proposal probabilities used by Markov chains. These feature extraction methods are similar to what convolution and pooling layers in CNN based architectures do. The only difference is that CNN based architectures do not depend on handcrafted features (e.g. hand-tuning Canny edge detection parameters as part of the process).

On the task of satellite image-based classification, work by Persello and Stein [2017] shows that using FCN-based architectures for neighbourhood classification tasks such as slum detection gives better results than using SVMs or patch-based CNNs. Many of the machine learning models used for remote sensing tasks were initially created for small scale object detection purposes [Noh et al., 2015; Long et al., 2015; Chen et al., 2017]. Thus, these models are often relatively good at classifying objects such as cars, chairs, glasses etc, and are adapted to classify larger-scale remotely sensed objects such as lakes, residential neighbourhoods or even roads which have different structures and could span entire images [Liu et al., 2018; Zhou et al., 2018; Zhang et al., 2020]. For remote sensing tasks, architecture/model pipelines which can incorporate this image scaling factor often have an advantage [Liu et al., 2018]. For instance,

the winning land cover classification entry in the DeepGlobe 2018 [Demir et al., 2018] challenge uses a semantic segmentation model based on an FCN architecture [Tian et al., 2018]. The model, named Dense Fusion Classmate Network (DFCNet), fuses shallow and deep layers to capture global (image-level) and local (pixel level) information. DFCNet heavily relies on a clear road network dataset which was used to train its Classmate model. Since many of the townships, villages and informal settlements in South Africa do not have labelled roads, we used the simpler UNet [Ronneberger et al., 2015] architecture which only requires images and ground truth labels to train and takes into account the scaling factor.

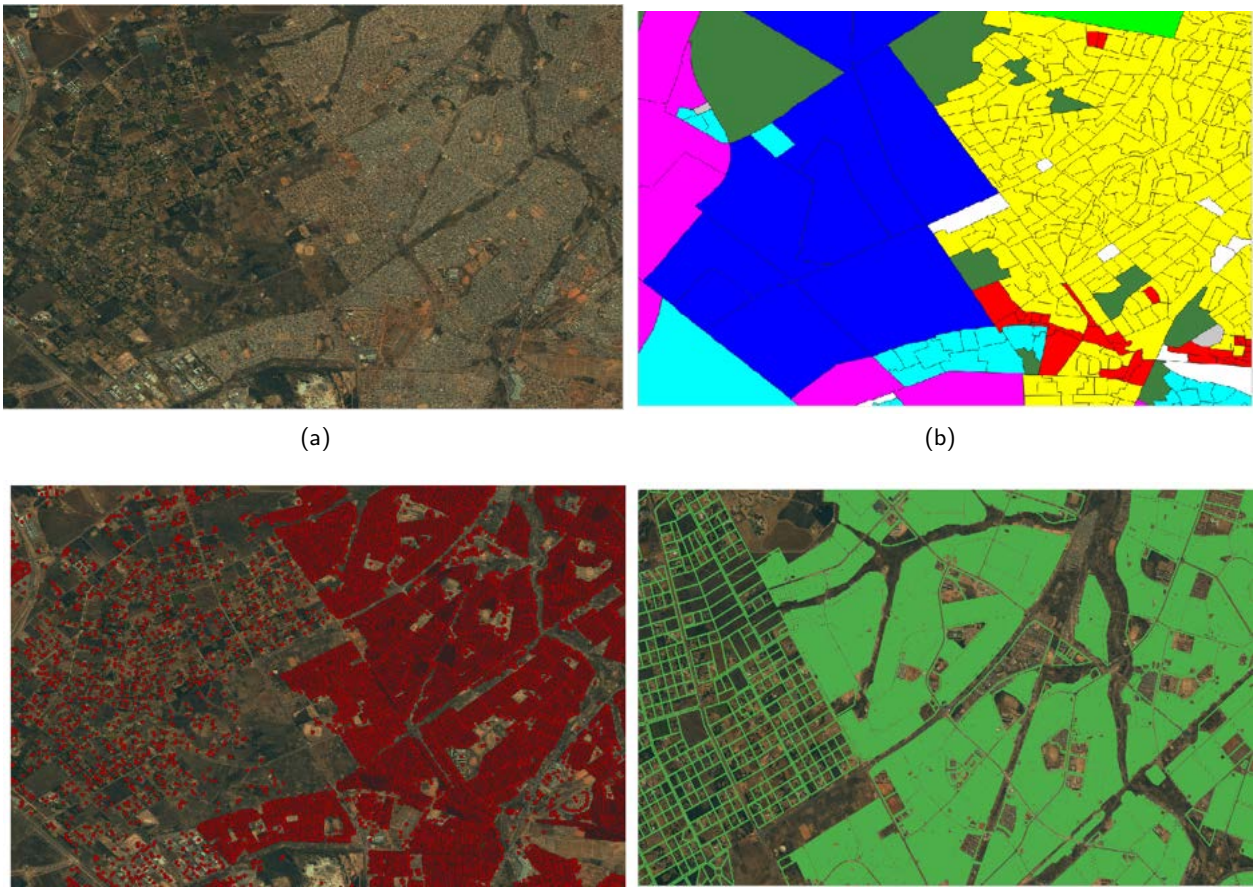
Our objective is to use satellite images to understand the spatial properties of neighbourhoods in South Africa. The spatial apartheid act influenced were different groups in South Africa lived. The objective was to segregate the people living in the country at the time by race so that they can allocate different budgets to the different groups for things like schools and hospitals per neighbourhood. Non-European groups were predominantly moved out of urban areas into places like townships where they were allocated uniformly sized small plots of land on the outskirts of cities and towns. We aim to show that township neighbourhoods are visually different from suburbs, which are also different from informal settlements and villages, by using spatial properties together with visual characteristics of these neighbourhoods. This work requires that 1) we can detect residential areas from satellite images and 2) that we can label the distinguished residential areas according to their classes i.e. as a township, suburb, etc to contextualize the spatial properties we see in South Africa for better understanding.

The rest of this dissertation outlines our methodology for creating a ground truth dataset for the task of neighbourhood classification in South Africa, how we evaluated the quality of the ground truth labels, and baseline experiments we performed on our dataset using a U-Net based neighbourhood segmentation model.

CHAPTER 3

A NEIGHBOURHOOD SEGMENTATION DATASET

3.1. A NEIGHBOURHOOD SEGMENTATION DATASET



(a) (b)

(c) (d)

Figure 3.1: The available 2011 dataset. Figure 3.1a: Satellite images (Suburban area on the left - these are usually sparsely populated and township area on the right - these are usually densely populated). Figure 3.1b: EA dataset covering the same area (Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light Green: Farm, Light grey: Collective living Quarters, Dark Grey: Village, Blue: Smallholdings, White: Background.). Figure 3.1c: Geo-referenced dataset of all the buildings in the image covering the same area. Figure 3.1d: Cadastral dataset of exact-size polygons representing real-estate ownership.

To use machine learning to analyse the effects of spatial segregation, we had to first create a dataset that can be used to train a model which can distinguish between the different types of neighbourhoods

at scale. We created this dataset using satellite images of South Africa captured between 2006-2017 and a combination of several vector datasets that were mostly available for 2011. These vector datasets allowed the construction of pixel-level labels for the satellite images annotated by the relevant class type each label represents. One result of the 2011 census survey is a vector dataset which shows how the South African government divided the entire land and labelled what each piece should be used for: e.g. farming, industrial activity, parks and recreational land, formal and informal residential areas, etc. This is the only freely available and most recent vector dataset labelling all the neighbourhoods in South Africa. The other vector datasets are used to locate the exact position of buildings on these pieces of land. The rest of this chapter describes the datasets used for the task of neighbourhood classification.

3.2. SATELLITE IMAGES

We obtained satellite images covering the entire country of South Africa from 2006-2017 from the South African National Space Agency [SANSA, 2019]. The satellite image dataset consists of images taken from the SPOT satellite sensor, with varying resolutions in different years as depicted in Table 3.1. Given that our ground truth labels were obtained in 2011, we also use satellite images from 2011 to assemble our labelled dataset. These images are at a resolution of 10m which means that each pixel represents 10 meters on the ground, and also use the EPSG:4326 (WGS 84/Latlong) projection. Each image is 21688×21688 pixels and the entire country is covered by a total of 550 images. Table 3.1 shows the number of images per year and the corresponding image resolutions. The top-left image from Figure 3.1a shows an example from the 2011 satellite image dataset. Figure 3.2 shows examples of satellite images for each type of neighbourhood.

3.3. ENUMERATION AREAS

To associate each pixel in our dataset of satellite images with the type of neighbourhood it depicts, we turned to the Enumeration Areas (EAs) dataset created in 2011 by Statistics South Africa (Stats SA) [SA, 2019] which is a South African government agency responsible for conducting the South African census. The dataset consists of land demarcations according to their government-specified use cases (e.g. farms, industrial areas, residential areas, etc.). EAs are geographical units consisting of 100-250 households, used to demarcate locations for which census data is aggregated (similar to census blocks

Table 3.1: The resolution of satellite images in our dataset and the number of images per year. We used the 2011 satellite images to compile the ground truth dataset.

Year	Resolution	Number of images
2006	7550 x 7250	625
2007	7550 x 7250	690
2008	21688 x 21688	545
2009	29406 x 29277	545
2010	21688 x 21688	545
2011	21688 x 21688	550
2012	21688 x 21688	531
2013	21688 x 21688	549
2014	35000 x 35000	549
2015	35000 x 35000	548
2016	35000 x 35000	548
2017	35000 x 35000	543

in the US), which are permanent land use demarcations made by the post-Apartheid government [SA, 2019]. The full list of $103,576$ EAs for all of the 9 provinces in South Africa, along with their types of land use are specified in Table 3.2. Each EA in our dataset is represented by a geo-referenced polygon in the EPSG:4148 - Hartebeesthoek94 projection which is different from the projection used for our satellite images–EPSG:4326 (WGS 84/Latlong). For these polygons to be correctly geo-collocated with our satellite images, we re-projected the polygons to the EPSG:4326 (WGS 84/Latlong) projection using the QGIS remote sensing software [QGIS Development Team, 2009]. This allows us to overlay the polygons on our satellite images, and label pixels according to the type of class each corresponding polygon belongs to.

One shortcoming of the EA dataset is that townships are grouped with suburbs under the label "formal residential areas", which does not allow us to distinguish townships from suburbs. However, each EA is associated with a name. To label the EAs as townships vs not, we looked up a list of South African townships on Wikipedia and labelled the EAs that appeared in the list as townships. To ensure that our labels were accurate, we recruited 10 students who lived in various townships around South Africa to ascertain that our labels matched the location of each township. Figure 3.3 shows EAs covering the Gauteng province along with their associated land-use labels.

(a) Cultivated farm land (b) Farm land (c) Vacant land

(d) Small-holding (e) Traditional residential/village (f) Formal residential/suburb

(g) Township (h) Informal settlement

Figure 3.2: Sample images from our satellite image dataset depicting residential building density in different neighbourhoods. From left-right: Cultivated farm land, Uncultivated farm land, vacant land, smallholding, village, suburb, township and then informal settlement.

3.4. GEOGRAPHICALLY REFERENCED (GEO-REFERENCED) BUILDINGS DATASET

While the EA dataset allowed us to label each pixel with the type of neighbourhood it was associated with, the label does not indicate whether or not the land was *actually* used in the manner indicated by

Table 3.2: Table showing the number of Enumeration Areas (EAs) per EA type per Province made during the 2011 census. In this dataset from Statistics South Africa, the Township class is a part of the Formal Residential class. These EAs cover the entire country.

Province	Formal residential	Informal residential	Traditional residential /Village	Farm	Parks and recreation	Collective living quarters	Indus-trial	Small holdings	Vacant	Comm-ercial	Total
Western Cape	8 136	714	0	674	89	198	167	96	493	251	**10 818**
Eastern Cape	4 952	608	9 150	578	62	171	106	48	2 678	129	**18 482**
Northern Cape	1 456	56	299	788	32	48	49	24	284	24	**3 060**
Free State	3 957	235	501	654	20	103	156	70	537	58	**6 291**
KwaZulu-Natal	6 773	1 011	6 979	687	114	180	287	60	1 236	203	**17 530**
North West	2 535	228	3 052	481	28	130	116	159	434	76	**7 239**
Gauteng	15 283	2 055	216	137	41	560	519	600	859	580	**20 850**
Mpumalanga	2 893	303	3 068	478	97	124	180	84	447	96	**7 770**
Limpopo	1 755	103	7 664	750	56	123	96	75	850	64	**11 536**
Total	**47 740**	**5 313**	**30 929**	**5 227**	**539**	**1 637**	**1 676**	**1 216**	**7 818**	**1 481**	**103 576**

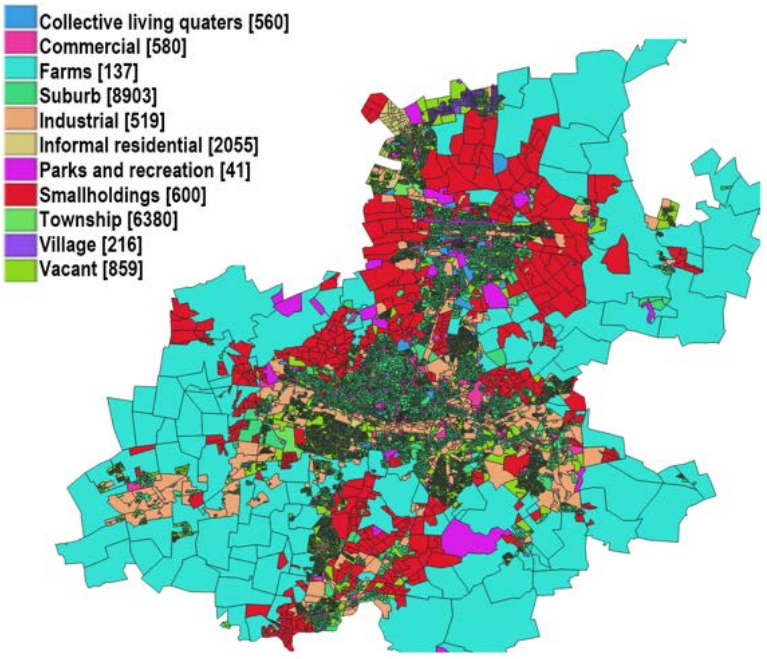
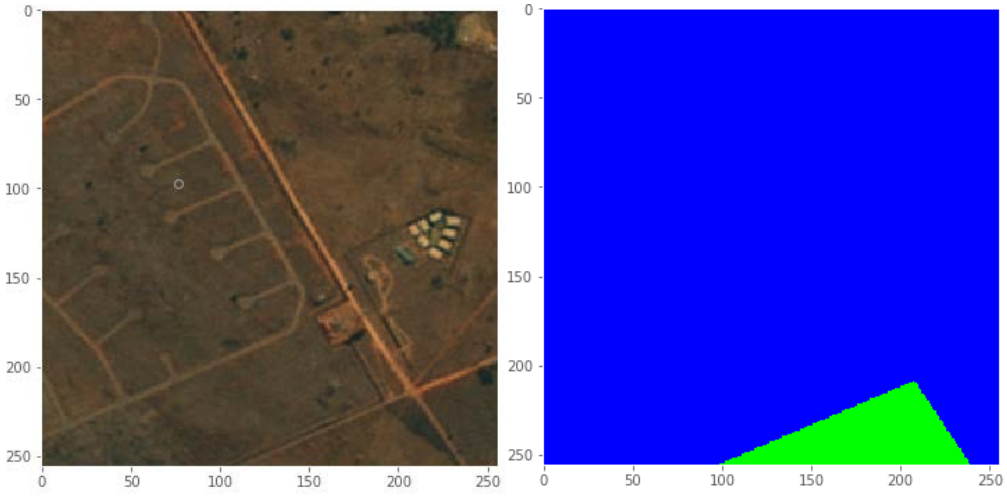


Figure 3.3: Enumeration Areas (EAs) polygons covering the entire Gauteng province which is one of the 9 provinces in South Africa, each colour depicts a certain neighbourhood class as labelled in the key on the left.

the government. For example, as shown in Figure 3.4, there may be no visible difference between land designated as farmland, vacant, or smallholding, if no buildings are indicating how the land was indeed used. The land in Figure 3.4 is designated for small-holding use but no houses have been built yet. We need some way of understanding how the land was *actually* used rather than what it was designated

for. The Geo-referenced buildings dataset tells us where all the buildings are in South Africa. This shapefile dataset was created by Eskom (which is a South African electricity public utility company) in partnership with the Council for Scientific and Industrial Research (CSIR), and consists of building count data in South Africa from 2006 to 2016 as shown in the bottom left image of Figure 3.1. These data points use the EPSG:4326 (WGS 84/Latlong) projection which is the same projection used for our satellite images. Thus, there is no need to perform re-projections to align our data. The dataset captures geographical coordinates of formal, informal and non-dwelling structures. All the data points are geo-referenced and have features such as class names (e.g. dwelling, school, airport etc.), province and municipality describing the building type and location.



