ENCODING REMOTELY SENSED TIME SERIES DATA AS TWO-DIMENSIONAL IMAGES FOR URBAN CHANGE DETECTION USING CONVOLUTION NEURAL NETWORKS

MARC DANIEL DUKES

Thesis presented in fulfilment of the requirements for the degree Master of Science in Geoinformatics at Stellenbosch University

SUPERVISOR: DR Z MUNCH
Department of Geography & Environmental Studies
Stellenbosch University

CO-SUPERVISOR: DR TL GROBLER
Department of Computer Science
Stellenbosch University

April 2022

DECLARATION

By submitting this report electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2022

SUMMARY

Urban expansion is the most pervasive form of land cover change in South Africa. A method that can effectively detect and indicate areas that have a higher probability of displaying urban change will therefore be a valuable asset to analysts. That is why it is critical to derive a rapid framework that can accurately map urban change. An alternative remote sensing approach that uses multi-temporal time series data and deep learning techniques has been proposed as a potential method for performing a successful urban change detection. The interdisciplinary scientific field of computer vision holds a framework for encoding time-series data as two-dimensional (2D) images for input to a convolution neural network (CNN).

Traditional image classifications techniques and more recent studies that have deployed machine learning and deep learning classifiers (namely support vector machine (SVM), random forest (RF), *k*-nearest neighbour (*k*NN), long short-term memory (LSTM) and CNN) have been used for urban land cover classification. In this study, a unique framework proposed within computer vision that exploits Gramian angular fields (GAF) and Markov transition fields (MTF) as the transformations for encoding time series data as 2D imagery prior to deep learning classification is investigated for urban change detection.

Two main experiments were carried out, both of which utilised the proposed framework for performing an effective urban change detection. The first experiment used coarse resolution data derived from Pretoria using MODIS 500m and 250m normalised difference vegetation index (NDVI). The proposed framework was then deployed, and Gramian angular summation field (GASF), Gramian angular difference field (GADF), and MTF transformations used to encode the time series data. A concatenated encoded image containing the information from all three transformations was formed and was run alongside the three individual transformations. Multiple pre-trained CNN architectures (namely ResNet, DenseNet, InceptionV3, InceptionResNetV2, VGG and MobileNet) were used, from which an urban change detection was derived. It was established that the concatenated images yielded the highest accuracy at 91% and 93% for the 500m and 250m resolution datasets, respectively. The proposed framework was compared to a current state-of-the-art time series classifier (LSTM) to illustrate the effectiveness of encoding and processing deep learning classifiers. The results also outperformed that of other urban change detections studies conducted in South Africa.

The second experiment made use of higher resolution Sentinel-2 data derived from a resampled 30m resolution NDVI product of Pretoria. Several investigations were made into the influencing elements that affect the performance of the urban change detection. These were the spatial and temporal resolutions, training data size and different classification schemes. Using the proposed

framework from the first experiment, the spatial and temporal resolutions were tested. The results showed that an increase in spatial or temporal resolution will have a positive effect on the performance. The 30m resolution dataset yielded a 4% increase over the 250m resolution data tested in the first experiment. Altering the time-series length (TSL) from 32 to 82, the accuracy increased from 96% to 98%, respectively. It was also illustrated that by increasing the amount of training data, one could improve the performance of the change detection. Multiple classifications were performed, and the accuracy assessed using a confusion matrix. It was established that a 70%+ minimum pixel probability and the majority ensemble classifier performed the best. The frameworks generalisability was tested at three different locations (Durban, Gqeberha, and Khayelitsha), and was able to generalise using the Durban dataset. However, the models were unable to generalise using the Gqeberha, and Khayelitsha datasets due to the diverse ecological and climatic properties.

The experiments showed that deploying a computer vision framework of encoding multi-temporal time series data as two-dimensional images for an urban change detection using CNN classifications is, in fact possible, and proved to be one of the most effective urban change detection methods performed in South Africa. However, it is recommended that further research deploys a signature extension approach for training the models in order to improve the generalisability. Additional research into using Landsat8 and increased TSL datasets is also recommended.

KEY WORDS

Remote sensing; urban change detection; time series classification; deep learning; convolution neural networks; computer vision; encoding time-series; Gramian fields; Markov fields:

OPSOMMING

Stedelike uitbreiding is die heersende vorm van grondbedekkingsverandering in Suid-Afrika. 'n Metode om gebiede met 'n groter waarskynlikheid van stedelike veranderinge te toon of effektief te kan kan opspoor en aandui, sal 'n waardevolle bate vir ontleders wees. Daarom is dit van kritieke belang om 'n minder tydrowende raamwerk op te stel wat stedelike verandering akkuraat kan karteer. 'n Alternatiewe afstandswaarnemingsbenadering wat multi-temporale tydreeksdata en diepleertegnieke gebruik, word voorgestel as 'n moontlike metode vir suksesvolle opsporing van stedelike veranderinge. Die interdissiplinêre wetenskaplike veld van rekenaarvisie bevat 'n raamwerk vir die kodering van tydreeksdata as tweedimensionele beelde wat as invoer dien vir 'n konvolusionele neurale netwerk (CNN).

Tradisionele beeldklassifikasietegnieke en meer onlangse studies wat masjienleer- en diepleerklassifiseerders (naamlik ondersteuningsvektormasjien (SVM), ewekansige woud (RF), k-naaste buurtklassifiseerder (kNN), lang-kort-termyn-geheue (LSTM) en CNN) word dikwels gebruik vir klassifikasie van stedelike grondbedekkings. In hierdie studie word 'n unieke raamwerk voorgestel wat binne rekenaarvisie ontwikkel is wat Gramian-hoekvelde (GAF) en Markov-oorgangsvelde (MTF) benut as ‘n transformasie in die kodering van tydreeksdata as tweedimensionele beelde voordat diepleerklassifikasie ondersoek word vir die opsporing van stedelike veranderinge .

Twee eksperimente is uitgevoer, wat beide die voorgestelde raamwerk gebruik het vir opsporing van stedelike veranderinge. Die eerste eksperiment het gegewens gebruik van growwe resolusie wat uit Pretoria verkry is, met behulp van MODIS 500m en 250m genormaliseerde verskil plantegroei-indeks (NDVI) data. Die voorgestelde raamwerk is daarna ontplooi deur Gramian hoeksomvelde (GASF), Gramian hoekverskilvelde (GADF) en MTF transformasies te gebruik om die tydreeksdata te kodeer. 'n Saamgevoegde gekodeerde beeld wat al drie transformasies bevat, is gemaak en saam met die drie individuele transformasies analiseer. Veelvuldige vooraf-opgeleide CNN-argitekture (naamlik ResNet, DenseNet, InceptionV3, InceptionResNetV2, VGG en MobileNet) is gebruik, waaruit die stedelike verandering afgelei is. Daar is vasgestel dat die saamgevoegde beelde die hoogste akkuraatheid gelewer het met 91% en 93% vir die datastelle van onderskeidelik 500m en 250m. Die voorgestelde raamwerk is vergelyk met 'n huidige moderne tydreeksklassifiseerder (LSTM) om die doeltreffendheid van kodering en verwerking van 'n diepleerklassifiseerder te illustreer. Die resultate was ook beter as dié van ander stedelike veranderingstudies in Suid-Afrika.

Die tweede eksperiment het gebruik gemaak van Sentinel-2-data met 'n hoër resolusie, ook afgelei van 'n NDVI-produk vir Pretoria, verwerk na 30m. Verskeie ondersoeke is gedoen om vas te stel

wat die faktore is wat die akkuraatheid van die opsporing van stedelike verandering beïnvloed, byvoorbeeld, die ruimtelike en temporale resolusies, die grootte van die opleidingsdata en verskillende klassifikasie skemas. Met behulp van die voorgestelde raamwerk van die eerste eksperiment, is die effek van ruimtelike en temporale resolusies getoets. Die resultate het getoon dat 'n toename in ruimtelike of temporale resolusie 'n positiewe uitwerking op die akkuraatheid sal hê. Die datastel met 'n resolusie van 30m het 'n toename van 4% opgelewer in vergelyking met die resolusiedata van 250m wat in die eerste eksperiment getoets is. Deur die tydreekslengte (TSL) van 32 na 82 te verander, het die akkuraatheid toegeneem van 96% tot 98%. Die studie het ook aangedui dat die akkuraatheid van veranderingopsporing sou verbeter kon word deur die hoeveelheid opleidingsdata te vermeerder. Veelvuldige klassifikasie skemas is uitgevoer en die akkuraatheid met behulp van 'n verwarringsmatriks getoets. Daar is vasgestel dat 'n 70%+ minimum pixelwaarskynlikheid en die meerderheidsensemble-klassifiseerder die beste gevaar het. Die veralgemeenbaarheid van die raamwerke is op drie verskillende plekke (Durban, Gqeberha en Khayelitsha) getoets, maar kon slegs in Durban veralgemeen word. Die modelle kon nie stedelike verandering met Gqeberha- en Khayelitsha -datastelle optel nie weens die uiteenlopende ekologiese en klimaatseienskappe.

Die eksperimente het getoon dat die implementering van 'n rekenaarvisie raamwerk vir die kodering van multi-temporale tydreeksdata as tweedimensionele beelde vir die opsporing van stedelike veranderinge met behulp van CNN-klassifikasies in werklikheid moontlik is en een van die mees doeltreffende opsporingstegnieke vir stedelike veranderinge in Suid-Afrika kan wees. Dit word egter aanbeveel dat verdere navorsing 'n uitbreidingsbenadering gebruik vir die opleidingsdata vir die modelle om die veralgemenbaarheid te verbeter. Bykomende navorsing oor die gebruik van Landsat8 en verhoogde TSL-datastelle word ook aanbeveel.

SLEUTELWOORDE

Afstandswaarneming; stedelike veranderingsopsporing; tydreeks klassifikasie; diepleer; konvolusionele neurale netwerke; rekenaarvisie; kodering van tydreekse; Gramian velde; Markov velde

ACKNOWLEDGEMENTS

I would like to sincerely thank:

- My supervisor, Dr Zahn Munch, for her continual guidance, excellent suggestions and encouragement thought out the duration of my research, as well as some financial aid for editing;
- My parents, Elke Dukes and Trevor Dukes who provided me with the opportunity to study and pursue my passion, and their constant love and support;
- Dr Trienko L Grobler for the assistance in developing the framework, the outstanding suggestions in the initial stages of my research and providing some financial aid;
- My partner, Jodi Cahi, for the endless love and support she provided, as well as the constant reassurance and motivation during the darker days;
- Aré van Schalkwyk for doing an exceptional job editing the thesis.
- All the staff members from the Department of Geography and Environmental Studies who gave constructive criticism, useful feedback and helpful comments during the report sessions;
- The Postgraduate Office for the partial scholarship in the year of 2020; and
- My fellow varsity colleagues who helped keep me on track and make the long hours in the lab somewhat bearable.

TABLE OF CONTENTS

TABLES

FIGURES

ACRONYMS AND ABBREVIATIONS

ACF	Autocorrelation function
AI	Artificial intelligence
ANN	Artificial neural networks
API	Application programming interface
ARVI	Atmospherically resistant vegetation index
AUC	Area under the receiver operating characteristic curve
CART	Classification and regression trees
CCA	Canonical correlation analysis
CDA	Change detection accuracy
CNN	Convolutional neural network
CP	Change pixels
CPU	Central processing unit
CV	Computer vision
DenseNet	Densely connected convolutional networks
DL	Deep learning
DNN	Deep neural network
DT	Decision tree
DVI	Difference vegetation Index
EMR	Electromagnetic radiation
EO	Earth observation
ES	Electromagnetic spectrum
FAR	False alarm rate
GAN	Generative adversarial networks
GADF	Gramian angular difference field
GAF	Gramian angular field

GASF	Gramian angular summation field
GEE	Google earth engine
GIS	Geographical information systems
GPU	Graphics processing unit
GTS	Ground truth samples
k-NN	*k*-nearest neighbours
KZN	KwaZulu-Natal
LDA	Linear discriminant analysis
LSTM	Long short-term memory
MaxL	Maximum likelihood
ML	Machine learning
MLP	Multilayer perceptron
MODIS	Moderate resolution Imaging spectroradiometer
MPP	Minimum pixel probability
MTF	Markov transition field
NDBI	Normalised difference built-up Index
NDVI	Normalised difference vegetation index
NIR	Near-infrared
NN	Neural network
OA	Overall accuracy
PA	Producer’s accuracy
Pan	Panchromatic
PCA	Principle component analysis
PVI	Perpendicular vegetation index
RAM	Random-access memory
RBF	Radial basis function networks
ResNet	Residual network

RF	Random forest
RFE	Recursive feature elimination
RGB	Red green blue
RNDSI	Ratio normalised difference soil index
RNN	Recurrent neural network
RS	Remote sensing
RVI	Ratio vegetation index
SAR	Synthetic aperture radar
SAVI	Soil-adjusted vegetation index
SLRA	Soil line atmospheric resistance index
SRI	Simple ratio index
STACD	Spatio-temporal ACF change detection
SVM	Support vector machine
TACD	Temporal autocorrelation change detection
TSARVI	Type soil atmospheric impedance vegetation index
TSL	Time-series length
UA	User’s accuracy
UAV	Unmanned aerial vehicles
USGS	United States geological survey
VGG	Visual geometry group
1D	One-dimensional
2D	Two-dimensional

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Informal settlements are growing at alarming rates as people move closer to cities for potential employment opportunities due to economic or environmental factors. The complex socio-economic process of urbanisation has altered the global distribution of population in urban and rural areas (UN 2018). Urbanisation has increased at a rapid rate over the past 50 years. Between 1950 and 2018, movement from rural to urban areas has risen by 25% and a predicted further 13% increase reaching a global urban total population of 68% by 2050 (UN 2018). Over 90% of the economic growth and activity occur in urban areas (Li, Gong & Liang 2015). It is then understandable that rural-urban migration is widely spread (Lopez, Shimoni & Grippa 2017).

Despite all the economic benefits of urbanisation and the increase in urban activity, significant air pollution, congestion, and food security arise (Zhou, Li & Pan 2018). The change in land cover and land use caused by urbanisation has amplified the heat island effect. It is causing irreversible changes to ecosystems (Sinha, Santra & Mitra 2018), increasing surface temperature and affecting net ecosystem carbon exchange, compounding the effects of climate change. The climate change worsens the current vulnerabilities such as vector-borne diseases (dengue fever and malaria) as well as water-borne diseases (dysentery and cholera) and, in addition, adds to the pressure on the environment (Bryan et al. 2009). South Africa is a developing country and has increased carbon emissions (Bryan et al. 2009). Continuous research into climate change and the carbon cycle in South Africa and urban planning will require timely land cover data (Gong, Li & Zhang 2019; Liu et al. 2018).

In Third World countries like South Africa, human settlement expansion is one of the most pervasive forms of land cover change (Kleynhans et al. 2012; Kleynhans, Salmon & Wessels 2017). The expansion of human settlements is more often unplanned and informal. Settlements categorised as “informal” are frequently expanding and encroaching on land previously covered by natural vegetation (Kleynhans et al. 2013). These newly developed informal settlements occur in random unplanned locations and do not provide essential services such as electricity, refuse removal, water-based sewage or running water (Kleynhans et al. 2015). The informal manner in which these settlements are developed primarily results in unplanned layouts (Palframan 2005). According to the United Nations (UN 2018) study, South Africa needs to be empowered to plan, develop, implement, and maintain human settlements.

Therefore, the ability to detect informal settlements is critical for the detailed mapping of these areas to provide local municipalities and regional governments with the correct data. A regular

update on land cover change is precious for urban planning. This data can accommodate newly expanded areas during the planning and help to get essential services into recently developed informal settlements. Remote sensing and, in particular, a time series of satellite data have proven to be an effective way to monitor and track land cover changes (De Beurs & Henebry 2005; Lu et al. 2004; Verbesselt, Hyndman, Newnham, et al. 2010).

Change detection, using remote sensing (RS) and geographical information systems (GIS), is a well-established method for understanding the alteration of an area over some time. The process of digital change detection assists in determining alterations associated with applications such as land cover change and settlement expansion. In remote sensing, multiple techniques are used to perform change detection (Lu et al. 2004), including the algebra method, transformation, classification, advanced models and visual interpretation (Lu et al. 2004). The post-classification approach for change detection has been used for many years and is an effective method (Tewkesbury et al. 2015). Researchers have used this method to investigate change detection for human settlement expansion around the world. In South Africa, both temporal and spatiotemporal autocorrelation analysis have been used with great success for human settlement detection and change analysis (Kleynhans et al. 2013; Kleynhans et al. 2012; Kleynhans, Salmon & Wessels 2017).

1.2 REMOTE SENSING

Image classification is when pixels are assigned to a specific class using various methods (Campbell & Wynne 2011). Pixel-based classification identifies the pixels as individual units and proceeds with classification using each pixel's spectral value (Campbell & Wynne 2011). In recent findings, machine learning (ML) algorithms have proven more effective than the traditional methods of image classification (Maxwell, Warner & Fang 2018). These algorithms can learn and improve automatically through experience (Maxwell, Warner & Fang 2018). One type of ML focuses on using a supervised learning approach for classification (Michalski, Carbonell & Mitchell 2013; Zhang 2020). The computer program and algorithm learn from the input training data to make new classified observations of unseen testing data (Camps-Valls 2009). A few ML classifiers that are currently being used are support vector machine (SVM), decision tree (DT), random forest (RF), nearest neighbour and neural network (NN) (Michalski, Carbonell & Mitchell 2013; Zhang 2020).

Li, Gong & Liang (2015) successfully built a classification framework to determine annual urban dynamics. They incorporated the normalised difference vegetation index (NDVI) in the classification scheme and applied a spatiotemporal filter to check for consistency. Using the RF as their primary classifier, they achieved high accuracies of 90%, 87%, 85% and 88.5% for the initial

classifications and increased these accuracies after temporal filtering. Zhou, Li & Pan (2018) demonstrated how RF can be used to accurately classify urban land cover in downtown Suzhou, China using multi-sensor data from both Landsat-8 OLI and Hyperion along with Sentinel-1A data. This is supported by Celik (2018), who showed that the RF classification technique could be used for an urban change detection.

In addition to the classification approach, autocorrelation analysis has also proven to be an effective technique used for change detection. Autocorrelation analysis is the degree of correlation of the values with a delayed copy; this is seen as a delay function (Kleynhans, Salmon & Wessels 2017). Using the observation, it is the similarity between the time lag (Kleynhans, Salmon & Wessels 2017). First piloted by Kleynhans et al. (2012) in Gauteng, South Africa, new human settlements were detected from 500 m moderate resolution imaging spectroradiometer (MODIS) time-series data through the use of a temporal autocorrelation function (ACF) and optimised thresholding. By adapting the thresholds based on the change properties of neighbouring pixels, Kleynhans et al. (2013) increased the accuracy of the change detection from 88% to 91% and decreased the false alarm rate (FAR) from 15% to 5%. The performance of this new spatio-temporal ACF change detection (STACD) method was further improved by modifying the change index (Kleynhans et al. 2015) to yield a 17% increase in accuracy and a 1% FAR (Kleynhans et al. 2015). Kleynhans, Salmon and Wessels (2017) developed a novel framework for parameter selection for the STACD method. They compared the results using different sampling frequencies such as daily, eight-daily, monthly, two-monthly, quarterly and semi-annually. They concluded that both the FAR for the daily and two-monthly data sets were nearly identical at 1% and that there was little performance difference between the two datasets (Kleynhans, Salmon & Wessels 2017).

1.3 DEEP LEARNING

Within the field of deep learning (DL), which is a broader part of ML premised on the methods using artificial neural networks (ANN), we find the concept of the convolutional neural network (CNN) (Stoian et al. 2019). Although DL has shown to be effective, it requires large amounts of training data to achieve high accuracy. When using time-series datasets, building a predictive model can prove challenging. Neural networks are challenging to train, which is why pre-trained models exist (Stoian et al. 2019).

Mboga et al. (2017) classified informal settlements using CNN, they built the CNN using a combination of spatial feature learning hyperparameters and training hyperparameters. They built the CNNs with eight convolutional layers and made layers 2, 3, and 4 fully connected. The SVM set the baseline classification at 68.84%, and all the CNNs (1-6) outperformed the SVM with

accuracies ranging from 86.32% to 91.53%. They proved that DL algorithms such as CNN can be used for practical classifications of informal human settlements.

Stoian et al. (2019) investigated the replacement of RF classifiers with fully CNN architectures in an operational context to identify which techniques are the most effective for operational purposes. They concluded that their model FG-Unet yielded improved results than pixel-based RF classifiers. However, this approach shows high variability in quality over the different landscapes. Hatami, Gavet & Debayle (2018) showed that CNN was successfully used for image recognition using a time-series dataset. They concluded that CNNs have a high performance on image classification time-series. Another comparison between RF and CNN was conducted to classify satellite images (Pelletier, Webb & Petitjean 2019). It was concluded that CNN could create a higher quality classification map than RF base methods and outperformed the RF accuracy score by 2%-5% (Pelletier, Webb & Petitjean 2019).

The success of these studies show that DL classifications (CNN) are becoming the next generation of state-of-the-art classification approach (Chen et al. 2019; Hatami, Gavet & Debayle 2018; Mboga et al. 2017; Pelletier, Webb & Petitjean 2019; Stoian et al. 2019). Whereas ML is still effective, DL is more accurate when trained with large volumes of data. The advantage of using a DL classifier is that the feature engineering process aspect is simplified. The process of using domain knowledge to gather features from the raw data is simplified.

1.4 COMPUTER VISION

Research has shown the success of encoding time-series data as images for classifications through the use of CNN. Wang and Oates (2015) demonstrated a novel framework for encoding time-series data as different types of images for visual inspection followed by a classification using tiled CNNs. Using Gramian angular fields (GAF) and Markov transition fields (MTF) enabled techniques from computer vision (CV) for classification (Wang & Oates 2015), high-level features could be extracted from the GAF, MTF and GAF-MTF images when using a DL feature extractor such as a CNN (Wang & Oates 2015). Wang and Oates (2015) concluded that their approach yielded competitive results when compared to other state-of-the-art methods. In addition to MTF, Yang, Chen and Yang (2020) employed Gramian angular difference fields (GADF) and Gramian angular summation fields (GASF) as the transformation methods for encoding time-series data. These encoded images were used to evaluate the performance as well as the complexity of CNN architectures (Yang, Chen & Yang 2020). The encoding methodology has had limited exposure to RS applications. However, Dias et al. (2020) deployed GASF, GADF and MTF transformations of pixelwise time-series data for a Eucalyptus region classification. Ten pre-trained CNNs were run using the encoded images and the results were evaluated and compared. It was concluded that

this novel framework of encoding time-series data as two-dimensional (2D) images for multiple CNN classification shows great potential for other RS applications.

A gap in the literature was identified and shows that no research has been conducted using the computer vision technique of encoding time-series data as 2D images for a CNN classification to detect human settlement expansion.

1.5 PROBLEM STATEMENT

Rapidly detecting informal settlements is critical for developing detailed maps of the expanding areas. Due to the rapid growth of urbanisation, accurate maps and regularly updated land cover data provide valuable information to local municipalities and regional governments for urban decision making and planning. Remote sensing and time-series data from satellites have proven to be an effective in monitoring and tracking land cover changes (De Beurs & Henebry 2005; Lu et al. 2004; Lunetta et al. 2006; Verbesselt, Hyndman, Newnham, et al. 2010). Human settlement expansion in the Gauteng province of South Africa has successfully been detected from 250 m resolution MODIS data (Kleynhans et al. 2013; Kleynhans et al. 2012) using a spatiotemporal autocorrelation change detection method. Classification of land cover data using standard ML algorithms such as random forest as well as more advanced DL methods (CNN) have also been implemented (Mboga et al. 2017; Stoian et al. 2019). DL is more accurate when trained with large amounts of data. The recently developed computer vision techniques allow the encoding of time-series data through different transformation techniques before classification. Wang and Oates (2015) concluded that the CNN classification using their encoded images through the GAF and MTF transformation could yield competitive results when compared to other state-of-the-art methods. Dias et al. (2020) and Yang, Chen and Yang (2020) conducted studies that provided the findings of Wang and Oates (2015) and showed the potential of utilising the novel framework within an RS application.

Despite the literature found, no apparent research has investigated the use of the computer vision method of encoding time-series data with a CNN classification for an urban settlement detection. A further gap in the literature is that there is no comparison between MODIS and Sentinel-2 data used for performing an urban change detection. From these apparent research gaps, the following research questions were formulated:

a) How effective is encoding time-series data as 2D images for CNN classification within the field of RS?

b) How valuable is the novel framework for performing an accurate urban change detection?

c) How beneficial is it to use higher resolution imagery such as Sentinel-2 over MODIS imagery for urban change detection?

1.6 AIM AND OBJECTIVES

This research aims to evaluate the potential of encoding time-series data as 2D images from MODIS and Sentinel-2 for an urban change detection through classification with convoluted neural networks.

To achieve the research aim, the following objectives have been set:

1) Review literature, specifically looking at urban change detections, deep learning CNN classifications and the novel framework of encoding time-series data.
2) Experiment 1:
    a. Evaluate and assess the effectiveness of encoding time-series data as 2D images for a CNN classification.
    b. Compare the performance of the novel framework to a baseline classification approach using the long short-term memory (LSTM) algorithm.
3) Experiment 2:
    a. Evaluate the effectiveness of increasing the spatial resolution of the data for the novel framework of encoding time-series data as 2D images.
    b. Assess the consequence of altering the temporal nature of the input time-series data.
    c. Evaluate the generalisability of the novel framework when testing with data gathered from three different geographical locations.
4) Synthesise the results of the two experiments to make further recommendations for performing an urban change detection using computer vision techniques and the DL classification apparatus.

1.7 SIGNIFICANCE AND RATIONALE

The transition from rural to urban has economic, social and political implications, exerting pressure on the capacities for urban management and planning. In South Africa, informal settlement expansion is rapid, requiring frequently updated land cover information to prevent problems with service delivery from local municipalities. By achieving the aim of the study, this research will yield an effective and novel framework for rapidly processing time-series images for urban change detection through encoding time-series image data and CNN classification. Should this research

prove successful, it will provide planners from local municipalities and regional governments with timeous information of nearby developing settlements and further assist with providing basic services. This research can also provide recommendations regarding which resolution satellite imagery would be more appropriate for human settlement detection.

1.8 RESEARCH METHODOLOGY AND AGENDA

This research is quantitative and deductive. By using experimental and evaluative techniques, an urban settlement detection will be performed by encoding image time-series data for input to a CNN classification. Secondary data extracted from the Google Earth engine (GEE) platform will be used as the input imagery for this research. The study will utilise both MODIS and Sentinel-2 imagery from a specified period. Classification will be conducted using a DL classifier, CNN. Statistical evaluation will ensure that the results yielded are successful (90%+) and comparable to current state-of-the-art techniques used for change detection.

The research agenda shown in Figure 1 illustrates the workflow of this thesis and the content of each chapter. Chapter 1 is the planning (proposal) phase of the research, followed by Chapter 2, the literature review, which will continue throughout the research. Chapters 3 and 4 describe the two experimental sections of the thesis. Chapter 3 investigates the use of computer vision techniques to encode the time-series data as 2D images for multiple CNN classifications. Chapter 4 focuses on the comparison of two different datasets while using the novel framework for an urban change detection. The final chapter will provide conclusions and make recommendations for future research.

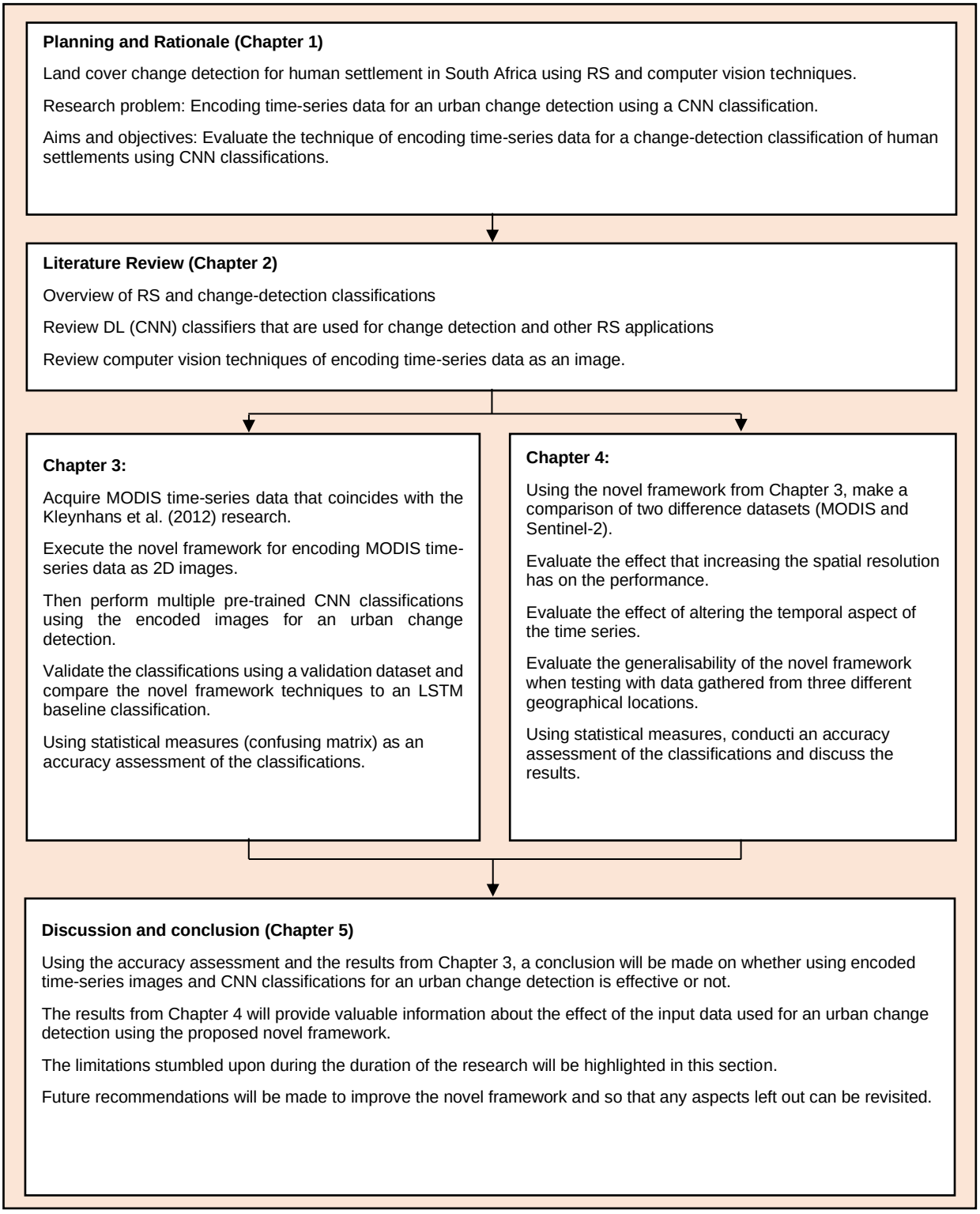


Figure 1.1: A workflow diagram illustrating an overview for each of the five chapters within this research thesis

CHAPTER 2: LITERATURE REVIEW

This chapter provides an overview of RS, artificial intelligence (AI) and computer vision. First discussed are some concepts of the electromagnetic spectrum (ES), reflectance properties and the importance of data pre-processing. Concentrating on RS and the study's objectives, the focus is placed on image classification and change detection. The fundamental concepts of AI, ML and DL with their respective linkages to the field of RS are addressed. This chapter concludes with a discussion of an interdisciplinary scientific field known as computer vision. This discussion includes a novel methodology of encoding time-series data as 2D images.

2.1 REMOTE SENSING

RS is the acquisition and process of gathering information about a phenomenon or object without making direct contact (Cracknell 2007; Lillesand, Kiefer & Chipman 2015; Mather & Koch 2011). Detecting and monitoring the physical characteristics of and object is done by analysing the electromagnetic radiation (EMR) reflected or emitted from the target (Campbell & Wynne 2011). This broad definition includes countless activities and applications across multiple scientific fields (Campbell & Wynne 2011; Chen & Campagna 2009; Crews & Walsh 2009; Dimitrios et al. 2012; Ng & Acharya 2009; Suzuki & Matsui 2012; Viana et al. 2017). Earth observation (EO) although RS interprets and analyses EMR emitted or reflected from objects located on the Earth surface (Lillesand, Kiefer & Chipman 2015). The data is recorded using airborne or space-borne instruments such as aeroplanes, unmanned aerial vehicles (UAV) and satellites (Mather & Koch 2011). These principles of RS for EO will be used as such throughout this thesis.

2.1.1 Electromagnetic radiation

Electromagnetic energy is a product of several mechanisms, including the change in electron energy levels, the electrical charges acceleration and the thermal movement of molecules and atoms (Lillesand, Kiefer & Chipman 2015). Electromagnetic energy has five characteristic properties; wavelength, frequency, amplitude, phase and speed (Campbell & Wynne 2011).

Reflected radiation from the surface is captured by sensors and used for analysis in RS. The RS field focuses on utilising visible light (blue, green and red, between 0.38 and 0.72 μm), infrared (IR, 0.72 to 1000 μm) and microwaves (1 mm to 30 cm) (Campbell & Wynne 2011; Chuvieco 2020).