(a) Actual frame (b) Ground truth label

Figure 3.4: Example of an image with an associated label of what the land should be used for, while the land is not yet developed for the intended use. In this case the blue and green denote small-holding and farm ground-truth labels respectively, the land is designated to be used for the small-holding type of neighbourhood but is still vacant.

3.5. CADASTRAL DATASET

The Cadastral dataset maps the exact sizes of real estate ownership and spatial information needed to describe the geographic extent of each property. This dataset was captured in 2018 and consists of polygons of plots of land throughout the country which are registered with the South African government. The dataset uses the EPSG:4148 - Hartebeesthoek94 projection which we re-projected to the EPSG:4326 (WGS 84/Latlong) projection to correctly geographically align our datasets. Figure 3.1d shows an

example from this dataset. Plots of land that are either in informal areas that are not registered or vacant land are not covered by the Cadastral dataset. This dataset allows us to see which polygons have buildings along with the exact positions of the buildings. We can then label polygons that are vacant as background (regardless of their designated land use). Note that unregistered plots that have buildings appear vacant in the Cadastral dataset but we do not label these as background.

CHAPTER 4

DATASET CREATION METHODOLOGY

4.1. GROUND-TRUTH DATASET CREATION

The ground-truth datasets we have are not typical ground-truth datasets used in land-cover classification tasks like those in Demir et al. [2018]; Yang and Newsam [2010b]. Those datasets have images (aerial/satellite) accompanied by land cover maps that perfectly cover the phenomena of interest in the images. It is important that the land cover maps have as little error as possible because semantic segmentation models are trained and evaluated using these maps. In this chapter, we discuss the methodology we used to construct our ground-truth dataset. The following subsection first discusses the model we used to iteratively create the ground truth dataset and the performance evaluation metrics which were used to guide us on what the dataset could be lacking for the task at hand. The rest of the chapter discusses the iterative process by which we constructed the dataset along with our results.

4.1.1. The model

One of this project's eventual goals is to be able to classify neighbourhoods precisely and measure the sizes of these neighbourhoods and shapes over time. This is why using a segmentation model was an obvious first choice for us - we want to be able to see where these neighbourhoods start and end. South African neighbourhoods, in particular, are usually divided by a road, which can mean a very wealthy neighbourhood next to a very poor neighbourhood, as depicted by the image in Figure 1.1. Using a classification model which labels an entire image with one class, we might miss many of these cases and not adequately capture these boundaries. In addition to manual inspection during the dataset construction process, we also used a deep learning model to help evaluate the quality of the data, assist in the creation of ground truth labels and guide the search for supplementary sources of data.

A U-Net is a convolutional autoencoder with skip connections between layers in the encoder and the decoder. These skip connections allow the decoder to retrieve information from prior layers in the encoder rather than solely depend on the low-resolution bottleneck (the output of the encoder). Given that we are working with satellite images where each pixel represents a large physical area, it is important for the decoder to reference a larger part of the original image if necessary.

U-Net based architectures have won several semantic segmentation challenges and performed state-of-the-art on neighbourhood classification tasks since its introduction in 2015 Kaggle [2019]; Helber et al. [2019]; Ansari et al. [2020], and its efficiency allows us to train and evaluate models quickly. This was particularly important while trying to understand the nature of our dataset. As we discuss in the sections below, we constructed our dataset using an iterative process where we first trained a model which allows us to see shortcomings in our training data while examining the results using our validation data, then augment/alter our dataset as necessary, and repeat the process.

To segment neighbourhoods in satellite images, we modified [Ronneberger et al., 2015]'s U-Net semantic segmentation architecture to accept input data sizes of 80×80, 256×256 and 2711×2711. The 80×80 pipeline took much longer to process with a lower performance, perhaps because the model sees a small region at a time given the input image size and the resolution of the satellite images (2.5m per pixel). The 2711×2711 pixels performed poorly perhaps because of the opposite reason (seeing a region that is too large). As a result, we used a model with a 256×256 image input to guide our dataset creation process.

Our U-Net model was first trained with the following parameters as defined in [Ronneberger et al., 2015]. It has 23 convolution layers, with the Relu activation function between the layers and the Softmax activation function on the last layer. The model was trained with the Adam optimizer at a learning rate of $1e-4$ [Kingma and Ba, 2014], and we applied batch normalization with batch sizes of 28. We used the categorical cross-entropy loss function since there are multiple classes and each pixel can only belong to exactly one of the classes [Koidl, 2013]. A model with these specifications was used to guide us in the dataset creation process and as a baseline model for experiments on the final dataset.

4.1.2. Performance Evaluation

To evaluate this model we decided to use the accuracy performance metric, the Cohen's Kappa metric and the confusion matrix. The accuracy tells us how many times the model makes a correct prediction and the confusion matrix gives more details for the predictions the model gets wrong. In cases such as ours with imbalanced datasets, performance on the less common classes can be much worse than the more common ones mainly because the model has not seen those instances as often as required for it to learn their characteristics. In addition to accuracy, we use the Cohen's Kappa metric [Kvålseth, 1989] which measures how well our classifier performs relative to what is expected by chance (i.e. if labels

were predicted using a random classifier):

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{4.1}$$

Let us assume that there are 2 classifiers: our classifier and a random classifier. P_o in equation 4.1 is the empirical probability of agreement between the two classifiers. This is the fraction of data points for which our classifier and the random classifier both make correct predictions, and both make incorrect predictions as compared to the ground truth labels. P_e, the expected probability of agreement in equation 4.1, is estimated by calculating the empirical agreement when both classifiers randomly assign labels. For N instances, the probability of a random classifier getting the correct labels is $\frac{1}{N}$ [Landis and Koch, 1977]. κ is always less or equal to 1. When the empirical agreement probability is equal to the random agreement probability, then the value of κ should be 0. This would mean that our classifier is not doing better than a random classifier. For instance, if our classifier obtained an accuracy of 0.5 on a binary classification task with balanced classes, we would expect the Cohen's Kappa value to be around 0 because a random classifier would also achieve an accuracy of 0.5. On the other hand, if we had a binary classification task with an imbalanced dataset consisting of 90% class A and 10% class B, we could train the classifier to always predict class A, and achieve 90% accuracy on a test set with the same fraction of data from class A and class B. However, the Cohen's Kappa value would still be close to zero. Thus, we believe the Cohen's Kappa metric to be a better indicator of performance while working with imbalanced datasets.

4.1.3. Data Preparation

As discussed before, most of the challenges encountered in this project pertained to data. These challenges ranged from the time and resources required to appropriately label the images in the dataset, to the computational resources required to process high-resolution images and create masks used for semantic segmentation. In addition to the steps taken above to alleviate these challenges, we sliced each satellite image of 21688×21688 pixel resolution to sets of size 256×256 pixels of images and corresponding masks for more efficient processing.

4.1.3.1. Vector data alignment

As mentioned in Chapter 3, our satellite images use the EPSG:4326 projection. Thus, to align all components of our dataset, we re-project datasets using different projection systems to EPSG:4326. This is important since accurate building masks can only be obtained if the datasets can be accurately overlaid. While the building dataset already used the same projection as the satellite images, the EA and Cadastral datasets had to be re-projected to EPSG:4326 using the QGIS software.

4.1.3.2. Data splits

While constructing and refining the dataset, we iterated on 19; 21688×21688 pixels of satellite images from Gauteng, Limpopo, North West, Free State and Mpumalanga provinces (Figure 4.1). This subset was chosen to help guide us in creating better ground truth labels during the data creation process. This process involved training and testing a model and then interpreting what kind of data should be added to the data in order to create a better ground truth dataset in the next iteration.

Splitting the dataset: We have chosen to split the data into a $60 : 20 : 20$ training: validation: test ratio. To achieve this we had to ensure that the same pixel does not appear in more than one set which also means that we have to divide the images such that the neighbourhood classes are well balanced according to that ratio. We counted the number of pixels per class per image then separated them into the different sets by attempting to have the $60 : 20 : 20$ ratio for each class. This was done by performing a grid search for a $60 : 20 : 20$ percentage of pixels per class split over our images. Out of the 19 images of size 21688×21688 pixels, 11 images are in the training set, 4 images in the validation set and 4 in the test set.

Although we had planned to perform our experiments only on the Gauteng province, Gauteng has a non-rectangular shape which means that the satellite images also cover parts of the neighbouring provinces (Limpopo province, North West province, Free state province and Mpumalanga province) as depicted in Figure 4.1.

We tile the 19 images of size 21688×21688 pixels into 134,064 images of size 256×256 for easy processing (Table 4.1). These splits are only used for experiments in the dataset creation process. Our final dataset (which we will discuss in later sections) has a total of 550 satellite images of size 21688×21688 pixels which equate to 3880800 satellite images of size 256×256 pixels. Thus the subset

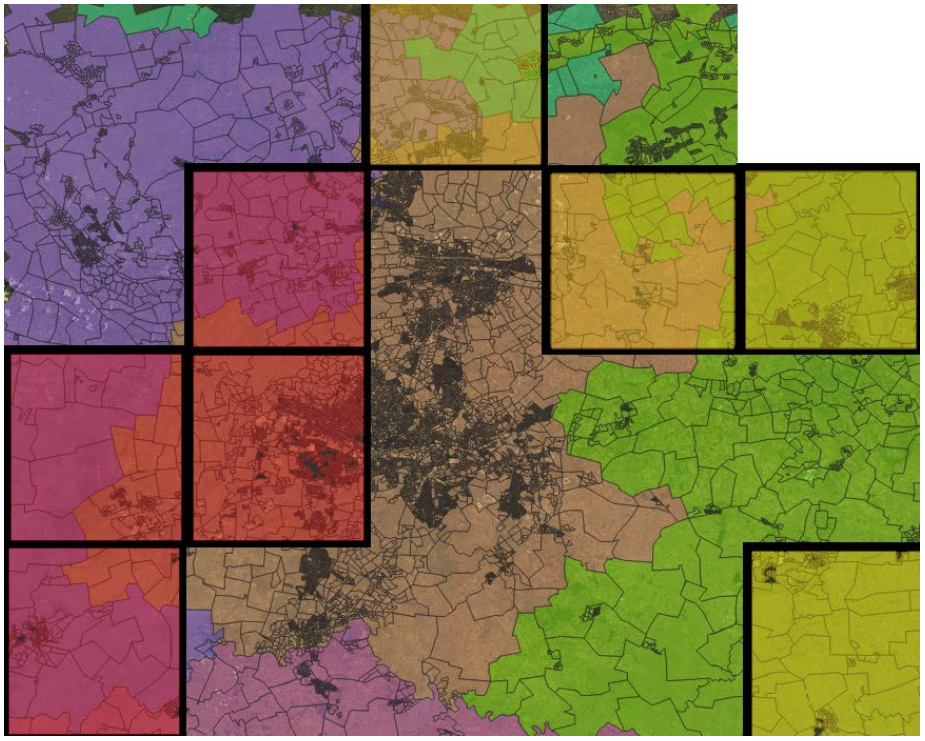


Figure 4.1: The subset of data chosen for creating the final dataset for training a neighbourhood classification model. Each block represents one 21688×21688 pixel satellite image and there is a total of 19 images in this subset. The blocks in red represent images in the test set, those in yellow represent images in the validation set and the rest of the data is in the training set.

we use to tune the dataset is only 3.45% of the entire 2011 satellite image dataset covering South Africa.

The splits in Table 4.1 are also visually illustrated in Figure 4.1. The test set covers the South West of the dataset while the validation set covers the North East and Southern parts of the dataset sample. Although we strove to have close to a $60 : 20 : 20$ split for all classes, as shown in Table 4.2, it was not possible to have the perfect split for all classes. In particular, for the "Parks and Recreation" class we have an $80 : 2 : 18$ split which was not our goal. For classes such as the "Background" class, however, we were able to obtain close to the desired split. Images in each set (train, test and validation) are unique to that set meaning that images in the test set, for example, do not exist in the training set. However, part of a city/neighbourhood in one set can exist in the other set if they share a boundary. We allowed for this neighbourhood overlap in the subset used for the data creation process. But we ensure that there is no overlap in neighbourhoods in the final dataset.

Table 4.1: Splitting the data to create models before scaling to the entire country.

Dataset	**Number of images (21688×21688 pixels)**	**Number of images (256×256 pixels)**
Training set	11	77 616
Validation set	4	28 224
Testing set	4	28 224
Total	19	134 064

Table 4.2: The number of pixels per class for our data subset with 12 classes labelled using a combination of the EA data and the building data.

	Number of pixels			**% of pixels**		
Class	**Train set**	**Validation set**	**Test set**	**Train set**	**Validation set**	**Test set**
Suburbs	67336557	15043710	35303660	57.22	12.78	30.00
Townships	204255256	24861153	65076127	69.43	8.45	22.12
Informal settlement	102382989	9833663	379950086	68.16	6.55	25.29
Village	85567053	32715623	12871645	65.24	24.94	9.81
Small holdings	12464368	2468759	5692240	60.43	11.97	27.60
Collective living quarters	7434193	1455496	3008361	62.48	12.23	25.28
Industrial land	41957836	7542567	13488518	66.61	11.97	21.41
Commercial land	14384681	1708257	2558774	77.12	9.16	13.72
Parks and recreation	3544161	103664	787369	79.91	2.34	17.75
Farms	75362779	43385159	62256169	41.64	23.97	34.39
Vacant	4575672	1050783	1608610	63.24	14.52	22.23
Background	4467376631	1709519230	1609041305	57.38	21.96	20.67

To quickly iterate on the dataset creation process we used a basic U-Net model like the one in [Ronneberger et al., 2015] trained and evaluated on the subset of the dataset described above. That is, we trained the model on 77616 satellite images of size 256×256 pixels and evaluated it on 28224 256×256 images (Table 4.1). We initially used the EA dataset with the 12 classes of neighbourhoods listed in Table 3.2 as labels, and report results on our test set in Table 4.3. Our goal is to build a dataset with ground-truth labels that allow us to train a neighbourhood segmentation model. Working with a smaller dataset and a basic model helps us iterate quickly on building the necessary dataset. The steps we took to create this dataset are outlined in the following subsections.

4.2. USING SOLELY THE EA DATASET AS GROUND-TRUTH

Land cover datasets like the Deepglobe dataset [Demir et al., 2018] and UC Merced dataset [Yang and Newsam, 2010a] require expensive manual labour to create ground-truth data. This limitation often results in datasets that do not cover large spaces [Yang and Newsam, 2010a], [Science and Laboratory,

2017 accessed February 12, 2020], [Demir et al., 2018]. We wanted to find a way to label pixels covering an entire country over multiple years so that we can allow for the evaluation of spatial changes as accurately as possible with already existing datasets.

4.2.1. Ground-truth data preparation

To prepare the ground-truth data, we used the polygons from the EA dataset described in Chapter 3, starting with all 12 classes: Township, Suburb, Commercial, Farms, Informal settlement, Industrial, Parks and recreation, Smallholdings, Village, Collective living quarters, Vacant and Other. The polygons were in shapefile format and therefore geo-referenced which means that we could perfectly overlay them on top of the geo-referenced satellite images.

At this stage, we used 27,636 polygons (which is a subset of the entire 103,576 EAs covering South Africa) labelled into different classes. The goal is to create a mask dataset corresponding to the satellite image dataset where each satellite image has an associated mask image. To do that, we first have to create blank images of the same size as each corresponding satellite image. We then have to read each polygon, spatially locate its corresponding group of pixel coordinates on the satellite image and label the corresponding pixels with the correct colour on the blank image. If one polygon spans two or more images, this algorithm looks for the pixels corresponding to the polygon on each image it spans and then labelling the corresponding blank images accordingly. One disadvantage of this process is that each polygon iterates each and every satellite image in the dataset and this means that we would have to do this read and write process for all 27,636 polygons until we have labelled all the satellite images accordingly.

Dissolve polygons: To make this process more efficient, we used the QGIS software to dissolve the polygons into the 12 class types we have. What this means is that we merge all the 6380 polygons representing township neighbourhoods in our dataset for example, into one big polygon representing townships. Merging individual polygons per class helps us reduce the amount of time the CPU spends reading polygons by a significant amount. This process of merging polygons by a particular attribute is called dissolving and is illustrated in Figure 4.8 as well as explained previously in Section 2.2.

Create masks: To finally create the masks, we overlaid the dissolved neighbourhood polygons onto blank images of the same size as the satellite images, and coloured in the corresponding pixels according

to a specified key set using a combination of the following 3 python-based libraries: GDAL, RasterIO and Geopandas. The images and corresponding masks are in the lossless PNG format because this format does not alter image pixel values while compressing them (as opposed to JPEGs for example). Since we use semantic segmentation models with specific pixel colours as labels, preserving the exact pixel values for the masks is crucial for our task. By the end of this process, we had ground-truth images like the one on the top-right position in Figure 3.1 corresponding to the satellite image data from 2011 like the one on the top-left in Figure 3.1.

4.2.2. Evaluation

We divided the training data into 42 mini-batches and trained the U-Net model for 10 epochs and 100 steps per epoch. The 100 steps per epoch hyper-parameter is used to define how many mini-batches of samples to use in one epoch. We divided our dataset into mini-batches because our experiments use large datasets and using mini-batches reduces training time given that each mini-batch can be processed in parallel on the GPU. Thus, the number of steps per epoch is calculated based on the size of the dataset and the size of the mini-batches.

Examining the results of this initial model uncovered some shortcomings in the dataset as discussed in the prior section. While pixels in the dataset were classified with 61% accuracy, this was a misrepresentation across the majority of the classes. This is due to the high imbalance of farmland pixels in the dataset resulting in the over-representation of farmland (81% of pixels in the training data). Many classes are incorrectly classified as farmland, and visual inspection of results, such as those in Figure 4.3 shows why some classes can be confused for farmland. Farmland that has not been cultivated does look like vacant land. The Cohen's Kappa value for this model was 0.0151 which is very low and tells us that we need to improve either the model or the dataset. Another problem was that since the EA labels only specify the designated land use; vacant, farm, commercial and industrial lands can be confused for each other because undeveloped land often looks like vacant land. It is very difficult even for humans to distinguish farmland from smallholding land in this image on Figure 3.4 for example, and many of these neighbourhoods have open spaces which look like farmland/vacant land. Using solely the EA data as labels, there is no way of tracking which land is developed and which is not. The model was also unable to distinguish between wealthy neighbourhoods with large yet undeveloped yards, a patch of which could look like a farm (column 2 of Figure 4.3), and farmland. A clear next step from this first

Table 4.3: Classification accuracy for various ground truth modifications. "EA" is an abbreviation for the Enumeration Area dataset consisting of land demarcations according to the type of use designated by the government.

Dataset	Accuracy	Cohen's Kappa
EA data: 12 classes	61%	0.0151
EA Dataset + Cadastral: 12 Classes	51.13%	0.36146
EA Dataset + Cadastral: 4 Classes	73.80%	0.5792
EA data + buildings: 12 classes	92.94%	0.6299
EA data + buildings:4 classes	96.14%	0.7578

iteration was to create ground truth labels which enable us to distinguish between background pixels and those representing buildings. Doing this would enable us to learn the visual characteristics of labelled clusters of buildings without the many false positives seen here.

4.3. USING THE EA DATASET + CADASTRAL DATASET AS GROUND-TRUTH

A neighbourhood with an EA label of township doesn't necessarily correspond to an associated polygon representing only houses. Many pixels do not represent buildings within the polygons and they even dominate the space in images of some neighbourhoods. If we have too many non-building pixels labelled as township, it is difficult for a segmentation model to learn the visual characteristics distinguishing a township from a different type of neighbourhood. As discussed above, vacant land designated as a township can be confused for vacant land designated for many other types of neighbourhoods. Thus we need to know which locations have been developed for their designated purposes. The cadastral dataset consists of geo-referenced polygons depicting the exact sizes of government registered real-estate ownership and spatial information needed to describe the geographic extent as shown in the bottom-left image of Figure 3.1. This dataset only labels land which is officially registered with the government. A disadvantage of this is that the dataset does not represent neighbourhoods such as informal settlements and other non-registered land occupancies (as seen by the open pockets of unlabelled built-up land on the right side of Figure 3.1d). To overcome this, we used polygons from the EA dataset for neighbourhoods that are not represented in the cadastral dataset (villages and informal settlements). Note that this means we do not know whether or not land designated for villages and informal settlements has been

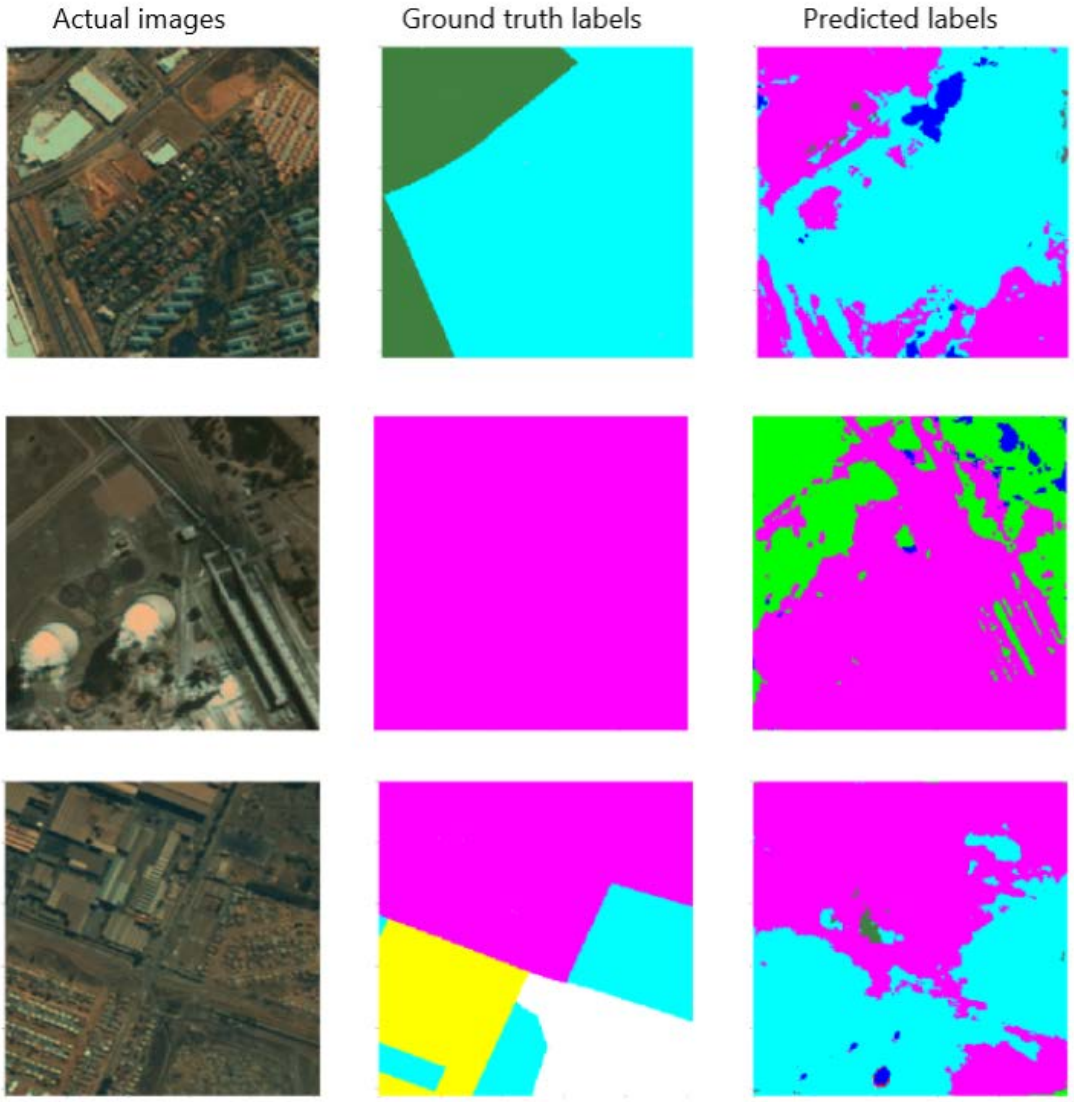


Figure 4.2: Results from training the U-Net model with labels from the EA Dataset. Column 1 shows the actual images, Column 2 shows the ground-truth data and column 3 shows the model predictions. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light green: Farm, Light grey: Collective living quarters, Dark grey: Village, Blue: Smallholdings, White: Background.

developed.

4.3.1. Ground-truth data preparation

To use the cadastral dataset for our purpose, we still need to know which neighbourhoods the real estate/house polygons in the dataset belong to. To do this, we computed the spatial intersection of the

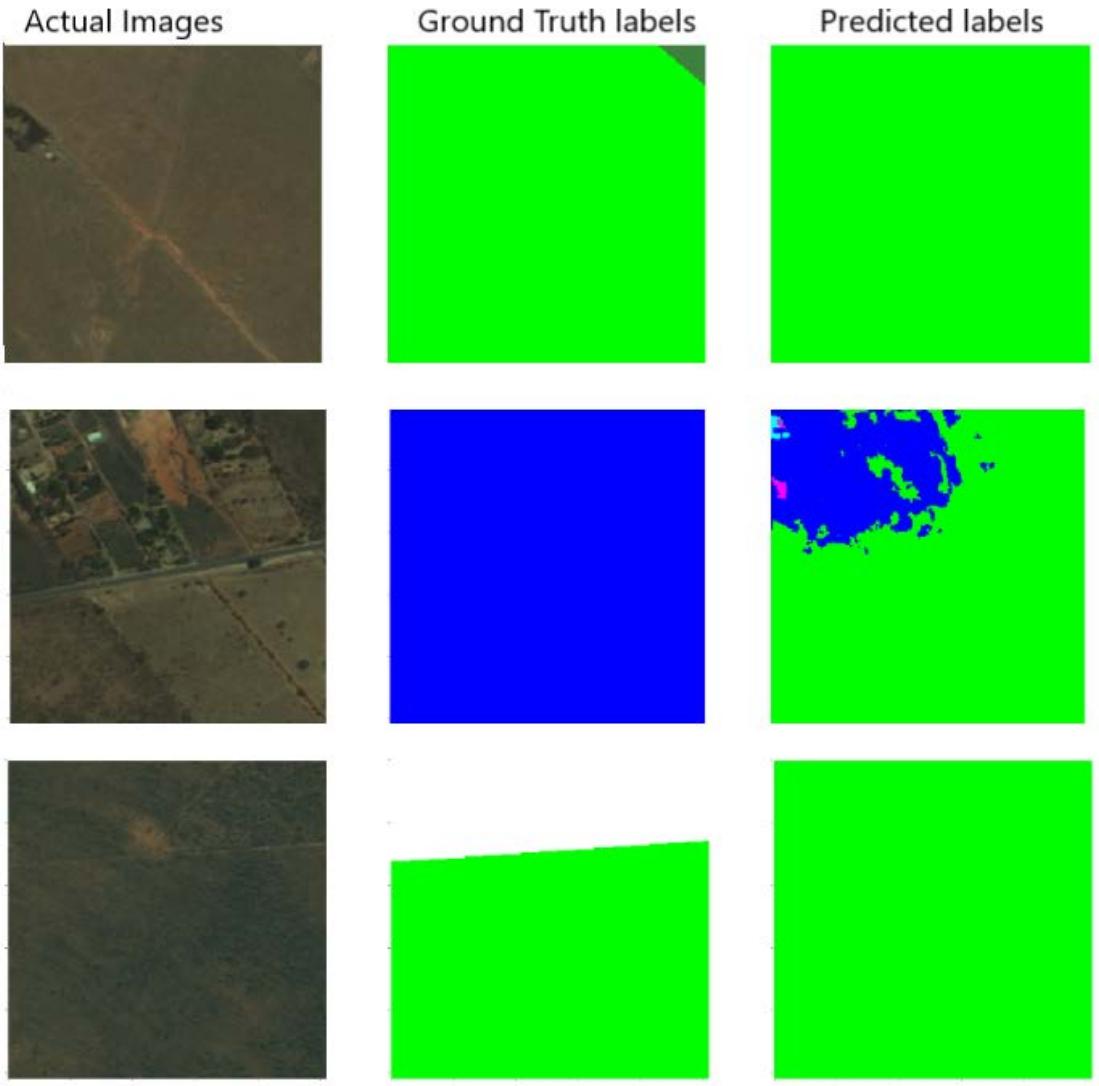
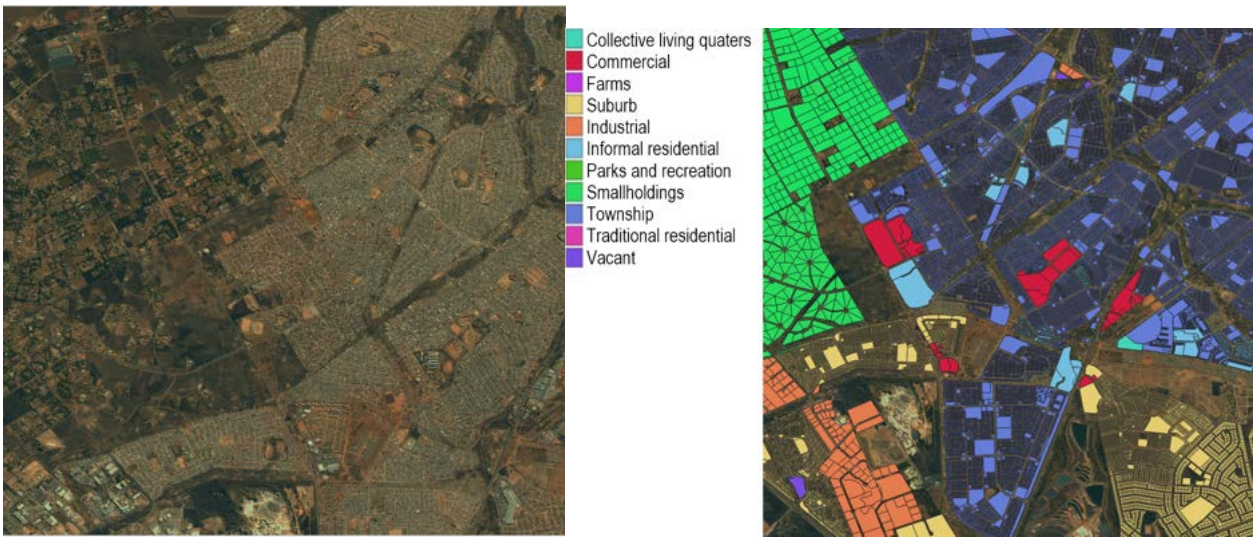


Figure 4.3: Examples of images from our 12 class dataset created using EA ground truth labels. The first column shows the input images, the second column depicts the ground truth labels, and the third column shows our model predictions. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light Green: Farm, Light grey: Collective living Quarters, Dark Grey: Village, Blue: Smallholdings, White: Background.