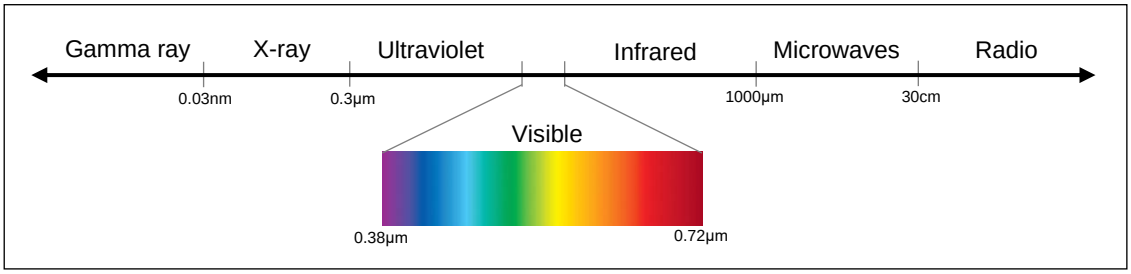


Figure 2.1: Electromagnetic radiation spectrum ranging from Gamma rays to Radio with their respective wavelengths

The EMR reflected by objects contains valuable information about the target's biological, chemical, and physical properties (Chuvieco 2020). An airborne or space-borne sensor captures the reflected radiation at different regions of the ES, otherwise known as bands (Patra 2010). Each band represents a specific portion of the ES and is limited by the designated wavelength cut-offs. The selection of a sensor and the bands collected depend on the intended target or RS application. The number of bands that an image adopts is referred to as spectral resolution, and is determined by the sensor (Patra 2010).

Spatial resolution refers to the area on the ground captured by the sensor at a specific altitude and point in time (Mather & Koch 2011). The spatial resolution ties into the number of pixels that can be found in the image. Sensors that can capture images at a higher spatial resolution will have significantly more pixels of a smaller size (Hsieh, Lee & Chen 2001). This will help to minimise mixed pixels and reduces the loss of information caused by the radiance averaging process (Jones & Sirault 2014).

Radiometric resolution refers to the recorded sensitivity levels caused by minor variations in the radiance (Campbell & Wynne 2011). Sensors with high radiometric resolutions have the pronounced ability to pick up slight variations in the object's radiance (Chuvieco 2020). The temporal resolution of imagery (De Beurs & Henebry 2005; Chuvieco 2020; Pelletier, Webb & Petitjean 2019) refers to the revisit period of a sensor for a particular location. Temporal resolutions often vary depending on the intention and objectives of the sensor (Campbell & Wynne 2011).

When opting for a particular sensor, it is crucial to consider the spectral, spatial, temporal and radiometric characteristics of the image. The objectives of the problem at hand determine the selection of the sensor as it will affect the classification and the respective results (Mather & Koch 2011). Although low spatial, spectral and radiometric resolutions will allow for faster processing of larger areas, they will not capture the finer details of the indented targets (Chuvieco 2020). To select a sensor for an assignment, one needs to understand the properties of surface reflectance and

how EMR interacts with the ground (Campbell & Wynne 2011). Section 2.1.2 will focus on the surface reflectance properties and interaction of EMR with water, soil and vegetation.

2.1.2 Surface reflectance

The EMR that can pass through the Earth's atmosphere and reach the surface will either be absorbed, transmitted, or reflected. Absorption of EMR results from an object consuming the light energy. In contrast, transmission occurs when the energy can pass through the object without major devitalisation (Campbell & Wynne 2011). Reflection is the process of redirecting light energy as it interacts with an object (Campbell & Wynne 2011). The extent to which absorption, transmission or reflection occurs depends on the wavelength of the light energy, angle of illumination, and the nature of the object's surface (Lillesand, Kiefer & Chipman 2015).

The two types of instruments used for gathering imagery in RS are either an optical-based sensor or a synthetic aperture radar (SAR) sensor (Chuvieco 2020). Optical-based sensors are passive sensors that utilise the sun's radiation to record the reflectance of the Earth's surface (Campbell & Wynne 2011). The drawbacks of optical sensors are that images cannot be collected during inclement weather or at night (Sahu 2006). Clouds and smoke present significant obstacles for collecting consistent imagery of the Earth's surface (Campbell & Wynne 2011). SAR sensors provide an alternative solution for the drawbacks that optical sensors present. However, the research in this thesis focuses on utilising optical imagery collected by the MODIS and Sentinel-2 instruments.

There are several optical sensors available today with varying spectral, spatial, radiometric and temporal resolutions. Multispectral RS systems have been commonly used for many EO applications such as land cover classification, change detections, vegetation and crop analysis, urban planning and fire tracking (Gong, Li & Zhang 2019; Hu, Dong & Batunacun 2018; Q Li et al. 2020; Liu et al. 2018; Ma et al. 2018; Phalke & Özdoğan 2018; Salmon et al. 2013; Usman et al. 2015). The sensors selected are dependent on the individual applications; some may opt for higher spatial resolution systems, where others may focus on the temporal resolution aspect of the data. The primary characteristics from several space-borne multispectral RS systems are shown in Table 2.1. The spatial resolution of the panchromatic (Pan) and multispectral bands are given alongside the total number of bands. The revisit time, also known as the temporal resolution, is provided with the launch year and the references for each sensor. There were two Sentinel-2 sensors launched; each sensor had a revisit time of 10 days. However, when both are used in conjunctions, the temporal resolution decreases from 10 days to five days.

Table 2.1: Eight spaceborne sensors with their respective properties

Sensors	Total bands	Spatial resolution (Pan)	Spatial resolution (MS)	Revisit time	Launch year	Reference
GeoEye-1	5	0.41m	1.65m	1.7 – 4.6 Days	2008	(GeoEye 2008)
IKONOS	5	0.82m	4m	± 3 Days	1999	(ESA 2020a)
Landsat 7	8	15m	30m, 60m	16 Days	1999	(Masek 2017)
Landsat 8	11	15m	30m, 100m	16 Days	2013	(Masek 2013)
MODIS	36	-	250 m, 500 m, 100m	1 – 2 Days	1999 & 2002	(USGS 2018)
Sentinel-2	13	-	10m, 20, 60m	5 Days	2015 & 2016	(ESA 2015)
SPOT-7	5	15m	6m	1 Day	2014	(ESA 2020c)
WorldView-3	9	0.3m	1.24m	1 Day	2014	(DigitalGloble 2014)
Quickbird	5	0.65m	2.62m	3 Days	2001	(DigitalGloble 2001)

RS data can be expensive, and alternative open-source platforms have easily accessible high-resolution imagery. The United States geological survey (USGS) Global Visualization Viewer (Glovis), USGS Earth Explorer, the Copernicus Open Access Hub and GEE are all open-source platforms that allow easy access to imagery from several sensors (Giuliani et al. 2018; Gong, Li & Zhang 2019; Hu, Dong & Batunacun 2018; Sundarakumar et al. 2016).

MODIS-derived imagery has a high temporal resolution with a one-day revisit time; however, it lacks spatial resolution (Table 2.1). MODIS 500 m resolution data has been used in many RS applications and has shown great success (Duong 2004; Grobler et al. 2013; Kleynhans et al. 2012; Wong et al. 2008). The MODIS sensor offers a higher resolution image with a 250 m x 250 m pixel size (Kleynhans, Salmon & Wessels 2017). However, spectral products derived from the 250 m resolution data compromise the revisit time, resulting in a temporal resolution change from 8 to 16 days (Lunetta et al. 2006). Low-resolution imagery such as MODIS allows processing over a large area (Broxton et al., 2014; Ryu et al., 2018). Landsat 7 and 8 and Sentinel-2 imagery have been implemented in countless RS studies as the higher resolution data are freely available (Daudt et al. 2018; Q Li et al. 2020; Stoian et al. 2019). However, before processing any RS imagery, the data requires pre-processing to remove any imperfections.

2.1.3 Pre-processing of imagery

Pre-processing is the removal and correction of the flaws and imperfections present in the RS imagery (Campbell & Wynne 2011; Mather & Koch 2011). These flaws and errors must be addressed before any processing or analysis (Chuvieco 2020).

Receiving stations can correct a few of the errors that occur when gathering RS imagery; however, the remaining errors may need to be addressed by the analyst before any processing. The necessary pre-processing steps include radiometric (atmospheric correction) and geometric (orthorectification) correction (Campbell & Wynne 2011).

Orthorectification corrects the geometric distortions caused by the variations in sensor velocity and altitude (Campbell & Wynne 2011; Lillesand, Kiefer & Chipman 2015). These variations are related to the curvature of the Earth's surface, relief displacement, atmospheric refraction, and panoramic distortions (Lillesand, Kiefer & Chipman 2015). The resulting product is a geometrically correct image. Ground control points are gathered and used to link geographic coordinate systems to the image coordinate system (Mather & Koch 2011). Once an image has been geometrically corrected and has a geographic coordinate system, resampling can be executed (Chuvieco 2020).

Atmospheric correction accounts for the downward solar irradiance and the upward radiance leaving the Earth's surface (Chuvieco 2020). The absorption and scattering of EMR within the atmosphere alter the magnitude of the radiance leaving the ground (Chuvieco 2020; Mather & Koch 2011). Atmospheric correction is when the true reflection is simulated by accounting for several factors (Lillesand, Kiefer & Chipman 2015). These include satellite geometries (zenith angle), the solar zenith angle, azimuth angles, the slope of the ground surface, topographic features, atmospheric gas parameters, as well as the atmospheric conditions such as aerosol optical thickness (AOT) (Lillesand, Kiefer & Chipman 2015; Mather & Koch 2011). Atmospheric correction is a critical pre-processing step and needs to be accounted for by the analysts. A platform such as GEE is beneficial as it provides a final pre-processed product that can immediately be used (Celik 2018). GEE has become a popular platform for performing RS tasks alongside open-source software as it provides users with pre-processed ready-to-go imagery (Celik 2018).

2.1.4 Image classification

Digital image classification is used to assign pixels to informational classes (e.g. land cover) from RS imagery (Campbell & Wynne 2011; Lillesand, Kiefer & Chipman 2015). There are several different approaches to perform image classification, namely unsupervised, supervised, pixel-based, object-based and rule-based classifications. This section will also provide commonly used ML classifiers and their success within the field of RS.

The unsupervised classification approach focuses on utilising a clustering method to identify natural groups or structures in a multispectral image(Campbell & Wynne 2011; Liu & Mason 2016). This classification approach is beneficial when classifying data without prior knowledge of

the area or region (Liu & Mason 2016). However, the spectral classes formed still require a verification process by the analyst (Mather & Koch 2011).

A supervised classification approach is based on the idea that there is prior knowledge of the study area, either through fieldwork or secondary sources (Campbell & Wynne 2011). The underlying principle of supervised classification is that unknown pixels are classified using samples of pixels with known identify (Campbell & Wynne 2011; Chuvieco 2020). These samples are commonly referred to as training samples and are collected by the analyst (Chuvieco 2020). The critical aspect of supervised classifications is collecting adequate training samples (Campbell & Wynne 2011; Mather & Koch 2011). The main objective here is to obtain a significant number of samples to compensate and accurately represent the variation in spectral information for each class or category. The drawbacks of a supervised classification approach are that the output tends to be biased because analysts usually assign training samples before considering the spectral characteristics (Campbell & Wynne 2011). Another limitation is the time for which adequate training data is collected (Mather & Koch 2011). However, if the training data is collected correctly, supervised classifications are often more effective than unsupervised classification (Enderle & Weih 2005; Mohd Hasmadi, Pakhriazad & Shahrin 2009; Nijhawan, Srivastava & Shukla 2017).

The idea behind a pixel-based classification is that the classifier will apply decision logic to each pixel (Castillejo-González et al. 2009; Lillesand, Kiefer & Chipman 2015). This type of classification works on a per-pixel basis, where each pixel is isolated (Lillesand, Kiefer & Chipman 2015). Pixel-based approaches are usually practical for datasets that show a relationship between the information classes (e.g. land cover types) and the spatial resolution (Castillejo-González et al. 2009; Duro, Franklin & Dubé 2012; Gao & Mas 2008). However, a decrease in the effectiveness will begin to show when the target features are more significant in size than the pixels in the dataset (Castillejo-González et al. 2009). This leads us to the alternative approach known as an object-based classification.

Unlike the pixel-based approach, object-based classifiers use a segmentation algorithm that aggregates image pixels into homogeneous objects that do not intersect (Castillejo-González et al. 2009; Liu & Xia 2010; Myint et al. 2011). The resulting product is a multi-resolution segmentation image from which the objects are classified (Comer & Delp 1995; Comer & Delp 1999; Liu & Xia 2010). Benefits of an object-based approach include additional information such as the spatial relationships between neighbouring objects, shape, texture and other spatial-related data (Hussain et al. 2013; Liu & Xia 2010). Additionally, this approach helps reduce the spectral variations within classes and severity of the “salt-and-pepper” effect (Liu & Xia 2010).

Although object-based classifiers have been shown to outperform pixel-based classifiers and achieve a higher overall accuracy (OA) (Araya & Hergarten 2008; Cleve et al. 2008; Myint et al. 2011; Riggan & Weih 2009), both have limitations associated with mixed pixels. The understanding behind mixed pixels is that they present themselves when a pixel cannot ideally occupy one homogenous class (Campbell & Wynne 2011; Jones & Sirault 2014). They display the average brightness value from several classes rather than just one (Campbell & Wynne 2011). Mixed pixels have an indirectly proportional relationship to the spatial resolution and will increase as the resolution decreases (Campbell & Wynne 2011; Jones & Sirault 2014).

The interface between wildland and urban was classified using object-based and pixel-based classifiers (Cleve et al. 2008). Cleve et al. (2008) deducted that the object-based approach performed better when representing built areas. However, the pixel-based classifier could yield similar accuracies for the shrub/tree and shadow classes (Cleve et al. 2008).

2.1.4.1 Rule-based classification

The rule-based classification approach uses a set of rules applied sequentially to discriminate between the different categories (Mendel 2017). These targeted categories are classified based on whether they meet the thresholds of the ruleset (Chuvieco 2020). Fuzzy rule-based classification systems and classification and regression trees (CART) are both well known for this classification approach (Ishibuchi & Nakashima 2001; Lawrence & Wright 2001; Mendel 2017). The CART algorithm is commonly used to build DT, which continuously divide features until reaching a desired level of homogeneity at the terminal nodes (Lawrence & Wright 2001; Mendel 2017). To validate and improve the results, a cross-validation method is applied using unfamiliar samples that were not used in the construction of the DT (Chuvieco 2020).

2.1.4.2 Parametric classifiers

Parametric classifiers greatly simplify the learning process by assuming that the training data follow a Gaussian or normal distribution (Hubert-Moy et al. 2001; Jain, Duin & Mao 2000). This allows the classifiers to learn and summarise data through calculated parameters (Jain, Duin & Mao 2000). These classifiers present a limitation as they assume classes in a multispectral space are symmetric (Hubert-Moy et al. 2001). Parametric classifiers also assume that the boundaries for the classes use a fixed-form decision (Hubert-Moy et al. 2001). The maximum likelihood (MaxL) is a commonly used and well-known parametric classifier (Myburgh & Niekerk 2013; Strahler 1980; Wei & Mendel 2000). Araya & Hergarten (2008) performed a comparative study to distinguish a land cover classifying using a pixel- or object-based classification. The maximum

likelihood classifier is known to be too sensitive and can be affected by the quality of training data and shows a decrease in accuracy when more input features are used (Myburgh & Niekerk 2013).

2.1.4.3 Machine learning classifiers

RS data typically does not have a normal distribution and requires a classifier that can deal with the versatility of the data (Jain, Duin & Mao 2000). Non-parametric classifiers are an ideal solution as they do not make any assumptions regarding the training data distributions, nor do they estimate the parameters (Jain, Duin & Mao 2000). Several ML classification algorithms have been implemented for RS and have shown great success (Brovelli, Sun & Yordanov 2020; Camps-Valls 2009; Celik 2018; Maxwell, Warner & Fang 2018; Shang & Chisholm 2014). These classifiers include DTs, SVM, ***k***-nearest neighbours (***k***-NN), RF and ANN (Maxwell, Warner & Fang 2018; Pal & Mather 2005).

DT classifiers can identify relationships between dependent and independent variables by learning simple decision rules deduced from the data features (Priyam et al. 2013; Quinlan 1996; Swain & Hauska 1977). The architecture of a DT classifier consists of one or more branches, where each branch contains a set of rules. The rules are used to process and assign the most probable class to the unknown instance (Lawrence & Wright 2001).

A simple non-parametric classifier, such as ***k***-NN, uses distance-based labelling unknown instances (Cover & Hart 1967; Cunningham & Delany 2020; Islam et al. 2008). ***K***-NN utilises known neighbouring instances to assign classes to the unknown instances (Cover & Hart 1967; Cunningham & Delany 2020). ***k*** represents the number of nearest neighbours, which is the core factor in determining the final class (Cunningham & Delany 2020). This simple non-parametric classifier is effective for data that does not have a normal distribution. In RS and particularly the process of high-resolution imagery, ***k***-NN is effective (Li & Cheng 2009). Li and Cheng (2009) illustrated the success of the ***k***-NN classifications as they achieved a minimum accuracy of 84% for their five class classification (bare land, green-land, road, settlement and water). However, a comparative study showed that the ***k***-NN classifier was outperformed by other ML classifiers such as DTs and SVM (Qian et al. 2015)

SVM classifiers focus primarily on the training samples located near the edge of class descriptors (Amarappa & Sathyanarayana 2014; Tzotsos & Argialas 2008). This allows the SVM classifier to determine the most optimal separating hyperplane amongst the classes (Novack et al. 2011; Shao, Chen & Deng 2014). SVM classifiers have shown their efficiency within RS for numerous applications including urban mapping (Bazi & Melgani 2006; Cao et al. 2009; Pal 2008; Petropoulos, Kalaitzidis & Prasad Vadrevu 2012). Myburgh and Niekerk (2013) performed an

object-based land cover classification for five features using three different classifiers; an SVM, ***k***-NN and ML classifier. The SVM achieved the highest OA at 90.5%, 20.6% higher than the ***k***-NN and 0.8% higher than the ML classifier (Myburgh & Niekerk 2013). The SVM classifier mapped the more complex *bare ground and build-up* class more accurately than the other classifiers (Myburgh & Niekerk 2013). Cao et al. (2009) also expressed that the SVM based approach does not only present comparable results to the local threshold method, but helps remove the trial-and-error procedure, concluding that this new approach is a simple alternative for detecting urban extents.

The RF classifier has shown to be effective for performing image classifications of RS data (Celik 2018; Lawrence & Wright 2001; Novack et al. 2011; Poona & Ismail 2014; Rodriguez-Galiano et al. 2012). The RF algorithm is based on using multiple DTs (Maxwell, Warner & Fang 2018). Each DT is generated using random features sampled separately from the input vectors (Breiman 1996; Breiman 2001; Pal 2005). An uncorrelated forest of DTs is formulated (Maxwell, Warner & Fang 2018), and a vote is cast at the individual trees (Breiman 2001; Pal 2005). Each DT contributes to a vote that will determine the assignment of the input variables (Rodriguez-Galiano et al. 2012). Training sets are generated for feature selection, which helps with reducing RF classifier sensitivity to training set sizes (Rodriguez-Galiano et al. 2012). RF has been successfully implemented for a sematic classification of urban buildings, it was also concluded that this approach was effective and accurate (Du, Zhang & Zhang 2015). Ghosh, Sharma & Joshi (2014) also illustrated the success of the RF classifier for an urban landscape. Akar & Güngör (2012) conducted a land-cover classification of the urban and rural features, it was illustrated that the RF classification outperformed the SVM algorithm by 10% with respect to the urban data. The RF classification showed to be more effective with the urban data producing a OA of 85.63% (Akar & Güngör 2012).

DL classifiers fall under a subsection within ML where the models start to introduce a sophisticated approach to ML (Ongsulee 2018; Zhang et al. 2017). These complex multi-layer neural networks (NN) allow data to pass between nodes in a highly connected manner (Ongsulee 2018). DL classifiers are developing and have shown to be extremely powerful when trained correctly (Ongsulee 2018; Zhang et al. 2017). RS has recently started to adopt several DL classifiers for many applications including urban detection (De et al. 2018; W Li et al. 2019; Ma et al. 2019; Pan et al. 2020; Zhang et al. 2016; Zhang, Zhang & Kumar 2016; Zhu et al. 2017). Further details regarding the architectural structure of the classifiers and the application and the success of these DL classifiers are provided in the AI section (2.2) below.

2.1.4.4 Feature selection

Feature selection is a core aspect of ML and impacts the model's performance (Brownlee 2020; Kuhn 2018; Tang, Alelyani & Liu 2014). It reduces the input variables for a predictive model (Kuhn 2018; Kuhn & Johnson 2019). Reducing the number of input variables reduces the computation cost and potentially increases the model's performance (Tang, Alelyani & Liu 2014). Data cleaning and feature selection are essential steps and should be taken to achieve the best possible results (Kuhn & Johnson 2019). The three main feature selection techniques include the embedded, wrapper and filter feature selection methods (Kuhn 2018; Kuhn & Johnson 2019). Three top-performing feature selection wrappers include the area under the receiver operating characteristic curve of the RF (AUC-RF), recursive feature elimination (RFE) and Boruta (Poona et al. 2016). Feature selection has shown to play a significant role in optimising the performance of classifications for urban applications in RS (Georganos et al. 2018).

Dimensionality reduction does not form part of the feature selection process and is an alternative method (Kuhn 2018; Tang, Alelyani & Liu 2014). Different ideologies project the model's input data into a lower-dimensional feature space (Kuhn 2018; Kuhn & Johnson 2019). However, feature selection and dimensionality reduction techniques are related in that they both seek fewer input variables for predictive models (Kuhn 2018). The difference is that dimensionality reduction techniques form entirely new input features by creating a data projection, whereas feature selection merely selects features to remove or keep in the dataset (Kuhn 2018; Kuhn & Johnson 2019).

Dimensionality reduction has been a popular option for reducing and removing noisy and redundant features (Tang, Alelyani & Liu 2014). Three popular techniques include linear discriminant analysis (LDA), canonical correlation analysis (CCA) and principle component Analysis (PCA) (Tang, Alelyani & Liu 2014). PCA has successfully been deployed in several cover change detection applications (Deng et al. 2008; Dharani & Sreenivasulu 2021; Qin et al. 2013).

2.1.4.5 Accuracy assessment

Accuracy assessment and validation are essential processes in classifying RS data (Lewis & Brown 2001). Validating the results with ground truth data allows one to determine the success of the classification and detect any errors (Ariza-López, Rodríguez-Avi & Alba-Fernández 2018). The standard accuracy assessment for evaluating the success of image classification is usually conducted using a confusion matrix (Ariza-López, Rodríguez-Avi & Alba-Fernández 2018; Lewis & Brown 2001). Standard error metrics or confusion matrices contain the OA, Kappa, errors of commission and omission, and the user and producer accuracies (Ariza-López, Rodríguez-Avi &

Alba-Fernández 2018; Lewis & Brown 2001). To formulate a confusion matrix, ground truth samples (points) need to be generated and must be representative of the classes (Ariza-López, Rodríguez-Avi & Alba-Fernández 2018). Several different sampling schemes can be deployed, such as random sampling, systematic, stratified-random and stratified systematic unaligned. The OA is a widely used and simple measurement of the overall proportion of correctly classified samples (Salmon et al. 2015). However, OA must not be used alone as big classes or big number of samples may skew the results (Salmon et al. 2015). The Kappa (KHAT or k) is a statistical estimate of the classification performance when compared to a random classification (Salmon et al. 2015). The Kappa value is an indication of the reliability of the classification (Equation 2.2):

Equation 2.2

$$\widehat{K} = \frac{N\sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r}(x_{i=} \cdot x_{+i})}{N^2 - \sum_{i=1}^{r}(x_{i+} \cdot x_{+i})}$$

Where r is the number of rows in the error matrix;

x_{ii} is the number of observations in row column i;

x_{i+} is the total observations in row i;

x_{+i} is the total observations in column i; and

N is the total number of observations in the matrix.

The user and producer accuracies provide the estimated accuracy per class (Salmon et al. 2015). The user accuracy refers to the number of samples correctly assigned to the class that they belonged to (Lewis & Brown 2001; Salmon et al. 2015), whereas the producer's accuracy indicates how many samples belong to a class and are assigned to that respective class (Salmon et al. 2015). Errors of commission and omission are the invert of the user and producer accuracies, respectively. A confusion matrix is an effective tool for performing an accurate assessment of a classification.

Supervised classifications have an additional validation measure. It has become second nature in a supervised classifications approach to split the data and utilise a portion for training, followed by testing (Tan et al. 2021). The ratio for which the data is spilt into training and test depends on the analyst, however, it has become common practice to use a 70/30 or 80/20 split for training and test, respectively. GIS software and Python packages offer an evaluation function that utilises the testing data from the train-test split (TensorFlow 2021). The evaluation function, in most instances, represents the OA of the classification, however (TensorFlow 2021), it is still important to investigate the other elements of the confusion matrix.

K-fold cross-validation is a standard model evaluation procedure that uses the train-test split aspect of ML. The split of the input dataset can occur at random and as a result, the model is trained with a specific portion of the data (Figure 2.2) (Dias et al. 2020; Rodríguez, Pérez & Lozano 2010). It is suggested that by running and retraining the algorithms multiple times, each time on a different portion of the dataset, one is validating the model (Fushiki 2011; Wong & Yeh 2020). Analysts set out several folds, represented by the term k (Rodríguez, Pérez & Lozano 2010). As shown in Figure 2.2, different portions of the datasets are used to train and test the models. Each fold is run through the classifier, and results are recorded (Rodríguez, Pérez & Lozano 2010).

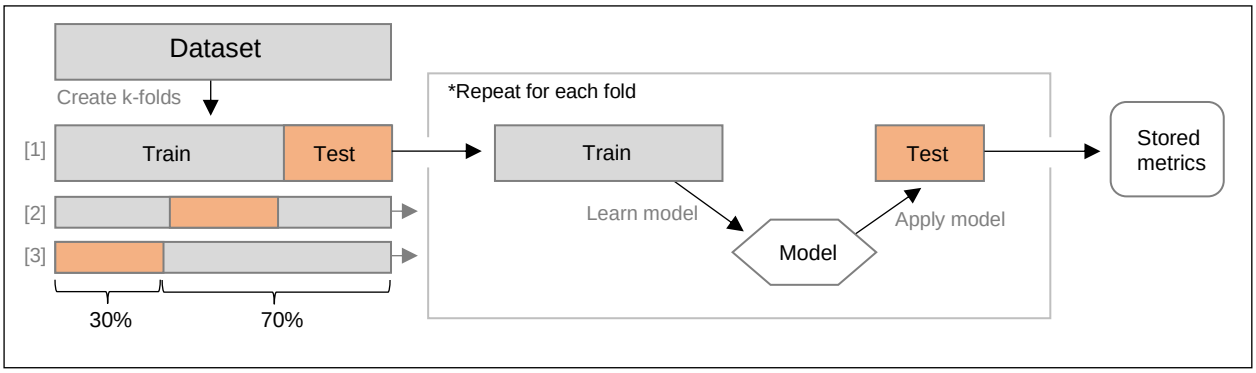


Figure 2.2: K-fold cross-validation diagram illustrating the evaluation procedure of running multiple models using different portions of the dataset

2.1.5 Land cover change detection

Conducting a change detection requires two or more images from separate dates (Kleynhans et al. 2012). Each image is then classified and compared to the previous, either at the pixel or object-based scale (Moser, Serpico & Vernazza 2007; Radke et al. 2005). A binary change detection infers that the result consists of two classes: change and no change (Moser, Serpico & Vernazza 2007; Radke et al. 2005). Multi-class change detection will classify the two images and compare the land cover classes. However, a comparison with only two images can often prove unreliable, as the land cover of a similar nature may appear different at various stages in their seasonal growth cycle (Lunetta et al. 2006). Time-series data can be introduced to the process to improve results (Section 2.1.6.1).

There are several ways for performing change detection, either by altering the source of the data (SAR or optical) or selecting a preferred method. This section will focus on studies that have utilised optical data to perform change detection. However, it is essential to recognise SAR imagery's success with land cover classifications and change detections that are focused around urban areas (Hu & Ban 2014; Lopez, Shimoni & Grippa 2017; Sinha, Santra & Mitra 2018). In addition to the classification methodology whereby ML and DL algorithms are used to classify images (Daudt et al. 2018; Yin et al. 2017), an ACF can be used for monitoring the surface

reflectance over time (Kleynhans et al. 2013; Kleynhans et al. 2012). It is essential to understand all methods and data sources used in performing accurate change detections.

2.1.5.1 Time series analysis

An increase in the temporal frequency could help distinguish change events from natural phenological cycles (Lunetta et al. 2006). Time-series data is currently being deployed as an effective and reliable dataset for distinguishing changes (Grobler et al. 2013; Kleynhans, Salmon & Wessels 2017; Lunetta et al. 2006; Salmon et al. 2013). Gathering a high temporal dataset is critical for accurate change detection, although it is not the only aspect to consider. When working with passive optical sensors, spectral resolution is an essential factor to consider.

2.1.5.2 Spectral analysis

The spectral response is constructive in detecting changes in land cover types (Espinoza-molina et al. 2017; Tran et al. 2018; Xue & Su 2017) as each part of the ES has a unique interaction with each different land cover type (Campbell & Wynne 2011; Lillesand, Kiefer & Chipman 2015). Band selection is essential for gathering the correct data for performing a change detection (Polykretis, Grillakis & Alexakis 2020; Sinha, Sharma & Nathawat 2015). To track changes in land cover, different land cover types must be effectively classified (Gašparović, Zrinjski & Gudelj 2019). Each band or combination of bands, known as indices, can detect a specific land cover type (Espinoza-molina et al. 2017; Polykretis, Grillakis & Alexakis 2020; Xue & Su 2017). Indices have effectively been deployed for several applications regarding land cover classifications and change detections, most of which involve vegetation, soil and urban monitoring (Abbas et al. 2013; Alexander 2020; Espinoza-molina et al. 2017; Li 2020; Qian et al. 2015; Tran et al. 2018).

Vegetation and stress monitoring utilise several indices such as the NDVI, simple ratio index (SRI) and the soil-adjusted vegetation index (SAVI). Healthy vegetation has a high reflectance in the green region of ES and a lower reflectance in the blue and red regions. The same applies to non-stressed and stress vegetation, respectively. The near-infrared (NIR) regions of the ES can detect the stress levels, where high NIR reflectance illustrates healthy vegetation.

NDVI (with a dynamic range of -1 to1) has been one of the more effective indices in RS and has been used in many applications, not just the monitoring of vegetation stress (Baluja et al. 2012; Hu, Dong & Batunacun 2018; Kim et al. 2011; Kleynhans et al. 2012; Lunetta et al. 2006). NDVI is defined as follows (Rouse et al. 1974):

Equation 2.4

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Where $NDVI$ is the vegetation index;

NIR is the NIR band; and

Red is the red band.

Several other vegetation indices have been developed and deployed for RS applications. These include the ratio vegetation index (RVI), difference vegetation index (DVI), perpendicular vegetation index (PVI), atmospherically resistant vegetation index (ARVI), soil line atmospheric resistance index (SLRA), type soil atmospheric impedance vegetation index (TSARVI) and several others (Xue & Su 2017). Indices have also been developed and deployed for urban monitoring (Li 2020). The normalised difference built-up index (NDBI) is used for mapping built-up or urban areas (Zha, Gao & Ni 2003). Zha, Gao and Ni (2003) concluded that NDBI contributed to the 92.6% accuracy of mapping urban areas. The ratio normalised difference soil index (RNDSI) is an urban mapping index usually used alongside the NDBI and NDVI (Deng et al. 2015; Li 2020).

Several studies have successfully implemented spectral vegetation indices for urban mapping by monitoring the change in vegetations (Grobler et al. 2012; Grobler et al. 2013; Kleynhans et al. 2011; Kleynhans et al. 2012). Grobler et al. (2013) and Kleynhans et al. (2012) illustrated that NDVI and the individual bands from MODIS could be used to perform urban change detection.

2.1.5.3 Urban change detection

Developing countries such as South Africa have an increasing rate of urbanisation as people are migrating to major cities in search of employment (UN 2018). Urban expansion is the largest and most pervasive land cover change in South Africa (Kleynhans et al. 2013). Several studies have investigated the best way to perform urban change detection within South Africa using freely available imagery. Grobler et al. (2012) performed a land cover separability analysis that uses a harmonic oscillator and combines it with a mean-reverting stochastic process. Deploying these two mathematical processes could produce urban change detection; however, with lower than acceptable accuracies. Salmon et al. (2013) illustrated a successful urban change detection method using internal covariance matrices from an Extended Kalman Filter. The urban change detection yielded accuracies over 90% for processing MODIS time-series data in Gauteng province of South Africa (Salmon et al. 2013).

Kleynhans et al. (2012) provided an alternative approach for performing an urban change detection in South Africa, which used an ACF applied to 500 m MODIS NDVI imagery. The temporal ACF method comprised two stages: the off-line optimisation phase and the operational phase

(Kleynhans et al. 2012). The offline optimisation phase used the simulated change data with the no-change data to determine the parameters appropriate for the applications (Kleynhans et al. 2012). These parameters consisted of the band, lag, and threshold selection (Kleynhans et al. 2012). The temporal ACF yielded a high OA of 88.46% for detecting urban change in Gauteng, South Africa. Kleynhans et al. (2013) improved on the temporal ACF method using a spatiotemporal ACF approach. The spatiotemporal ACF considered the neighbouring N pixels around the located pixel (Kleynhans et al. 2013). Using several mathematical equations and the Euclidean distance between pixel centres and their mean values, a spatiotemporal ACF was constructed (Kleynhans et al. 2013). Kleynhans et al. (2013) concluded that the spatiotemporal ACF improved the previous method (Kleynhans et al. 2012) and achieved a 90.98% OA using the same dataset. Kleynhans, Salmon & Wessels (2017) then performed a temporal ACF using a higher resolution dataset (MODIS 250 m), increasing the change detection accuracy (CDA) by 0.41%. Kleynhans, Salmon & Wessels (2017) conducted further investigations into the dataset's temporal aspect and sampling frequency. It was concluded that over the six-year observation period, the daily, eight-daily, monthly, and two-monthly sampling frequency had no significant effect on the CDA (Kleynhans, Salmon & Wessels 2017). However, when the sampling frequency was reduced beyond two-monthly, there was a noticeable decrease in the change detection performance (Kleynhans, Salmon & Wessels 2017).