EA dataset and the Cadastral dataset to obtain the neighbourhood type label from the EA dataset as illustrated in column 2 of Figure 4.5. For example row 2 of Figure 4.5 shows that the informal settlement on the right is not represented in the Cadastral dataset (the informal settlement label shown in red comes from the EA dataset as shown by the lack of precise land demarcations in the masks created using the cadastral dataset depicted in yellow). Figure 4.4 illustrates what this ground truth dataset looks like.



(a) Satellite image (b) Corresponding ground truth label

Figure 4.4: Example image from this dataset. Figure 4.4a shows the satellite image and Figure 4.4b shows the corresponding mask.

4.3.2. Results

We trained this U-Net model for 30 epochs and 100 steps per epoch–these hyper-parameters were found through a grid search on our validation set. The error and validation accuracy did not change significantly for values higher than these numbers. Training and evaluating the U-Net model on this dataset, we obtained an accuracy of 51.13% and a Cohen's Kappa value of 0.36146. Looking at the confusion matrix, we saw that most of the confusion was between classes with similar visual characteristics like farmland and vacant land and it is unlikely that even a human would be able to distinguish between these classes visually. To alleviate this confusion, we decided to collapse the 12 classes in our data set into 4 visually distinct categories: background, wealthy neighbourhoods, non-wealthy neighbourhoods and non-residential building clusters. The goal here was to combine classes with similar visual characteristics as guided by the confusion matrix. The 4 classes are: background (combining vacant, parks and recreational areas and farms), commercial space/non-residential buildings (commercial areas and industrial areas), wealthy residential areas (suburbs and smallholdings) and non-wealthy residential areas (townships, informal settlements and villages). We decided on these classes after investigating the results of the 12 class confusion matrix. In addition to collapsing the classes, we also reduced the number of background pixels by first removing all the images in which at least 70% of the pixels belong to the background

class. The value 70% was chosen arbitrarily. Table 4.4 shows the number of images in this new subset. Training this model on these new classes gave us an accuracy of 73.80% and a Cohen's Kappa value of 0.5792. Looking at the confusion matrix on Table 4.6 for the collapsed classes dataset, there was much less confusion between the classes and the overall accuracy value was more representative of the individual class accuracy for each of the 4 classes.

We compare our model trained and evaluated on 4 classes with one first trained on 12 classes and then evaluated on the same 4 by merging the classes after the fact. To collapse these classes, we merged the values from the corresponding labels in the class confusion matrix which was generated from the model trained on 12 classes. As stated above, the 4 classes are: background (combining vacant, parks and recreational areas and farms), commercial space/non-residential buildings (commercial areas and industrial areas), wealthy residential areas (suburbs and smallholdings) and non-wealthy residential areas (townships, informal settlements and villages).

Table 4.5 shows the performance of the 12-class model evaluated on 4 classes. The model which was trained on 4 classes performs much better than the one trained on 12 classes but evaluated on the 4 classes. For instance, the model trained on 4 classes achieves an accuracy of 75.5% on non-wealthy neighbourhoods, and that trained on 12 classes but evaluated on 4, achieves an accuracy of 54.8%– classifying a large portion of non-wealthy neighbourhoods as background (40.9% compared to 21.1% for the 4-class model). Since we kept the number of training examples constant while training each model, one reason for this difference in performance could be that the 4-class model had more training examples per class as compared to the 12-class model.

Table 4.3 shows that the 12 class neighbourhood segmentation is a difficult task. While an accuracy of 61% is achieved in our first attempt of creating ground truth labels, the Cohen's Kappa value is very low. The over-representation of farmland pixels in the dataset allowed more of them to be correctly predicted hence the high accuracy, but many other classes were incorrectly classified showing the low κ value. As the dataset was more balanced, we see an increased κ value due to the improvement of classification accuracy across classes, even though the overall accuracy decreased. Collapsing classes in conjunction with balancing the dataset and creating improved ground truth labels focused on occupied land results in increased values of both κ and accuracy. As shown in Table 4.6, the model predicts more classes accurately.

Table 4.4: The number of EAs per collapsed class. The number of Background polygons is at most 1,797 due to the reduction of images consisting of only farm land pixels.

Attribute	**Number of EAs per class**	**Number of pixels/class**	**Number of images per class in test set**
Background	1 797	34 533 815	527
Non-Residential Neighbourhoods	1334	3 652 323	56
Wealthy Residential Neighbourhoods	12 754	10 324 170	158
Non-Wealthy Residential Neighbourhoods	11 690	17 025 692	260

Table 4.5: Confusion matrix of the results of the model trained on 12-class labels but evaluated on the 4-class labels of ground truth labels augmented with Cadastral dataset.

	Non-Residential Neighbourhoods	**Wealthy Neighbourhoods**	**Background**	**Non-Wealthy Neighbourhoods**
Non-Residential Neighbourhoods	**0.5**	**0.193**	**0.196**	**0.113**
Wealthy Neighbourhoods	**0.1008**	**0.584**	**0.2857**	**0.022**
Background	**0.0019**	**0.06275**	**0.758**	**0.1888**
Non-Wealthy Neighbourhoods	**0.006**	**0.0229**	**0.409**	**0.548**

Table 4.6: Confusion matrix of a model trained on collapsed classes and ground truth labels augmented with Cadastral data. Accuracy $= 73.8\%$.

	Non-Residential Neighbourhoods	**Wealthy Neighbourhoods**	**Background**	**Non-Wealthy Neighbourhoods**
Non-Residential Neighbourhoods	**0.4478**	**0.0993**	**0.2955**	**0.1575**
Wealthy Neighbourhoods	**0.031**	**0.6575**	**0.0363**	**0.0363**
Background	**0.0162**	**0.0655**	**0.7842**	**0.1342**
Non-Wealthy Neighbourhoods	**0.0067**	**0.0269**	**0.2111**	**0.7553**

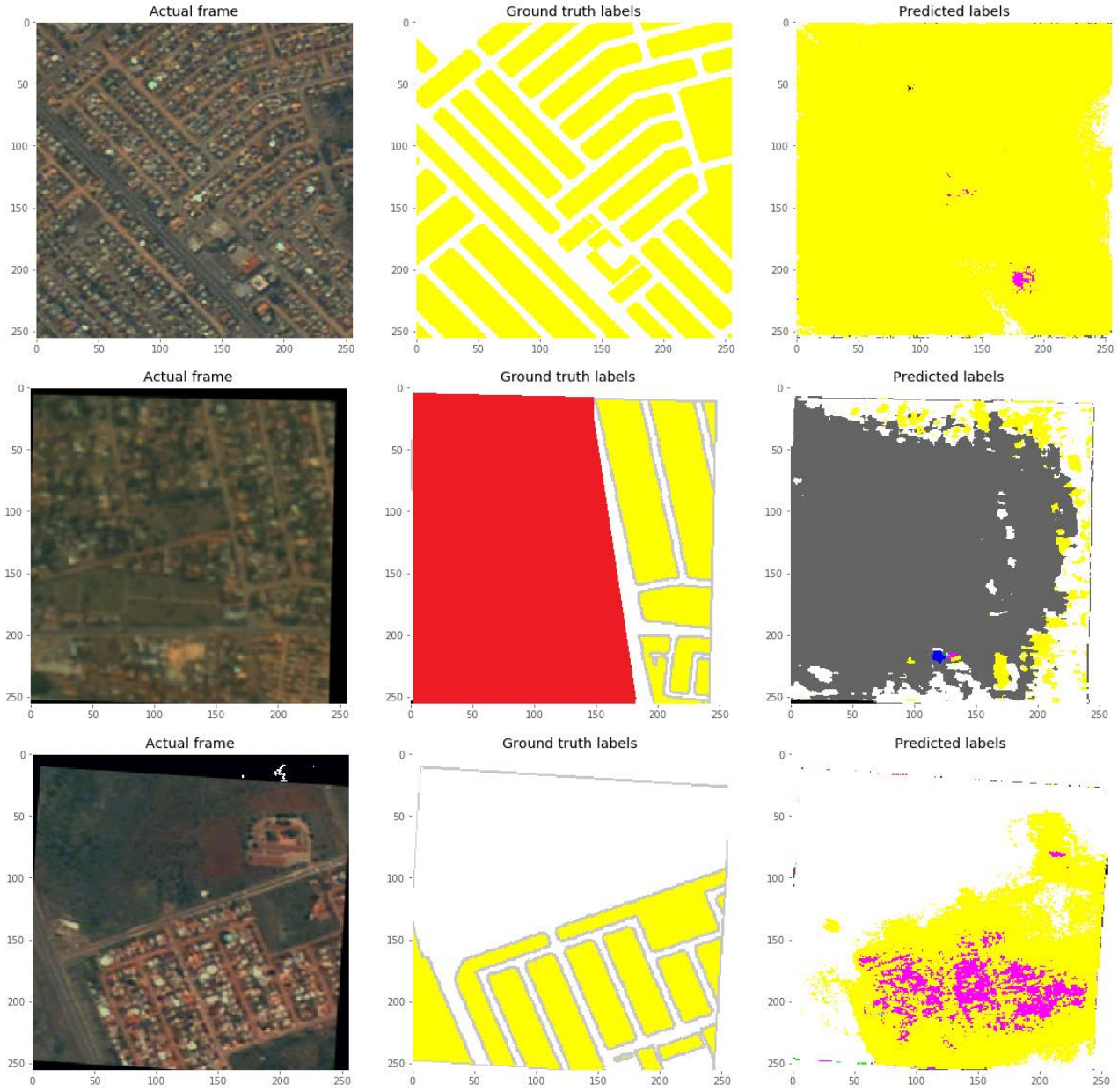


Figure 4.5: Results from training the U-Net model using labels from the EA Dataset + Cadastral Dataset. Column 1 shows the actual images, Column 2 shows the ground-truth data and column 3 shows the model predictions. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light green: Farm, Light grey: Collective living quarters, Dark grey: Village, Blue: Smallholdings, White: Background.

4.4. USING THE EA DATASET AND BUILDING DATASETS AS GROUND-TRUTH

The cadastral dataset together with the EA dataset covers the exact sizes of buildings and yards and defines the neighbourhoods they belong to. This dataset also distinguishes between neighbourhoods that have actually been used for their designated purposes versus those that have not. However, the cadastral dataset was captured in 2018 and the EA dataset in 2011. To train a segmentation model using satellite images from 2011, we would have to use labels from 2018 although there may be significant differences in real estate between 2011 and 2018. Thus we needed yet another way of creating ground truth labels that give us information regarding real estate in 2011. Our final dataset uses the geo-referenced buildings dataset described in chapter 3 rather than the cadastral dataset. As a reminder, this dataset consists of building locations in South Africa from 2006 to 2016, capturing formal, informal and non-dwelling structures.

4.4.1. Ground-truth data preparation

To use the building dataset for our neighbourhood classification task, we followed the procedure below to create masks that can be overlaid on top of our satellite imagery, assigning each pixel to the desired class.

Buffing points into polygons: The first step was to use the buffer algorithm in Section 2.2 to transform each point (a single latitude and longitude coordinate pair) into a circular polygon of a specific radius. In our case, we inflated the points by a distance of 0.0007 decimal degrees. We arrived at this number through a trial and error search, looking for polygons which covered an average suburban house together with its yard (Figure 4.6). Buffing allows large swaths of vacant land to be labelled as background.

Each circle can either be too big in informal areas where there are small compounds or too small for the building as in the case of the industrial neighbourhood class. A polygon that is slightly too large for the compound will still allow our neighbourhoods to be appropriately labelled as in row 2, column 2 of Figure 4.11. But a circle around a larger compound, e.g. in smallholdings or industrial areas, only usually covers the building as shown in row 1, column 2 of Figure 4.11. We assume that a semantic segmentation model such as U-Net that has access to images at different scales should be able to learn

visual characteristics distinguishing these classes given enough examples.

Spatial Intersection: To label which neighbourhood each building belongs to, we computed the spatial intersection of the EA dataset and buffed building dataset as demonstrated in Figure 4.7, joining the datasets at points where they overlap.

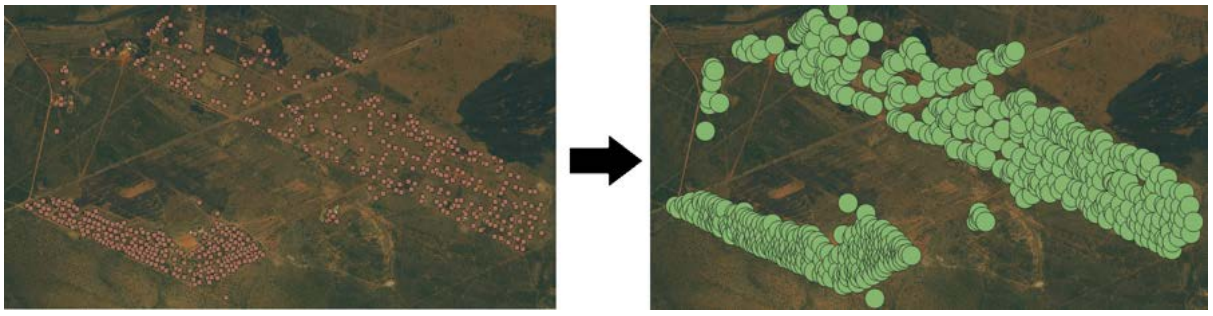


Figure 4.6: Converting building point data to polygons using a buffer algorithm so that we can approximate the space covered by the building. The images illustrated here are at a resolution of 2.5m per pixel and size of 786 x 386 pixels.

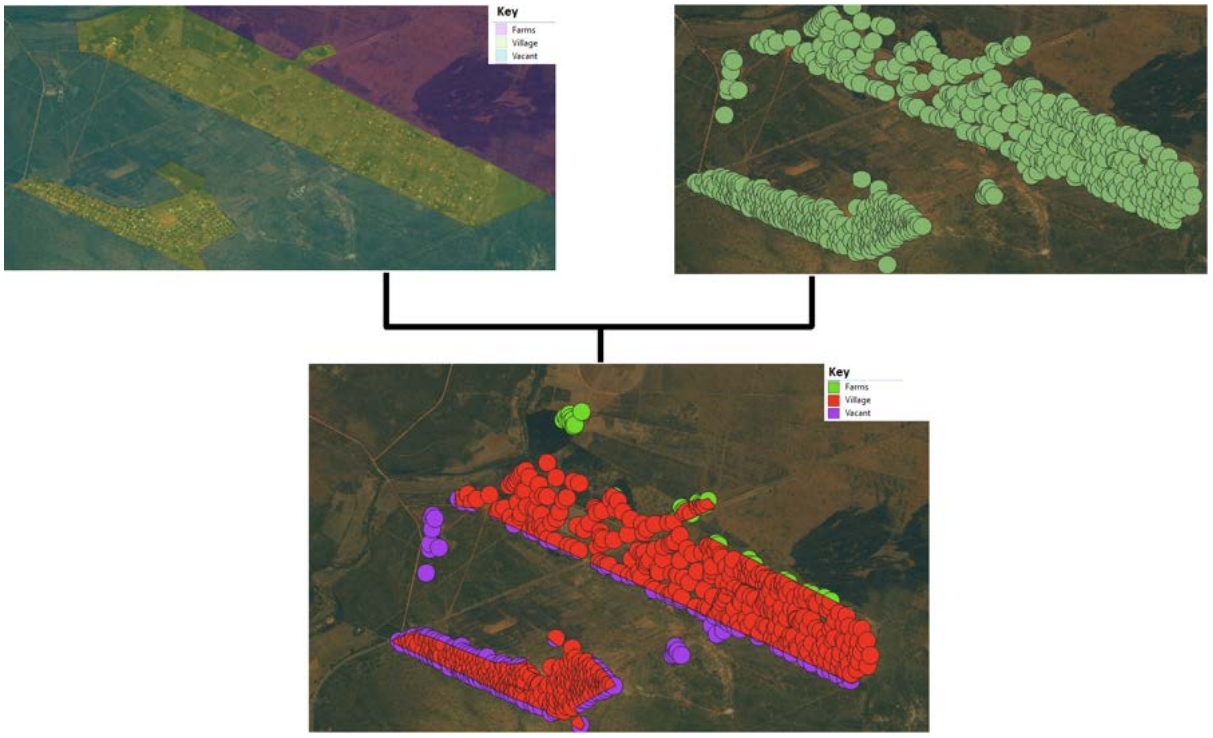


Figure 4.7: Computing the spatial intersection between the land use labels from the EA dataset and the buffed building polygons so that we can know the neighbourhoods in which these houses belong.