DL algorithms have recently been implemented for RS applications such as change detections (Seydi, Hasanlou & Amani 2020; Shi et al. 2021; Stoian et al. 2019; Zhang et al. 2018). However, to understand why there is an uptake in the use of DL algorithms and how they can perform the given tasks effectively, one needs to comprehend the complexity of the architecture of the algorithm. The following section will provide an in-depth discussion of AI and, more importantly, DL algorithms.

2.2 ARTIFICIAL INTELLIGENCE

AI falls under a wider branch within the computer science field, which focuses on building intelligent machines and algorithms capable of executing tasks that would typically require a human (Leslie 2019; McCarthy. 2007). It is the intelligence displayed and demonstrated by machines or models (Jackson 2019). Al can be any system that takes action and perceives the environment to maximise the system's ability to achieve the goals set out at the beginning (Jackson 2019; Leslie 2019). AI has previously been divided up depending on the application, however, applications have become so extensive that this is no longer applicable (Nilsson 1982). In this section, one aspect of AI (i.e. ML) will be broken down and discussed.

2.2.1 Machine learning

AI and ML are both a part of the computer science field, however, AI is the broader concept where intelligent machines are created to simulate human thinking (Michalski, Carbonell & Mitchell 2013; Ongsulee 2018). ML is a subfield of AI that allows machines to learn from the data and perform tasks without being explicitly programmed to do so (Michalski, Carbonell & Mitchell 2013; Ongsulee 2018). ML consists of several algorithms to learn and improve data predictions and outcomes (Jordan & Mitchell 2015; Zhang 2020). The algorithms are designed to spot patterns using statistical techniques and then perform tasks based on these patterns (Jordan & Mitchell 2015). One of the sophisticated tasks ML algorithms are capable of is the classification of data or imagery. Several of these classifiers and the applications for which they were deployed have been mentioned in Section 2.1.5.6. RS has adopted the use of ML and AI for the processing and classification of imagery. Land cover monitoring and change detection have also implemented ML algorithms to improve performance and accuracy (Section 2.1.6). The following section will focus on DL fundamentals, different algorithm architectures and the applications where they have been deployed within RS.

2.2.2 Deep learning

DL is a subfield of ML with next-generation algorithms (Goodfellow, Bengio & Courville 2016). DL models are designed to analyse data and make relative predictions independently without human input (Goodfellow, Bengio & Courville 2016; Lecun, Bengio & Hinton 2015). This can be achieved using a layered algorithm structure known as ANN. The ANN structure and design are inspired by a human brain and the biological neural networks present (Goodfellow, Bengio & Courville 2016; Lecun, Bengio & Hinton 2015). As a field itself, DL is consistently growing and being used for countless applications across academic fields (Deng & Yu 2014; Najafabadi et al. 2015). As a result, several DL algorithms are currently being deployed (Shrestha & Mahmood 2019). Each has a unique architectural structure and can process different types of input data effectively.

A few commonly used DL algorithms include CNNs, LSTMs, recurrent neural network (RNN), generative adversarial network (GAN), radial basis function networks (RBF), and multilayer perceptron's (MLPs) (Goodfellow, Bengio & Courville 2016; Lecun, Bengio & Hinton 2015; Shrestha & Mahmood 2019). The sections following this will discuss CNNs, RNNs and LSTMs, and the details regarding their respective architectural structures.

2.2.3 Convolutional neural network (CNN)

CNNs are powerful and compelling image-processing algorithms that can perform descriptive and generative tasks (Albawi, Mohammed & Al-Zawi 2018; Kim 2017). These algorithms can recognise and classify features within an image and are widely used for analysing images (Albawi, Mohammed & Al-Zawi 2018; Kim 2017). The idea of the term convolution denotes the mathematical function of convolution, which refers to the linear operation whereby the multiplication of two functions produces a third function (Albawi, Mohammed & Al-Zawi 2018; Kim 2017). CNNs consist primarily of three layers: a convolutional layer, a pooling layer, and a fully connected layer (Albawi, Mohammed & Al-Zawi 2018) (Figure 2.3).

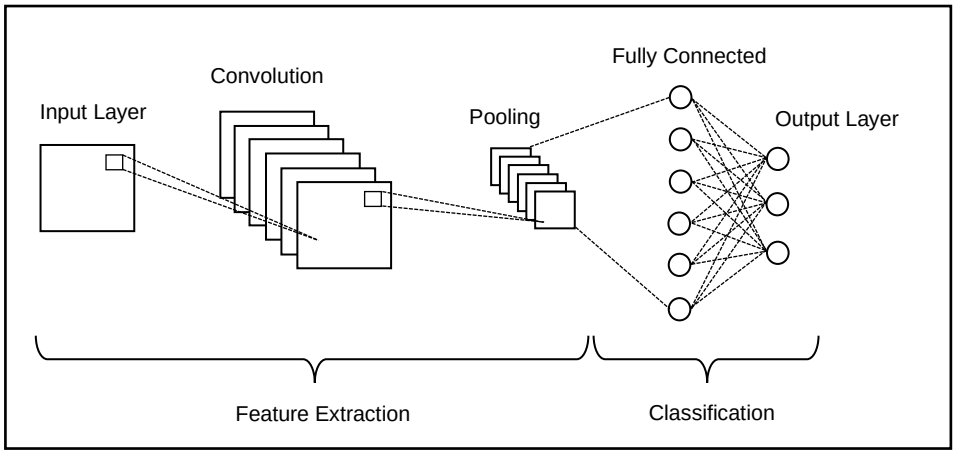


Figure 2.3: A basic CNN architecture showing the convolutional layer, pooling layer, and fully connected layer

The convolution layer extracts features from the input image (Albawi, Mohammed & Al-Zawi 2018; Song et al. 2019). This is the layer where the mathematical function of convolution occurs between the image and a specified filter with a particular size (Albawi, Mohammed & Al-Zawi 2018; Kim 2017). The filter slides over the image and records the dot product between the filter and the image with respect to its size (Kim 2017). The result is a feature map that provides information regarding the corners and edges of the image. The feature map is then fed through other layers to extract additional features (Albawi, Mohammed & Al-Zawi 2018; Kim 2017; Song et al. 2019). A convolutional layer is typically followed by pooling layers (Song et al. 2019). The objective of a pooling layer is to reduce the size of the feature map, which helps with decreasing computational costs (Albawi, Mohammed & Al-Zawi 2018; Pelletier, Webb & Petitjean 2019). This is done by reducing the connections between layers and independently operating for each feature map (Albawi, Mohammed & Al-Zawi 2018; Song et al. 2019). Several pooling operations are available and are dependent on the method of choice. These operations include max pooling, which selects the most prominent element from the feature map, average pooling, which formulates the average of the elements within the predefined size, and sum pooling, which

calculates the total value from all elements within that predefined size (Albawi, Mohammed & Al-Zawi 2018; Song et al. 2019). The fully connected layers form the last few layers before the output layer (Song et al. 2019). These fully connected layers contain weights and biases and neurons used to connect two different layers (Song et al. 2019). The output images from the previous layers are flattened and passed through the fully connected layers, whereby the flattened vector undergoes several mathematical function operations (Song et al. 2019). It is at this stage where the classification process takes place.

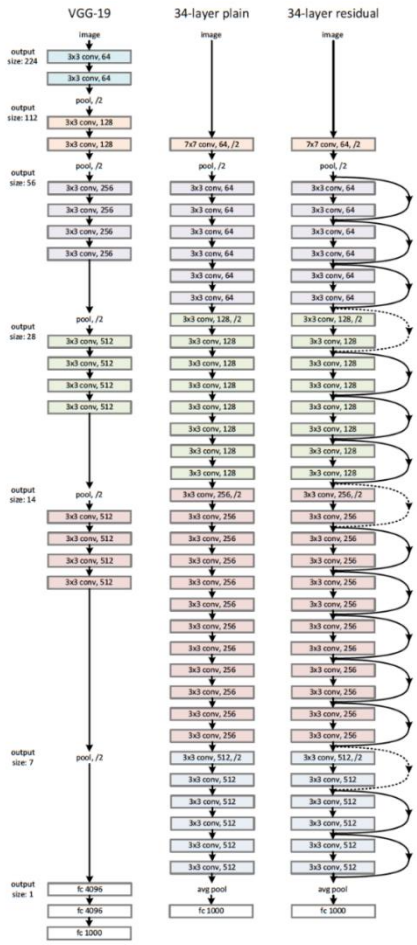
A critical parameter of the CNN architecture is the activation function. These functions approximate and learn from the continuous and highly complex relationships between the network and its variables (Song et al. 2019). The activation function will decide which information shell continues through the network. These activation functions include ReLU, Softmax, Sigmoid and tanH (Sharma, Sharma & Anidhya 2020; Song et al. 2019). Depending on the specific usage and the intention of the algorithm, an activation function is selected (Sharma, Sharma & Anidhya 2020; Song et al. 2019). Softmax is generally the preferred choice for multi-classifications, where the Sigmoid is used for binary classification.

Although it has been stated that there are three layers, CNN commonly to contain several convolutional and pooling layers. The operations mentioned above are therefore repeated multiple times within the model before reaching the fully connected layers. As a result, there are numerous variations of well-known algorithms that have been developed for particular applications. However, an increase in layers does not necessarily increase performance.

2.2.3.1 Pre-trained CNNs

CNN models require substantial training data to produce accurate and effective classifications (Kim 2017). However, one often does not have the time to collect or even possess the available data for such extensive training datasets. As a result, several CNN architectures have been pre-trained on large-scale image databases such as ImageNet (Dias et al. 2020; Fei-fei et al. 2021; Zhang et al. 2021). These large-scale image databases contain over 14.1 million images and provide an adequate training base for the models. Commonly used pre-trained CNN architectures include residual network (ResNet) (He et al. 2016; Kaiming et al. 2015; Zhang et al. 2021), Densely connected convolutional networks (DenseNet) (Huang, Liu & Van Der Maaten 2017; Ruiz 2018b; Zhang et al. 2021), visual geometry group (VGG) (He et al. 2016), Inception network (Szegedy et al. 2016), and MobileNet (Howard et al. 2017). Each of these pre-trained CNNs has a different architectural structure.

Keras provides an open-source NN library that contains ResNet V1 with 50, 101 and 152 layers; DenseNet with 121, 169 and 201 layers; VGG16 and VGG19; InceptionV3; InceptionResNetV2; and MobileNetV. Keras enables fast experimentations with several deep neural networks (DNNs) (Keras 2020a). The inception architecture is a micro-architectural design that allows for deeper convolutional layers (Szegedy et al. 2016). The inception module performs a multi-level feature extraction by calculating the 1x1, 3x3 and 5x5 convolutions (Längkvist, Karlsson & Loutfi 2017; Szegedy et al. 2016). These outputs are stacked alongside the channel dimension before passing through the next network layer. The VGG network was introduced by Simonyan & Zisserman (2015) and is characterised as a simplistic CNN. The VGG algorithms only use 3x3 convolutional layers and the max-pooling to account for the volume size (He et al. 2016). The two fully connected layers contain 4096 nodes each, with the Softmax activation function. The values 16 and 19 from the VGG16 and VGG19 algorithms refer to the number of weighted layers for the respective CNNs (He et al. 2016). Figure 2.4 shows the architectural breakdown of the VGG19 algorithm in comparison to the 34-layer ResNet (He et al. 2016).

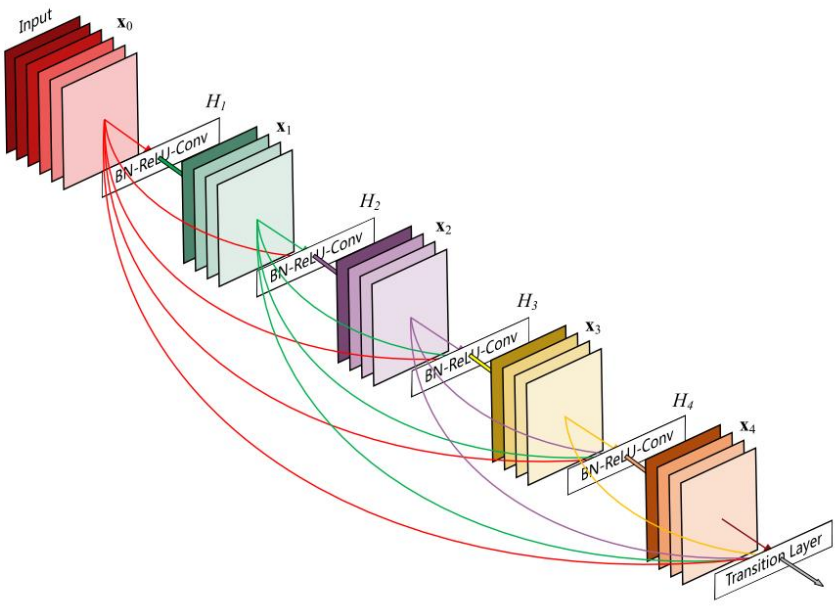


Source: He et al. 2016

Figure 2.4: The architectural structure and breakdown of the VGG19, 34-layer plain network and the 34-layer ResNet

The ResNet architecture uses multiple residual blocks in a forward direction. Unlike other architectures, the ResNet relies upon micro-architecture modules (residual blocks) to construct the CNN (He et al. 2016). The ResNet is also commonly referred to as a network-in-network architecture. ResNet uses a plain 34-layer network inspired by the VGG19 shown in Figure 2.4 (He et al. 2016). Shortcut connections are added to form the residual network (ResNet) with 34 layers. Additional layers may be added to the ResNet architecture, causing an increase in the number of parameters and the depth of the algorithm (He et al. 2016; Ruiz 2018b).

DenseNet was proposed by Huang et al. (2017) as a novel architecture that exploits the shortcut connections of the ResNet. The algorithm directly connects all layers and passes the resulting feature maps through each subsequent layer (Ruiz 2018a; Zhang et al. 2021). DenseNets has multiple deeply connected layers that concatenate the output feature maps with the feature maps of the respective incoming layer (Huang, Liu & Van Der Maaten 2017; Ruiz 2018a; Zhang et al. 2021) (Figure 2.5).



Source: (Huang, Liu & Van Der Maaten 2017)
Figure 2.5: Five dense layer blocks with shortcut connections between each feature map and a growth rate of k=4

2.2.3.2 Applications of CNNs

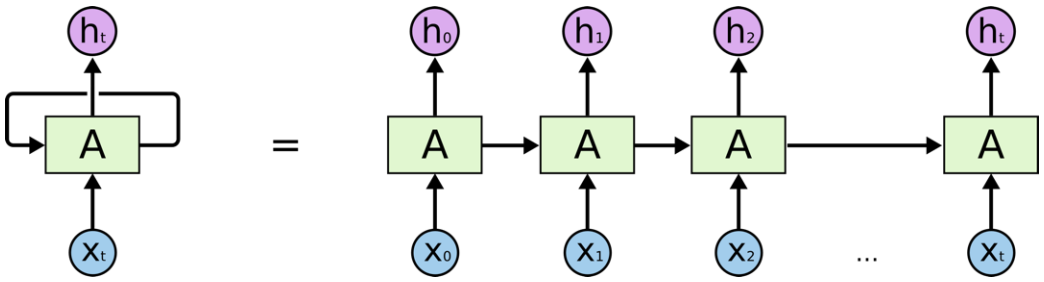
When trained and applied correctly, DL algorithms such as CNN are compelling image-classifiers. CNNs have been implemented across academic fields for numerous applications such as facial recognition, medical image classification and the evaluation of computer vision techniques (Coskun et al. 2017; Feng, Geng & Qin 2020; Hatami, Gavet & Debayle 2018; Wang & Oates 2015). CNNs are considered one of the most effective image classifiers currently deployed, and as a result, RS applications have utilised these algorithms. Chen et al. (2019) illustrated the fast unsupervised DNN's success when performing a change detection using multitemporal SAR

imagery. Land cover maps have effectively been produced using a temporal CNN classification (Pelletier, Webb & Petitjean 2019). Pelletier, Webb and Petitjean (2019) concluded that the temporal CNN could yield an OA of 93% and outperformed the state-of-the-art methods at the time. The findings from Stoian et al. (2019) confirmed those of Pelletier, Webb & Petitjean (2019), where the CNN was the top-performing classifier for land cover mapping using high-resolution imagery (Stoian et al. 2019). Informal settlements have also successfully been detected using CNNs and high-resolution imagery (Mboga et al. 2017). Mboga et al. (2017) illustrated that the DL classifier (CNN) achieved a 91.71% OA and outperformed the SVM ML classifier.

Pre-trained architectures such as ResNet and VGG Networks have been deployed to classify and evaluate 2D encoded coloured images (Yang, Chen & Yang 2020). It was concluded that the simple ResNet could yield satisfactory results for the experiment (Yang, Chen & Yang 2020). A comparative study by Dias et al. (2020) deployed multiple pre-trained CNN algorithms for a pixel-wise Eucalyptus classification. The DenseNet121, DenseNet169, DenseNet201, InceptionV3, InceptionResNetV2, ResNet50, VGG16, VGG19, MobileNetV1 and the XceptionV1 were all processed and evaluated using encoded time-series images (Dias et al. 2020). All pre-trained CNNs yielded high accuracies, with the RenNet50 and DenseNet201 producing the highest result at 97.8% (when using 250 points) and 96.4% (when using 500 points), respectively (Dias et al. 2020). These pre-trained CNN algorithms provide a fast, powerful, and effective classification without extensive training data.

2.2.4 Recurrent neural network (RNN)

RNN is a subclass of neural networks that effectively models sequence data (Pang et al. 2019). The concept of being a neural network implies that RNNs exhibit behaviour similar to that of a human brain. RNNs can produce predictive results within a sequential dataset that most other algorithms cannot (Pang et al. 2019; Wang & Tax 2016; Yin et al. 2017) - forward feeding network process the information by passing it through in one direction. This process starts from the input, continues through the hidden layers and ends at the output layer (Pascanu et al. 2014). The nodes are never touched twice, and information only moves through the architecture in one motion. However, RNNs work differently by processing information through a loop in the architecture (Colah 2015; Zaremba, Sutskever & Vinyals 2014). Decisions are made by considering the current input and what was learnt from the input received previously (Colah 2015). Figure 2.6 displays the RNNs loop and how it equates to the chain structure of repeating modules.



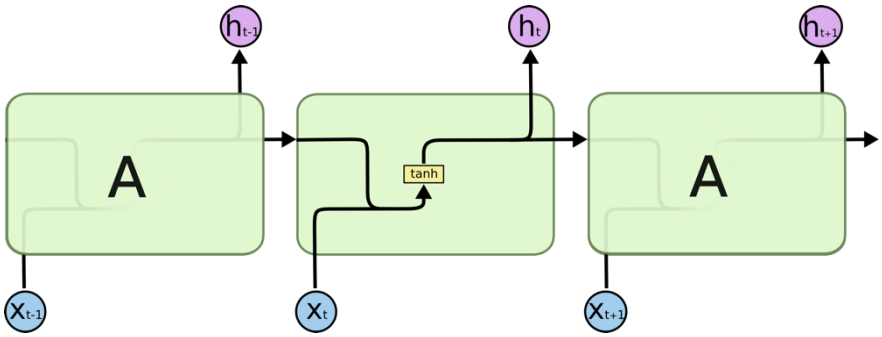
Source: Colah 2015

Figure 2.6: An unrolled RNN that displays the loop as a chain structure

RNN is typically deployed when using sequential data for either speech recognition, visual sequential applications and solving computer vision problems (Karita et al. 2019; Pang et al. 2019; Wang & Tax 2016). RNN algorithms have shown issues regarding the gradients of the model (Donges 2021). Exploding gradients is when the model assigns a high importance weight for a reason (Donges 2021). This can be overcome by squashing or truncating the gradients. Vanishing gradients has also been an issue with RNN, where the loop stops learning because of the low gradient value (Donges 2021; Karim et al. 2017; Zaremba, Sutskever & Vinyals 2014). The issue of vanishing gradients has been solved through the proposed LSTM architecture.

2.2.5 Long short-term memory (LSTM)

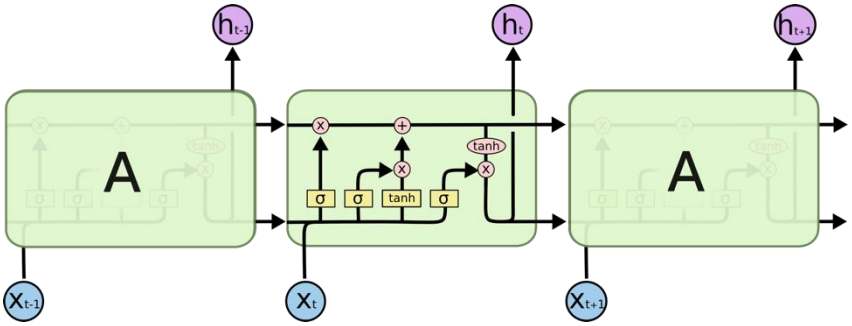
The LSTM architecture is a type of RNN that can learn the memorising long-term dependencies (Colah 2015; Donges 2021; Karim, Majumdar & Darabi 2019). What differentiates RNNs and LSTMs is that LSTM is about recalling past information for more extended periods than that of RNN architecture (Colah 2015; Karim et al. 2017). LSTMs are adequate as they can retain information over time and apply it when ended. LSTMs have a similar chain-like structure to an RNN, however, contain four interacting layers that have unique wat of communicating (Breuel 2015; Colah 2015). Figure 2.7 shows the chain structure of the repeating modules within an RNN.



Source: Colah 2015

Figure 2.7: The chain-like structure of an RNN with the repeating modules

Whereas Figure 2.8 displays the same chain-like structure with repeating modules, however, illustrates the interacting layers found within the LSTM (Colah 2015). LSTMs also avoid the long-term dependency issues that come about from remembering information for too long (Breuel 2015; Colah 2015; Karim et al. 2017). The four interacting layers are the reason for the success of the LSTM over the traditional RNN. The LSTM's ability to remember information over extended periods allows for the effective time-series data process (Breuel 2015; Donges 2021). LSTMs have been known for their high performance and classification of time-series data. The LSTM architecture is the preferred choice for processing time-series data for speed recognition, financial forecasting and weather forecasting(Cao, Li & Li 2019; Karevan & Suykens 2020; Sagheer & Kotb 2019; Soltau, Liao & Sak 2017).



Source: Colah 2015
Figure 2.8: The chain-like structure of repeating modules of the LSTM architecture that contains four unique interacting layers

2.2.6 Computer vision

Computer vision (CV) is a field of AI that deals with how computers gain a high-level understanding from digital videos and images (Szeliski 2011). It seeks to comprehend and automate the tasks that the visual system of humans can do. These tasks include processing, acquiring, analysing, understanding and extracting high-dimensional data to produce symbolic or numerical information. Image classification is a large part of a CV, and DL algorithms have actively been trying to increase the accuracy and effectiveness of the classifications (Szeliski 2011). In CV, there are currently different methods being deployed for processing large image datasets. Initially, each image from a temporal dataset was passed through the algorithm for classification (Hatami, Gavet & Debayle 2018; Pelletier, Webb & Petitjean 2019; Stoian et al. 2019). Change detection would then utilise these classified images and determine where the change has accrued. However, large datasets would result in high computation costs and will require an extensive amount of time. Using a pixel-wise approach, individual time series can be used to represent temporal datasets. The recent developments in the field of CV have shown ways to encode time-series data as 2D images. The reconstructed encoded images can then be passed

through classification algorithms. The idea of combining CV technology with DL classifiers is inspired by the rapid development of these two respective fields. Although LSTM algorithms have shown to be a practical approach for classifying time-series data (Section 2.2.5), CNNs have gained analysts' interest for their exceptional image classification performance. Experimental studies have been deployed to investigate the methodology of encoding time-series data as 2D images for CNN classifications (Dias et al. 2020; Wang & Oates 2015; Yang, Chen & Yang 2020).

2.2.6.1 Encoding time-series data

Various transformations have been proposed in CV for encoding time-series data as 2D images (Yang, Chen & Yang 2020). This is in the hope that the resulting encoding 2D images can reveal additional features and patterns than the original one-dimensional (1D) (Yang, Chen & Yang 2020). The two popular encoding transformations include the GAF and the MTF (Wang & Oates 2015). Each of these transformations follow a unique procedure to compute a matrix that represents the encoded images.

The first step of the GAF is to rescale (normalise) the original input time series to a [-1, 1] scale factor. The time series $X = \{x_1 x_2, x_3, \cdots, x_n\}$ with n real observations will be passed through Equation 2.7 to normalise the data (Wang & Oates 2015).

Equation 2.7:

$$\tilde{x}_i = \frac{(x_i - max(X) + (x_i - min\,(X))}{max(X) - min\,(X)}$$

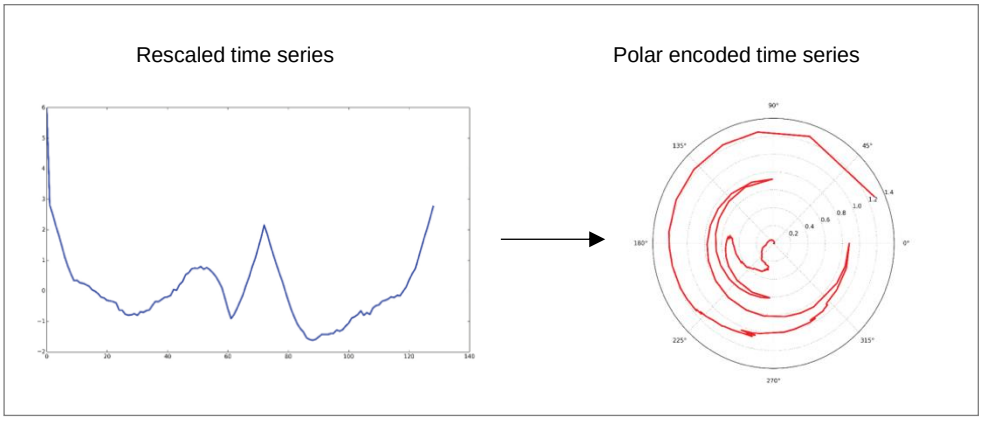
Following on from this, the rescaled time series $\tilde{X}$ can now be represented in polar coordinates. This is done by encoding the rescaled time series values as the angular cosine and the time stamp as the respective radius (Wang & Oates 2015). Equation 2.8 illustrates the polar encoding process of the time-series data.

Equation 2.8:

$$\begin{cases} \emptyset = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases}$$

The time stamps are represented by $t_i$in the equation. The constant factor *N* helps to regularise the span aspect of the polar coordinate system (Wang & Oates 2015). This polar encoding process is a novel way to view and understand the time series (Wang & Oates 2015). Figure 2.9 visually

represents the rescaled time series and the respective polar encoded data shown on a polar coordinate system.



Source: Wang & Oates 2015
Figure 2.9: Rescaled time series and the respective polar-encoded dataset displayed on a polar coordinate system

Using the transformed polar-encoded time-series data, the angular perspectives are exploited by considering trigonometric difference/sum between each point to identify and capture the temporal correlation between different time intervals (Dias et al. 2020; Wang & Oates 2015). The GAF matrix formed (from top-left to bottom-right) corresponds to the original time-series data and is symmetrical along the main diagonal (Yang, Chen & Yang 2020). The GAF can generate two different images using the separate equations (2.9 and 2.11) and the unit row vector. Equations 2.9 and 2.10 illustrate the GASF where Equations 2.11 and 2.12 illustrate GADF (Dias et al. 2020; Wang & Oates 2015; Yang, Chen & Yang 2020).

Equation 2.9:

$$GASF = \begin{pmatrix} \cos(\emptyset_1 + \emptyset_1) & \cdots & \cos(\emptyset_1 + \emptyset_n) \\ \cos(\emptyset_2 + \emptyset_1) & & \cos(\emptyset_2 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_n + \emptyset_1) & \cdots & \cos(\emptyset_n + \emptyset_n) \end{pmatrix}$$

Equation 2.10:

$$GASF = \tilde{x}' \cdot \tilde{x} - \sqrt{I - \tilde{x}^2}' \cdot \sqrt{I - \tilde{x}^2}$$

Equation 2.11:

$$GADF = \begin{pmatrix} \sin(\emptyset_1 + \emptyset_1) & \cdots & \sin(\emptyset_1 + \emptyset_n) \\ \sin(\emptyset_2 + \emptyset_1) & & \sin(\emptyset_2 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \sin(\emptyset_n + \emptyset_1) & \cdots & \sin(\emptyset_n + \emptyset_n) \end{pmatrix}$$

Equation 2.12:

$$GADF = \sqrt{I - \tilde{x}^2}' \cdot \tilde{x} - \tilde{x}' \cdot \sqrt{I - \tilde{x}^2}$$

The MTF transformations use transitional probability statistics to preserve detail within the time domain (Yang, Chen & Yang 2020). The original time series are discretised by being split into quantile bins and are followed by the construction of the Markov transition matrix. The Markov transition probabilities M_{zx} of the quantile bin q_z moves the q_x, for the time stamps at z and x, respectively (Dias et al. 2020; Wang & Oates 2015; Yang, Chen & Yang 2020). When time series $X = \{x_1 x_2, x_3, \cdots, x_n\}$ and $Q = \{q_1 q_2, q_3, \cdots, q_n\}$, the size of Q will have an effect on the Markov transition matrix (w) size (Wang & Oates 2015; Yang, Chen & Yang 2020). The MTF is illustrated in Equation 2.13

Equation 2.13:

$$M_{zx} = \begin{pmatrix} w_{zx}|x(1) \in q_z, x(1) \in q_x & \cdots & w_{zx}|x(1) \in q_z, x(n) \in q_x \\ w_{zx}|x(2) \in q_z, x(1) \in q_x & & w_{zx}|x(2) \in q_z, x(n) \in q_x \\ \vdots & \ddots & \vdots \\ w_{zx}|x(n) \in q_z, x(1) \in q_x & \cdots & w_{zx}|x(n) \in q_z, x(n) \in q_x \end{pmatrix}$$

The three different encoded transformations (GASF, GADF and MTF) are used to generate 2D images from 1D time-series data (Wang & Oates 2015). An example of each of the encoded images can be seen in Figure 2.10. The resulting 2D images are then classified using any image classifying algorithms desired by the analyst.

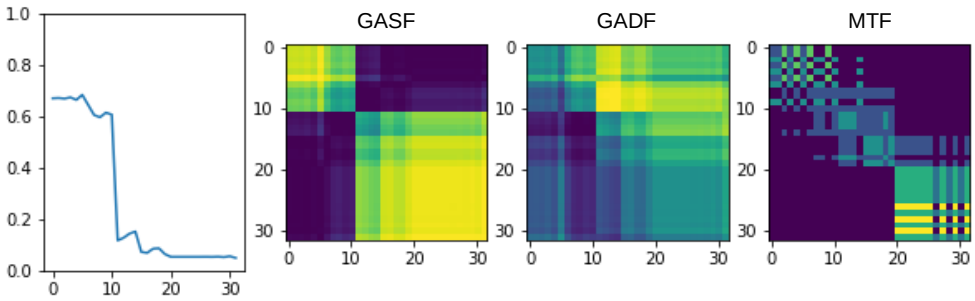


Figure 2.10: Rescaled time series and its respective GASF, GADF and MTF encoded images

The methodology of encoding time-series data through the GAF and MTF for image classification was proposed by Wang and Oates (2015). After this, only a handful of applications have adopted this novel framework (Dias et al. 2020; Yang, Chen & Yang 2020; Yuan et al. 2021). It has only recently gathered traction and has started being implemented. However, RS and specifically change detection applications have not had significant exposure to this unique methodology. The study by Dias et al. (2020) is one of the few studies that have implemented encoding of time-series data using GADF, GASF and MTF for a CNN classification. Dias et al. (2020) utilised the novel framework for a pixel-wise Eucalyptus region classification and achieved high accuracies using pre-trained CNN architecture.

2.3 LITERATURE SUMMARY

The literature reviewed shows that combining RS, DL, and computer vision holds great potential for urban change detection. Traditional RS techniques and methodologies have been used to effectively process imagery and perform land cover classification, and in particular urban land cover classification and change detection. Non-parametric classifiers, such as ML algorithms, DTs, SVM, ***k***-nearest neighbours (***k***-NN), RF and ANN have become popular RS methods for classification. An essential attribute in deriving a change detection is the use of multi-temporal data. Time-series data can therefore provide additional information required for performing a successful change detection.

Recent DL developments have shown that CNN algorithms are extremely powerful image classifiers when trained correctly. As a result, a novel framework was proposed to combine methods from computer vision in RS classification and change detection. Input derived by first encoding temporal data as 2D images are processed through a CNN classification. This demonstrates how the advancement of scientific knowledge often requires collaboration between fields. Although several other time-series methodologies have been implemented in South Africa to perform urban change detection, the novel framework of encoding time series data as a 2D image processed through multiple CNN architectures has not been applied for urban change detection.

CHAPTER 3: THE VALUE OF A NOVEL COMPUTER VISION BASED CHANGE DETECTION TECHNIQUE FOR URBAN AREAS USING COARSE RESOLUTION IMAGERY

3.1 INTRODUCTION

Human settlement expansion is one of the most pervasive forms of land cover change worldwide, as in South Africa (Kleynhans, Salmon & Wessels 2017). Due to population growth, economic and employment opportunities, urban areas are rapidly expanding (Kleynhans et al. 2013), encroaching on the natural environment (Kleynhans et al. 2015). Informal expansion coupled with unplanned developmental activities are treated reactively and place a burden on infrastructure services (Nassar & Elsayed 2018). Timely and accurate change information in the urban environment is therefore essential for successful planning and management (Jensen & Im 2007). Consequently, change detection is a crucial step for analysing temporal EO sequences. Time-series satellite remote sensing has provided a consistent source for change detection over space and time (De Beurs & Henebry 2005; Chen et al. 2019; Hu & Ban 2014; Liu et al. 2018; Lunetta et al. 2006; Verbesselt, Hyndman, Newnham, et al. 2010). Furthermore, the use of hyper-temporal satellite data alongside time-series analyses has successfully been applied in South Africa for land cover change detection (Grobler et al. 2012; Grobler et al. 2013; Kleynhans et al. 2013; Kleynhans et al. 2012; Kleynhans et al. 2015; Kleynhans, Salmon & Wessels 2017; Salmon et al. 2013). To exploit time-series data available through EO, a temporal autocorrelation change detection (TACD) method was used to detect new settlements in areas typically covered by natural vegetation in South Africa (Kleynhans et al. 2012). An advancement of this method was then found in Kleynhans et al. (2013), where a spatio-temporal autocorrelation change detection (STACD) was performed. The proposed STACD method is based on the premise of a TACD, however uses a per-pixel autocorrelation change index with that of the neighbouring pixel index to increase performance. The pixel-based temporal function could be improved by 17% when considering spatial autocorrelation (Kleynhans et al. 2015). The stability of the time-series means variance over time, when compared to a threshold value, is used as a measure of per-pixel land cover change (Kleynhans et al. 2015). The pixel-based TACD and STACD method uses an autocorrelation Function (ACF) applied to a normalised difference vegetation index (NDVI) time series derived from moderate resolution imaging spectroradiometer (MODIS) 250 m and 500 m imagery. NDVI has proved successful in change detection studies (Hu, Dong & Batunacun 2018; Kleynhans et al. 2011; Lunetta et al. 2006; Usman et al. 2015).

Due to the requirement of large amounts of training data, ML algorithms for image analysis have not been used extensively for change detection (Daudt et al. 2018) and have mostly been designed

to generate a different image to which a manual threshold is applied (Zhan et al. 2017). Leveraging techniques and insights brought by developments in computer vision, Wang and Oates (2015) developed a novel framework to encode time-series data as an image to enable ML to recognise and classify the time series. Each time series is encoded as an individual image using a gramian angular field (GAF) and a Markov transition field (MTF) before classification using a tiled convolutional neural network (CNN) (Wang & Oates 2015). Captured into an image, GAF represents a time series of observations in a matrix containing the temporal correlation between observations in different time intervals, while MTF represents transition probabilities (Dias et al. 2019). CNN is a leading ML classifier in image recognition that uses 2D image data as input (Abdel-Hamid, Mohamed & Jiang 2014; Szegedy et al. 2015). This image-based framework considers image transformation using Gramian angular summation field (GASF), Gramian angular difference field (GADF), and MTF encoding as a feature engineering technique for DL approaches (Yang, Chen & Yang 2020). Dias et al. (2019) showed that GASF/GADF encoding preserved temporal relationships, while MTF captured transition probabilities among different time-series states. They successfully transformed time-series data to encoded images before the classification, which made use of 11 different pre-trained deep-learning-based feature extractors to identify Eucalyptus (Dias et al. 2019). Dias et al. (2019) established that the best-performing transformation (GASF, GADF, or MTF) was a combination of the three. A concatenated image was formed by combining the encoded images from each of the transformations before the deep-learning-based classifications (Dias et al 2019). Similarly, Yang, Chen and Yang (2020) evaluated the impact of using GASF/GADF and MTF transformation methods on multivariate time-series as well as the sequences of concatenating images on classification accuracy using a CNN. With the advancement of DL and CNNs dominating the field as a state-of-the-art classifier, it has successfully been applied to various applications (Barra et al. 2020; Hatami, Gavet & Debayle 2018; C Li et al. 2020; Mboga et al. 2017; Stoian et al. 2019; Yang et al. 2019). There are currently several different architectures that all perform at a high level using a residual learning framework (He et al. 2016; Huang et al. 2017). With the development of the ImageNet database, 1.2 million images are used to pre-train different CNN architectures (He et al. 2016). A dense convolutional network was proposed by Huang et al (2017), where each layer is a connected layer in a feed-forward fashion. These DenseNets are also contained with a shorter connection between layers at the output and inputs of the network (Huang et al. 2017). The residual learning framework has shown promising results (Dias et al. 2020; Yang, Chen & Yang 2020).

This paper presents a novel supervised DL method applied to the change detection problem of human settlement expansion in South Africa. MODIS NDVI time-series data are encoded as 2D images using GASF, GADF, and MTF transformations. State-of-the-art CNN architectures are

then trained using this data after which classification is performed. To assess the influence of resolution, a comparison is made between two MODIS resolution datasets (250 m and 500 m). This research will illustrate the use of a DL apparatus for pixel-based change detection using encoding techniques alongside feature extraction from time-series data.

3.2 DATA DESCRIPTION

3.2.1 MODIS data

Pixel-based time series were derived from the MODIS instrument. NDVI was computed to obtain the time-series values. Two datasets were collected for the period ranging from 2001/01/01 to 2009/02/28 (2979 days) over the Gauteng Province, South Africa (Kleynhans, Salmon & Wessels, 2017). The first dataset was obtained from the MCD43A4 Daily 500 m product (USGS 2018). The near-infrared (NIR) (867-876 nm) and red (620-670 nm) bands were used to calculate the NDVI values for each image as (NIR - Red) / (NIR + Red) (Lunetta et al. 2006). Due to hardware restrictions, the temporal resolution was reduced to six-daily to decrease the dataset size from 2 976 to 501 images (Table 3.1). The second dataset was obtained from the MOD13Q1 16-day 250 m NDVI product (USGS 2018). It has a lower temporal resolution (16-daily) and a higher spatial resolution of 250 m (Table 3.1). Both the 250 m and 500 m MODIS datasets contained training samples from all three classes. The 250 m dataset consisted of 866 no-change and 433 changed pixels, whereas the 500 m dataset contained 244 no-change and 122 changed pixels (Table 3.1). Both these datasets were split into training and validation samples using a 70:30 ratio. The testing pixels were used to perform an image classification and change detection.

Table 3.1: Dataset properties for the two datasets, MODIS 500 m and 250 m

	MODIS 500 m	MODIS 250 m
Resolution	500 m	250 m
Band	NDVI	NDVI
Temporal resolution	6-daily	16-daily
Time-series length	501 images	206 images
No-change pixels	244	866
Change pixels	122	433
Model training pixels [70%]	255	909
Model validation pixels [30%]	110	390
Model testing pixels	1050	5712

A representation of a single NDVI pixel from the two datasets can be seen in Figure 3.1. All data were collected and extracted using the Google earth engine (GEE) platform as it is freely available (Gorelick et al. 2017). Due to cloud cover, the missing values in the time-series were filled using Slinear interpolation. It should be noted that the study area does not have prolonged periods where there is extensive cloud cover.

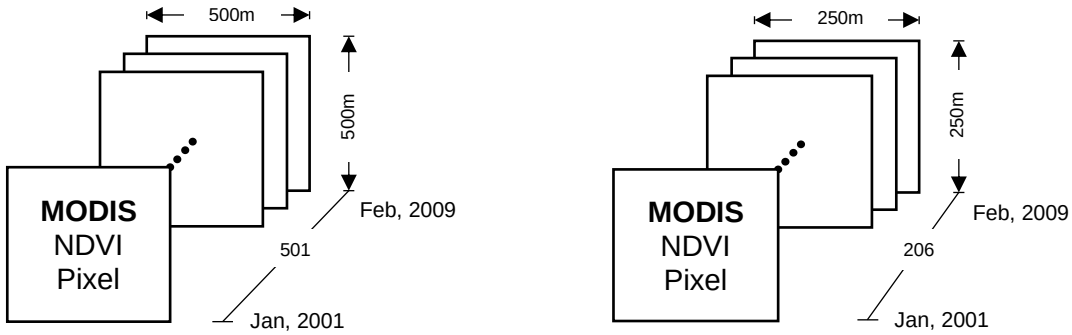


Figure 3.1: Single-pixel representation for 250 m and 500 m NDVI time-series datasets

3.2.2 Study area

In the northern part of South Africa, the Gauteng province has seen high levels of urbanisation. During the period from 2001 to 2009, there was significant human settlement expansion. Illustrated in Figure 3.2 is the location of the Gauteng province, and the ground truth samples (GTS). The GTS data is used for training and validating the CNN models. The area represented by the red square, known as the testing site, is the location where an image classification and change detection will be conducted.

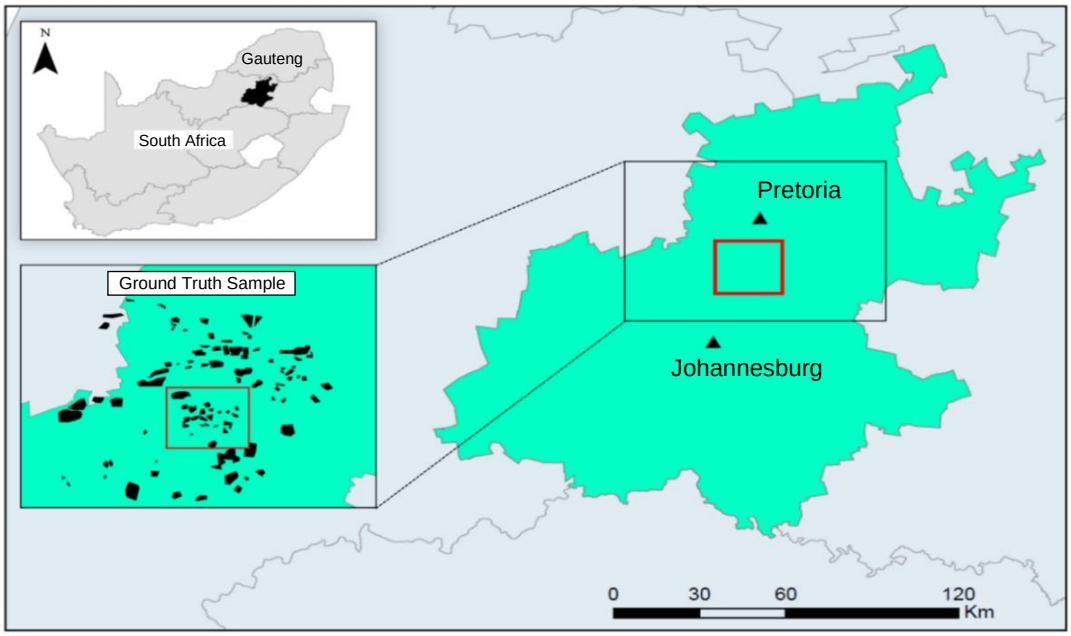


Figure 3.2: Provincial map showing the location of the Gauteng province in South Africa. Zoomed in map showing the location of the Study area where MODIS data was collected and used to train the models

Focusing on the area around Pretoria with the black square in Figure 3.2, the GTS were established. This was done using Landsat 7 ETM+ data from 2001 to 2008 in GEE. Three classes of data were digitised. The first two classes consisted of “no-change” pixels for the urban and vegetation areas. The urban class represented urban pixels that did not change from 2001 to 2009, where the vegetation class represented any vegetation type that remained vegetation for that duration. The third class represented changed pixels corresponding to pixels that change from vegetation to urban in the period from 2001 and 2008.

Two high-resolution Quickbird images from Google Earth clearly show the urban settlement encroaching on natural vegetation from 2001 to 2009 (Figure 3.3). The MODIS 500 m dataset is illustrated by images 1 and 2 overlaid by a 500 m MODIS pixel grid, while images 3 and 4 display the same high-resolution images overlaid with the 250 m pixel grid.

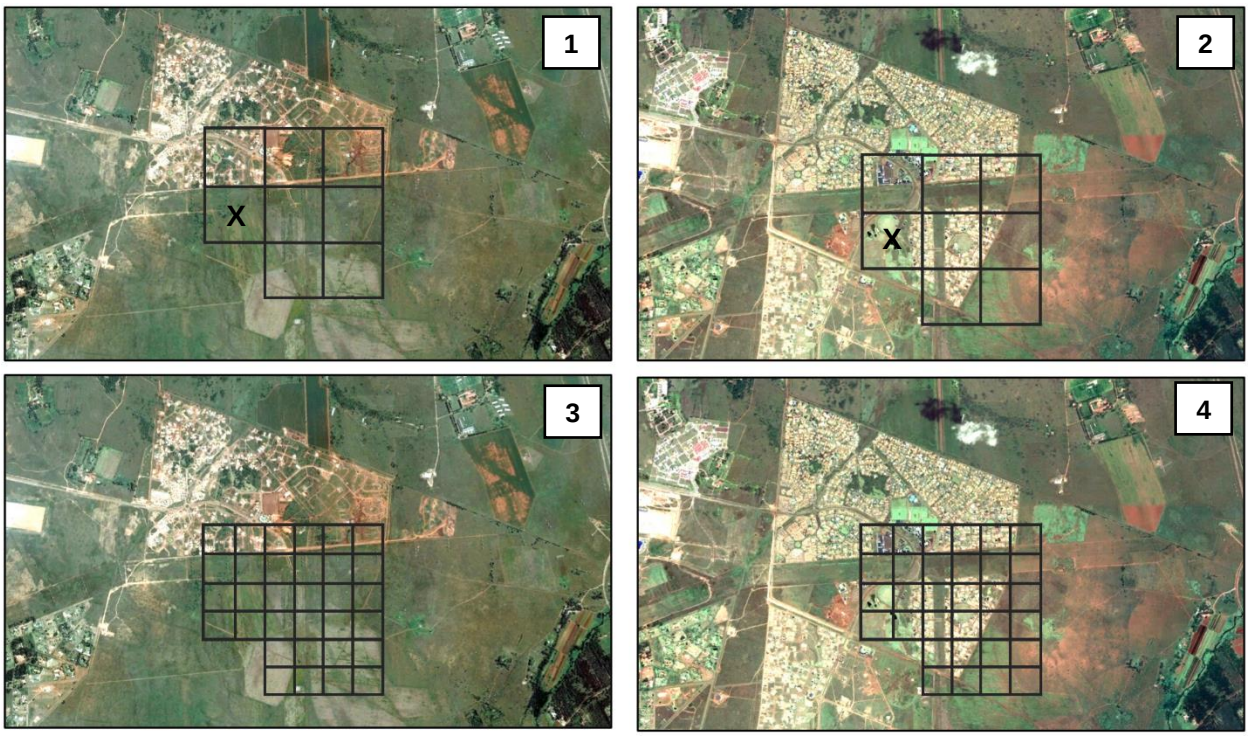


Figure 3.3: Google Earth Quickbird imagery showing urban expansion overlaid with 250 m and 500 m resolution MODIS grids. Images (1) and (2) correspond with 500 m pixels and images (3) and (4) illustrate the 250 m pixels

Each pixel has a corresponding time series that is made up of NDVI values. Figure 3.4 illustrates a “changed” and “no-change” pixel-based time series from 2001 and 2009. The NDVI values range from -1 to 1. The time series that represents “change” in Figure 3.4 corresponds to a pixel that underwent change between 2005 and 2006. An example of this can be seen by pixel X in Figure 3.3. However, the second time series of “no-change” demonstrates a pixel that remained constant throughout the duration. The abrupt change that occurs in a pixel will be represented by that

respective pixel's time series. Through the use of encoding techniques and state-of-the-art feature extractors, these time series will be used to perform a change detection.

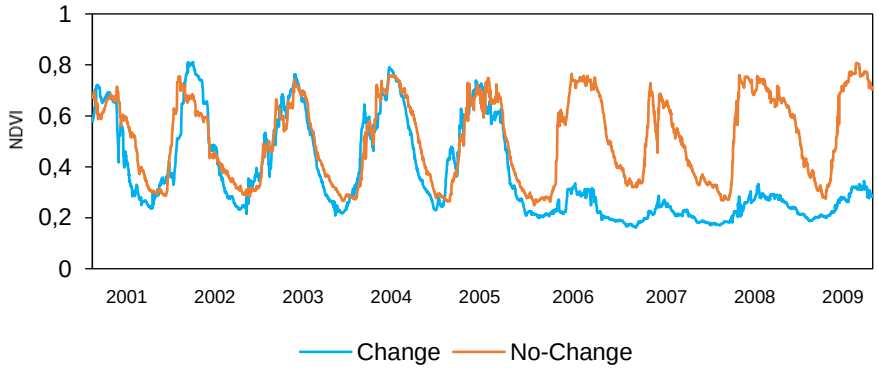


Figure 3.4: MODIS NDVI time series of the changed and no-change pixels over nine years

3.3 METHODS

In this study, pixel-based time series derived by 500 m and 250 m MODIS imagery were used. The long short-term memory (LSTM) classification algorithm was applied to the original time series as a baseline to compare the effect of the encoding framework. Encoding of the original time series was performed using GADF, GASF, and MTF transformations. Additionally, a concatenated image was created using all three encoding techniques. These datasets were used as the inputs to 11 feature extractors. A validation process was implemented and the results were compared. The top-performing feature extractor was selected and an image classification was conducted. An accuracy assessment was used to verify the classification and change detection.

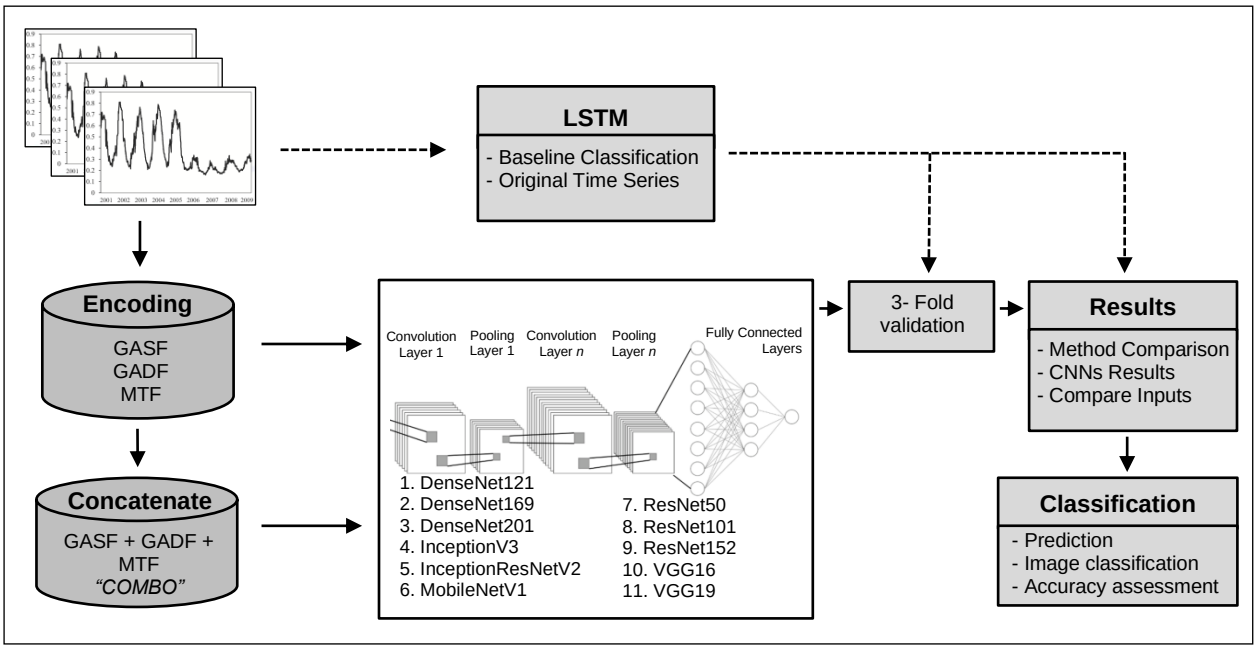


Figure 3.5: Workflow diagram illustrating the process of performing a change detection using encoding transformations with feature extractors

3.3.1 Baseline LSTM classification

The LSTM was used as the baseline recurrent neural network (RNN) classification and was conducted on the original time series. RNN algorithms perform well on sequential data such as time series due to their architecture (Karim, Majumdar & Darabi 2019; Tan et al. 2019). An RNN contains layers that are connected with a recurring feed that allow it to remember information as it moves through the layers (Yin et al. 2017). For the baseline classification with the LSTM algorithm, 365 time series extracted from 500 m resolution MODIS NDVI data were selected. Model parameters were set up in Python and connected to the Keras application programming interface (API) to allow an LSTM architecture to run (Keras 2020b). The model was trained on 70% of the data and validated on the remaining 30%. The LSTM model used a Softmax activation function and contained three dense layers. Categorical cross-entropy was used as a loss function, with a batch size of 1, and an epoch of 15. LSTM has proven to be an effective and state-of-the-art time-series classifier (Graves & Schmidhuber 2005; Karim et al. 2017; Karim, Majumdar & Darabi 2019; Tan et al. 2019; Yildirim et al. 2019). As a result, the LSTM was chosen as a baseline classifier for the original time series before encoding.

3.3.2 Encoding time series as images

Gramian Angular Field (GAF) and Markov Transition Field (MTF) pixel-wise transformations were executed for each time series in both datasets (250 m and 500 m). GAF uses a polar coordinates-based matrix to represent the temporal correlation between observations in different time intervals (Dias et al. 2019; Wang & Oates 2015; Yang, Chen & Yang 2020). The original time-series per pixel, consisting of NDVI values scaled between [-1, 1] was converted to polar coordinates. Two quantities were considered in the encoding to GAF, the scaled NDVI (x) and its corresponding timestamp (i). Variable x was expressed as an angle, computed by $arccos(x)$, whereas the radius represents the relative position in the time series with length N (Dias et al. 2019).

Using the polar encoded results, GASF and GASD matrices were created. The GASF is based on cosine functions and GADF is based on sine functions (Yang, Chen & Yang 2020) as can be seen in equations 3.1: (A) and (B).

Equation 3.1

$$A) \quad GASF = \begin{pmatrix} \cos(\emptyset_1 + \emptyset_1) & \cdots & \cos(\emptyset_1 + \emptyset_n) \\ \cos(\emptyset_2 + \emptyset_1) & & \cos(\emptyset_2 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_n + \emptyset_1) & \cdots & \cos(\emptyset_n + \emptyset_n) \end{pmatrix}$$

$$B) \quad GADF = \begin{pmatrix} \sin(\emptyset_1 + \emptyset_1) & \cdots & \sin(\emptyset_1 + \emptyset_n) \\ \sin(\emptyset_2 + \emptyset_1) & & \sin(\emptyset_2 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \sin(\emptyset_n + \emptyset_1) & \cdots & \sin(\emptyset_n + \emptyset_n) \end{pmatrix}$$

GASF and GADF matrices used trigonometric summation and difference of $\emptyset_i$ for each point in the time series. The matrices were then converted into colour images for display purposes, as shown in Figure 3.6 (Dias et al. 2019; Yang, Chen & Yang 2020).

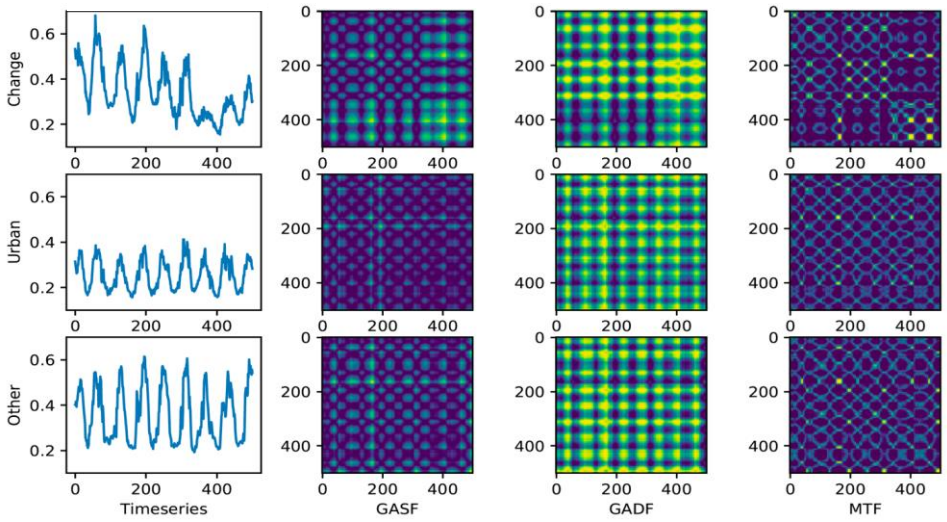


Figure 3.6: The respective GASF, GADF, and MTF encoded colour image for each of the time-series classes (Change, Urban, and Other)

MTF captures transitional probabilities sequentially to preserve information in the time domain. (Dias et al. 2019; Wang & Oates 2015). The time series was discretised by splitting into quantile bins, after which a Markov transition matrix was built, from which transition probabilities were computed and W represented the transitions between quantile bins. The final MTF represents Markov transition probabilities sequentially in an N x N matrix, where N represents the total number of elements in the time series. The encoded Markov transition statistics were then also represented as colour images, illustrated in Figure 3.6.

To increase the extraction of discriminative features through CNNs, a set of concatenated images were formed by combining the GASF, GADF, and MTF before feeding the CNNs (Yang et al. 2019). The CNNs deployed have three input data channels, red green blue (RGB) each require information. Each transformation (GASF, GADF, and MTF) was assigned to the respective RGB

input channel from the concatenated image. The three transformations were constructed in such a way to match the input requirements of the CNNs. The concatenated images were called $COMBO_{500}$ and $COMBO_{250}$ for the MODIS 500 m and 250 m resolution datasets respectively.

3.3.3 Deep-learning feature extractors

Data-driven feature extractors find effective representations without specific knowledge of the application (Dias et al. 2019). Designers of the CNNs exploit the properties that initial layers have for computing simple patterns like edge gradient or other simple patterns, however, this is done without the need for specific knowledge (Dias et al. 2020; Wang & Oates 2015). Eleven CNN architectures pre-trained on ImageNet were used as feature extractors (Fei-fei et al. 2021; Keras 2020a). There are over 1.4 million images in the ImageNet database that are used to pre-train the feature extractors responsible for performing the image classifiers (Fei-fei et al. 2021). These feature extractors consist of several max-pooling and convolutional layers. The training process allows the feature extractor to assign weights to layers that can be later implemented for a second task, such as a classification. The architectural structure of these pre-trained CNNs can be found in Section 2.2.3.1. There are six main architectures (DenseNet, InceptionV3, InceptionResNetV2, MobileNet, ResNet, VGG) that all have a unique structure and process the flow of information differently. Each one of these architectures has several variations that use different amounts of convolutional and pooling layers. The top-performing architecture was then used to conduct a classification. The 11 architectures considered were DenseNet121, DenseNet169, DenseNet201 (Huang et al. 2017), InceptionV3 (Szegedy et al. 2016), InceptionResNetV2 (Längkvist, Karlsson & Loutfi 2017), MobileNetV1 (Howard et al. 2017), ResNet50 (Kaiming et al. 2015), ResNet101, ResNet152 (Keras 2020a), VGG16, VGG19 (Simonyan & Zisserman 2015).

All CNNs were trained and validated with a 70:30 split for each encoder (GASF, GADF and MTF) using both 250 m and 500 m resolution datasets. These three transformations datasets were then used as the input data for each feature extractor using the 500 m resolution MODIS data. Additionally, the $COMBO_{500}$ concatenated dataset was processed through all feature extractors. However, only the $COMBO_{250}$ concatenated images were used as the input dataset for the 250 m resolution MODIS data due to the high performance of the $COMBO_{500}$.

3.3.4 Model evaluation protocol

The same evaluation protocol was applied for all experiments at both 250 m and 500 m resolution datasets: all input data were split into training (70%) and validation sets (30%) (Table 3.1). A three-fold cross-validation was run for all feature extractors, as well as the baseline RNN model.

3.3.5 Image classification

The best-performing feature extractor was selected to classify the testing site to perform a change detection (Figure 3.2). The $COMBO_{250}$ dataset was used to train the top-performing feature extractor. The time-series extracted from 5 712 pixels of 250 m resolution MODIS pixels were encoded using the encoding framework (Section 3.3.2). Using the trained feature extractor Densenet121, each pixel was classified based on the probability of being assigned to a class. The prediction function produced three confidence values per pixel. Each value represents the probability of that respective pixel being assigned to each of the three classes (urban, vegetation, change). The pixel was then assigned to the class with the highest probability, resulting in a three-class classified image. A simplified binary classification image was created to show change and no-change pixels from 2001 to 2009. The result of the classification was a change detection map that illustrated that pixel change occurred.

3.3.6 Classification evaluation protocol

To verify the success of the classification, an accuracy assessment was implemented (Dervisoglu, Bilgilioglu & Yagmur 2020). The first step was to create stratified random points over the binary classification. The stratified random sampling strategy creates points that are randomly distributed over each class (Stehman 1996). However, the number of points created was proportional to the relative area of the class. Seventy randomly distributed points were created within the “no-change” class and 51 within the “change” class. Each point was verified using a 2001 and 2009 Landsat 7 image to determine whether change occurred at that location. Using the predicted and actual classes a confusion matrix was created. Overall accuracy (OA), user’s accuracy (UA) and producer’s accuracy (PA), Kappa, and positive predictive power were formulated (Dervisoglu, Bilgilioglu & Yagmur 2020). The OA represents the percentage of cases correctly allocated to their respective classes while the family of kappa indices have traditionally been used to accommodate for the effects of chance agreement (Pontius & Millones 2011). The positive predictive power, also called precision, is a metric that quantifies the number of correct positive predictions made and was computed by dividing true positives by the sum of true positives and false positives. True positives are data points classified as “change” by the model that are actual “change”, while false positives are data points classified as “no-change” that belong to the “change” class (Juba & Le 2019). Precision can be thought of as a measure of a classifier’s exactness. A low precision can also indicate many false positives.

3.3.7 Robustness

To test the robustness and generalisability of the model, new unseen data from a different location was introduced. The highest-performing feature extractor (DenseNet121) was selected and trained with the $COMBO_{250}$ dataset from Pretoria. Assessing the generalisability of the model, unseen data from Maputo was introduced. This data consisted of a 250 m resolution MODIS NDVI pixel-based time series that was then encoded using the same technique applied in Section 3.2. The concatenated images from Maputo formed a dataset that was used to evaluate the robustness of the model.

3.4 RESULTS

3.4.1 Training the classifiers

The performance of all 11 CNN classifiers trained on each of the input transformations of the MODIS 500 m resolution dataset (GASF, GADF and MTF) as well as the combined concatenated image was evaluated on overall accuracy and prediction error (loss function). Table 3.2 shows the average balanced accuracy and loss for the classifiers with corresponding results for the four different input datasets (GASF, GADF, MTF, and COMBO).

Table 3.2: Accuracy and loss assessment for CNNs with highest-performing classifier per input dataset highlighted (MODIS 500 m)

	GASF		GADF		MTF		$COMBO_{500}$	
MODIS 500 m	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
ResNet50	0.9061	0.2353	0.9242	0.2546	0.8970	0.3706	0.9242	0.2371
ResNet101	0.8909	0.3524	0.9061	0.2223	0.9152	0.2610	0.9364	0.1847
ResNet152	0.9000	0.2742	0.9045	0.2729	0.9091	0.2715	0.9333	0.1770
DenseNet121	0.9000	0.3086	0.9212	0.2475	0.8364	0.3675	0.9394	0.1720
DenseNet169	0.9091	0.2075	0.9091	0.2682	0.8606	0.4064	0.9282	0.1547
DenseNet201	0.9182	0.2325	0.9212	0.2596	0.9061	0.3145	0.9333	0.1728
InceptionV3	0.8879	0.3733	0.9061	0.2858	0.8818	0.4029	0.8697	0.4552
InceptionResNetV2	0.8727	0.3202	0.8939	0.2835	0.8273	0.4930	0.9091	0.2371
VGG16	0.8455	0.4243	0.8939	0.3830	0.8091	0.4090	0.9000	0.3132
VGG19	0.8727	0.4100	0.9000	0.3077	0.8667	0.3525	0.9152	0.2650
MobileNetV1	0.8455	0.6421	0.8970	0.2719	0.8515	0.4860	0.8636	0.3016
Mean	0.8862	0.3436	0.9070	0.2779	0.8692	0.3759	0.9139	0.2428

When using the GADF encoded images as input data, ResNet50 had the highest effective accuracy (92.42%), whereas GASF and MTF encoders performed lower at 91.82% and 91.52% respectively using the DenseNet201 and ResNet101 models. The concatenated input images (COMBO) trained on the DensesNet121 classifier was the highest-performing model with an accuracy of 93.94%, making it the single most effective classifier. The highest mean accuracy over the 11 CNN classifiers of 91.39% was also achieved for the concatenated (COMBO) input dataset.

Similarly, Table 3.3 shows the effective performance of the 11 CNN classifiers applied to the best-performing input dataset from Table 3.2, the COMBO concatenated dataset, using MODIS 250 m resolution. The COMBO concatenated input dataset consists of GASF, GADF and MTF transformed images from the 250 m resolution dataset. The mean and standard deviation scores for the accuracy and loss for each classifier are illustrated in Table 3.3

Table 3.3: Accuracy and loss assessment for all CNNs using the COMBO250 concatenated input images for the MODIS 250 m resolution dataset

	Accuracy			Loss		
MODIS 250 m	*Mean* ± *Stdev*			*Mean* ± *Stdev*		
ResNet50	0.9427	±	0.0090	0.1666	±	0.0139
ResNet101	0.9419	±	0.0090	0.1883	±	0.0358
ResNet152	0.9273	±	0.0259	0.2181	±	0.0682
DenseNet121	0.9444	±	0.0053	0.1897	±	0.0252
DenseNet169	0.9308	±	0.0089	0.2008	±	0.0472
DenseNet201	0.9350	±	0.0097	0.2387	±	0.0212
InceptionV3	0.9419	±	0.0039	0.1635	±	0.0244
InceptionResNetV2	0.9273	±	0.0131	0.2091	±	0.0722
VGG16	0.9410	±	0.0044	0.2081	±	0.0168
VGG19	0.9333	±	0.0092	0.2464	±	0.0212
MobileNetV1	0.9256	±	0.0068	0.2304	±	0.0224
Mean	0.9356	±	0.0096	0.2054	±	0.0335

As with the 500 m resolution input data (Table 3.2), the DenseNet121 model applied to the higher resolution 250 m MODIS imagery had the highest mean effective performance accuracy of 94.44± 0.53%. Figure 3.7 presents the comparison of the effective performances of the CNN classifiers using the COMBO concatenated images as input for both the MODIS 250 m and 500 m resolution datasets.