Dissolve polygons by neighbourhood types: Before the dissolving step, the building polygons con-

sisted of over 10 million data points each saved in shapefiles, resulting in computationally expensive read and write operations to convert them to image masks. This step reduces the over 10 million polygons significantly by grouping those that belong to the same neighbourhood together.

Create masks: We overlaid a combination of the EA and building datasets on top of the satellite imagery to create labels designating each pixel as one of the 12 classes or background if no building was built on the land (Figure 4.9).

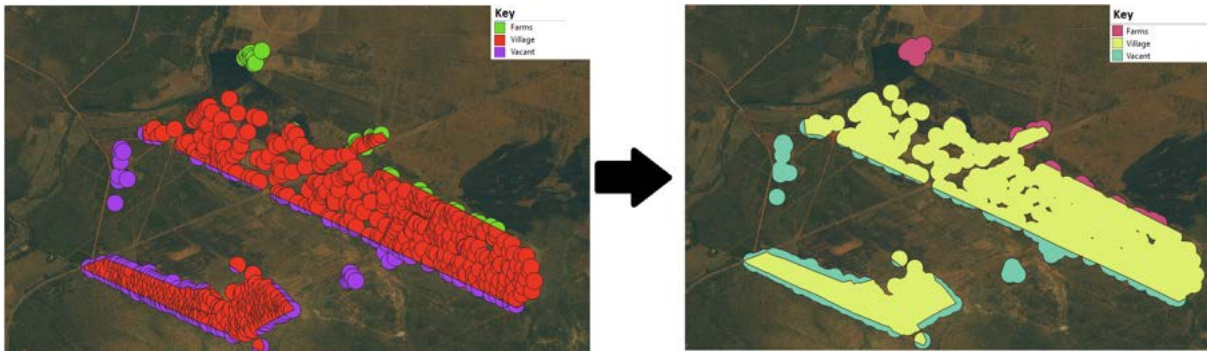


Figure 4.8: The process of dissolving overlapping building polygons by neighbourhood.

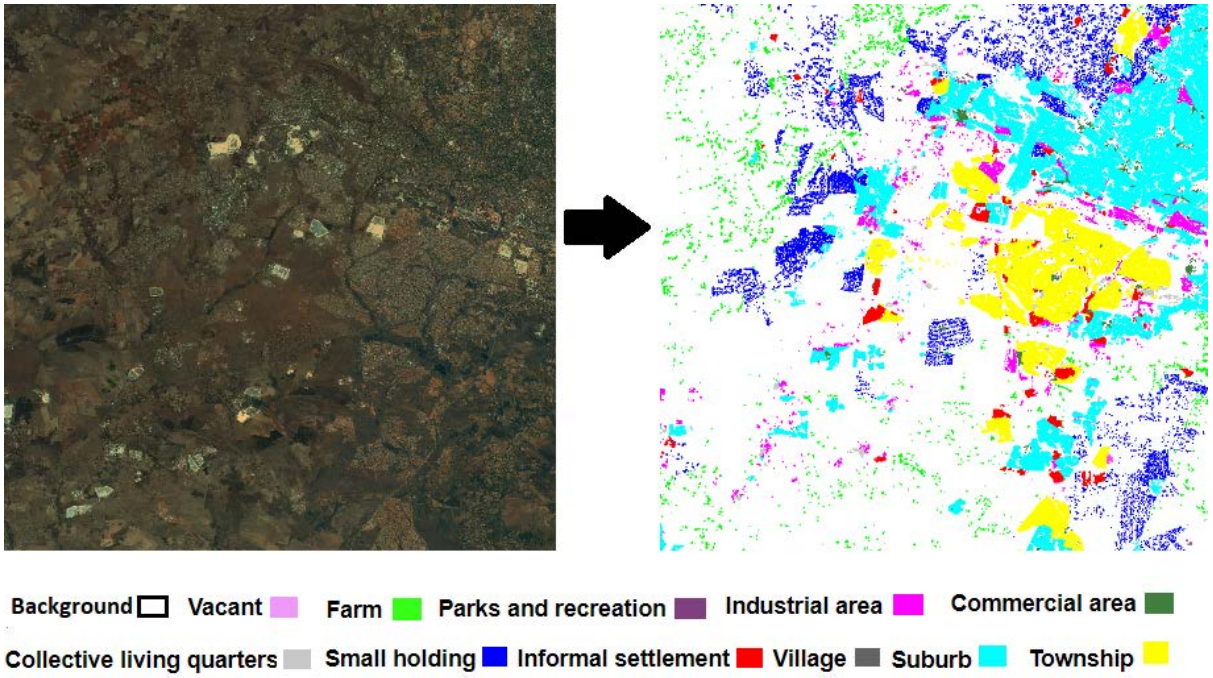


Figure 4.9: A 21688 x 21688 pixels satellite image with the corresponding ground truth mask.

4.4.2. Segmentation results

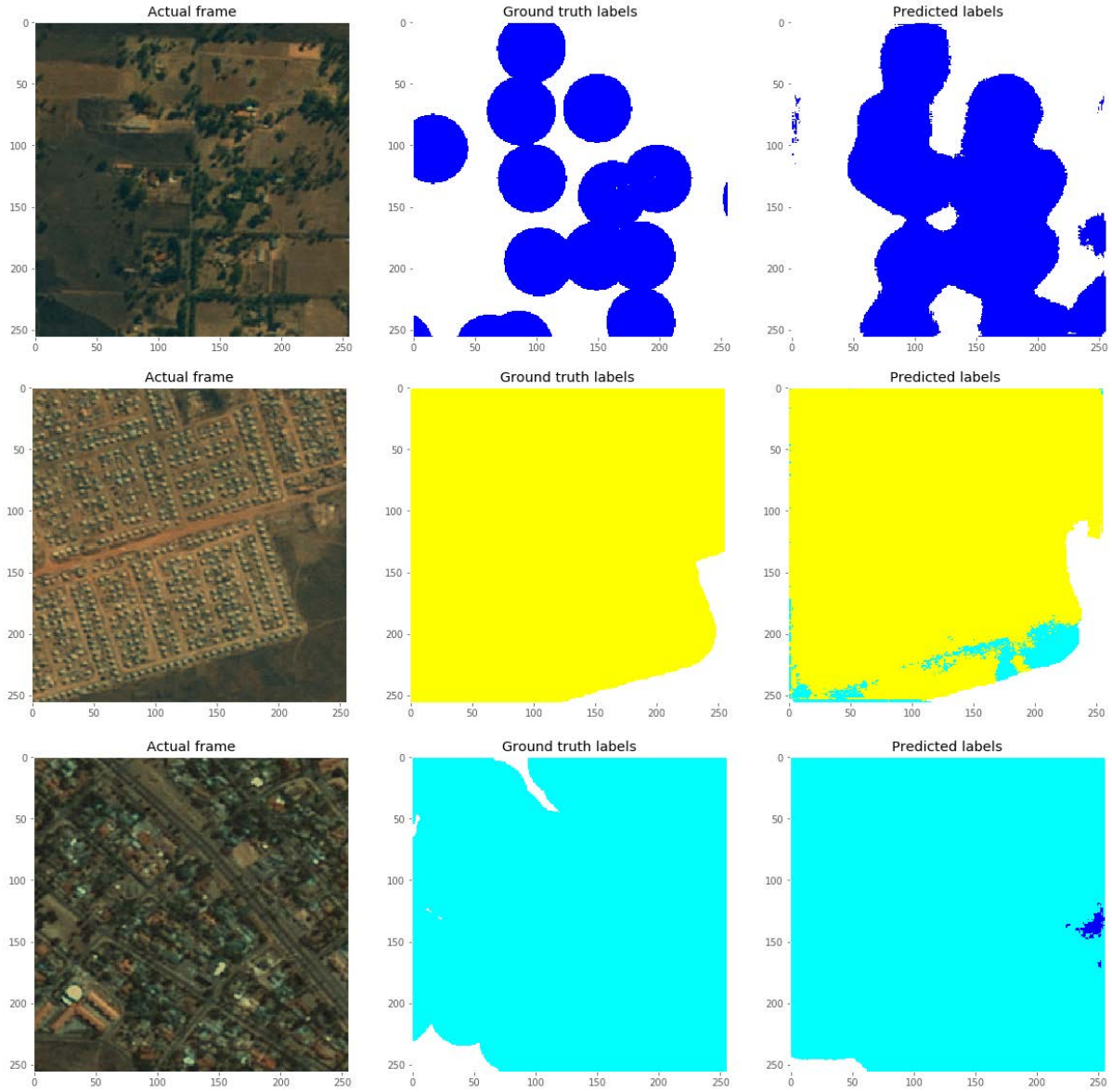


Figure 4.10: Results from training the U-Net model using labels from the EA and buildings datasets as Ground-truth. Column 1 shows the images, column 2 shows the ground-truth labels and column 3 shows the model predictions. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light green: Farm, Light grey: Collective living quarters, Dark grey: Village, Blue: Smallholdings, White: Background.

We trained a U-Net model for 30 epochs and 100 steps per epoch on this dataset as before. Training a U-Net model on this dataset achieves an accuracy of 92.92% and a Cohen's Kappa value of 0.6299

on the 12 classes, a significant improvement from the previous values of 0.0151 and 0.36146 on the 12 class classification tasks (results summarized in Table 4.3).

Since the building polygon dataset was created using building data points, labels on farmland, for example, depict the buildings on farms and not the farm itself. Because of that, the farm will be labelled as background and the buildings on the farm will be labelled as farmland. The same is true for classes representing vacant land and parks and recreational areas as depicted in the bottom right image pair of Figure 4.11.

Since the building masks are now represented with circular polygons, the model has also learnt to mimic this by identifying pixels representing buildings and undeveloped land around them which fall within the circles' radius as illustrated by row 1 of Figure 4.11. This model better distinguishes undeveloped land from houses, and the high Cohen's Kappa value of 0.6299 indicates that the model performs much better than random chance predictions.

4.5. FINAL DATASET COMPOSITION

We have now created a dataset of ground truth labels which we have used to build a neighbourhood segmentation model as illustrated in Figure 4.11. It consists of 12 classes representing neighbourhood types such as townships, suburbs, informal settlements and villages spanning the entire country. The objective is to be able to distinguish neighbourhoods in South Africa using their visual characteristics, which would allow us to eventually study how they evolve over time. Do some townships look like suburbs over time? Do their sizes shrink or expand? While we have images from 2006-2017, we only have labels for images in 2011. Thus, a neighbourhood segmentation model can be trained and evaluated on data from 2011, and this model can then be applied to our images in other years. The results cannot be accurately evaluated for other years since we do not have neighbourhood labels for years other than 2011. We can however visually inspect the predicted changes and see what we find.

4.5.1. Final dataset size

There are 550 satellite images covering the entire country, however, most of these images are of farmland/non-built up land which is why we sampled our final dataset from regions with more residential neighbourhoods. In total, the final dataset consists of 100 images of size 21688×21688 pixels

from the year 2011. As illustrated in Table 5.1 this equates to approximately 700000 images of size 256×256.

4.5.2. Final dataset train/validation/test split

The training data consists of provinces in northern South Africa (which includes all the images we used to create the dataset), and the KwaZulu-Natal province located on the eastern coast of South Africa. This is 60 images of size 21688×21688 pixels from the following 6 provinces: Gauteng, North West, Limpopo, Free State, Mpumalanga and Kwa-Zulu Natal. The test set consists of 20 images of size 21688×21688 pixels from both the Western Cape and Northern Cape provinces. Finally, our validation set consists of 20 images of size 21688×21688 pixels from the Eastern Cape province.

Table 4.7: Train, test and validation splits for the final dataset.

Dataset	**Number of images (21688x21688 pixels)**	**Number of images (256x256 pixels)**
Training set	60	402,192
Validation set	20	148,176
Testing set	20	148,176
Total	100	698,544

While creating the train/validation/test splits, we ensured that a single neighbourhood did not span multiple splits. This was done by having each split sampled by province, i.e. the validation set only spans the Eastern Cape province. We further illustrate this visually in Figure 4.12 where we show exactly where the samples are on the map.

During the compilation of the training dataset specifically, we picked images which cover different sceneries in the dataset such as densely/sparsely populated areas, mainland/coastal land and different ecosystems like forests/grassland. This is because we would like to train a neighbourhood segmentation model that generalizes across the whole country.

Conclusion

The goal of this work is to study the spatial makeup of neighbourhoods in South Africa using machine learning. To achieve that, we first need a ground truth dataset which can be used to train a model. From

the previous chapters, we have learnt that creating such a dataset can be expensive and that the expense is directly proportional to the size of land. We have explored using readily available vector datasets and satellite images to create a ground truth dataset which consists of 12 different neighbourhood types.

As described in the previous subsections, we started by converting polygons of land-use as defined by the government in Stats SA's EA dataset into ground truth image masks. The problem with this technique was that undeveloped land was also classified as Suburb for example, and there was no way to distinguish between developed and undeveloped land as depicted in Figure 4.13b. In the next iteration, we considered using the cadastral dataset which consists of polygons depicting real estate ownership as registered officially with the government. The problem with this dataset was that it did not document unregistered land occupancy, i.e. real estate in informal settlements and villages was not documented in this dataset. There was also no way to verify whether or not the registered land was developed. Figure 4.13c shows the same plot as in Figure 4.13a but overlayed with the cadastral data. The undeveloped land on the left side is captured as part of the ground truth mask which is misleading. Combining a modified version of the building and EA datasets we created a dataset which captures clusters of buildings in each neighbourhood. The modified building dataset only captures buildings across all classes and the EA dataset labels the building clusters into the different neighbourhood types they belong to. This allows us to more accurately track the location of background pixels (as depicted in Figure 4.13d) and train machine learning models that focus on the visual characteristics of buildings and their surrounding areas.