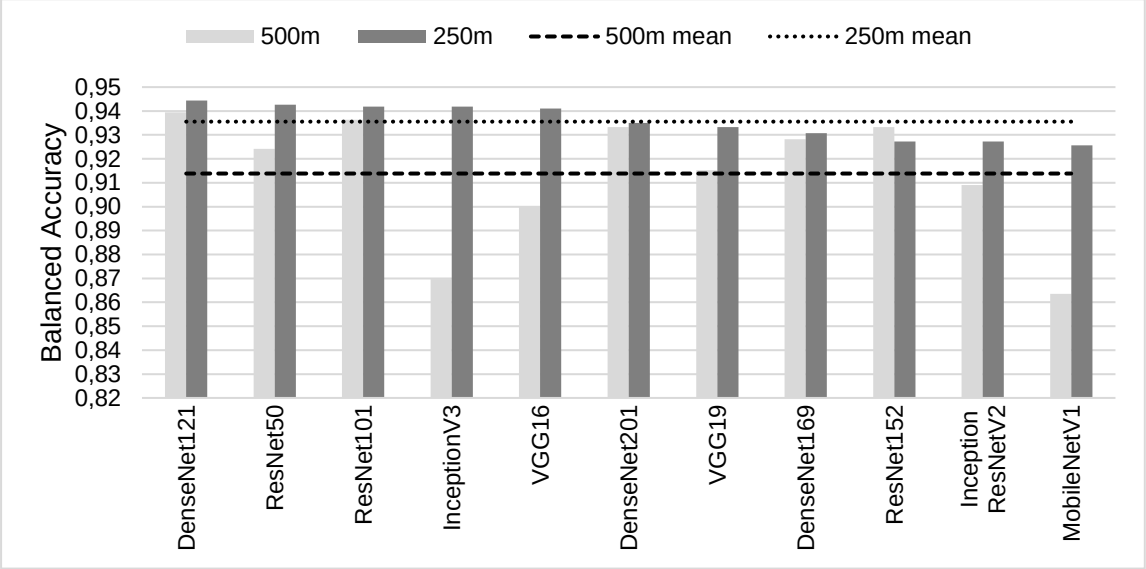


Figure 3.7: Comparison of classifiers based on resolution

As shown in Table 3.2 and 3.3, DenseNet121 is the highest-performing classifier when using COMBO images as the input data. DenseNet121 had an increase in effective performance from 93.94% to 94.44% when the MODIS resolution was increased from 500 m to 250 m. Table 3.3 also shows the mean accuracy of the classifiers for each dataset. The increased resolution resulted in a 2% increase in the performance accuracy of the CNN models.

3.4.2 Baseline classifier

The LSTM baseline classification of the original time series was compared to the four top-performing CNN classifiers trained on the encoded time-series data for each raster resolution (Figure 3.8). Performance was measured on training accuracy and loss.

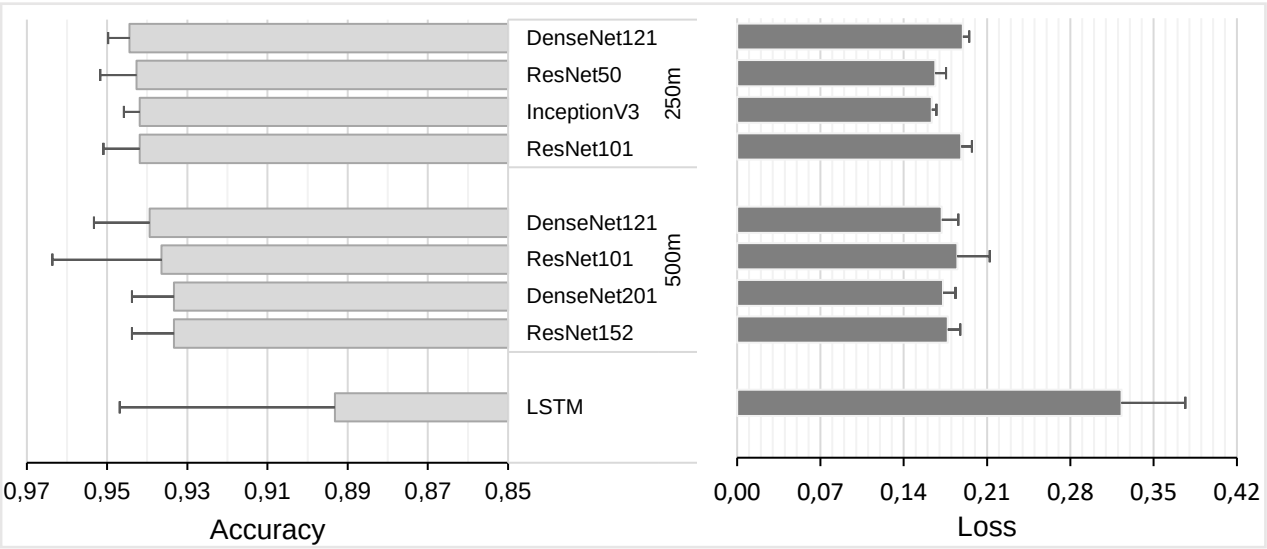


Figure 3.8: Performance of baseline LSTM classifier on original time series vs the four top-performing CNN models using encoded images for both 250 m and 500 m MODIS datasets. Error bars show standard deviation

The DenseNet121 model outperformed the baseline classification by ±5%. The trained and validated DenseNet121 was therefore the CNN classifier selected (Figure 3.7) to perform binary image classification on MODIS 250 m NDVI encoded time series to identify change and no-change pixels between 2001 and 2009 for the study area marked in red in Figure 3.2.

3.4.3 Image classification

The binary classification image (Figure 3.9) was produced by the best-performing CNN classifier in Table 3.3. The black pixels in Figure 3.9 represent areas of no change, whereas the transparent pixels illustrate areas of change, allowing visualisation of the result from the highest-performing CNN model. When compared with a high-resolution Google Earth image, it is qualitatively clear that the model was able to accurately identify changed areas.

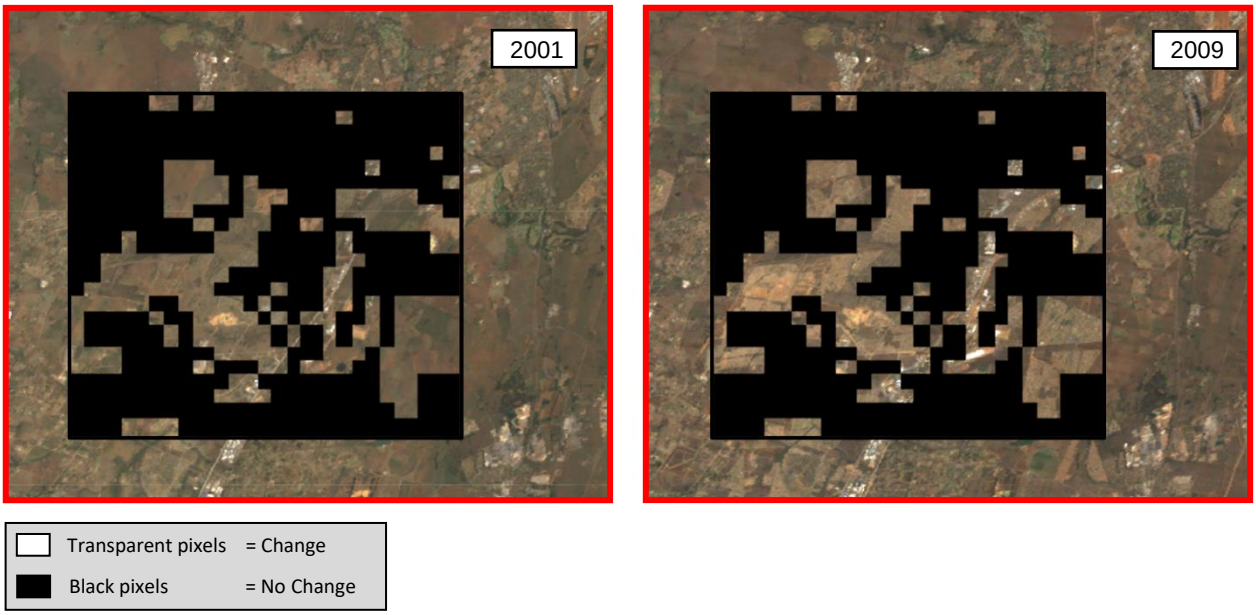


Figure 3.9: Binary image classification with DenseNet121 classifier to illustrate change and no-change pixels between 2001 and 2009 using MODIS NDVI 250 m resolution data

To assess the accuracy of the binary classification, a confusion matrix was constructed using the 121 stratified random points, where 51 points represent the “change” class and 70 the “no-change” class. Table 3.4 shows the results for the overall accuracy (OA), Kappa, PA, UA, error of omission and commission.

Table 3.4: Confusion matrix showing overall accuracy (OA), Kappa, and positive predictive power

Predicted Class

Actual Class	No Change	Change	Total	Producer's Accuracy	Omission Error
No Change	69	5	74	0.9324	0.0676
Change	1	46	47	0.9787	0.0213
Total	70	51	121		
User's Accuracy	0.9857	0.9020			
Commission Error	0.0143	0.0980			

Overall Accuracy	0.9500
Kappa	0.8972
Positive Predictive Power	0.9020

An OA of 95% (Kappa 0.8972) was achieved. The image classification resulted in high producer's (PA>0.93) and user's accuracies (UA>0.90) for both classes. Precision or positive predictive power was computed as 90.2%.

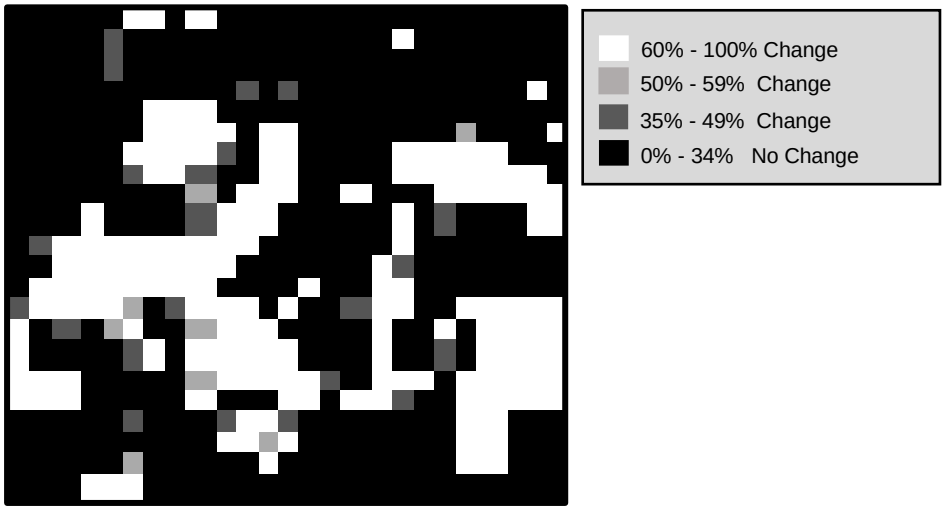


Figure 3.10: Graphically presents the prediction probability percentage for the "change" class, representing model confidence

Three colours are used to illustrate the different levels of confidence for the changed pixels: white represents model confidence above 60%; light grey illustrates model confidence ranging from 50% to 59% and; dark grey pixels represent a confidence level of below 50%, generally bordering the "no-change" class (black) (Figure 3.10).

3.4.4 Generalisability

The DenseNet121 CNN model was used to test the generalisability of the CNN classifiers. The graph (Figure 3.11) shows the accuracy for both training and testing data over 100 epochs, with training data from Pretoria in blue, and validation data from Maputo in orange.

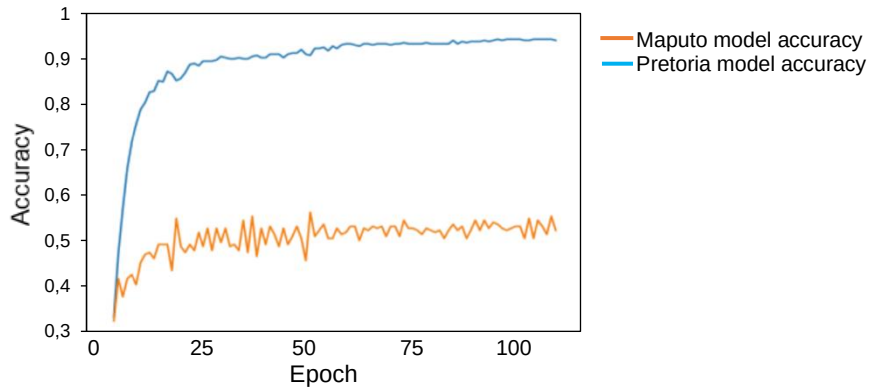


Figure 3.11: DenseNet121 generalisability results using training and validation data from Pretoria and Maputo respectively

Although high model accuracies were accomplished using the training data from Pretoria (blue line), this was not the case for the model validation accuracy undertaken on the Maputo data (orange line). A validation accuracy of 59% was achieved, which is significantly lower than the 94% model training accuracy.

3.5 DISCUSSION

This study presented a novel approach to change detection through data engineering by encoding MODIS 500 m and 250 m NDVI time-series data as 2D images using GASF, GADF, and MTF transformations. Eleven CNN models were trained and validated. In this section, the efficacy of GAF and MTF encoded images with CNNs as a method for urban change detection will be discussed. After classification using the best-performing classifier and resolution, an accuracy assessment was performed to draw a comparison between this framework and other change detection methods. Furthermore, the generalisability and robustness of the selected model will be discussed before concluding the discussion with a look at the limitations of the research.

The pre-trained CNNs achieved consistently high mean accuracies ranging from 87 - 91% (Table 3.2). The DenseNet and ResNet architectures achieved the best results which is in line with the literature (He et al. 2016; Huang, Liu & Van Der Maaten 2017). DenseNet201, ResNet50 and ResNet101 obtained the highest accuracies when using GASF, GADF, and MTF input datasets respectively (Table 3.2). This is in agreement with the findings by Dias et al. (2019) that both these architectures were top-performing feature extractors. However, DenseNet was more beneficial when applied to larger datasets (Dias et al. 2019). In this study, the DenseNet121 feature extractor

achieved the highest accuracy of ~94% when using the $COMBO_{500}$ and $COMBO_{250}$ datasets. As expected, the GASF, GADF, MTF concatenated input images produced a higher mean accuracy when evaluating the model performance (Table 3.2), outperforming the three individual transformations (Dias et al. 2019; Yang, Chen & Yang 2020).

The combination of the DenseNet121 feature extractor and the concatenated input images was found to be the most successful when using MODIS imagery (Figure 3.7). Moreover, Kleynhans, Salmon & Wessels (2017) found that an increase in the MODIS resolution would positively affect the accuracy of an urban change detection. Evidence of this can be seen in the higher mean accuracy for the higher resolution MODIS $COMBO_{250}$ (Table 3.3) and the comparison in Figure 3.7. DenseNet121 remains the most favourable feature extractor for this application and was therefore chosen to perform the change detection on the study area (Figure 3.2)

The LSTM algorithm, a state-of-the-art time-series classification approach used for urban change detection (Karim et al. 2017; Yildirim et al. 2019), when applied to the original time-series, only achieved an accuracy of 89.32±5.37% (Figure 3.8) in training and validation. The mean accuracy for the LSTM was much lower than achieved using an encoded time-series and CNNs with a much higher standard deviation.

The DenseNet121 CNN applied to the 2001-2009 time series, encoded as GADF, GASF and MTF concatenated images, produced a three-class change detection map with classes “urban”, “vegetation” and “change”. The model predicted the probability of belonging to each class. The binary change map (Figure 3.9) only illustrates the class of highest probability with “urban” and “vegetation” collapsed to “no-change”. The accuracy assessment substantiated the results (Table 3.4). The confusion matrix illustrates an OA of 95% which is close to the model’s evaluation accuracy of 94.44%. However, the lower positive predictive power of 0.9020 could indicate that the model may produce false positives. This is illustrated by the fact that the model classified five pixels as “change” when in fact no change had occurred at that exact location (Table 3.4). The misclassification could be due to the low resolution, which results in mixed pixels. An area of pixel size 62500 m^2 would require extensive land cover change to register a pixel change from vegetation to urban. If only a portion of the pixel changed, the signature would differ from that of a pure vegetation pixel. The model may identify this change in the spectral response and classify it accordingly. An increase in spatial resolution would reduce the number of false positives and increase the performance of the model. Nevertheless, the model presents a high true negative rate (also called specificity) of 93%. This indicates that the model does not often misclassify “no-change” pixels. As seen in Table 3.4, only one pixel of 70 was misclassified as “no change” when

in fact change occurred. This is a noteworthy when looking at the accuracy assessment, as it shows that the model will seldomly fail to detect areas of change.

The confidence with which the model could detect changed pixels is illustrated in Figure 3.10, which illustrates the probability of a pixel belonging to the “change” class. The majority of pixels were classified with 60% or more confidence. However, lower confidence was noted for pixels that are located on the outskirts of urban areas. These pixels fell within the 35%-49% confidence category, which is related to the concept of mixed pixels. The model was still able to distinguish the correct class just at a lower confidence level but would benefit from higher resolution imagery.

On the other hand, the novel framework of encoding the time series for a CNN classification produced significantly higher accuracies using both 500 m and 250 m resolution datasets when comparing at the model performance level before the image classification. A comparison at this level can only be done as the DesneNet121was the only architecture used to produce an image classification. The framework of encoding time series and performing a CNN classification can outperform the state-of-the-art LSTM classifier. Extensive research has shown that there is a lack of literature concerning the framework of encoding time-series data for an urban change detection. This novel framework developed by Wang and Oates (2015) has been implemented in several studies with various applications except for an urban change detection (Barra et al. 2020; Dias et al. 2020; Huang, Chakraborty & Sharma 2020; C Li et al. 2020; Yang, Chen & Yang 2020).

By comparison with a temporal autocorrelation time-series based change detection method deployed in South Africa (Kleynhans et al. 2013; Kleynhans et al. 2012; Kleynhans et al. 2015; Kleynhans, Salmon & Wessels 2017), the OA of 94% achieved in this study using a similar dataset outperformed the TACD and STACD methods, which achieved OAs of 88% and 76% respectively. This implies that applying the framework of encoding time-series imagery as features for input to a CNN classification, derived by Wang and Oates (2015), can successfully be implemented for an urban change detection.

The success of this framework originates from the combination of encoding time-series input data using GAF and MTF transformations, as well as state-of-the-art feature extractors. The encoding is the critical step that transforms the per-pixel time series into an image format that matches the input requirements for the feature extractors.

Despite the successful change detection, there are still several aspects that need to be noted for future research. The generalisability of the model is poor Figure 3.11. The model could not be trained and tested in two different locations. The input time series are specific to a location and would need to be generalised to allow a more robust model. A key aspect would be to remove the seasonality of the time series and perform a line smoothing. If a general trend time series for urban

change was to be found, the model would potentially be transferable. Other aspects that would help increase the performance of the model are increasing the resolution further to reduce the number of mixed pixels on the boundaries of land cover types, as well as establishing the effect that the temporal aspect has on the model performance.

3.6 CONCLUSION

This study evaluated the effectiveness of encoding pixel-wise time series derived from MODIS data for a CNN classification. GASF, GADF and MTF encoding transformations were performed on time series generated from 500 m and 250 m resolution MODIS NDVI imagery. $COMBO_{500}$ and $COMBO_{250}$ concatenated datasets were created by combining the three transformation outputs. Eleven pre-trained feature extractors were used to process the encoded time series. The different CNN architectures performed at various levels depending on the specific transformation applied to the relevant input data. The concatenated $COMBO_{500}$ was the top-performing dataset when compared to the individual GASF, GADF and MTF transformations, achieving an average accuracy of 91.39%. This is primarily because of its ability to combine the information from all three encoding transformations. The increase in the resolution of the input data (COMBO250) yielded a high average accuracy of 93.56%. It is concluded that an increase in spatial resolution will indeed help increase the performance of the feature extractors. However, the extractors are slightly overfitted as their loss values do not decrease. It was concluded that at the end of this research, the novel framework of encoding time-series data as 2D images for an urban change detection with the use of multiple CNN classifications was effective concerning the current state-of-the-art method deployed. Further research regarding a drastic increase in the spatial resolution is recommended, as well as to investigate the temporal aspect of the input data.

CHAPTER 4: FACTORS INFLUENCING THE PERORMANCE OF URBAN CHANGE DETECTION USING HIGH-RESOLUTION IMAGERY AND TIME-SERIES ENCODING

4.1 INTRODUCTION

Urbanisation continues due to economic and employment opportunities (Asongu et al. 2020; Gunter 2021) causing the expansion of human settlements to be one of the most pervasive forms of land cover change in South Africa (Kleynhans et al. 2012). A practical framework for urban change detection that processes data in a timely and accurate manner is therefore essential for urban planning and management (Jensen & Im 2007). Kleynhans, Salmon and Wessels (2017) demonstrated an effective way of monitoring urbanisation using a spatiotemporal ACF. Spatial analysis of earth observation (EO) data from satellites and sensors can help stakeholders track and understand urban development over a wide range of spatial and temporal scales (Al-Bilbisi 2019). Time-series RS data have proven to be a trustworthy source in performing change detection (De Beurs & Henebry 2005; Chen et al. 2019; Liu et al. 2018). At pixel level, a remotely sensed time series usually contains trend and seasonal components or intra-annual fluctuations (Xu et al. 2019). Changes that occur in the trend component can be ascribed to disturbances such as fire, deforestation or urbanisation (Verbesselt, Hyndman, Newnham, et al. 2010; Verbesselt, Hyndman, Zeileis, et al. 2010). New developments in the field of computer vision have introduced a novel framework of encoding a time series of data as an image (Liu & Wang 2016; Wang & Oates 2015) using Gramian angular summation field (GASF), Gramian angular difference field (GADF) and Markov transition field (MTF) transformations. The resulting encoded images are suitable for use in a convoluted neural network (CNN) classification (Liu & Wang 2016; Wang & Oates 2015). The novel framework of processing encoded time-series images for DL classifications has only recently gathered traction in fields such as weather and financial forecasting, fault diagnosis, human activity recognition, and sensor classifications (Barra et al. 2020; Huang, Chakraborty & Sharma 2020; C Li et al. 2020; Qin et al. 2020; Yang, Chen & Yang 2020). However, the methodology has not yet had significant exposure in the field of spatial sciences. A study conducted by Dias et al. (2020) utilised the framework for a pixel-wise Eucalyptus region classification using MODIS imagery, while Chapter 3 proposed a methodology of implementing this novel framework within a spatial science context in South Africa. Pixel-wise MODIS time-series data were collected and encoded before CNN classification testing of multiple classifiers for urban change detection. Accuracies of higher than 90% were achieved, proving the novel

framework superior to current state-of-the-art methods (Chapter 3). To improve accuracy, further investigation into the data and classification schemes were recommended (Chapter 3).

Chapter 3 demonstrated a correlation between spatial resolution and the accuracy of the classifications. The increased resolution from MODIS 500 m to 250 m resulted in a one per cent increase in the change detection accuracy (CDA) in agreement with the literature (Kleynhans, Salmon & Wessels 2017). The increased accuracy contradicts findings by Chen, Stow & Gong (2004), who stated that high-resolution imagery would more likely result in a decrease in classification accuracy for an urban environment. However, Chen, Stow and Gong (2004) concluded that the land cover characteristics would play a role in the performance. Therefore, the relationship between resolution and classification accuracy will vary depending on the study site and the resolution level of data (Chen, Stow & Gong 2004).

In addition to resolution, other elements may affect the performance of the classifier. One such aspect to consider is the length of the time series. Rispens et al. (2014) investigated the impact of time-series length (TSL) on the accuracy and precision of algorithms and concluded that an increase in TSL could positively affect accuracy (Hills et al. 2014; Rispens et al. 2014). A second element to consider is the quantity of available training data. Previous studies (Cho et al. 2015; Dunnmon et al. 2019)have shown the importance of training dataset size and its correlation to the accuracy of the class assignment. A consensus across the literature is that a more extensive training dataset is more beneficial to the success of the classification (Cho et al. 2015; Dunnmon et al. 2019; Zhong et al. 2018). Zhong et al. (2018) found that CNN classifications performed better with larger training datasets. The biproduct of a CNN prediction is a value that illustrates the probability of an object (pixel) being assigned to a class (Segal-Rozenhaimer et al. 2020). Classification is a result of applying the stipulated minimum pixel probability (MPP) value. Altering the MPP value will ultimately affect the final classification. In addition to applying a single classifier, recent studies have utilised an ensemble of CNN classifications (Chen et al. 2017; W Li et al. 2019; Vasan et al. 2020). An ensemble of CNN classifications uses information from multiple architectures, providing an advantage over a single classification (Vasan et al. 2020)and has proven more effective than the individual CNN classifications (Chen et al. 2017).

Despite the success of DL classifications, their generalisation ability remains unclear (Jakubovitz, Giryes & Rodrigues 2019). Although removal of the seasonal trend of the time series may improve the performance of the classification (Zhang & Qi 2005), DL algorithms can accommodate seasonal trends in the data if these trends are well presented (Hamzaçebi 2008).

This chapter describes various factors that can affect change detection using CNNs with input from encoded pixel-wise time-series NDVI data gathered from Sentinel-2 imagery. GAF and MTF are

applied to transform the time series before classification. Firstly, the relationship between spatial resolution and the performance of the CNN-based change detection algorithm is investigated, with a comparison to results from Chapter 3. Secondly, due to the varying periods for which spatial data can be gathered, an experiment is conducted using different training set sizes and TSL datasets. The top-performing framework for change detection is determined by varying the MPP for image classifications and testing an ensemble of CNN classifications. After the generalisability of the change detection algorithm is tested using standard and seasonally detrended datasets, image classification is performed. The accuracy of the classifications is measured using a confusion matrix. The workflow for this chapter can be seen in Figure 4.1.

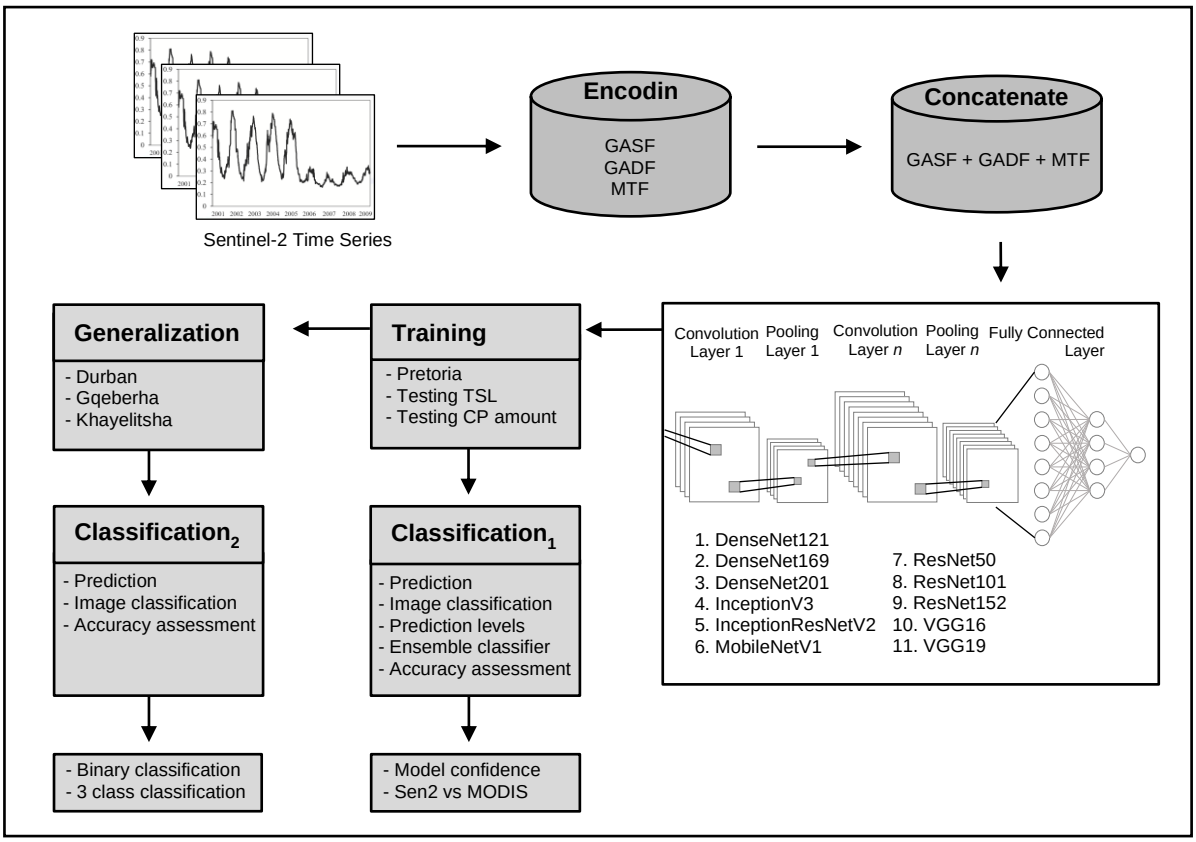


Figure 4.1: Workflow diagram illustrating the process for implementing a change detection through DL feature extractors and encoding transformation for multiple locations

4.2 MATERIALS AND METHODS

4.2.1 Study area

Illustrated in Figure 4.2: is a national map of South Africa that indicates the locations of the four testing sites (Pretoria, Durban, Gqeberha, and Khayelitsha), each located in a different province: Gauteng, KwaZulu-Natal (KZN), Eastern Cape, and Western Cape, respectively. The Pretoria site was used for model development, training (ground truth data), testing as well as prediction (Chapter 3). High levels of urbanisation with encroachment on natural vegetation continues in this

study area. The other study sites, Durban, Gqeberha, and Khayelitsha testing sites, have been included to illustrate the ability of the DL model to generalise.

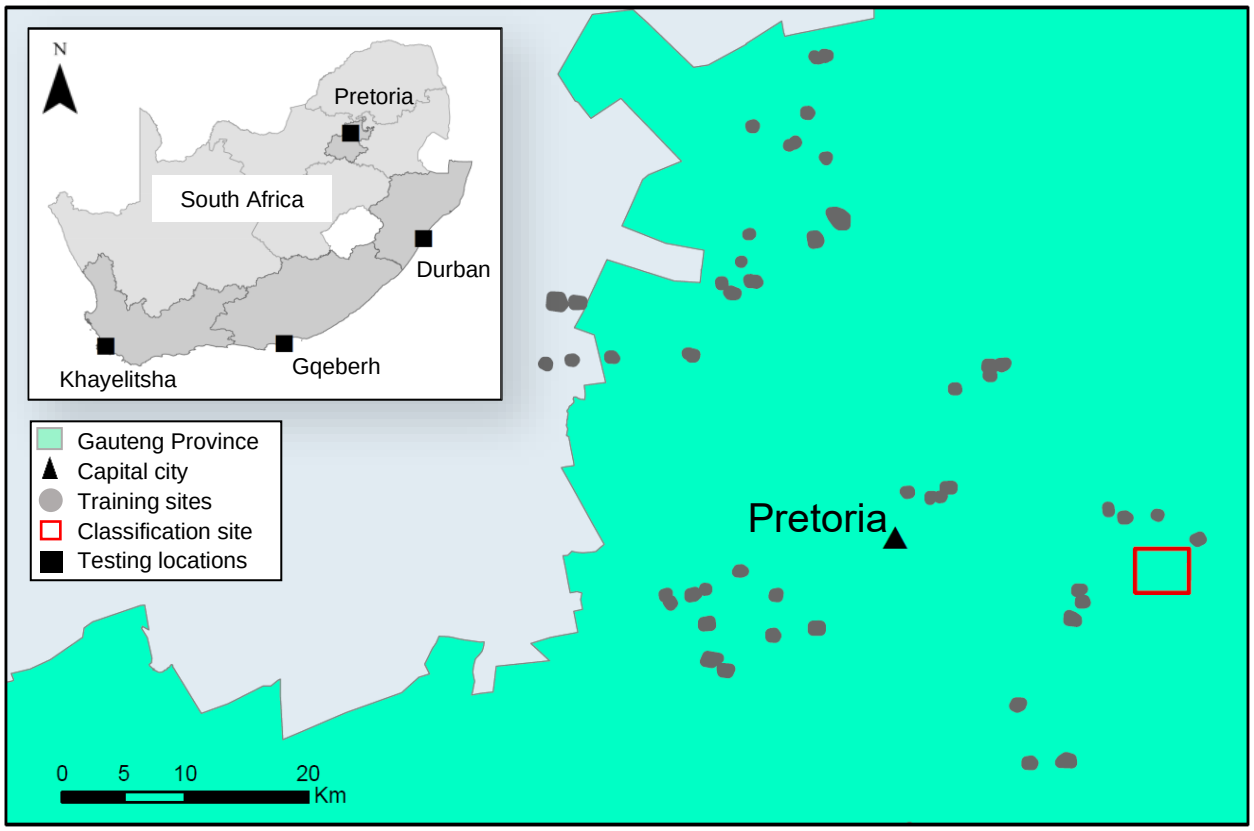


Figure 4.2: Provincial map showing testing locations in their respective provinces, training points and classification site for accuracy assessment

The location of the ground truth data can be seen as grey dots in Figure 4.2:. The red square shows the Pretoria site used to test model accuracy for change detection and accuracy assessment.