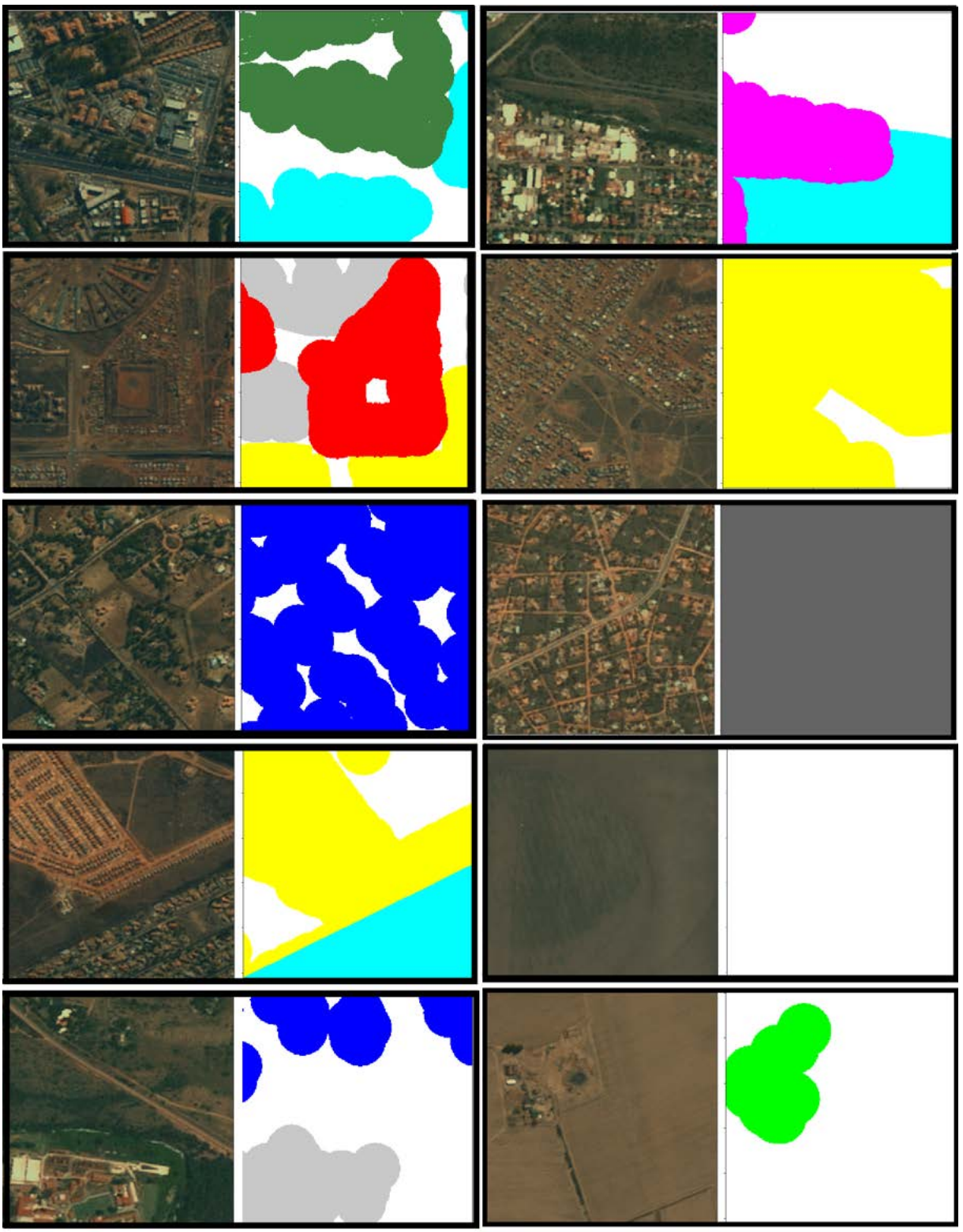


Figure 4.11: Samples from the final dataset which consists of image and mask pairs. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light Green: Farm, Light grey: Collective living Quarters, Dark Grey: Village, Blue: Smallholdings, White: Background.

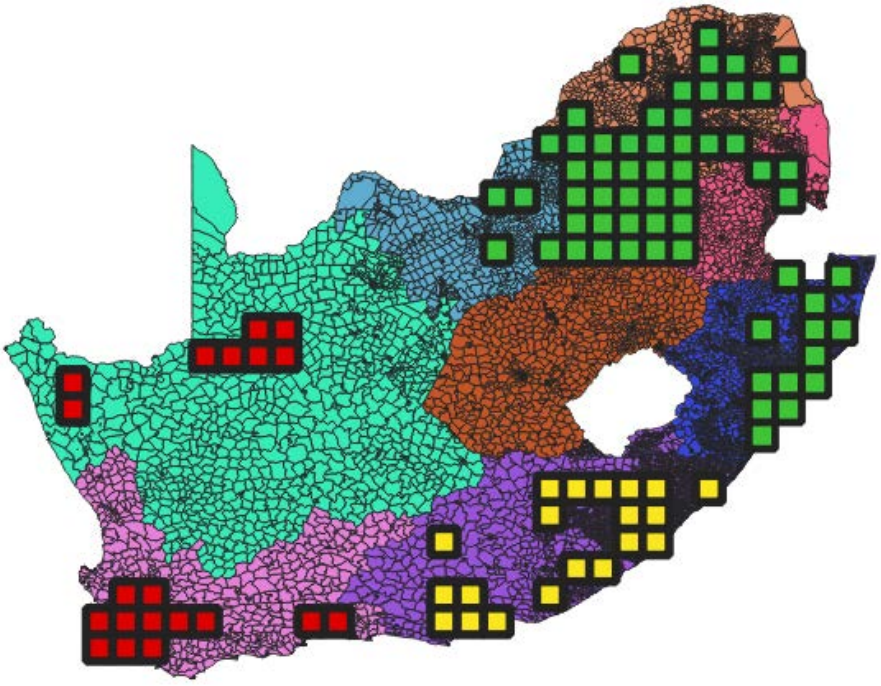
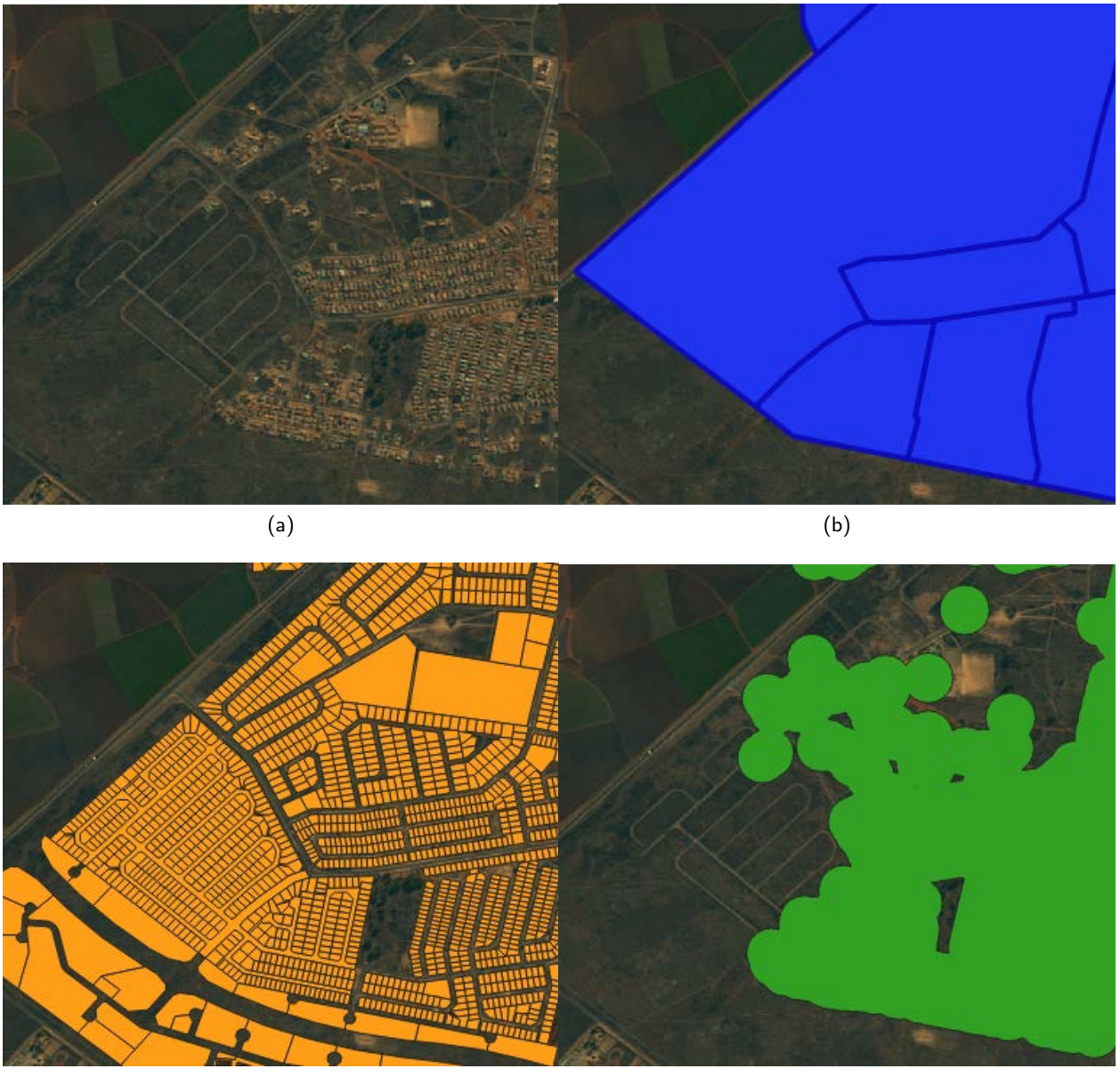


Figure 4.12: The final dataset for training a neighbourhood classification model. Each block represents one 21688×21688 pixels satellite image and there is a total of 100 images in this subset. The blocks in red represent images in the test set, those in yellow represent images in the validation set and those in green represents the images in the training set.



(a) (b)

(c) (d)

Figure 4.13: Given the actual image in Figure 4.13a, the aim was to create ground truth data which captures clusters of buildings defined as a neighbourhood. We iterated from the ground truth images in Figure 4.13b using just EA data to Figure 4.13c, using a combination of EA data and Cadastral data and then finally settling on Figure 4.13d using a combination of building data and EA data.

CHAPTER 5

BASELINE RESULTS AND FUTURE WORK

After following the procedures described in prior chapters, we now have a dataset of labelled satellite images enabling us to train and evaluate models for South African neighbourhood segmentation. In this chapter, we describe preliminary experiments on the final dataset, what we have learned from inspecting the results of these experiments, and future work which we hope to be enabled by our dataset.

5.1. EXPERIMENTS

Table 5.1: Classification accuracy after training a model using the final dataset. Unbalanced refers to the training set as is, while balanced refers to results after balancing the training data.

Dataset	**Accuracy**	**Cohen's Kappa**
Final dataset:unbalanced	91%	0.29
Final dataset: balanced	57.45%	0.4326

5.1.1. Technical setup

For both experiments described below, we used a similar hyper-parameter setup like the one described in Chapter 4.1–using a grid search on the validation set to tune the hyper-parameters. Our model consists of a U-Net architecture with 42 layers, batch normalisation at batch sizes of 42 and a categorical cross-entropy loss function with the Adam optimizer at a learning rate of $1e-4$. The only difference is the number of epochs we trained for and the number of steps per epoch. A grid search on our validation set found that with this larger dataset, increasing the number of epochs to 60 gave the best validation accuracy and similarly, increasing the number of steps per epoch to 1240 gave better results. Our grid search showed that training past $2,000$ epochs results in overfitting. While training for 60 epochs in all our experiments, we use the early stopping technique which allows us to stop training if there is no change in the validation loss, and save the best weights for the network.

5.1.2. Experiment 1: Baseline model on our final dataset

We trained a U-Net baseline model on our final dataset described in section 4.5. To recap, the dataset consists of $402,192$ training images of size 256×256, $148,176$ validation images and $148,176$ images in the test set. We perform experiments on the un-collapsed version of the dataset in our experiments with

the 12 classes as specified in Chapter 4. Training this model took nine and a half hours on an NVidia Titan V Graphical Processing Unit (GPU) on a machine with 16GB RAM. One significant difference from the testing phase of the data creation step with this technical setup however was performing inference during the testing phase. Semantic segmentation requires inference on individual pixels. $148,176$ images with 256×256 pixels each result in approximately 4.1 billion pixels that have to be forward passed through the model to obtain predictions. This is 5 times the number of pixels we used during the dataset tuning process. Storing the prediction versus ground truth arrays of length 4.1 billion is infeasible using our computational resources. We thus decided to store the prediction vs ground truth arrays in a Comma Separated Value (CSV) file on disk. Saving the neighbourhood class prediction names as digits instead of strings (e.g. 11 instead of "Township") also reduced our memory usage because much more memory is allocated for a string variable than for an integer variable.

Since the majority of South Africa consists of undeveloped land, our training data also consists of mostly vacant land. As a result, the model learned how to recognise vacant land pixels with much better accuracy than other classes.

As expected this model performed very poorly, with a Cohen's Kappa value of 0.29 and accuracy of 91%. This high accuracy value was a result of the number of vacant land pixels that the model accurately recognized, with very low accuracy for other classes. Our training and evaluation data, as well as the choice of loss of function, encouraged the model to maximize the overall validation accuracy. And in our case, this could be done by maximizing the accuracy of the majority class which is vacant land.

5.1.2.1. Experiment 2: Baseline model after training on a balanced version the final dataset

To enable the model to classify non-vacant land, we balanced the training and validation data by reducing the number of vacant land pixels so that the model can learn to distinguish the other classes. We used the same technique as described in Chapter 4 to balance the training data, keeping images which do not have vacant land pixels only or if they do, ensuring that these pixels cover at most 70% of the image. Although this method improved results during the ground truth data construction process, it eliminates many of the images consisting of buildings like farm-houses and those representing parks and recreational land. This is because buildings like farm-houses are usually surrounded by farms/vacant land. Thus, over 70% of the pixels in an image of a farm-house will consist of vacant land resulting in the image being discarded from the training data. Since our main goal is to detect and classify

residential neighbourhoods we decided to adhere to this process nevertheless.

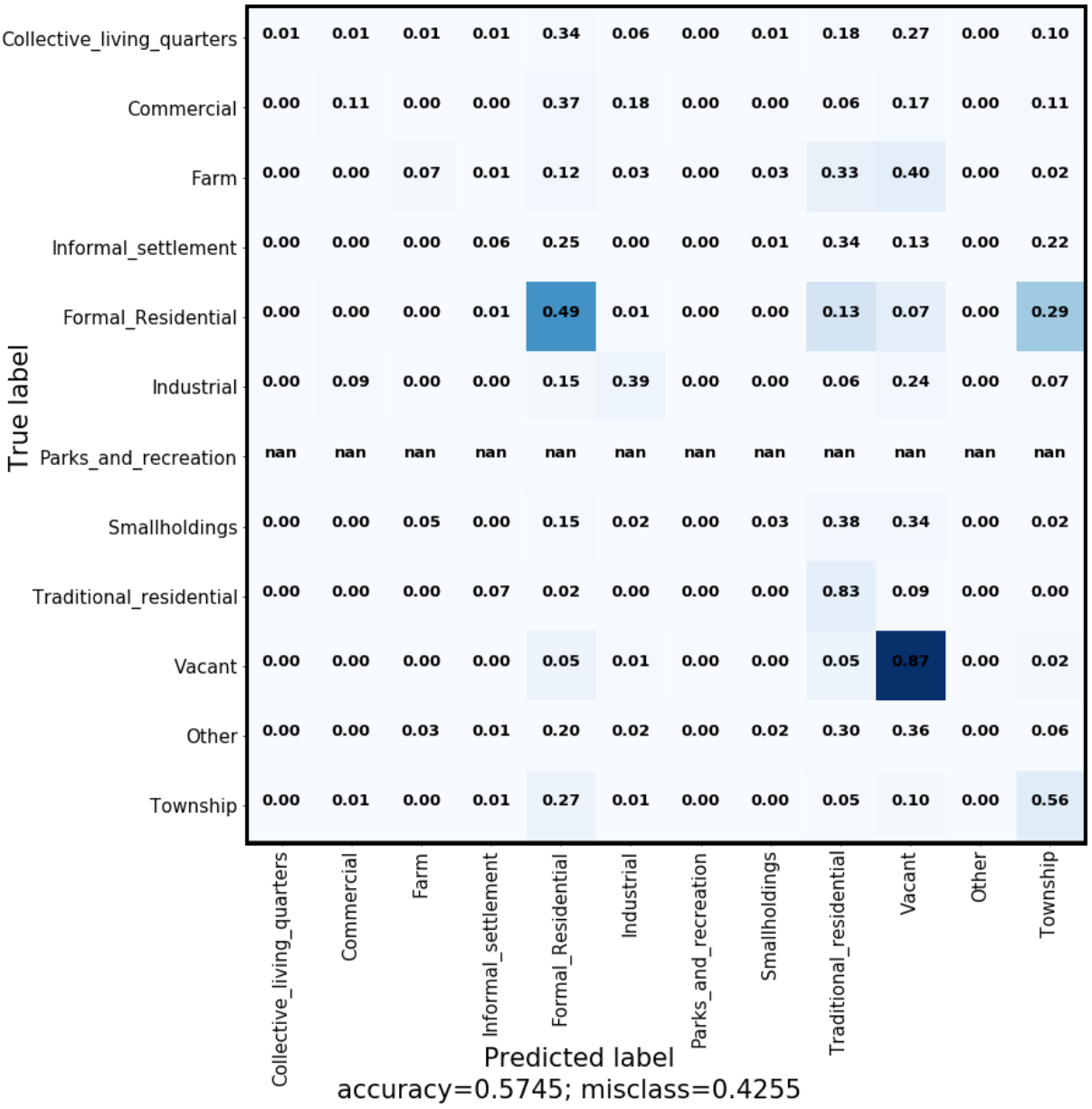


Figure 5.1: Confusion matrix of the U-Net model trained on images from the balanced training set.