4.2.2 Data collection

A per-pixel time series of spatially aligned NDVI pixels was calculated using Equation 2.4 and was derived from the Sentinel-2 MSI: Multispectral instrument, Level-2A (USGS 2018) using the GEE platform (Evans & Malcom 2021; Gorelick et al. 2017; Chapter 3). NDVI was computed using the equation (NIR - Red)/(NIR + Red) (Lunetta et al. 2006) where red represents band 4 and NIR represents band 8, the NIR band. Sentinel-2 was chosen over Landsat 8 for its higher temporal resolution to account for high cloud cover over the respective locations. However, the 10 m resolution Sentinel-2 data was resampled to 30 m to correspond with Landsat 8. This was done so that if one does not have access to Sentinel-2 data or if one would like to perform the methodology during a period for which Sentinel-2 had not been established, one could do so. With no atmospherically corrected imagery available before December 2018, GEE access to Sentinel-2 Level-2A limited the study period from 2019 to 2021. Training and test datasets for change

detection were collected for the period ranging from 2019/01/01 to 2021/06/30 (911 days). The presence of cloud cover and the seven-day temporal resolution resulted in a maximum time-series length (TSL) of 82. All models were trained using encoded data (Section 4.2.3) collected from Pretoria, located in the Gauteng Province, for comparison with other time series-based studies (e.g. Kleynhans, Salmon & Wessels 2017; Chapter 3). Testing took place at four locations across South Africa (Figure 4.2:).

Multiple datasets (Table 4.1) were collected and encoded to conduct the experiments (Section 4.2.5) to test model performance. The sensor, resolution, size of the training dataset represented by the number of changed pixels (CP) and TSL for each training and testing site along with their respective dataset names can be found in Table 4.1.

Table 4.1: Datasets for training and testing with sensor, resolution, CP and TSL per dataset

	Dataset names	Satellite	Resolution	CP	TSL
Training	$Pretoria_1$	Sentinel-2	30m	433	82
	$Pretoria_2$	Sentinel-2	30m	547	82
	$Pretoria_3$	Sentinel-2	30m	547	57
	$Pretoria_4$	Sentinel-2	30m	547	32
	$Pretoria_5$	MODIS	250 m	122	82
Testing	$Pretoria_6$	Sentinel-2	30m	Unknown	82
	$Pretoria_7$	MODIS	250 m	Unknown	82
	$Durban_1$	Sentinel-2	30m	20	65
	$Gqeberha_1$	Sentinel-2	30m	115	50
	$Khayelitsha_1$	Sentinel-2	30m	234	50
	$Durban_2$	Sentinel-2	30m	20	32
	$Gqeberha_2$	Sentinel-2	30m	115	32
	$Khayelitsha_2$	Sentinel-2	30m	234	32

To capture change between 2019 and 2021, three classes of training and testing data were digitised from 10 m resolution Sentinel-2 Level-2A data (ESA 2015). Two classes (vegetation and urban) of pixels without change (no-change) and one class of CP were saved in the $Pretoria_1$ dataset (Table 4.1). A total of 433 CPs was collected for comparison with the number of MODIS CPs used in Chapter 3. A MODIS NDVI dataset (MOD13Q1 resolution 250 m) (USGS 2018) labelled $Pretoria_5$ was collected over the exact location and time frame (TSL 82) as $Pretoria_1$, however, the number of comparative CPs was only 122, decreasing the training sample size. In $Pretoria_2$, 547 CPs were collected to further test the effect of increasing the training sample size. Based on $Pretoria_2$ with 547 CPs, in $Pretoria_3$ and $Pretoria_4$ the TSL was decreased to 57 and 32 respectively. Test datasets Pretoria6 (Sentinel-2) and Pretoria7 (MODIS) were used for image classification, to test the effect of resolution, MPP and ensemble modelling.

A comparison between the MODIS 250 m, Sentinel-2 10m and Sentinel-2 30m pixel grids is illustrated in Figure 4.3. The yellow grid represents the MODIS 250 m pixels, while (1) and (2) display the grid of 10 m and 30 m resolution pixels respectively (Figure 4.3).

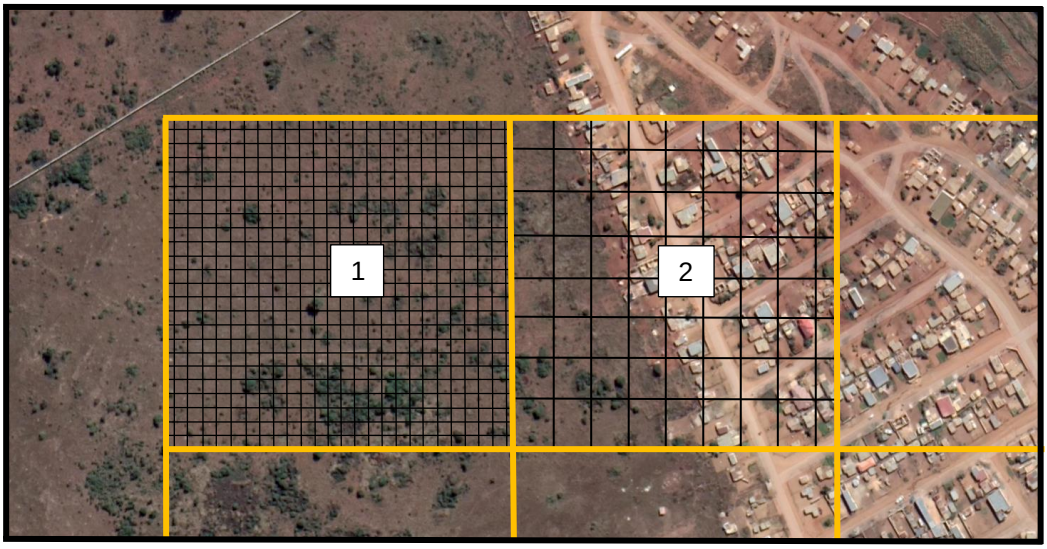


Figure 4.3: Quickbird imagery overlaid with MODIS 250 m resolution yellow grid pattern. Pixel (1) and (2) correspond with Sentinel-2 10 m resolution and Landsat8 30 m resolution grid patterns respectively (Source: Google Earth).

A per-pixel time series is represented in Figure 4.4. The blue line shows pixels where a change from vegetation to urban has occurred, while the orange line in Figure 4.4 shows a strong seasonal vegetation response.

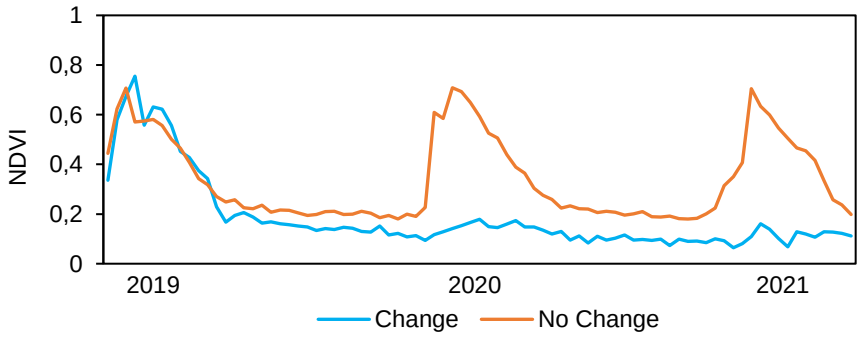


Figure 4.4: Sentinel-2 NDVI time series for a changed and unchanged pixel 2019-2021

In the changed pixel (blue line), the NDVI is initially high, but decreases due to urbanisation and shows a consistently low NDVI for the remaining period. The consistent low is an indication of no change for the urban class. The time-series data collected at the Gqeberha and Khayelitsha testing sites showed a different phenological curve without strong seasonal patterns, related to climatic conditions and rainfall regimes at the sites.

Removing seasonality from the training and testing data may potentially increase model performance (Yan 2012). For the datasets Pretoria4, Durban2, Gqeberha2 and Khayelitsha2 (Table

4.1), seasonality was removed to establish the effect of the seasonal patterns on model performance (Martínez et al. 2018; Nelson et al. 1999; Yan 2012)and generalisability. This reduced the TSL to 32, the minimum input size required by neural networks (Keras 2020a). Figure 4.5 shows the time series for the pixels in Figure 4.4, while Figure 4.6 illustrates the changed pixels at the testing sites with the seasonal pattern removed.

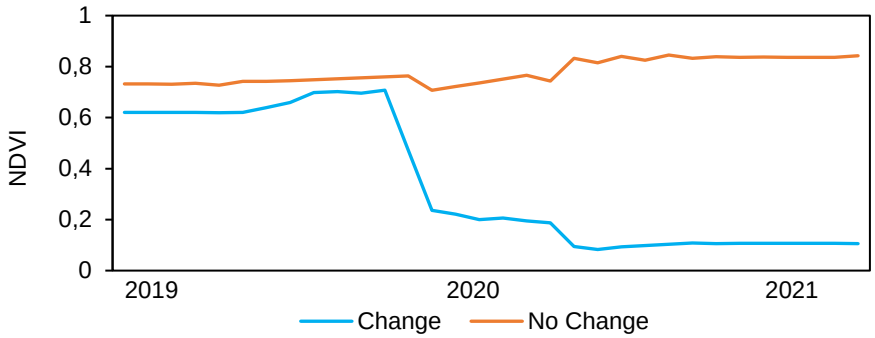


Figure 4.5: Seasonality removed from the Sentinel-2 NDVI time series for a change and no-change pixel

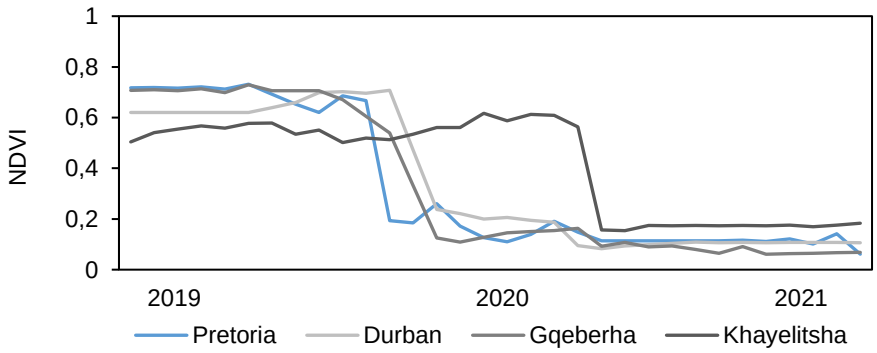


Figure 4.6: Seasonality removed from Sentinel-2 NDVI time series for a changed pixel at each of the four test sites (Pretoria, Durban, Gqeberha, Khayelitsha)

Although removing the seasonality should increase the similarity between the time series, there is still a difference between the training and testing data.

4.2.3 Encoding time series as image

GAM and MTF pixel-wise transformations were executed for each time series in all datasets (Table 4.1). A polar coordinate-based matrix can display the temporal correlation between observations in different time intervals (Dias et al. 2020; Wang & Oates 2015; Yang, Chen & Yang 2020; Yang et al. 2019). Polar coordinates are formulated from Equation 1 (Chapter 3). Cosine and sine functions are then used with the angles formulated from the polar coordinates to produce the GASF and GADF matrices (Yang, Chen & Yang 2020; Yang et al. 2019). Equation 2 (Chapter 3) formulae are used to generate the MTF, which preserves information in the time domain by capturing transitional probability statistics. The time series is discretised by splitting it into quantile bins, after which a Markov transition matrix is built (Dias et al. 2020; Wang & Oates 2015). Colour

images are created to visually display the matrices for the GASF, GADF and MTF encoded images for each of the three classes (Figure 4.7:). The original input time series for the encoding used a TSL of 82 and is illustrated in Figure 4.7:a, while Figure 4.7:b shows the time series with seasonality removed.

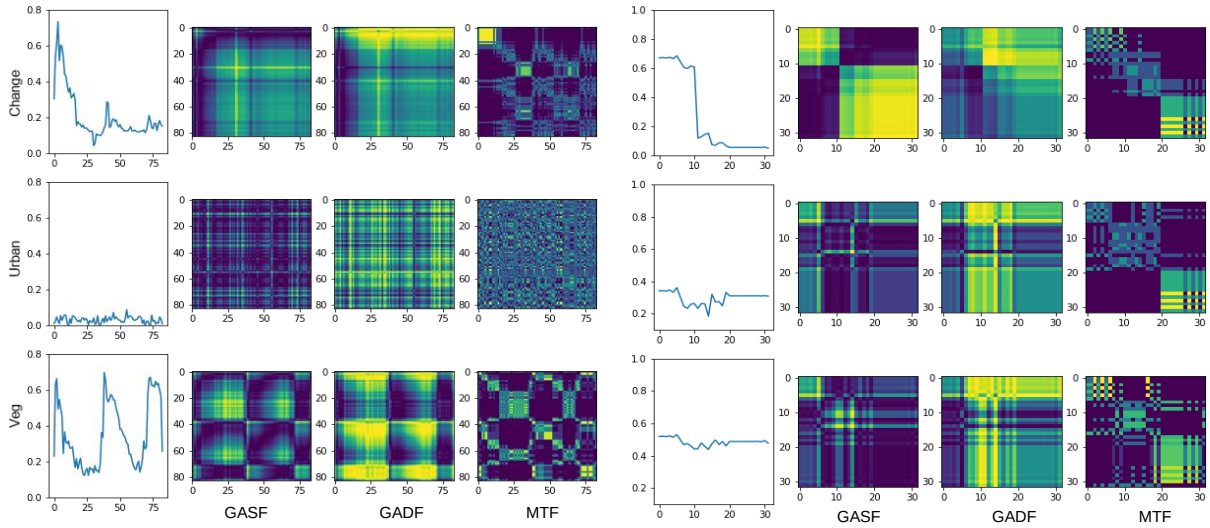


Figure 4.7: GASF, GADF, and MTF encoded colour images generated from the Sentinel-2 NDVI time series for the three classes (urban, vegetation, change)

Seasonality was removed from the TSL82 datasets to form the Pretoria$_4$, Durban$_2$, Gqeberha$_2$, and Khayelitsha2 test datasets. These datasets have a TSL of 32, which affects the size of encoded matrices (Figure 4.7:b). The GASF, GADF and MTF encoded images were concatenated to increase the extraction of discriminative features (Yang et al. 2019; Chapter 3) before input to the CNN feature extractors.

4.2.4 Deep-learning feature extractors

Eleven CNN architectures, pre-trained on ImageNet, were used as feature extractors to classify the per-pixel encoded time series (Fei-fei et al. 2021; Keras 2020a). Each feature extractor contains several convolutional and max-pooling levels. By training the feature extractors and assigning weights to the layers, a secondary task, such as classification, can be implemented. The 11 architectures that were considered include DenseNet121 DenseNet169, DenseNet201, InceptionV3, InceptionResNetV2, MobileNetV1, ResNet50, ResNet10, ResNet152, VGG16, and VGG19 (Dias et al. 2020; Chapter 3).

4.2.5 Experiments

Using datasets Pretoria1, Pretoria2, Pretoria3, and Pretoria4 (Table 4.1), three experiments were set out to test the effect of resolution (4.2.5.1), varying training set size (4.2.5.2) and time-series length (4.2.5.3) on model performance. In each of the three experiments, the input dataset was split 70:30 for training and validation purposes. A three-fold validation process gathered an average accuracy for comparison. Experiment 4 (0) involved testing the generalisability of the models to unseen data at the different testing sites. In experiment 5 (4.2.5.5), several classifications were conducted using the classification site in Figure 4.2: (red square) using the Pretoria2, Pretoria5, Pretoria6 and Pretoria7 input datasets (Table 4.1).

4.2.5.1 Experiment 1: resolution

The first experiment tested the effect of an increase in resolution from 250 m MODIS to 30 m Sentinel-2. Eleven feature extractors were used to process the Pretoria1 dataset. A three-fold validation process gathered an average accuracy for comparison between the use of MODIS and Sentinel-2 imagery.

4.2.5.2 Experiment 2: training set size

The second experiment tested the effect of increasing CP when training the feature extractors while keeping the TSL constant while experiment 3 examined the effect of varying TSL. Figure 4.8 shows the testing workflow for model evaluation using different CP and TSL. In experiment 2, a comparison was made between the Pretoria1 dataset with CP 433 and TSL of 82, and the Pretoria2 dataset with an additional 114 CPs was introduced to the training process when the Pretoria2 dataset was utilised.

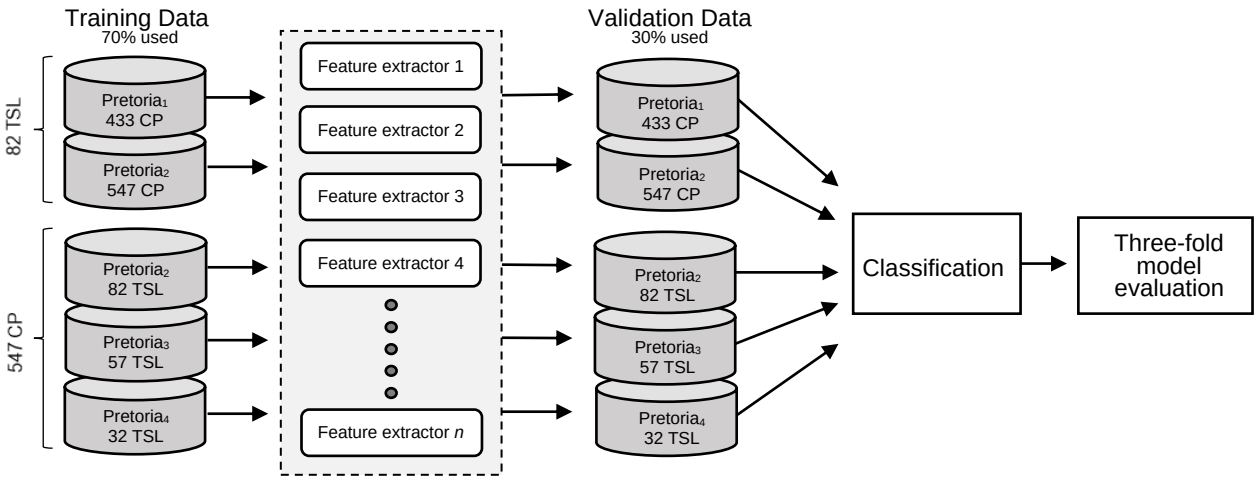


Figure 4.8: Data-testing workflow showing the split of data for the model evaluation using different CP and TSL

4.2.5.3 Experiment 3: time-series length

The third experiment determined the effect of TSL on feature extractors performance. The Pretoria2, Pretoria3, and Pretoria4 datasets all contained 547 CP, but varying time series lengths (Figure 4.8). Each dataset was processed through the feature extractors using the 70:30 split for training and validation. The average performance for each feature extractor was gathered by performing a three-fold validation process (Figure 4.8). Two of the eleven CNN architectures required a minimum TSL of 75 for the input dataset. Therefore, the Pretoria3 and Pretoria4 datasets could not be processed through InceptionV3 and InceptionResNetV2 feature extractors.

4.2.5.4 Experiment 4: generalisability

Testing the generalisation of the feature extractors required unseen data from different locations where urban change was present. Three testing sites were identified in three different provinces. Multiple datasets were gathered (Table 4.1) and used for three different experiments. Three-fold validation was performed for each experiment before image classification. In these experiments, nine feature extractors were trained using the Pretoria2 dataset, while testing was performed on unseen data from three locations: Durban1 (TSL 65), Gqeberha1 (TSL 50) and Khayelitsha1 (TSL 50) (Table 4.1) as well as datasets with seasonality removed (Durban2, Gqeberha2, and Khayelitsha2 with TSL 32).

4.2.5.5 Experiment 5: image classification

The first image classification using the Pretoria5 and Pretoria7 datasets (MODIS 250 m) was a baseline classification. DenseNet121, the top-performing MODIS feature extractor (Chapter 3) was selected and trained with all the Pretoria5 data and then used to predict three classes (urban, vegetation, change) on the Pretoria7 dataset. Each pixel was then assigned to the class for which it had the highest probability. A simple binary classification was also derived to illustrate change and no-change pixels (combined urban and vegetation classes) at coarser resolution. Figure 4.9 illustrates the workflow applied to Sentinel-2 (resampled to 30 m) to test the effects of probability constraints (Segal-Rozenhaimer et al. 2020) and the application of an ensemble of classifiers (Chen et al. 2017; S Li et al. 2019; Vasan et al. 2020).

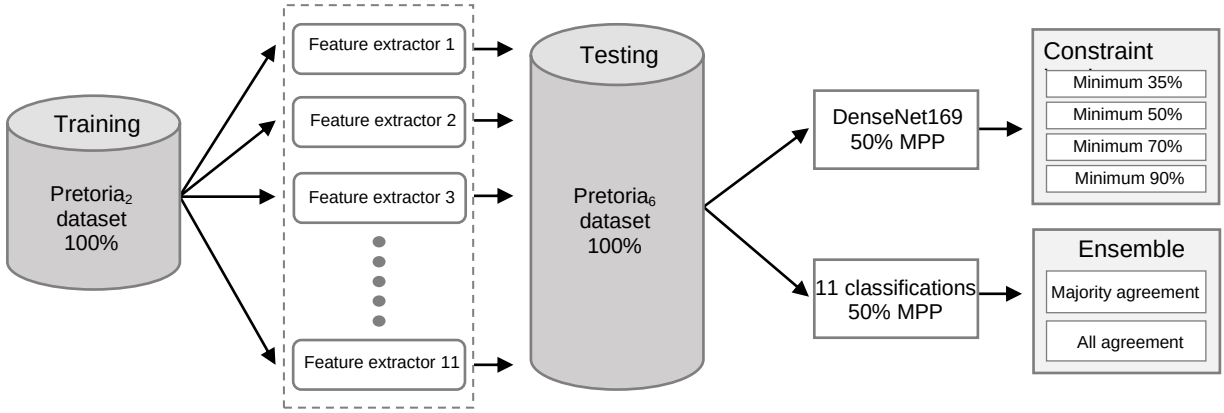


Figure 4.9: Training CNNs with the Pretoria$_2$ dataset to test minimum pixel probability constraints and an ensemble of CNNs for classification on the Pretoria$_6$ dataset

The Pretoria2 data was used to train the 11 CNN feature extractors to apply image classification on the Pretoria6 dataset (Sentinel-2 30m). The top-performing Sentinel-2 feature extractor (DenseNet169) was used to conduct four binary classifications, each altering the MPP level for which a pixel is assigned to a class. Four classifications were run: (1) the pixel was assigned to the class with the highest probability; (2) a 50% MPP constraint was applied, assigning pixels to a class if the prediction value was higher than 50%; (3) a 70% MPP constraint was employed, whereas (4) the final classification utilised a 90% MPP.

Two ensemble classifications were conducted using multiple feature extractors trained on Pretoria2 while classifying Pretoria6. The first ensemble classifier used a majority agreement rule. This rule required a pixel class agreement from six or more feature extractors to classify a pixel as changed. If this agreement was not met, the pixel was then assigned to the no-change class. The second ensemble classification required a pixel class agreement from all feature extractors. This meant that all 11 feature extractors had to classify the pixel as changed for that pixel to be assigned to the changed class. In addition to the ensemble rules, the 50% MPP constraint was applied.

A pixel-wise probability level classification was performed to show the individual probabilities of the pixels found in the classification of the Pretoria6 dataset. This classification utilised the traditional method of assigning the pixel to the class with the highest probability. Four probability levels were established, the first being 0% to 34% for no-change pixels. The next level is consistent with pixel probabilities ranging from 35% to 49%. The final two levels used the 50% to 59% and 60% to 100% probability range. These three levels represented pixels that were classified as changed.

To test the effect of generalisability on prediction, a three-class classification (urban, vegetation, change) and binary classification (change, no change) using 50% MPP constraint was performed for datasets Durban1, Gqeberha1, and Khayelitsha1 for all 11 feature extractors using a model

trained on Pretoria2. A further binary classification with 50% MPP constraint was performed on the seasonality removed datasets (Durban2, Gqeberha2, and Khayelitsha2).

4.2.6 Image classification evaluation protocol

To verify the success of the classifications, an accuracy assessment was performed using the confusion matrix (Dervisoglu, Bilgilioglu & Yagmur 2020). This image classification evaluation protocol was employed for each of the classifications performed. Stratified random points (150) were created for each binary classification (Stehman 1996). Each point was cross-referenced with a 2019 and 2021 Sentinel-2 10 m resolution image pixel to assign the class (change, no change). From the confusion matrix, overall accuracy (OA), Kappa, user's accuracy (UA), and producer's accuracy (PA) were formulated and recorded (Dervisoglu, Bilgilioglu & Yagmur 2020).

4.3 RESULTS

4.3.1 Experiment 1: resolution

The Pretoria1 dataset was used to train 11 CNN classifiers with 433 CP using Sentinel-2 30 m resolution data. Figure 4.10 shows the individual average accuracy for training each CNN classifier comparing Sentinel-2 30 m with MODIS 250 m and MODIS 500 m resolution encoded images (Chapter 3).

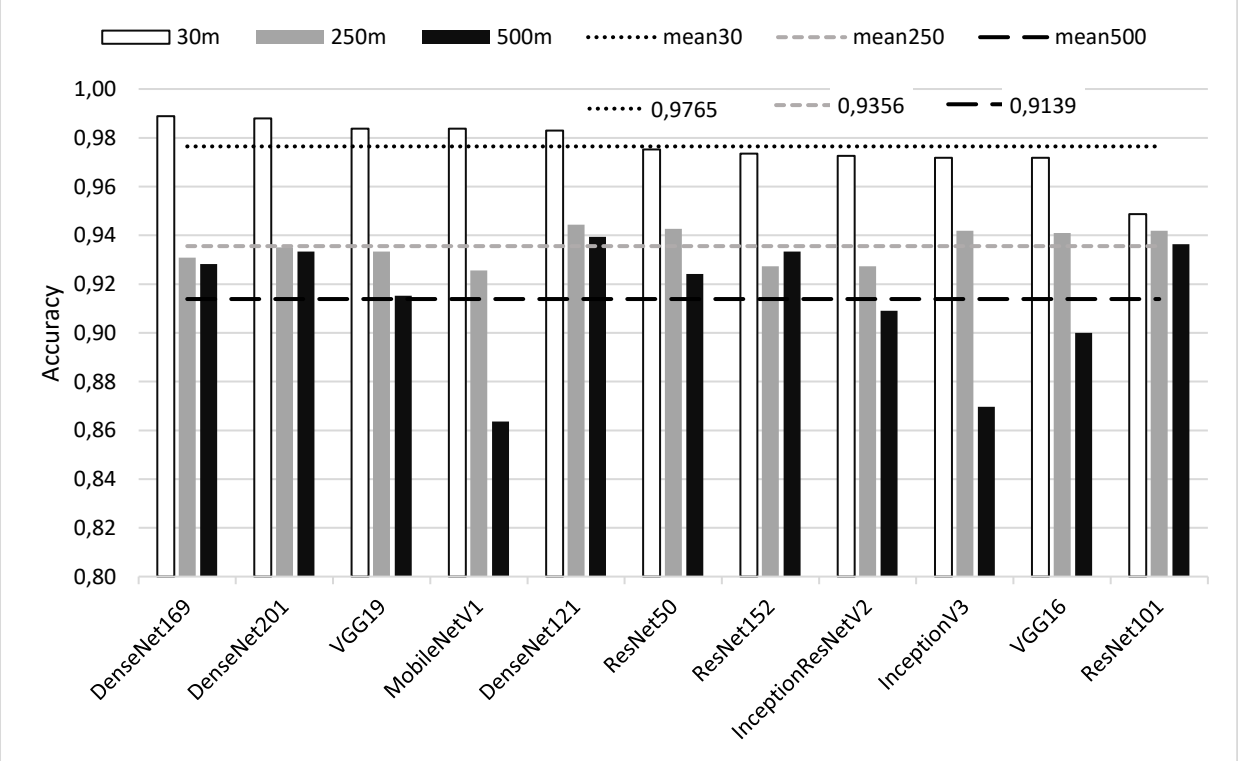


Figure 4.10: Effect of resolution on the training of 11 CNN feature extractors

The DenseNet169 classifier produced an accuracy of almost 99% for this Sentinel-2 30 m dataset. The mean accuracy over all 11 CNN models was 97.65%. By comparison to the findings in Chapter 3, a four per cent increase in accuracy was recorded when using Sentinel-2 30 m when compared to MODIS 250 m resolution data. Although DenseNet121 was the best-performing CNN model in Chapter 3, it was only the fifth-best performer on the higher resolution data, however, the training accuracy was still higher than 98%.

4.3.2 Experiment 2: additional training data

The Pretoria2 dataset contained an additional 114 CP compared to the Pretoria1 dataset, resulting in 547 CP (Table 4.1). Figure 4.11 illustrates the performances of the CNN classifiers when trained using 547 CP.

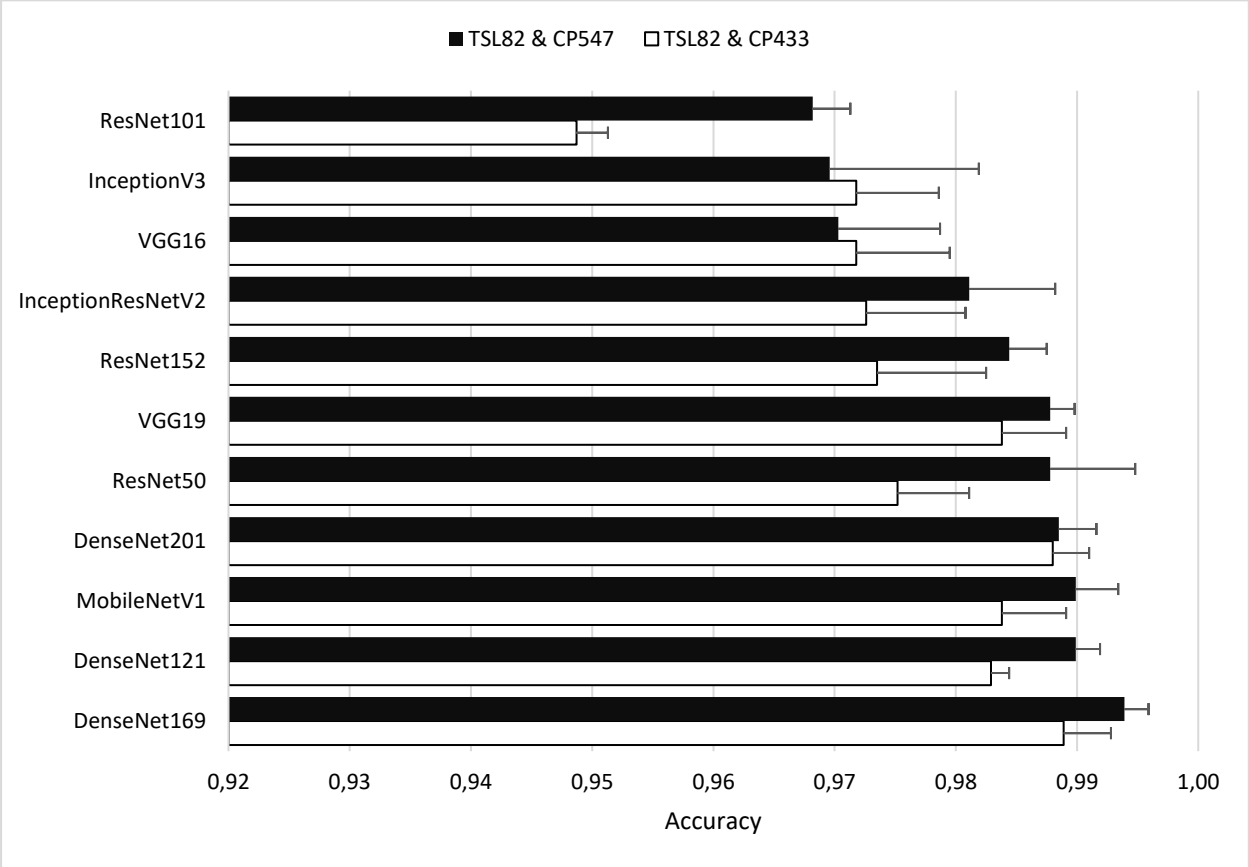


Figure 4.11: Training accuracy when using 11 CNNs with larger training set size (Pretoria 2: 547 CP; TSL 82) compared to Pretoria1 (433 CP; TSL 82)

Mean training accuracy of 98.29% was achieved over all CNN classifiers. The DenseNet169 classifier achieved the highest accuracy of 99.39%, while DenseNet121 scored 98.99%. This illustrates an 0.64% increase in the mean accuracy of the 11 CNN classifiers when additional CPs are used for training.

4.3.3 Experiment 3: TSL

The results shown in Figure 4.12 represent the individual performance of the CNN classifiers when varying the TSL during training.

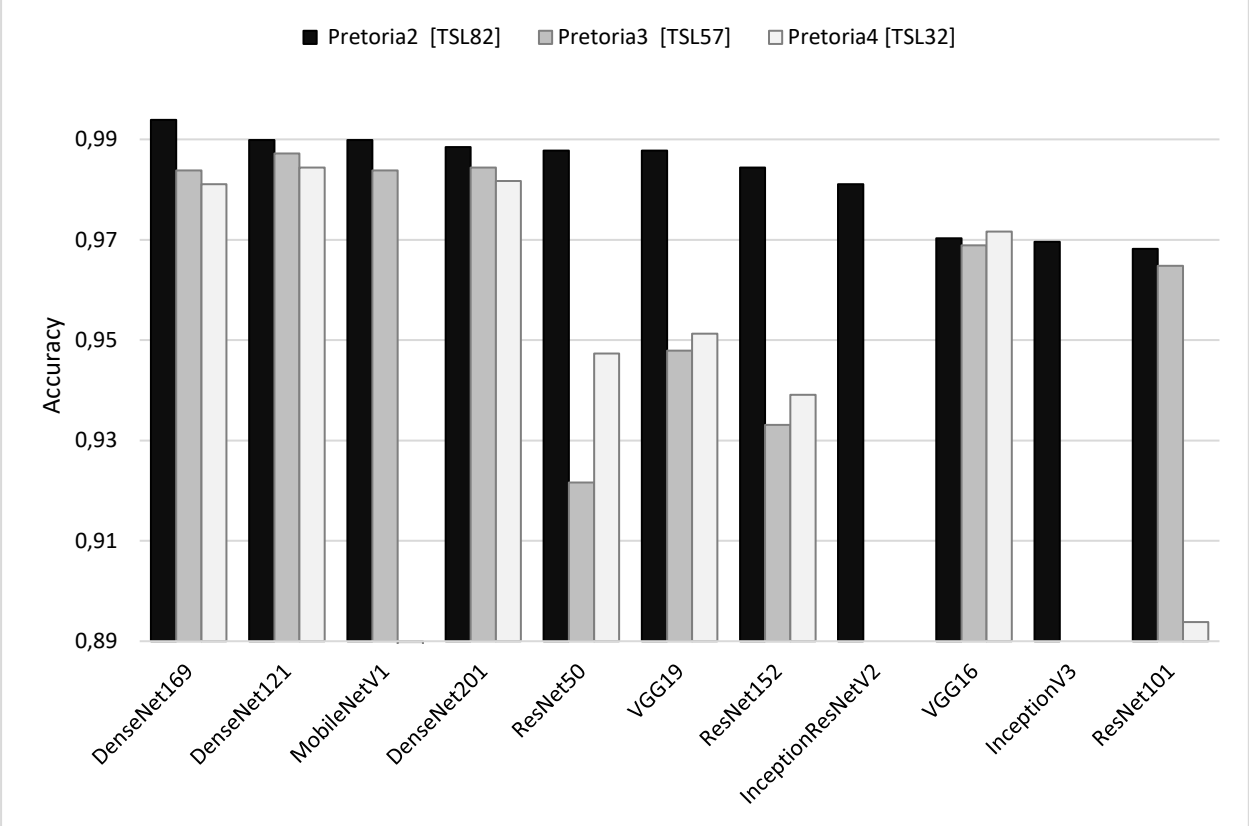


Figure 4.12: Performance of CNNs at 82, 57, and 32 TSL using the resampled Sentinel-2 30m resolution dataset with 547 CP

A mean accuracy of 93.2% was recorded when the classifiers were trained with the Pretoria4 dataset with a TSL of 32. The Pretoria3 dataset contained a TSL of 57 and achieved a mean performance accuracy of 96.4%, a more than 3% increase over the TSL 32 dataset. The mean accuracy increased a further 1.9% when trained with the Pretoria2 with TSL of 82. The DenseNet121 classifier was the top-performing CNN when considering both Pretoria4 and Pretoria3 datasets, and second-highest on Pretoria2. However, DenseNet169 outperformed both by 0.67% when trained with a TSL of 82 (Pretoria2). No results were recorded for Pretoria3 and Pretoria4 when using the InceptionV3 and InceptionResNetV2 frameworks as they require a minimum TSL of 75 in contrast to the ResNet, DenseNet, VGG and MobileNet frameworks that require a minimum TSL of 32 (Keras 2020a).

4.3.4 Experiment 4: generalisability

The results shown in Table 4.2 illustrate the generalisability of the nine CNN models tested on Pretoria2 and applied to Durban1 (TSL 65), Gqeberha1 and Khayelitsha1 (TSL 50 each). A

comparison is drawn between the application of the models for two classes (no-change, change) or three classes (urban, vegetation, change).

Table 4.2: Testing generalisability of nine CNNs on binary and three-class classifications at Durban, Gqeberha, and Khayelitsha trained on Pretoria2

	Durban$_1$ *(TSL65)*		Gqeberha$_1$ *(TSL50)*		Khayelitsha$_1$ *(TSL50)*	
Trained on 547CP	*binary*	*3-class*	*binary*	*3-class*	*binary*	*3-class*
ResNet50	0.9255	0.6687	0.8921	0.3583	0.6327	0.3773
ResNet101	0.9281	0.4417	0.8273	0.2853	0.7312	0.3662
ResNet152	0.9389	0.4458	0.9002	0.3852	0.8164	0.2810
DenseNet121	0.8466	0.6605	0.8848	0.3538	0.8063	0.4925
DenseNet169	0.8909	0.7301	0.8637	0.3762	0.7429	0.3923
DenseNet201	0.8470	0.6442	0.8925	0.3544	0.8266	0.3681
VGG16	0.9018	0.5112	0.9012	0.3986	0.7646	0.4658
VGG19	0.9230	0.6155	0.9075	0.3884	0.7863	0.5442
MobileNetV1	0.9287	0.7832	0.8999	0.3871	0.6778	0.5985
Mean	0.9046	0.6112	0.8855	0.3653	0.7661	0.4318

The performance of the CNN classifiers for binary classification is notably higher than that of the three-class classification. ResNet152 produced a high of 94% for binary when tested in Durban, while DenseNet201 produced 83% for Khayelitsha. The MobileNetV1 classifier produced the highest accuracies for three classes on Durban1 and Khayelitsha1 of 78% and 60% respectively. At Gqeberha VGG19 had a high of 91% for two classes, but only produced a high of 40% with the VGG16 classifier on three classes. Not only was there an individual increase in accuracy for a binary classification but an increase in all three mean accuracies. Gqeberha showed the largest improvement using a binary classification over the three-class classification.

Table 4.3 shows the performance for a binary classification at each testing location using the time-series datasets with seasonality removed, Durban2, Gqeberha2, and Khayelitsha2, TSL 32. DenseNet169 achieved the highest performance when testing in Durban and Khayelitsha, while DenseNet201 performed best in Gqeberha.

Table 4.3: Binary classification (no-change, change) performance of generalisability of nine CNN models for Durban, Gqeberha, and Khayelitsha

Binary Classification	$Durban_2$	$Gqeberha_2$	$Khayelitsha_2$
Trained on 547CP	*TSL32*	*TSL32*	*TSL32*
ResNet50	0.9693	0.9175	0.7510
ResNet101	0.8773	0.9133	0.7349
ResNet152	0.8344	0.9110	0.7410
DenseNet121	0.9755	0.8848	0.8279
DenseNet169	0.9767	0.9040	0.8479
DenseNet201	0.9500	0.9290	0.8378
VGG16	0.9632	0.9185	0.8199
VGG19	0.9693	0.9204	0.8260
MobileNetV1	0.8037	0.7775	0.8127
Mean	0.9244	0.8973	0.7999

4.3.5 Experiment 5: image classification

Table 4.4 represents the summarised confusion matrix for the image classifications on the Pretoria datasets (Figure 4.8 and Figure 4.9) Overall accuracy (OA), Kappa, producer's accuracy (PA) and user's accuracy (UA) are reported for classification of Pretoria7 dataset with DenseNet12, the best MODIS performer, and DenseNet169, the best Sentinel-2 performer on dataset Pretoria6. For each class (change, no change), 150 stratified random points were sampled.

Table 4.4: Confusion matrix results for multiple binary classifications using different pixel probability constraints while training and testing with the Pretoria TSL82 & CP547 dataset

Accuracy Assessment		All Pixels		Changed Pixels		No-Change Pixels	
Classifier	Probability constraints	OA	Kappa	PA	UA	PA	UA
DenseNet121	MODIS 35%+	0.9225	0.8041	0.8919	0.8250	0.9333	0.9608
	Sen2 35%+	0.9542	0.8962	0.9524	0.9091	0.9551	0.9770
DenseNet169	Sen2 50%+	0.9596	0.9160	0.9659	0.9341	0.9556	0.9773
	Sen2 70%+	0.9686	0.9344	0.9659	0.9551	0.9704	0.9776
	Sen2 90%+	0.9731	0.9432	0.9432	0.9881	0.9926	0.9640
Ensemble	Majority agreement	0.9776	0.9528	0.9545	0.9882	0.9926	0.9710
	All-in-agreement	0.9731	0.9430	0.9318	1	1	0.9574

The MODIS 250 m (Pretoria7) dataset achieved an OA of 92%. Using the higher resolution Sentinel-2 30 m data (Pretoria6), an increase of 3% was noted. As the prediction constraint

increases from 35% to 90%, the OA also increases, producing a high of 97.31% OA for the 90% constraint. The Kappa statistic demonstrated a similar pattern. However, the PA for changed pixels is inversely proportional to the OA, decreasing from 96.59% to 94.32% for constraint increase from 70% to 90%. The same pattern was seen with the user accuracy (UA) for the no-change pixels. The majority agreement ensemble classifier achieved the highest OA of 97.76% as well as the highest Kappa value of 0.9528. The all-in-agreement ensemble classifier produced the lowest PA of the five Sentinel-2 classifications.

Figure 4.13 shows a comparison of the binary classification of the 250 m Pretoria7 dataset (a) with four (b)-(e) probability constraint binary classifications (Pretoria6) within the study site (red square in Figure 4.2). The classifications (b), (c), (d) and (e) in Figure 4.13 each used different minimum-prediction probability values, increasing from 35% to 50%, then to 70% and finally 90% respectively. The white pixels illustrate change whereas the black pixels represent no change.

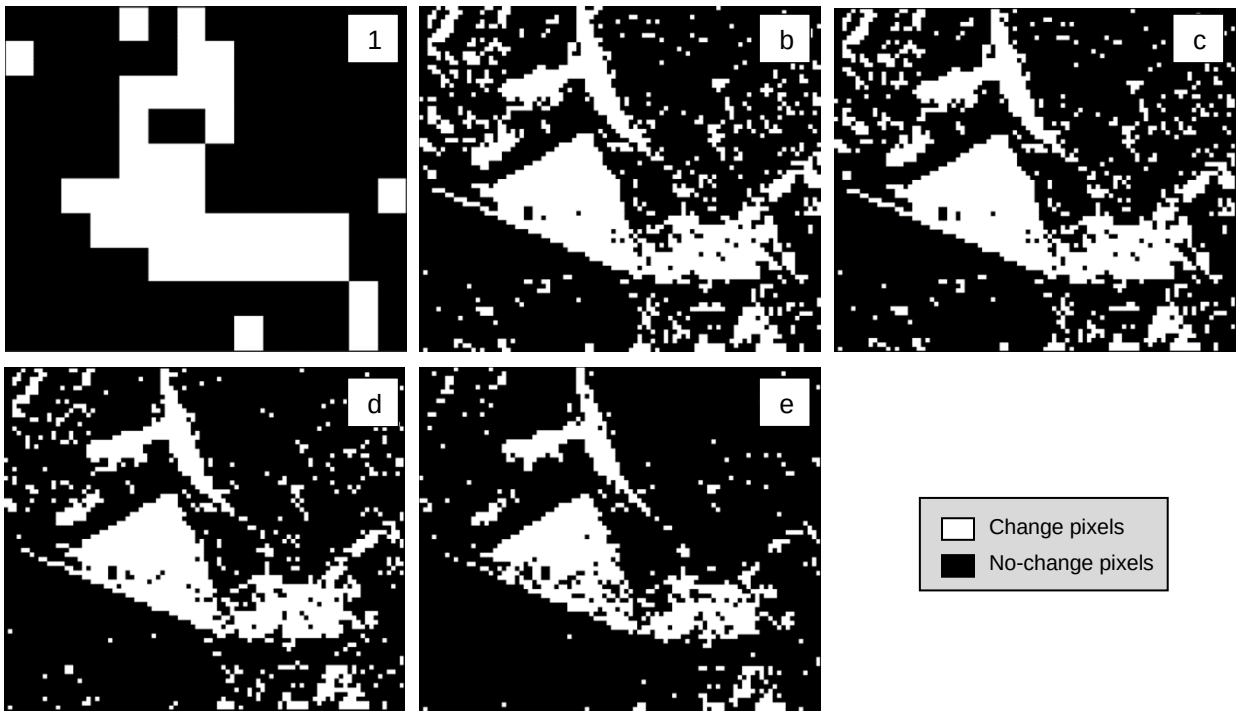


Figure 4.13: Binary classification of (a) 250 m resolution MODIS dataset (Pretoria7) and four 30 m resolution probability constrained Sentinel-2 (Pretoria6) at the study site. (b), (c), (d) and (e) represent the classification at pixel probability levels 35%+, 50%+, 70%+, and 90%+ respectively

Of the Pretoria7 dataset in Figure 4.13(a), 39 pixels were classified as change, an area of 2.4 km^2. For the 35%+ classification (b), 2047 changed pixels (1.8 km^2) were recorded. This is 0.6 km^2 less than that of the MODIS classification (a). The 90%+ MPP classification (e) contains far fewer changed pixels than the 35%+ classification (b), with only 1 237 pixels (1.1 km^2) in the change class. This area is less than half of the changed area in the MODIS classification (a).

The two ensemble image classifications shown in Figure 4.14 are binary classifications using (a) the majority agreement and (b) the all-in-agreement ensemble classifier. The majority agreement is the top classifier and has the most accurate display of change pixels.

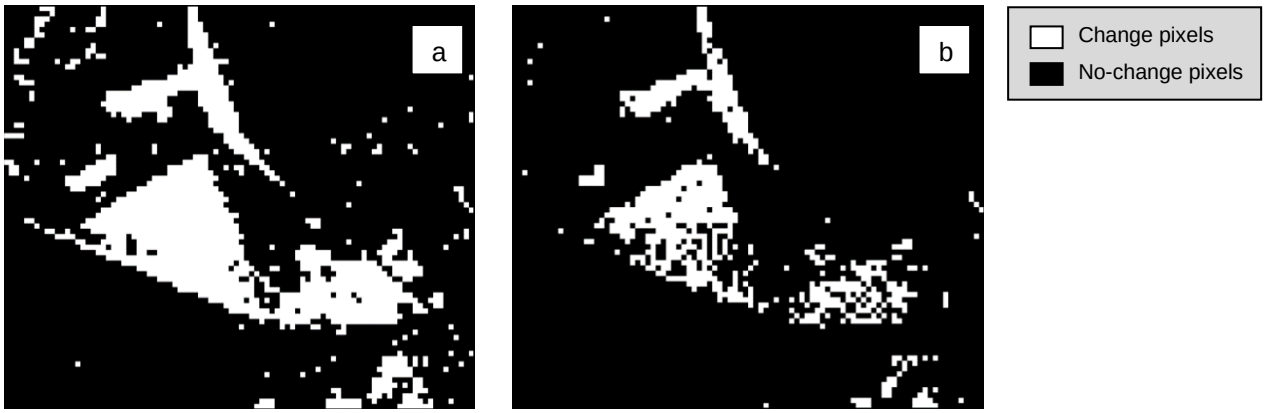


Figure 4.14: Ensemble classification representing (a) majority agreement and (b) all-in-agreement

The all-in-agreement classification (Figure 4.14b) displays fewer changed pixels. From visual inspection, the all-in-agreement classification consists of a large proportion of missed classified no-change pixels. The evidence of this is shown in Table 4.4, where the UA achieved for no-change pixels is the lowest of all six classifications. The majority-agreement ensemble classification has the highest OA and Kappa (Table 4.4) and is therefore the top-performing classifier. Figure 4.15 shows a subset of the majority-agreement classification (Figure 4.14a) illustrating for each pixel the probability level class obtained from the ensemble classification.

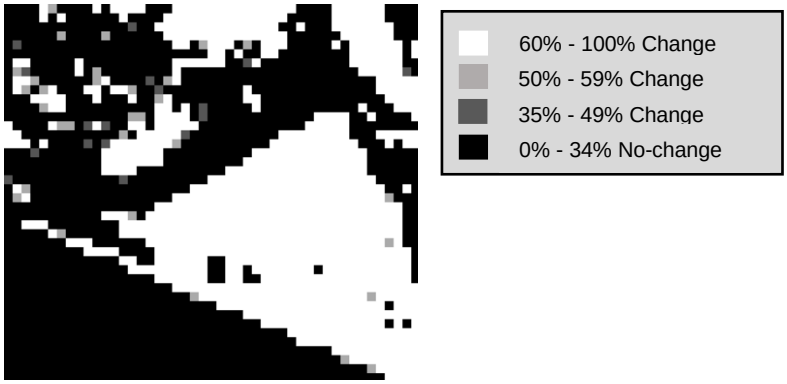


Figure 4.15: Pixelwise probability-level classification

The black pixels classified as no change have less than 34% probability of change, while the white pixels were predicted as having >60% probability of change by six or more of the CNN models. Scales of grey represent probabilities between 35% and 59%, bordering larger bodies of change pixels, however, these make up a tiny proportion when compared to 60%+ pixels.

Table 4.5 presents the confusion matrix computed from an accuracy assessment applied to the model generalisability experiment. OA, PA and UA are recorded at Durban, Gqeberha and Khayelitsha for all data.

Table 4.5: Confusion matrix for binary classifications demonstrating generalisability at the three testing locations using both the normal and seasonality removed datasets

Accuracy Assessment		All Pixels	Changed Pixels		No-Change Pixels	
Seasonality	Datasets	OA	PA	UA	PA	UA
Normal	$Durban_1$	0.9207	1	60.61	90.97	1
	$Gqeberha_1$	0.8848	0.4783	1	1	0.8712
	$Khayelitsha_1$	0.7771	0.1654	1	1	0.7668
Removed	$Durban_2$	0.9877	1	0.9091	0.986	1
	$Gqeberha_2$	0.9155	0.6174	1	1	0.9022
	$Khayelitsha_2$	0.7871	0.203	1	1	0.7749

The OA reflects similar results for the binary classification to those in Table 4.2 and Table 4.3. However, the PA for predicting changed pixels at Gqeberha and Khayelitsha is very low, showing that the model was not able to map the situation on the ground. The high UA shows that all changed pixels in the reference data were identified for these two locations. The confusion matrix confirms that the models were not able to generalise well to different locations from Pretoria training data.

4.4 DISCUSSION

This study converted per-pixel high-resolution Sentinel-2 NDVI time-series data into matrix representations, GAF and MTF images, from which features were extracted to represent urban, vegetation and change classes using a series of pre-trained CNN classifiers. The resolution, TSL and CP played an important role in the success of change detection. Classifications at different probability levels were explored, while the effect of using an ensemble of CNNs for classification was analysed.

Using three-channel concatenated encoded images generated from the GASF, GADF and MTF transformations, 11 CNN classifiers were trained and validated. The Pretoria1 dataset contained 433 CP (Table 4.1), selected to directly compare the MODIS 250 m (Chapter 3) and Sentinel-2 30 m results. In Chapter 3, a mean accuracy of 93.56% was achieved using the MODIS 250 m dataset, with the DenseNet121 feature extractor achieving the highest accuracy of 94.44%. By comparison, the Pretoria1 dataset showed a 4% increase in mean accuracy (Figure 4.10). DenseNet169 was the top-performing feature extractor with an accuracy of 98.89 (Figure 4.10). Utilising the same framework and number of CPs in training allows a direct comparison of datasets. The higher-resolution dataset (Sentinel-2 30 m) consistently achieved higher accuracies and outperformed the MODIS 250 m dataset. This follows the results from Chapter 3 (MODIS 500 m vs MODIS 250

m) and agrees with Kleynhans, Salmon & Wessels (2017). In this study, the increase in resolution positively affected the accuracy, and higher-resolution datasets are recommended.

When implementing DL algorithms such as CNNs, the training data plays a significant role in the success of the classification (Campbell & Wynne 2011). The use of higher-resolution imagery provides grounds for gathering additional pixels. With the ability to increase the number of CPs used in training and the sensitive nature of the algorithms, the Pretoria2 dataset consisted of 114 additional CPs (25% more training points) which led to an increase in mean accuracy of 0.64% (Figure 4.11). DenseNet169 remained the top-performing feature extractor and obtained a new individual high accuracy of 99.39% (Figure 4.11). The experiment proved that an improvement in model performance can be induced by increasing the sample size of the training data (Cho et al. 2015; Dunnmon et al. 2019; Zhong et al. 2018)

A multi-temporal change detection requires data from more than two points in time (Campbell & Wynne 2011). The length of the time series generated has been shown to affect the performance of the algorithms (Fonseca-Pinto et al. 2009; Hills et al. 2014). With the limited time frame of available data from the Sentinel-2 Level-2A dataset, a maximin TSL of 82 was assigned to Pretoria1 (Table 4.1), however, the TSL was reduced to 57 and 32, whilst keeping the same number of CPs. The feature extractors achieved significant decreases in overall accuracy when processing the additional two datasets. Mean accuracy of 96.39% and 93.22% were recorded for the TSL 57 and TSL 32 respectively, compared with 98.29 for TSL 82. A noteworthy comment is that the minimum input image size for most of the CNNs (ResNet, DenseNet, VGG, MobileNet) is 32x32 (Keras 2020a). When utilising the framework of encoding time series as 2D images, a minimum TSL of 32 is required. However, the InceptionV3 and InceptionResNetV2 required a TSL of 75 to be processed, hence the *null* values for Pretoria3 and Pretoria4 datasets (Keras 2020a). As found by Fonseca-Pinto et al. (2009) and Hills et al. (2014), this study found that the performance of the feature extractors is strongly dependent on the TSL of the training data.

Using the top-performing feature extractor (DenseNet169) with the optimal training dataset (457CP & 82TSL), a classification of unseen data was conducted, assigning pixels to respective classes based on varying minimum probability constraints. A baseline classification was established by utilising the traditional method of assigning pixels to the class with the highest probability (Figure 4.13b). The baseline OA of 95.42% (Kappa 0.8962) was computed from 150 random stratified points per class (Table 4.4). Further classifications employed a 50% and 70% MPP constraint, which both showed slight improvements across the evolution board (OA, PA, UA, and Kappa) compared to the baseline (Table 4.4). The 90% MPP classification achieved the highest OA and Kappa at 97.32% and 0.9432 respectively (Table 4.4). This was expected as the

classification requires extremely high pixel confidence from the DenseNet169 feature extractors predictions. Comparing the four change detection images in Figure 4.13b-d, it is clear that as the MPP increases, there is a decrease in the speckle and misclassification. Although the recorded OA was the highest, the PA for change pixels (CP) had decreased by 2.27% compared to the 70%+ MPP classification (Table 4.4). Although the OA may be high, it is critical to evaluate the PA of the changed pixels when determining the success of the change detection. A trade-off between OA, PA and UA is needed to determine the top classification scheme, as seen in Figure 4.16.

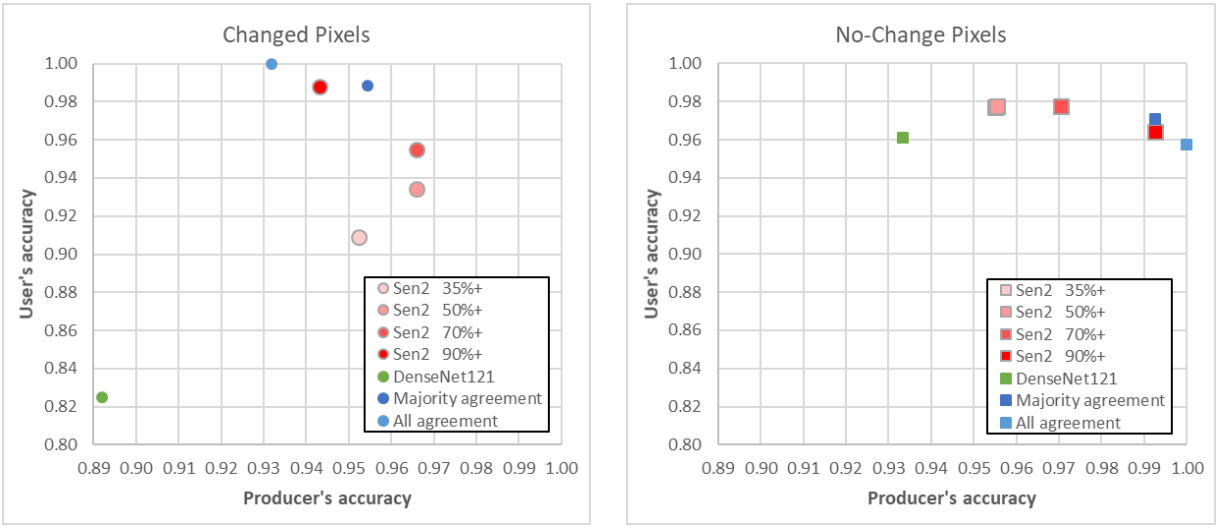


Figure 4.16: Comparison of PA and UA for changed and no-change pixels

In the context of this study, it was concluded that the 70% MPP classification produced superior change detection. Research has shown that an ensemble of CNN classifications may increase the performance (Chen et al. 2017; W Li et al. 2019; Vasan et al. 2020). Further investigations into two ensembles of CNN classifiers confirmed that a majority agreement rule, where six of the 11 CNNs must agree on a class assignment, produced the highest OA record in the study at 97.76% (Table 4.4). The important accuracy to note was the PA of 95.45% of the changed pixels (Figure 4.16). The second ensemble applied a simple all-in-agreement rule, where all 11 feature extractors had to agree on the class to which the pixel would be assigned. Although this ensemble achieved a high OA, the PA for changed pixels was significantly lower than all other change detections (Figure 4.16). This was due to the optimistic rule that all 11 CNNs had to agree with each other. This study used several diverse DL architectures with varying numbers of convolution layers and parameters. As a result, it is unlikely that all will agree.

The conclusion was made that Sentinel-2 30 m resolution datasets outperform MODIS 250 m in change detection, based on the premise that feature extractors in Chapter 4 achieved higher accuracy results than those in Chapter 3. This was confirmed by a direct comparison of a MODIS 250 m dataset processed over the same classification site (Figure 4.2). As expected, the accuracy of this classification (Table 4.4) is significantly lower and cannot compete with that of the Sentinel-

2 datasets. This is due to the significant decrease in mixed pixels, which is part of the inherent nature of using higher-resolution imagery (Campbell & Wynne 2011).

The final experiment undertaken focused on testing the generalisability of the feature extractors using unseen data (Samala et al. 2019)from three testing locations (Durban, Gqeberha, and Khayelitsha) through a three-class classification as well as a binary classification. The model validation results for three classes was very low with all sites achieving less than 62% accuracy (Table 4.2). These accuracies increased to above 76% when a binary classification was tested (Table 4.2). It was clear that the feature extractors struggled to distinguish between no-change vegetation and no-change urban classes at foreign locations. Merging these classes and evaluating the performance based on binary (change verse no-change) classification improved accuracy. Removing the seasonal pattern of the time series (Hamzaçebi 2008; Zhang & Qi 2005) further improved the mean model evaluations to above 80% (Table 4.3), however, the confusion matrix (Table 4.5) of the image classification shed a different light on the generalisability results.

Both Gqeberha and Khayelitsha locations achieved a low PA for the changed pixels with 48% and 17% respectively (Table 4.5). Removing the seasonality in the time series produced slight improvements, resulting in a PA of 20% and 62% for Gqeberha and Khayelitsha respectively (Table 4.5). As the removed seasonality data consistently achieved higher accuracies, this showed potential as a pre-processing requirement for model generalisation (Hamzaçebi 2008; Zhang & Qi 2005). The performance of the feature extractors at the Durban location showed much higher accuracies and a PA of 100% was achieved (Table 4.5). Although the model evaluations in Table 4.2 and Table 4.3 and the OA in Table 4.5 are relatively high, PA for the classification indicated the actual performance of the change detection at each location.

The nine feature extractors were unable to generalise for the Gqeberha and Khayelitsha locations. Seasonality, solar angle variation, landcover complexity, climatic conditions, and the distance from the training scene are all major factors that allow for the generalisation of the models (Olthof, Butson & Fraser 2005; Phalke & Özdoğan 2018; Verhulp & Van Niekerk 2017; Woodcock et al. 2001). Although the seasonality was accounted for, the CNN models could not overcome the remaining factors. Khayelitsha, located furthest from the training site, with additional ecological, topographic and climatic differences, resulted in the worst-performing change detection (Verhulp & Van Niekerk 2017). An increase in geographical distance in the north-south direction would result in poorer performance than the distance in the east-west direction (Olthof, Butson & Fraser 2005). This would explain the improved performance of the Gqeberha datasets over that of the Khayelitsha sites. The geographical distance will also alter the climatic and ecological conditions (Phalke & Özdoğan 2018). A combination of these factors may explain why the feature extractors

were able to generalise well on the Durban datasets. The data acquired in Durban had similar ecological and climatic properties to Pretoria (e.g. summer rainfall) and was located the closest to the training data (Olthof, Butson & Fraser 2005; Owolawi, Afullo & Malinga 2009). Studies have suggested a framework of performing a signature extension in order to increase the generalisation of models (Q Li et al. 2020; Olthof, Butson & Fraser 2005; Phalke & Özdoğan 2018; Verhulp & Van Niekerk 2017). This proposed framework extracts signatures from known features to perform successful classification (Olthof, Butson & Fraser 2005). Gathering portions of training data from the testing locations and covering all climate zones will increase the generalisation of the feature extractors (Q Li et al. 2020). A recommendation for future studies is to adopt the signature extension framework for a generalisable model.

4.5 CONCLUSION

In this chapter, the framework of encoding time-series data as 2D images for multiple CNN classification using Sentinel-2 30m resolution imagery was investigated. A comparison experiment between MODIS 250 m and Sentinel-2 30 m data for urban change detection was conducted. As expected, the Sentinel-2 30 m datasets outperformed the MODIS datasets, achieving significantly more accurate results. Further results were presented on the effects of the TSL and the number of CPs used for training. It was concluded that model performance was directly proportional to the number of CPs used in training and the TSL of the dataset. It is recommended to process datasets with a longer TSL and as many training samples as possible. Multiple classification schemes were investigated, and a 70%+ MPP was selected as the optimal pixel classification constraint alongside the DenseNet169 feature extractor. A majority agreement ensemble classification scheme produced competitive results for the 70%+ MPP classification. Concerning the classification OA, the majority agreement ensemble classifier produced the top-performing change detection. However, the 70%+ MPP classification achieved the highest PA for changed pixels with slightly lower OA. Both classification schemes are highly effective and compare favourably with other urban change detection methods tested in South Africa. The generalisability of the models was investigated at three testing locations: Durban, Gqeberha, and Khayelitsha. Disappointing results were found at Gqeberha and Khayelitsha. Due to ecological, topographic and climatic differences and the geographical distance in the north-south direction, the models at these locations did not generalise well. However, satisfactory results were achieved at the Durban testing site. The study found that converting time-series data into GAF and MTF images, followed by feature extraction using pre-trained CNN classifiers could successfully be used for change detection in an urban context.

CHAPTER 5: DISCUSSION AND CONCLUSION

This chapter will reflect on the research aim and objectives and summarise the findings from Chapters 3 and 4. The findings from encoding time series data for a CNN classification to perform an urban change detection are discussed alongside the limitations that emerged during the experiments. This chapter will also provide suggestions for future research and a summary of conclusions drawn.

5.1 REFLECTION ON RESEARCH OBJECTIVES

This research aimed to evaluate the potential of encoding time-series data as 2D images from MODIS and Sentinel-2 for urban change detection through classification with complex neural networks. Multiple per-pixel time-series datasets of different resolutions (500m, 250m and 30m) were collected using the MODIS and Sentinel-2 instruments. All the datasets were encoded using two Gramian angular fields (GADF and GASF) and the Markov transition fields. The 2D encoded images were then processed through several different pre-trained CNN architectures, and the results evaluated. After analysis, the top-performing CNN architecture for the respective datasets was selected to perform a classification to conduct an urban change detection. Accuracy assessments of each classification allowed comparison between the performances of the urban change detections. The main objective of this research was to investigate the methodology of performing urban change detection through a framework of encoding time series data as a 2D image for a CNN classification.

The first objective was to review the literature on the background concepts and principles of RS, image classifications and land cover change detection that primarily focused on urban areas (Chapter 2). A review of the literature for AI, different DL classification techniques and novel computer vision (CV) methodologies were also carried out in Chapter 2. The literature review showed the success of applying ML and DL classifications algorithms for urban change detections and the recent development in the CV technique of encoding time series data as 2D images. The literature showed that combining a DL classification algorithm (i.e. CNN) and the novel CV framework of encoding holds potential for RS applications. Several methodologies, including the CNN algorithm, have been deployed for urban classifications and change detections. Although a previous study has shown the success of implementing the novel CV framework of encoding time series data for an RS application, none have deployed this methodology for urban change detection.

Experiment 1 (Chapter 3) consisted of two objectives (2.a and 2.b). The purpose of this experiment was to evaluate the novel framework of encoding time series data as 2D images for a specific RS

application such as urban change detection. Objective 2.a specified that the novel CV framework of encoding would be assessed and evaluated based on its effectiveness to perform an urban change detection when the encoded 2D images are processed through multiple pre-trained CNNs. Objective 2.b was set in place to make a comparison between the performance of the novel framework and a baseline classification approach using the LSTM algorithm. Therefore, in Chapter 3, the novel framework was evaluated and compared to other state-of-the-art approaches.

Experiment 2 falls within the scope of Chapter 4 and consists of three additional objectives (3.a, 3.b and 3.c). Objective 3.a was set out to evaluate the effectiveness of increased spatial resolution on the framework proposed in Chapter 3. Chapter 4 experimented with higher resolution Sentinel-2 imagery (resampled to 30m) and illustrates its effect on urban change detection performance. Objective 3.b assessed the consequences of altering the temporal nature of the input time series. Although Sentinel-2 has a higher spatial resolution, it has a lower temporal resolution when compared to that of the MODIS instrument. Sentinel-2 also has less available imagery as it is only available from 2019, and the coastal areas present extensive amounts of cloud cover. All these factors have an impact on the temporal nature of the generated time-series. The proposed encoding framework is strongly influenced by the length of the input time-series, and it was important to assess the consequences of altering the temporal nature. Objective 3.c was derived to evaluate the generalisability of the proposed framework when testing with data from three different geographical locations. A critical aspect of developing an effective model is to test the generalisability using unseen data. Issues were expected. However, it was essential to understand how the model works and performs with unknown data. Further actions could then be implemented to increase the performance and generalisability of the framework.