.

This model achieved an accuracy of 57.45%, Mean Intersection over Union of 45.85% and an improved Cohen's Kappa value of 0.4326. Pixels from traditional residential areas/villages were classified with an accuracy of 83% which could perhaps mean that the visual representation of villages in the Western Cape and Northern Cape (provinces in our test set) are similar to those in our training and validation

Table 5.2: Confusion matrix of the results of the model trained on the 12-class labels but evaluated on the 4-class labels.

	Non-Residential Neighbourhoods	Wealthy Neighbourhoods	Background	Non-Wealthy Neighbourhoods
Non-Residential Neighbourhoods	**0.38**	**0.26**	**0.2045**	**0.15**
Wealthy Neighbourhoods	**0.015**	**0.335**	**0.23**	**0.415**
Background	**0.015**	**0.105**	**0.5**	**0.2**
Non-Wealthy Neighbourhoods	**0.0226**	**0.225**	**0.15**	**0.61**

sets. The township and suburb classes are predicted at 56% and 49% accuracy respectively. Figure 5.1 shows the confusion matrix. We can see that approximately 33% of townships are confused for suburbs and 10% for vacant land. Similarly, 29% of pixels representing suburbs are misclassified as townships, 13% as traditional residential neighbourhoods and 7% as vacant land.

Figure 5.2: The suburb neighbourhoods enclosed in the blue boundary show how these neighbourhood categories can have varying sub-types.

Segmenting south African neighbourhoods is not an easy task. From visual inspection, we see that

the same neighbourhood class can have different visual characteristics depending on the province. For instance, Figure 5.5 shows images from Western Cape and Northern Cape provinces. The neighbourhood types enclosed in the blue boundaries are all suburbs. However, the suburbs in the two larger boundaries towards the right have different characteristics from those on the left. The suburb clusters on the left look more like townships because of their smaller yards. And the large yards make those on the right look more like small-holdings (the neighbourhood surrounding the suburbs with large yards) or even villages. We have seen this occur in other classes too: some townships in other provinces have even smaller yards and may look like informal settlements. Or they may have larger yards which make them look like suburbs from afar. This variance in the visual characteristics of neighbourhoods across the country shows that careful consideration must be given to the construction of the train and test splits of the dataset. Our future work plans to explore visual similarity metrics to inspect which cities have similar visual characteristics.

Figure 5.3 shows example failure cases. The first row, for instance, shows that suburbs with larger yards are confused with traditional residential neighbourhoods. And from row 2 we can see that those with very small yards are confused with townships. Small-holdings are also often misclassified as traditional residential neighbourhoods or as farmland. Smallholdings usually have large houses and even larger yards–both of which are larger than those found in suburbs. Row 3 of Figure 5.3 illustrates some of these failure cases and row 4 shows how traditional residential neighbourhoods typically look on satellite images. The model confused small-holdings pixels with traditional residential pixels 38% of the time and misclassified small-holding pixels as vacant land 34% of the time. This was not surprising as small-holding landowners usually have a lot of open lands used for small scale farming. Since we have no way of accurately quantifying this open land consistently across the country, we labelled the location of the building with a small-holdings mask and the surrounding area as vacant land. Thus, this confusion is one that we anticipated.

5.1.3. Experiment 3: Leave one province out test experiment

Finally, we perform an experiment to better understand our dataset and the difference between the visual characteristics of the 9 provinces in South Africa. In order to do this, we train and validate on 8 provinces, and test on the 9^{th}. Between the 8 provinces, we randomly choose 6 of them as training data and 2 for validation. We perform this experiment in all 9 provinces, using the other 8 for training

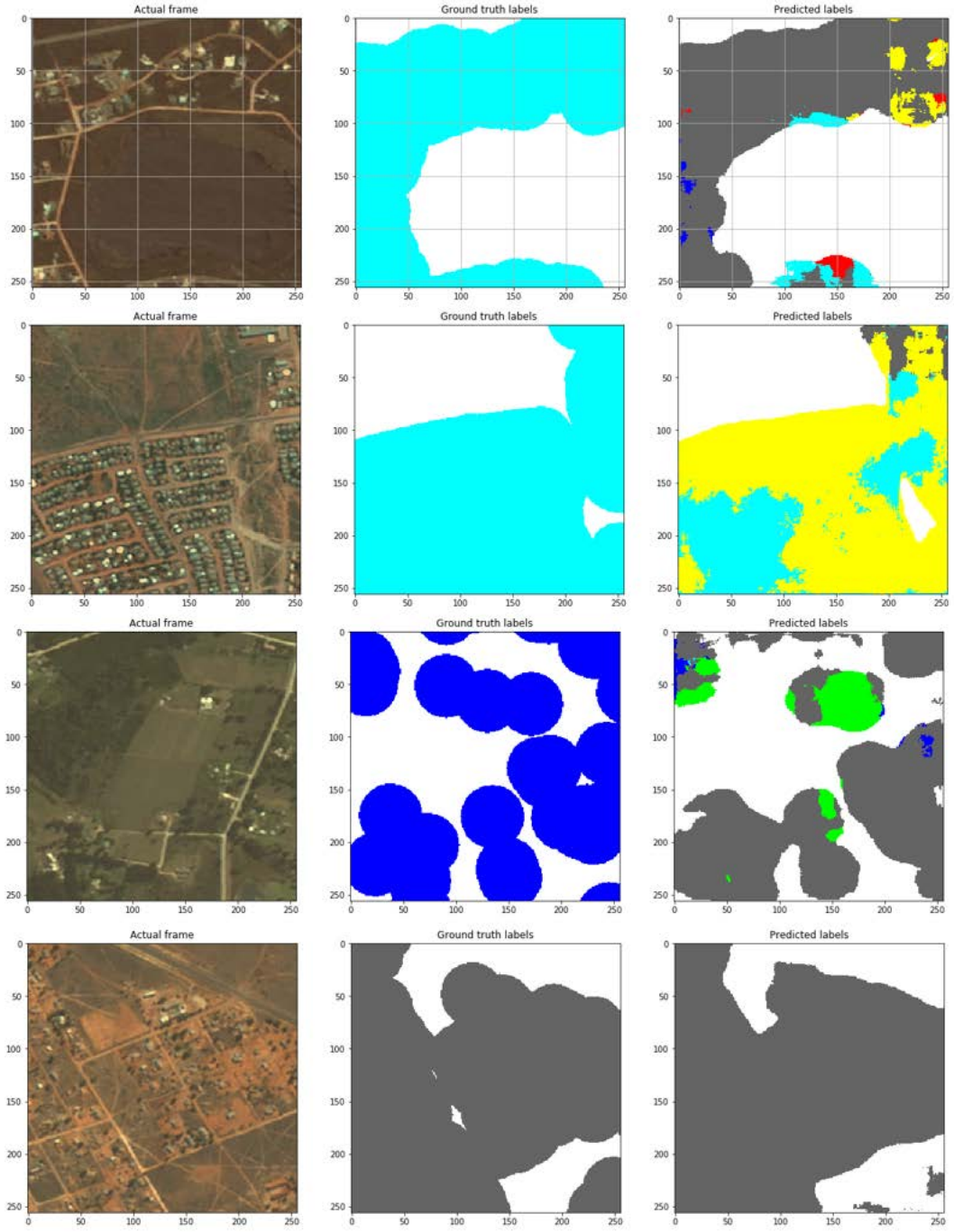


Figure 5.3: Some of the common failure cases. Yellow: Township, Light blue: Suburb, Pink: Industrial area, Olive green: Commercial land, Red: Informal area, Light green: Farm, Light grey: Collective living quarters, Dark grey: Village, Blue: Smallholdings, White: Background.

and validation. As specified in Section 4.5.2 our final dataset was composed by sampling images which cover different sceneries such as densely/sparsely populated areas, mainland/coastal land and different

ecosystems like forests/grassland and across different provinces.

For this experiment, we used the balanced version of the dataset described in 5.1.2 as training data, and show results on both balanced and unbalanced versions of the data in each province. In all experiments, we ensure that there are no overlapping images between the provinces. Table 5.3 shows the exact number of images per province and the corresponding accuracy while testing on that province and training on the rest. The test data is balanced using the same technique described in Chapter 4, only keeping images which do not have vacant land pixels or if they do, ensuring that these pixels cover at most 70% of the image. Balancing the classes meant that we lost several images with pixels representing classes such as parks, recreational areas, farms and smallholding areas because buildings from these classes are usually sparsely populated per image.

Looking at results on the balanced dataset, provinces with relatively high test accuracy such as Mpumalanga and Free State are located in the inland part of the country. Figure 5.4 shows that the model tested on Mpumalanga and trained using images from other provinces performs relatively well. It is particularly good at distinguishing between suburbs, villages/traditional residential and background pixels. Much of the model's confusion came from misclassifying townships, informal settlements and farms as traditional residential neighbourhoods. This could be because residential plot sizes have larger sizes in Mpumalanga than those in provinces containing more populous cities.

We can also see from Table 5.3 that the accuracy for Gauteng is much lower than the other provinces. Recall that throughout the dataset creation process we performed many experiments training and testing on the Gauteng province. Our experiments yielded high accuracy values and corresponding Cohen's Kappa values (see Chapter 4). The low testing performance on Gauteng from a model trained on all other provinces could be because neighbourhoods in the Gauteng Province look very different from other provinces in South Africa. Spatially, Gauteng is the smallest province in South Africa but it is densely populated because of the high economic activity in the province. The visual characteristics of Gauteng could be so distinct that a model only trained on images from provinces outside of Gauteng does not generalise to it.

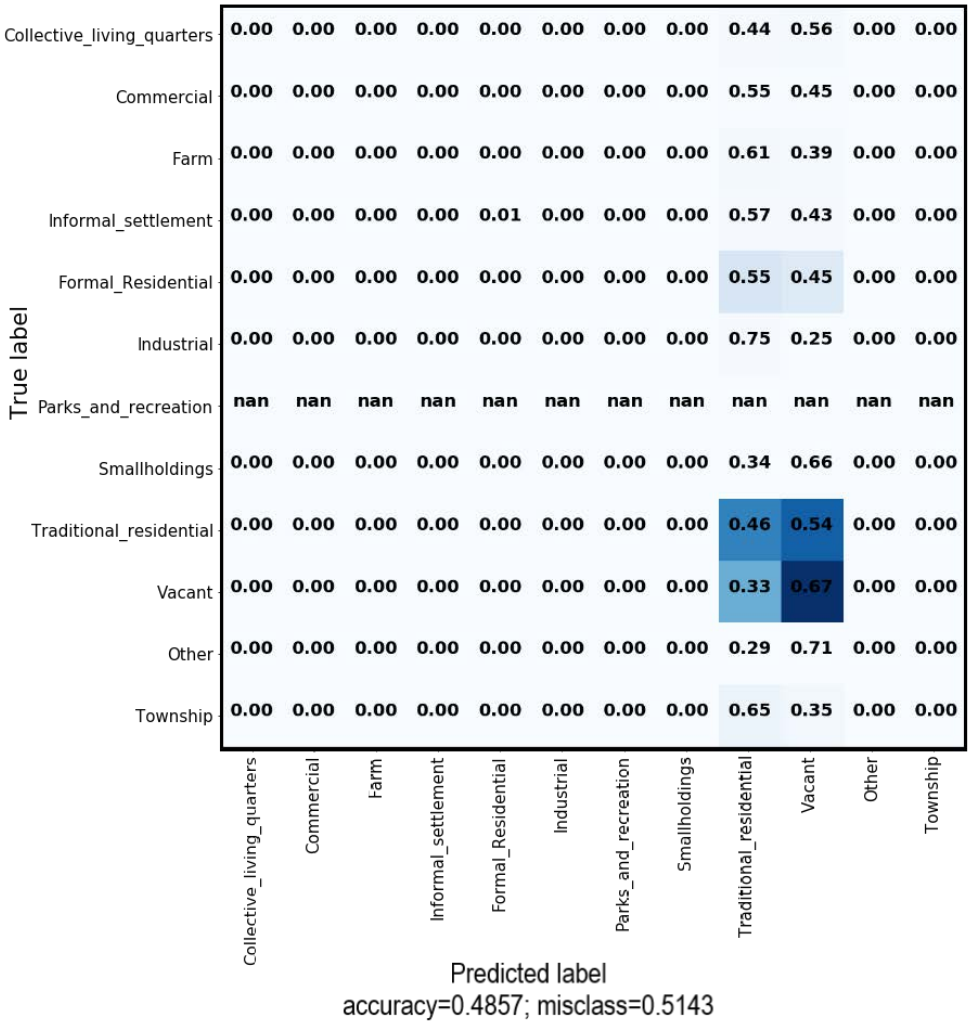


Figure 5.4: The confusion matrix from the model trained on sampled images from the 8 Provinces of South Africa and tested on Mpumalanga province.

5.2. FUTURE WORK

Our baseline experiments have shown us that provinces in South Africa have differing visual characteristics. Thus, we have to perform further experiments to create a train/test split that enables better model generalization.

Moving forward, we plan to refine our segmentation model to improve classification accuracy. Given the

Table 5.3: Table showing the exact number of images used per province and their corresponding accuracy values while testing on that province and training on the rest of the 8 provinces, on both the balanced and unbalanced versions of the test set.

Province	**Number of images**	**Overall Accuracy -without a balanced test set %**	**Number of images -Balanced test set**	**Accuracy -with a balanced test set**
Gauteng	49 392	91.02	8 221	11.08
Mpumalanga	49 392	92.82	3 920	63.49
Limpopo	84 672	93.04	6 601	8.69
North West	91 728	91.09	8 894	28.8
Free State	35 280	93.45	2 367	67.48
KwaZulu-Natal	91 728	89.18	12 841	45.59
Eastern Cape	141 120	92.64	13 192	48.57
Western Cape	91 728	95.54	3 997	31.17
Northern Cape	63 504	99.01	597	35.08

variance in the visual characteristics of neighbourhoods throughout the country, we would like to perform experiments on different regions to have a better understanding of model performance on this dataset. Our test set consisted of The Western Cape and Northern Cape provinces whose visual characteristics may be outliers from those in the rest of the country. Training from scratch and testing on different parts of the country could give a better indication of how the visual characteristics of different neighbourhoods vary by province.

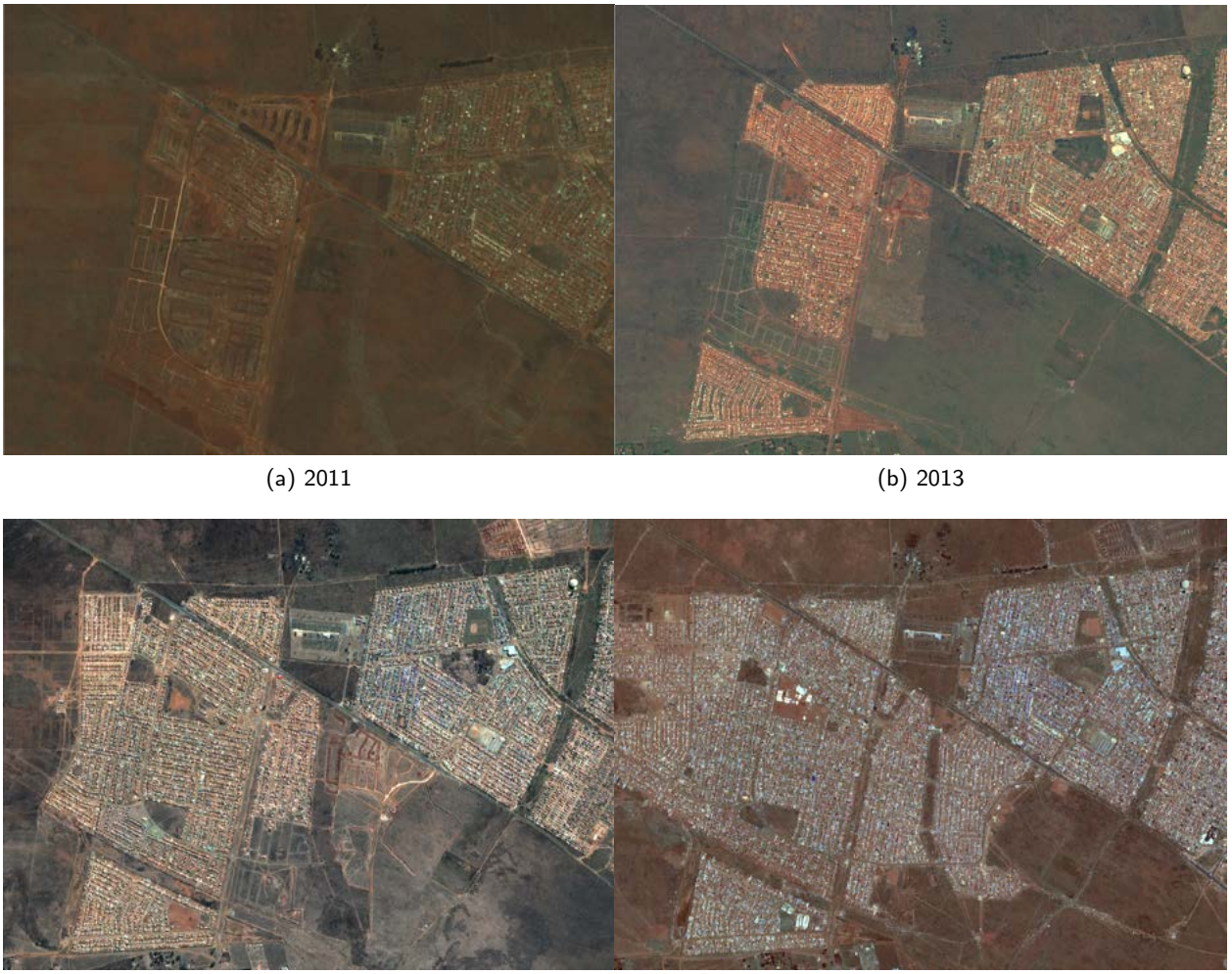
Our goal is to apply our segmentation model across multiple years to understand how neighbourhoods are changing and perform this analysis across the entire country of South Africa. Figures 5.5 and 5.6 show example cases of this where Figure 5.5 shows time-lapse images of a wealthy neighbourhood next to Johannesburg from our satellite image repository. The neighbourhood circled in red in Figure 5.5b looks like a smallholding neighbourhood type because of the sparsity of buildings and large plot sizes. However, this neighbourhood seems to change into what appears to be a suburb over time with a higher density of houses and smaller plot sizes. Figure 5.6 shows the growth of a township through time, we consistently see the small plots of land and uniform visual characteristics throughout the period. We hope to discover these types of phenomena using our dataset and segmentation model throughout the country. Coupling our analysis with census data could give further information on how the demographic makeup of the neighbourhoods has changed, and working with policymakers could help us understand

which policies help desegregate neighbourhoods.

(a) 2011 (b) 2013

(c) 2015 (d) 2017

Figure 5.5: These are time-lapsed images from our satellite image repository of a wealthy neighbourhood next to Johannesburg. The structure at the top-right of the image developing over time is the biggest mall in Africa (Mall of Africa) and around it developing is a wealthy neighbourhood. The neighbourhood circled in red in Figure 5.5b looks like a smallholding neighbourhood type because of the sparsity of buildings and bigger plot sizes, however, this neighbourhood seems to change into what appears to be a suburb over time with a higher density of houses and smaller plot sizes.



(a) 2011 (b) 2013

(c) 2015 (d) 2017

Figure 5.6: These are time-lapsed images from our satellite image repository of part of the Soweto Township in Johannesburg as it grows through time.

CHAPTER 6

CONCLUSION

6.1. CONCLUSION

Although satellite images are readily available in most parts of the world, ground truth datasets labelling the objects of interest, in our case neighbourhoods, are not. This is especially true for countries in the developing world. Our work takes the first steps required to use satellite imagery to analyse the evolution and effects of spatial apartheid in South Africa, by assembling an appropriate training and evaluation dataset and creating a model as a proof of concept for segmenting residential and non-residential areas in South Africa. The dataset consists of satellite images and corresponding ground truth images mapping 12 different neighbourhood types in South Africa. These classes are: Collective living quarters, Commercial, Farm, Informal settlement, Formal Residential, Industrial, Parks and recreation, Smallholdings, Traditional residential, Vacant, Township and Other. The first 10 classes were defined by Stats SA in their 2011 survey. But the Township class was not defined as an independent neighbourhood type–it was instead combined with suburbs and defined as a formal residential area. As described in Chapter 4, we separated these classes and labelled all the townships in the country. This was an important step in our analysis because townships were crucial to Spatial Apartheid as this was where the government placed non-European people. We performed an iterative process to assemble our dataset as described in Chapter 4 and trained and evaluated a U-Net segmentation model to classify image pixels according to the 12 classes. This was done with an accuracy of 57.45% and a Cohen's Kappa value of 0.4326.

While we did not anticipate spending 1.5 years assembling a dataset, this process has shown us the importance of this step and the complexity involved in it. Recently, machine learning researchers have called for the community to teach data annotation and collection practices as a speciality, and valuable contributions in this area as much as model related innovations [Lawrence, 2017; Jo and Gebru, 2020]. This lack of incentives to create new datasets and reliance on building models suited for existing benchmarks has resulted in a limited understanding of which parts of the machine learning pipeline require the most amount of time and resources [Lawrence, 2017]. It has also resulted in a lack of innovation in the design and procedure of dataset annotation and collection practices [Jo and Gebru, 2020; Lawrence, 2017]. Our dataset creation process took much more time than expected and presented us with many

unexpected challenges which required us to create an iterative dataset construction methodology using a machine learning model to tune the dataset.

We hope that, as one of the few neighbourhood segmentation models focused on the developing world, and the only one covering South Africa, this dataset will help enable interdisciplinary research in many areas. We plan to use this dataset to study the evolution of neighbourhoods across South Africa over time and hope to partner with policy researchers to perform further investigations using our dataset.

REFERENCES

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian conference on computer vision*, pages 180–196. Springer, 2016.

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

Ruha Benjamin. *Race After Technology: Abolitionist Tools for the New Jim Code*. John Wiley & Sons, 2019.

Nigel Worden. *The making of modern South Africa: Conquest, segregation and apartheid*. Blackwell Oxford, 1994.

Anthony J Christopher. Urban segregation in post-apartheid south africa. *Urban studies*, 38(3):449–466, 2001.

Michael Noble and Gemma Wright. Using indicators of multiple deprivation to demonstrate the spatial legacy of apartheid in south africa. *Social Indicators Research*, 112(1):187–201, 2013.

Johnny Miller. Apartheid's urban legacy, in striking aerial photographs. `https://bit.ly/3bB3fbo`, 2016 accessed January 7, 2020.

Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raska. Deepglobe 2018: A challenge to parse the earth through satellite images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 172–17209. IEEE, 2018.

Serena Yeung Fei-Fei Li, Justin Johnson. Lecture 11: Detection and segmentation. `https://stanford.io/2yn488i`, 2017 accessed April 19, 2020.

Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):657–673, 2002.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. *arXiv preprint arXiv:1712.02616*, 2017.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20:5, 2015.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

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

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

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

Kaggle. Land Cover Classification Kaggle Challenge. `https://www.kaggle.com/c/data-science-bowl-2018/discussion/54741`, 2019.

Vladimir Iglovikov, Sergey Mushinskiy, and Vladimir Osin. Satellite imagery feature detection using deep convolutional neural network: A kaggle competition. abs/1706.06169, 2017.

Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

Joshua Blumenstock, Gabriel Cadamuro, and Robert On. Predicting poverty and wealth from mobile phone metadata. *Science*, 350(6264):1073–1076, 2015.

Vongani Hlavutelo Maluleke, Sebnem Er, and Quentin R Williams. Estimating poverty using aerial images: South african application. *Data Science and Applications*, 1(1):29–36, 2018.

Naledzani Mudau, Willard Mapurisa, Thomas Tsoeleng, and Morwapula Mashalane. Towards development of a national human settlement layer using high resolution imagery: a contribution to sdg reporting. *South African Journal of Geomatics*, 9(1):1–12, 2020.

John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

N Mudau. Spot building count supporting informed decisions. *Position IT*, 10:51–52, 2010. URL `https://www.ee.co.za/wp-content/uploads/legacy/PositionIT2009/PositionIT2010/SPOT.pdf`.

Peter Hofmann et al. Detecting informal settlements from ikonos image data using methods of object oriented image analysis-an example from cape town (south africa). *Jürgens, C.(Ed.): Remote Sensing of Urban Areas/Fernerkundung in urbanen Räumen*, pages 41–42, 2001.

Guy Blanchard Ikokou and Julian Smit. Investigating the potential of common earth observation satellite imagery for automated multi-criteria mapping of urban landscape at municipal level in south africa. In *Earth Observations and Geospatial Science in Service of Sustainable Development Goals*, pages 171–179. Springer, 2019.

L Ngcofe, Thabisile Rambau, M McCalachan, Fezekile Hantibi, and Nale Mudau. Application of semi-automated settlement detection for an integrated topographic map information system update in south africa. *South African Journal of Geomatics*, 6(3):308–320, 2017.

P Hurskainen and P Pellikka. Change detection of informal settlements using multi-temporal aerial photographs–the case of voi, se-kenya. In *Proceedings of the 5th African Association of Remote Sensing of the Environment conference, Nairobi, Kenya, unpaginated CD-ROM*. Citeseer, 2004.

Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585. IEEE, 1994.

Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, 19(6):1657–1663, 2010.

S Karakış, AM Marangoz, and G Büyüksalih. Analysis of segmentation parameters in ecognition software using high resolution quickbird ms imagery. In *ISPRS Workshop on Topographic Mapping from Space*, pages 14–16, 2006.

Nicholus Mboga, Claudio Persello, John Ray Bergado, and Alfred Stein. Detection of informal settlements from vhr images using convolutional neural networks. *Remote sensing*, 9(11):1106, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Frans van den Bergh. The effects of viewing-and illumination geometry on settlement type classification of quickbird images. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, pages 1425–1428. IEEE, 2011.

L Mdakane and F Van den Bergh. Extended local binary pattern features for improving settlement type classification of quickbird images. 2012.

Lizwe Mdakane. *Settlement type classification using aerial images.* PhD thesis, 2014.

LP Abeigne Ella, Frans van den Bergh, Barend J van Wyk, and Michaël Antonie van Wyk. A comparison of texture feature algorithms for urban settlement classification. In *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages III–1308. IEEE, 2008.

Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

T Kemper, N Mudau, P Mangara, and M Pesaresi. Towards an automated monitoring of human settlements in south africa using high resolution spot satellite imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(7):1389, 2015.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279. ACM, 2010a.

The International Society for Photogrammetry and Remote Sensing. Apartheid's urban legacy, in striking aerial photographs. `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`, 2020 accessed February 12, 2020.

Defence Science and Technology Laboratory. Dstl satellite imagery feature detection. `https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/data`, 2017 accessed February 12, 2020.

Monika Kuffer, Karin Pfeffer, and Richard Sliuzas. Slums from space—15 years of slum mapping using remote sensing. *Remote Sensing*, 8(6):455, 2016.

Claudio Persello and Alfred Stein. Deep fully convolutional networks for the detection of informal settlements in vhr images. *IEEE geoscience and remote sensing letters*, 14(12):2325–2329, 2017.

Rizwan Ahmed Ansari, Rakesh Malhotra, and Krishna Mohan Buddhiraju. Identifying informal settlements using contourlet assisted deep learning. *Sensors*, 20(9):2733, 2020.

Thomas Stark. Using deep convolutional neural networks for the identification of informal settlements to improve a sustainable development in urban environments. *Technische Universität München*, 2018.

Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peerapon Vateekul. Semantic segmentation on remotely sensed images using an enhanced global convolutional network with channel attention and domain specific transfer learning. *Remote Sensing*, 11(1):83, 2019.

Francois PS Luus, Frans Van den Bergh, and Bodhaswar TJ Maharaj. The effects of segmentation-based shadow removal on across-date settlement type classification of panchromatic quickbird images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3):1274–1285, 2013.

Bodhiswatta Chatterjee and Charalambos Poullis. Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks. *arXiv preprint arXiv:1912.09216*, 2019.

Karishma Busgeeth, André Brits, and JB Whisken. Potential application of remote sensing in monitoring informal settlements in developing countries where complimentary data does not exist. 2008.

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 Conference on Computer Vision and Pattern Recognition*, pages 12726–12735, 2019.

Collin Homer, Jon Dewitz, Limin Yang, Suming Jin, Patrick Danielson, George Xian, John Coulston, Nathaniel Herold, James Wickham, and Kevin Megown. Completion of the 2011 national land cover database for the conterminous united states–representing a decade of land cover change information. *Photogrammetric Engineering & Remote Sensing*, 81(5):345–354, 2015.

Warren C Jochem, Tomas J Bird, and Andrew J Tatem. Identifying residential neighbourhood types from settlement points in a machine learning approach. *Computers, environment and urban systems*, 69:104–113, 2018.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

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.

Yan Liu, Qirui Ren, Jiahui Geng, Meng Ding, and Jiangyun Li. Efficient patch-wise semantic segmentation for large-scale remote sensing images. *Sensors*, 18(10):3232, 2018.

Lichen Zhou, Chuang Zhang, and Ming Wu. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *CVPR Workshops*, pages 182–186, 2018.

Jing Zhang, Shaofu Lin, Lei Ding, and Lorenzo Bruzzone. Multi-scale context aggregation for semantic segmentation of remote sensing images. *Remote Sensing*, 12(4):701, 2020.

Chao Tian, Cong Li, and Jianping Shi. Dense fusion classmate network for land cover classification. In *CVPR Workshops*, pages 192–196, 2018.

SANSA. South African National Space Agency. https://www.sansa.org.za/, 2019.

Statistics SA. Statistics South Africa. http://www.statssa.gov.za/?page_id=3917, 2019.

QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009. URL http://qgis.osgeo.org.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010b.

Patrick Helber, Benjamin Bischke, Jörn Hees, and Andreas Dengel. Towards a sentinel-2 based human settlement layer. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5936–5939. IEEE, 2019.

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

Kevin Koidl. Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*, 2013.

Tarald O Kvålseth. Note on cohen's kappa. *Psychological reports*, 65(1):223–226, 1989.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

Neil D Lawrence. Data readiness levels. *arXiv preprint arXiv:1705.02245*, 2017.

Eun Seo Jo and Timnit Gebru. Lessons from archives: Strategies for collecting sociocultural data in machine learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 306–316, 2020.