The final objective of this research was to synthesise the results of the two main experiments (Chapter 3 and 4) and bring to attention the limitations that were found in order to make further recommendations for performing an urban change detection using the computer vision technique of encoding and the DL classification apparatus (i.e. CNN) (Chapter 5).

5.2 SYNTHESIS OF FINDINGS

The two experiments, which made up Chapters 3 and 4, were set out to investigate the effectiveness of deploying the CV technique of encoding time series data as 2D images for urban change detection. Multiple pre-trained CNN algorithms were used to perform the change detection classifications alongside several datasets with varying spatial and temporal resolutions. Several classification schemes were used in Chapter 4 to determine the optimal performance for urban change detections. Three different testing locations were used to test the generalisability of the proposed framework.

5.2.1 Application of time-series encoding to coarse resolution imagery

The novel framework proposed by Wang & Oates (2015), which used GAF and MTF encoding transformations for converting time series data into 2D images for a CNN classification, was implemented and then evaluated for its effectiveness in performing urban change detection. Focusing on the per-pixel time-series data derived from 500m resolution NDVI MODIS imagery and collected within the Gauteng province of South Africa, the three encoding transformations (GASF, GADF and MTF) were applied. Coinciding with the literature (Dias et al. 2020; Yang, Chen & Yang 2020), all three encoding transformations were processed, and a fourth concatenated image was formed containing the information from all three transformations. As suggested by Dias et al. (2020), several pre-trained CNN architectures (ResNet, DenseNet, InceptionV3, InceptionResNetV2, VGG and MobileNet) were deployed for multiple classifications. The mean accuracy from the concatenated encoded images outperformed the three remaining transformations, as expected (Dias et al. 2020). DenseNet121 was the single top-performing CNN algorithm at 93.94% accuracy using 500m resolution imagery. The second experiment in Chapter 3 utilised 250m resolution NDVI MODIS data and it was processed using the same framework as before. Although, in this experiment, only the concatenated encoded images were used, due to their superior performance with the 500m resolution dataset (Dias et al. 2020). The results achieved using the 250m resolution data were in line with the findings of Kleynhans, Salmon & Wessels (2017). They illustrated that increasing the spatial resolution would increase the performance of the urban change detection. The mean accuracy for all 11 CNN architectures increased from 91.39% to 93.56% when the 250m resolution dataset was implemented. DenseNet121 remained the top classifier for the second dataset as well and yielded an accuracy of 94.44%.

Several studies were identified to illustrate the success that DL techniques in performing change detection and urban change detection (Daudt et al. 2018; Zhan et al. 2017). However, no apparent literature can stipulate the success of encoding time series data for urban change detection. Working on the premise that change detections have successfully been conducted using multi-temporal MODIS NDVI imagery (Grobler et al. 2013; Kleynhans et al. 2012; Kleynhans et al. 2015; Lunetta et al. 2006), a comparison was made between the novel framework and a current state-of-the-art time series classifier. The LSTM classifier was selected to perform a baseline classification using 250m resolution per-pixel MODIS NDVI time-series data. The results from the LSTM classifier were successful and yielded competitive accuracies to that of the findings in other studies that performed urban change detection within South Africa (Grobler et al. 2013; Kleynhans et al. 2012; Kleynhans et al. 2015). However, the accuracies from the LSTM could not

rise and match that of the proposed framework, for which the same multi-temporal data was encoded and processed through a CNN classifier. With the locality of the testing site within proximity to that of Kleynhans, Salmon & Wessels (2017), a fair comparison between methodologies was made and the proposed framework deployed in this research outperformed the existing study (Kleynhans, Salmon & Wessels 2017) for the Gauteng province of South Africa.

A noteworthy comment regarding the CNN loss values would assist in a deeper understanding of the classification results. To avoid overfitting, the loss value, recorded alongside the accuracy as the model runs, should be considered as the spatial resolution increases. Several limitations were noticed while deploying the proposed framework. The first major issue was that the models did not generalise well. All CNN models were trained using data from one location and tested on unseen data from a separate location (Maputo). The performance result was significantly lower, and urban change detection was not possible. This issue regarding model generalisability is discussed again with the respective results from Chapter 4. A limiting factor of running this proposed framework with data derived over ten years was the computing hardware on which the algorithms were processed. The random-access memory (RAM) capped out at 32 Gigabytes when processing encoded images derived from time series for large TSLs. The exponential nature of the transformations resulted in significantly large, encoded images, which throttled the central processing unit (CPU), graphics processing unit (GPU) and RAM.

5.2.2 Factors affecting accuracy and generalizability of urban change detection method

Following on from Chapter 3 and the success yielded by the proposed methodology, several experiments were set out to optimise the performance of the classification and potentially maximise the accuracies of the urban change detections. Chapter 4 used the same methodology to encode per-pixel time-series remote sensing data prior to a CNN classification as in Chapter 3. However, the prior knowledge that the concatenated encoded images outperformed the three remaining transformations affected the desired input dataset. The relationship between the spatial resolution and the performance of the change detection was further investigated (Chapter 3, Kleynhans, Salmon & Wessels 2017), especially to limit the mixed pixel effect of a lower resolution dataset (250m). Various open source platforms allow for easy access to high-resolution imagery (Daudt et al., 2018). As a result, the first experiment set out to evaluate the performance of the proposed framework using Sentinel-2 imagery resampled to 30m. The 30m resolution dataset outperformed both the 500m and 250m datasets by achieving a mean accuracy of 97.65%. That corresponds to a 4.09% and 6.26% increase in accuracy compared to the 250m and 500m resolution datasets, respectively. All three classifications utilised the same number of changed pixels (433) to allow for direct comparison between the spatial resolution the performance of the

classification. The trend once again corresponds to findings by Kleynhans, Salmon & Wessels (2017). The second experiment investigated the effect of increasing the training samples and its impact on classification accuracy. As expected, the results compared well with the findings in literature (Cho et al. 2015; Dunnmon et al. 2019; Zhong et al. 2018)where an additional 25% of changed pixels used in training allowed for a 0.64% increase in classification accuracy.

The resolution and specified training data are both critical elements to consider when trying to increase the change detection performance. An essential factor to consider when deploying this encoding framework is the length of the input time-series (temporal nature). Previous research has illustrated that the TSL may play a role in the performance of an algorithm (Fonseca-Pinto et al. 2009; Hills et al. 2014). As a result, experiments were set out to investigate how the TSL affects the proposed framework and the resulting change detection. By using GASF, GADF, and MTF transformation, the resolution of the encoded image directly relates to the temporal nature of the input time series. The greater the TSL of the input data, the greater the resolution of the encoded image. The experiment conducted in Chapter 4 illustrated that the classification performance would decrease as the TSL of the input data decreases. This was a critical aspect to understand, as it affects the choice of input data. The temporal nature of the derived time series is critical for the success of change detection using the proposed framework. Kleynhans, Salmon & Wessels (2017) stated that the temporal aspect of the data does not play a significant role in altering the performance unless the time stamp between images is greater than two months. The findings from Chapter 4 suggest otherwise and contradict the initial part of that statement. Any increase in the temporal nature of the input data will play a beneficial role in the outcome of the change detection.

Although Sentinel-2 has a higher temporal resolution than its competitor Landsat-8, the time from when the data was available becomes the issue. The results have shown that the performance increase can be derived by utilising a dataset with a greater TSL. However, with the high cloud cover at all locations and restricted Sentinel-2 data, the TSLs were limited to a maximum of 82, 65, 50 and 50 for Pretoria, Durban, Gqeberha, and Khayelitsha, respectively. This may increase the likelihood of selecting Landsat-8, as it has more available data.

The generalisation of the models was briefly touched upon in Chapter 3. Additional experiments in Chapter 4 test the application of the algorithms to unseen data from different locations. All models were trained using data derived from Pretoria and tested with unique datasets. Due to localised climates, the TSL for all testing locations was ±35% lower than the Pretoria dataset. However, the CNN algorithms produced moderately high OA’s. These accuracies were further increased by 2% - 3.5% when the seasonal pattern was removed. However, the OA’s do not represent the change detection performances at these testing locations. When focusing on the PA

of the change pixels, Durban achieved a 100% accuracy, whereas Gqeberha and Khayelitsha produced significantly lower PAs of 61.74% and 20.3%, respectively. As a result, Durban was the only testing location with effective urban change detection. The results yielded at the other two locations showed that the model did not generalise well and could not successfully detect urban change. The suggested reason for the poor performances relates to the geographic distribution of the sites and the distance from the training sites. The geographic distance between the sites results in altered climatic and ecological conditions (Phalke & Özdoğan 2018; Verhulp & Van Niekerk 2017). Although seasonality was accounted for, the CNN models could not overcome all the factors that play a role in spectral signature separation. Adding training data gathered from all locations would drastically help the models generalise (Q Li et al. 2020). The generalisability of the framework remains a limitation to this research. An additional limitation is that model evaluations are insufficient to determine the success of the model, and a confusion matrix is required to formulate PA and UA for each class. This is an essential step in verifying the performance of the urban change detections.

The final classification is a result of applying a MPP value to assign a pixel to a class. With varying MPP constraints, 35%+, 50%+, 70%+ and 90% +, four different classifications were produced. The 90% MPP achieved the highest OA although, the PA for CP was too low. The best overall performing classification constraint was the 70%+ MPP, as it had a high OA of 96.86% and the highest recorded PA of 96.59%. It was clearly illustrated that the number of misclassified changed pixels decreased as the MPP increased. Two ensemble classification schemes were also deployed; the majority agreement ensemble scheme yielded competitive results to the 70+ MPP classification scheme. It was able to produce a 97.76% OA with a 95.45% PA. The typical trend in all six different classification schemes is a trade-off between the OA and the PA of the change pixels. Depending on what an individual is looking for, they need to make a judgment call on selecting the appropriate classification scheme. However, a limitation for performing the ensemble classification was that all 11 CNN architectures needed to be trained and run, whereas the 70%+ MPP only requires the DenseNet169 algorithm.

5.3 SUGGESTION FOR FUTURE RESEARCH

It is essential to identify the limitations of the research and make suggestions for any future research. With the limitation mentioned in the previous two sections (5.2.1 and 5.2.2), recommendations will be made within this section.

All datasets for Chapters 3 and 4 use NDVI as the source, as suggested and implemented in several other studies. However, other indices and individual bands have also been effective in mapping urban change (NDBI, RNDSI and Red band) (Chapter 2). This is a noteworthy aspect, and these

indices and bands should be tested alongside NDVI in future research. As shown in Chapter 2, SAR imagery has proven valuable data sources for monitoring and mapping urban changes. It is recommended for future studies to potentially process SAR imagery using this framework or perform a data fusion of optical and SAR imagery prior to encoding the time-series. Combining state-of-the-art indices with SAR imagery could potentially have a significant impact on the performance of urban change detection. One of the main limiting factors identified within the scope of this research was the temporal nature and TSL of the derived from Sentinel-2 datasets in Chapter 4. One should therefore consider using a high-resolution instrument that has been active for a longer period so that the TSL is no longer a limiting factor. The research showed that the increase in the TSL of the input data would be beneficial to the performance of the urban change detection.

The removal of the seasonal pattern from the original time series proved successful in increasing the accuracies of the change detection. However, additional research into removing the seasonal trends is highly recommended for any future studies attempting to generalise the change detection models. Investigation into signature extension is also highly recommended to increase the generalisability of the models. By using small portions of training data from all testing locations, the models will transfer and generalise at a significantly high rate.

5.4 CONCLUSION

Urban change detections have successfully been implemented in previous studies and shown to be beneficial for local municipalities and regional governments for urban decision making and planning. Urban settlements are growing at alarming rates as people move closer to potential employment opportunities. Developing a fast and effective framework to accurately monitor urban expansion will allow regular updating of urban land cover data and valuable information about urban development. Autocorrelation functions currently stand as the leading methodology for performing a successfully urban change detection within South Africa. Although, despite the success, there was room for improvement. The advancement of scientific knowledge is essential and often requires collaboration between fields. The interdisciplinary scientific field of computer vision potentially possesses several ideologies and methodologies that have not yet been applied to RS.

This study investigated using a novel CV methodology that encoded time series data using GASF, GADF and MTF transformations as 2D images for a DL CNN classification. The proposed framework has yet to be been deployed in the field of RS to perform urban change detection. The focus of the study was first to evaluate the novel framework and assess the effectiveness of performing urban change detection. The encoding aspect was validated by comparing the

performance of an LSTM classifier using the original time-series data to the results yielded by the novel framework. The novel framework and CNNs outperformed the LSTM classifier and the current state-of-the-art urban change detection deployed in South Africa. This thesis demonstrated that the proposed CV framework alongside DL classifiers could effectively detect the urban change and track the changes. Addition experiments were set out to investigate the effect of the input dataset on the performance. The spatial and temporal resolutions both played a significant role in increasing the accuracy of the change detection. It was concluded that they possessed a directly proportional relationship to the framework's performance. Although the research presented great success, the models were unable to transfer and generalise well. More research is needed to fully optimise the performance and deploy the framework at an operational level.

The information displayed in this research should help convince others to branch out and investigate ideologies and methodologies developed within other fields, requiring the collaboration of knowledge. The success found within this thesis illustrates the new and improved method for performing highly accurate and effective urban change detection. This knowledge could help city planners, developers, and local municipalities understand and monitor urban expansion within South Africa.

REFERENCE LIST

Abbas A, Khan S, Hussain N, Hanjra MA & Akbar S 2013. Characterizing soil salinity in irrigated agriculture using a remote sensing approach. *Physics and Chemistry of the Earth* 55–57: 43–52.

Abdel-Hamid O, Mohamed A & Jiang H 2014. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Trans* 22: 1533–1545.

Akar Ö & Güngör O 2012. Classification of multispectral images using Random Forest algorithm. *Journal of Geodesy and Geoinformation* 1, 2: 105–112.

Al-Bilbisi H 2019. Spatial monitoring of urban expansion using satellite remote sensing images: A case study of Amman City, Jordan. *Sustainability (Switzerland)* 11, 8.

Albawi S, Mohammed TA & Al-Zawi S 2018. Understanding of a convolutional neural network. *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017* 2018-Janua: 1–6.

Alexander C 2020. Normalised difference spectral indices and urban land cover as indicators of land surface temperature (LST). *International Journal of Applied Earth Observation and Geoinformation* 86, November 2019: 102013.

Amarappa S & Sathyanarayana S. 2014. Data classification using Support vector Machine (SVM), a simplified approaCH. *International Journal of Electronics and Computer S cience Engineering*.

Araya YH & Hergarten C 2008. A comparison of pixel and object-based land cover classification: A case study of the Asmara region, Eritrea. *WIT Transactions on the Built Environment* 100: 233–243.

Ariza-López FJ, Rodríguez-Avi J & Alba-Fernández M V. 2018. Complete control of an observed confusion matrix. *International Geoscience and Remote Sensing Symposium (IGARSS)* 2018-July: 1222–1225.

Asongu SA, Agboola MO, Alola AA & Bekun FV 2020. The criticality of growth, urbanization, electricity and fossil fuel consumption to environment sustainability in Africa. *Science of the Total Environment* 712: 136376.

Baluja J, Diago MP, Balda P, Zorer R, Meggio F, Morales F & Tardaguila J 2012. Assessment of vineyard water status variability by thermal and multispectral imagery using an unmanned aerial vehicle (UAV). *Irrigation Science* 30, 6: 511–522.

Barra S, Carta SM, Corriga A, Podda AS & Recupero DR 2020. Deep learning and time series-To-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica* 7, 3: 683–692.

Bazi Y & Melgani F 2006. Toward an optimal SVM classification system for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 44, 11: 3374–3385.

De Beurs KM & Henebry GM 2005. A statistical framework for the analysis of long image time series. *International Journal of Remote Sensing* 26, 8: 1551–1573.

Breiman L 1996. Bagging Predictors. *Machine Learning*.

Breiman L 2001. Random Forests. *Machine Learning*.

Breuel TM 2015. Benchmarking of LSTM Networks.

Brovelli MA, Sun Y & Yordanov V 2020. Monitoring forest change in the amazon using multi-temporal remote sensing data and machine learning classification on Google Earth Engine. *ISPRS International Journal of Geo-Information* 9, 10: 1–21.

Brownlee J 2020. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery.

Bryan E, Deressa TT, Gbetibouo GA & Ringler C 2009. Adaptation to climate change in Ethiopia and South Africa: options and constraints. *Environmental Science and Policy* 12, 4: 413–426.

Campbell JB & Wynne RH 2011. *Introduction to Remote Sensing*. 5. The Guilford Press.

Camps-Valls G 2009. Machine learning in remote sensing dataprocessing. *Machine Learning for Signal Processing XIX - Proceedings of the 2009 IEEE Signal Processing Society Workshop, MLSP 2009*.

Cao J, Li Z & Li J 2019. Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications* 519: 127–139.

Cao X, Chen J, Imura H & Higashi O 2009. A SVM-based method to extract urban areas from DMSP-OLS and SPOT VGT data. *Remote Sensing of Environment* 113, 10: 2205–2209.

Castillejo-González IL, López-Granados F, García-Ferrer A, Peña-Barragán JM, Jurado-Expósito M, de la Orden MS & González-Audicana M 2009. Object- and pixel-based analysis for mapping crops and their agro-environmental associated measures using QuickBird imagery. *Computers and Electronics in Agriculture* 68, 2: 207–215.

Celik N 2018. Change Detection of Urban Areas in Ankara through Google Earth Engine. *2018 41st International Conference on Telecommunications and Signal Processing, TSP 2018*: 1–5.

Chen D, Stow DA & Gong P 2004. Examining the effect of spatial resolution and texture window size on classification accuracy: An urban environment case. *International Journal of Remote Sensing* 25, 11: 2177–2192.

Chen H, Jiao L, Liang M, Liu F, Yang S & Hou B 2019. Fast unsupervised deep fusion network for change detection of multitemporal SAR images. *Neurocomputing* 332: 56–70.

Chen J, Wang Y, Wu Y & Cai C 2017. An ensemble of convolutional neural networks for image classification based on LSTM. *Proceedings - 2017 International Conference on Green Informatics, ICGI 2017* 21, 1: 217–222.

Chen X & Campagna DJ 2009. Remote Sensing of Geology. In Warner TA Nellis MD & Foody GM (eds) *The SAGE Handbook of Remote Sensing*, 328–340. SAGE.

Cho J, Lee K, Shin E, Choy G & Do S 2015. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?

Chuvieco E 2020. *Fundamentals of Satellite Remote Sensing*. Third. CRC Press.

Cleve C, Kelly M, Kearns FR & Moritz M 2008. Classification of the wildland-urban interface: A comparison of pixel- and object-based classifications using high-resolution aerial photography. *Computers, Environment and Urban Systems* 32, 4: 317–326.

Colah 2015. Understanding LSTM Networks

Comer ML & Delp EJ 1995. Multiresolution image segmentation. *IEEE xplore* 4, 317: 2415–2418.

Comer ML & Delp EJ 1999. Segmentation of textured images using a multiresolution Gaussian autoregressive model. *IEEE Transactions on Image Processing* 8, 3: 408–420.

Coskun M, Ucar A, Yildirim O & Demir Y 2017. Face recognition based on convolutional neural network. *Proceedings of the International Conference on Modern Electrical and Energy Systems, MEES 2017* 2018-Janua: 376–379.

Cover TM & Hart PE 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* 13, 1: 21–27.

Cracknell AP 2007. *Introduction to Remote Sensing*. Second. Taylor and Francis Group.

Crews KA & Walsh SJ 2009. Remote Sensing and the Soical Sciences. In Warner TA Nellis MD & Foody GM (eds) *The SAGE Handbook of Remote Sensing*, 437442. SAGE.

Cunningham P & Delany SJ 2020. k-Nearest Neighbour Classifiers: 2. , 1: 1–22.

Daudt RC, Le Saux B, Boulch A & Gousseau Y 2018. Urban change detection for multispectral earth observation using convolutional neural networks. *International Geoscience and Remote Sensing Symposium (IGARSS)* 2018-July: 2115–2118.

De S, Bruzzone L, Bhattacharya A, Bovolo F & Chaudhuri S 2018. A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11, 1: 154–170.

Deng JS, Wang K, Deng YH & Qi GJ 2008. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. *International Journal of Remote Sensing* 29, 16: 4823–4838.

Deng L & Yu D 2014. Deep Learning: Methods and Applications. *Now Publishers Inc* 7.

Deng Y, Wu C, Li M & Chen R 2015. RNDSI: A ratio normalized difference soil index for remote sensing of urban/suburban environments. *International Journal of Applied Earth Observation and Geoinformation* 39: 40–48.

Dervisoglu A, Bilgilioglu BB & Yagmur N 2020. Comparison of Pixel-Based and Object-Based Classification Methods in Determination of Wetland Coastline. *International Journal of Environment and Geoinformatics* 7, 2: 213–220.

Dharani M & Sreenivasulu G 2021. Land use and land cover change detection by using principal component analysis and morphological operations in remote sensing applications. *International Journal of Computers and Applications* 43, 5: 462–471.

Dias D, Dias U, Menini N, Lamparelli R, Le Maire G & Torres RDS 2020. Image-Based Time Series Representations for Pixelwise Eucalyptus Region Classification: A Comparative Study. *IEEE Geoscience and Remote Sensing Letters* 17, 8: 1450–1454.

DigitalGloble 2001. *Data Shee: QuickBird*.

DigitalGloble 2014. *Data Sheet: WorldView-3*.

Dimitrios D, Agapiou A, Diofantos G & Sarris A 2012. Remote Sensing Applications in Archaeological Research. *Remote Sensing - Applications*.

Donges N 2021. A Guide to RNN: Understanding Recurrent Neural Networks and LSTM

Networks

Du S, Zhang F & Zhang X 2015. Semantic classification of urban buildings combining VHR image and GIS data: An improved random forest approach. *ISPRS Journal of Photogrammetry and Remote Sensing* 105: 107–119.

Dunnmon JA, Yi D, Langlotz CP, Ré C, Rubin DL & Lungren MP 2019. Assessment of convolutional neural networks for automated classification of chest radiographs. *Radiology* 290, 3: 537–544.

Duong ND 2004. LAND COVER MAPPING OF VIETNAM USING MODIS 500M 32-DAY GLOBAL COMPOSITES Nguyen. *International Symposium on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences 2004*.

Duro DC, Franklin SE & Dubé MG 2012. A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote Sensing of Environment* 118: 259–272.

Enderle DI. & Weih RC 2005. Integrating Supervised and Unsupervised Classification Methods to Develop a More Accurate Land Cover Classification. *Journal of the Arkansas Academy of Science* 59: 65–73.

ESA 2020a. IKONOS-2 Overview [online]. Available from: https://earth.esa.int/eogateway/missions/ikonos-2

ESA 2015. Sentinel-2 [online]. Available from: https://sentinel.esa.int/web/sentinel/missions/sentinel-2

ESA 2020b. SPOT-6 [online]. Available from: https://earth.esa.int/eogateway/missions/spot-6

ESA 2020c. SPOT-7 [online]. Available from: https://earth.esa.int/eogateway/missions/spot-7

Espinoza-molina D, Bahmanyar R, Ricardo D, Bustamante J & Datcu M 2017. Land-cover change detection using local feature descriptors extracted from spectral indices. : 6–9.

Evans MJ & Malcom JW 2021. Supporting habitat conservation with automated change detection in Google Earth Engine. *Conservation Biology* 35, 4: 1151–1161.

Fei-fei L, Deng J, Russakovsky O, Berg A & Li K 2021. ImageNet [online]. Available from: https://www.image-net.org/

Feng N, Geng X & Qin L 2020. Study on MRI Medical Image Segmentation Technology Based on CNN-CRF Model. *IEEE Access* 8: 60505–60514.

Fonseca-Pinto R, Ducla-Soares JL, Araújo F, Aguiar P & Andrade A 2009. On the influence of

time-series length in EMD to extract frequency content: Simulations and models in biomedical signals. *Medical Engineering and Physics* 31, 6: 713–719.

Fushiki T 2011. Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing* 21, 2: 137–146.

Gao Y & Mas J 2008. A comparison of the performance of pixel based and object based classifications over images with various spatial resolutions. *Online journal of earth sciences* 2, 8701: 27–35.

Gašparović M, Zrinjski M & Gudelj M 2019. Automatic cost-effective method for land cover classification (ALCC). *Computers, Environment and Urban Systems* 76, December 2018: 1–10.

GeoEye 2008. *Geoeye-1: The world's highest resolution commercial earth-imaging satellite*.

Georganos S, Grippa T, Vanhuysse S, Lennert M, Shimoni M, Kalogirou S & Wolff E 2018. Less is more: optimizing classification performance through feature selection in a very-high-resolution remote sensing object-based urban application. *GIScience and Remote Sensing* 55, 2: 221–242.

Ghosh A, Sharma R & Joshi PK 2014. Random forest classification of urban landscape using Landsat archive and ancillary data: Combining seasonal maps with decision level fusion. *Applied Geography* 48: 31–41.

Giuliani G, Chatenoux B, Honeck E & Richard JP 2018. Towards sentinel-2 analysis ready data: A Swiss data cube perspective. *International Geoscience and Remote Sensing Symposium (IGARSS)* 2018-July, 3: 8659–8662.

Gong P, Li X & Zhang W 2019. 40-Year (1978–2017) human settlement changes in China reflected by impervious surfaces from satellite remote sensing. *Science Bulletin* 64, 11: 756–763.

Goodfellow I, Bengio Y & Courville A 2016. *Deep Learning*. MIT press.

Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D & Moore R 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment* 202: 18–27.

Graves A & Schmidhuber J 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5–6: 602–610.

Grobler TL, Ackermann ER, Olivier JC, Van Zyl AJ & Kleynhans W 2012. Land-cover separability analysis of MODIS time-series data using a combined simple harmonic oscillator and a mean reverting stochastic process. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, 3: 857–866.

Grobler TL, Ackermann ER, Van Zyl AJ, Olivier JC, Kleynhans W & Salmon BP 2013. Using page's cumulative sum test on MODIS time series to detect land-cover changes. *IEEE Geoscience and Remote Sensing Letters* 10, 2: 332–336.

Gunter B 2021. Global Majority E-Journal. 12, 1: 1–75.

Hamzaçebi C 2008. Improving artificial neural networks' performance in seasonal time series forecasting. *Information Sciences* 178, 23: 4550–4559.

Hatami N, Gavet Y & Debayle J 2018. Classification of time-series images using deep convolutional neural networks. : 23.

He K, Zhang X, Ren S & Sun J 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem: 770–778.

Hills J, Lines J, Baranauskas E, Mapp J & Bagnall A 2014. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery* 28, 4: 851–881.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M & Adam H 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*.

Hsieh PF, Lee LC & Chen NY 2001. Effect of spatial resolution on classification errors of pure and mixed pixels in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing* 39, 12: 2657–2663.

Hu H & Ban Y 2014. Unsupervised Change Detection in Multitemporal SAR Images Over Large Urban Areas. *Image Processing for Remote Sensing* 7, 8.

Hu Y, Dong Y & Batunacun 2018. An automatic approach for land-change detection and land updates based on integrated NDVI timing analysis and the CVAPS method with GEE support. *ISPRS Journal of Photogrammetry and Remote Sensing* 146, October: 347–359.

Huang G, Liu Z & Van Der Maaten L 2017. Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*: 2261–2269.

Huang G, Liu Z, van der Maaten L & Weinberger KQ 2017. Densely Connected Convolutional Networks. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*: 4700–4708.

Huang T, Chakraborty P & Sharma A 2020. Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. 2018, December 2018.

Hubert-Moy L, Cotonnec A, Le Du L, Chardin A & Perez P 2001. A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote Sensing of Environment* 75, 2: 174–187.

Hussain M, Chen D, Cheng A, Wei H & Stanley D 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing* 80: 91–106.

Ishibuchi H & Nakashima T 2001. Effect of rule weights in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems* 9, 4: 506–515.

Islam MJ, Wu QMJ, Ahmadi M & Sid-Ahmed MA 2008. Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers. : 1541–1546.

Jackson PC 2019. *Introduction to Artificial Intelligence: Third Edition*. 3. New York: Dover Publications Inc.

Jain AK, Duin RPW & Mao J 2000. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1: 4–37.

Jakubovitz D, Giryes R & Rodrigues M. 2019. Generalization error in deep learning. *In Compressed Sensing and Its Applications*: 153–193.

Jensen J. & Im J 2007. Remote Sensing Change Detection in Urban Environments. *Geo-Spatial Technologies in Urban Environments*.

Jones HG & Sirault XRR 2014. Scaling of thermal images at different spatial resolution: The mixed pixel problem. *Agronomy* 4, 3: 380–396.

Jordan MI & Mitchell TM 2015. Machine learning: Trends, perspectives, and prospects. *Science* 349: 255–260.

Juba B & Le HS 2019. Precision-Recall versus accuracy and the role of large data sets. *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19) Precision-Recall*: 4039–4048.

Kaiming H, Xiangyu Z, Ren S & Sun J 2015. Deep Residual Learning for Image Recognition Kaiming. *CVPR*.

Karevan Z & Suykens JAK 2020. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks* 125: 1–9.

Karim F, Majumdar S & Darabi H 2019. Insights into lstm fully convolutional networks for time series classification. *IEEE Access* 7: 67718–67725.

Karim F, Majumdar S, Darabi H & Chen S 2017. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* 6: 1662–1669.

Karita S, Order A, Chen N, Hayashi T, Hori T, Inaguma H, Jiang Z, Someki M, Enrique N, Soplin Y, Yamamoto R, Wang X, Watanabe S, Yoshimura T & Zhang W 2019. A comparative study on transformer vs rnn in speech applications. *IEEE Xplore* 9, 2: 449–456.

Keras 2020a. Keras Applications [online]. Available from: https://keras.io/api/applications/

Keras 2020b. LSTM layer [online]. Available from: https://keras.io/api/layers/recurrent_layers/lstm/

Kim P 2017. *Convolutional Neural Network*. Apress, Berkeley, CA.

Kim Y, Glenn DM, Park J, Ngugi HK & Lehman BL 2011. Hyperspectral image analysis for water stress detection of apple trees. *Computers and Electronics in Agriculture* 77, 2: 155–160.

Kleynhans W, Olivier JC, Wessels KJ, Salmon BP, Van Den Bergh F & Steenkamp K 2011. Detecting land cover change using an extended kalman filter onMODIS NDVI time-series data. *IEEE Geoscience and Remote Sensing Letters* 8, 3: 507–511.

Kleynhans W, Salmon BP, Wessels KJ & Olivier JC 2013. A spatio-temporal autocorrelation change detection approach using hyper-temporal satellite data. *International Geoscience and Remote Sensing Symposium (IGARSS)*: 3459–3462.

Kleynhans W, Salmon BP, Wessels KJ & Olivier JC 2015. Rapid detection of new and expanding human settlements in the Limpopo province of South Africa using a spatio-temporal change detection method. *International Journal of Applied Earth Observation and Geoinformation* 40: 74–80.

Kleynhans W, Salmon BP & Wessels KJ 2017. A novel framework for parameter selection of the Autocorrelation Change detection method using 250m MODIS time-series data in the Gauteng province of South Africa. *South African Journal of Geomatics* 6, 3: 407.

Kleynhans W, Salmon BP, Olivier JC, Van Den Bergh F, Wessels KJ, Grobler TL & Steenkamp KC 2012. Land cover change detection using autocorrelation analysis on MODIS time-

series data: Detection of new human settlements in the gauteng province of South Africa. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, 3: 777–783.

Kuhn M 2018. *Applied Predictive Modeling*. 2. Springer.

Kuhn M & Johnson K 2019. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Taylor & Francis Group.

Längkvist M, Karlsson L & Loutfi A 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Pattern Recognition Letters* 42, 1: 11–24.

Lawrence RL & Wright A 2001. Rule-based classification systems using classification and regression tree (CART) analysis. *Photogrammetric Engineering and Remote Sensing* 67, 10: 1137–1142.

Lecun Y, Bengio Y & Hinton G 2015. Deep learning. *Nature* 521, 7553: 436–444.

Leslie D 2019. Understanding artificial intelligence ethics and safety systems: A guide for the responsible design and implementation of AI systems in the public sector. *The Alan Turing Institute*.

Lewis HG & Brown M 2001. A generalized confusion matrix for assessing area estimates from remotely sensed data. *International Journal of Remote Sensing* 22, 16: 3223–3235.

Li C, Xiong J, Zhu X, Zhang Q & Wang S 2020. Fault diagnosis method based on encoding time series and convolutional neural network. *IEEE Access* 8: 165232–165246.

Li Q, Qiu C, Ma L, Schmitt M & Zhu XX 2020. Mapping the land cover of africa at 10 m resolution from multi-source remote sensing data with google earth engine. *Remote Sensing* 12, 4: 1–22.

Li S, Song W, Fang L, Chen Y, Ghamisi P & Benediktsson JA 2019. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing* 57, 9: 6690–6709.

Li W 2020. Mapping urban impervious surfaces by using spectral mixture analysis and spectral indices. *Remote Sensing* 12, 1.

Li W, Liu H, Wang Y, Li Z, Jia Y & Gui G 2019. Deep Learning-Based Classification Methods for Remote Sensing Images in Urban Built-Up Areas. *IEEE Access* 7: 36274–36284.

Li X, Gong P & Liang L 2015. A 30-year (1984-2013) record of annual urban dynamics of Beijing City derived from Landsat data. *Remote Sensing of Environment* 166: 78–90.

Li Y & Cheng B 2009. An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification. *2009 17th International Conference on Geoinformatics, Geoinformatics 2009*: 2–5.

Lillesand T, Kiefer RW & Chipman J 2015. *Remote Sensing and Image Interpretation*. Seventh. John Wiley & Sons.

Liu D & Xia F 2010. Assessing object-based classification: Advantages and limitations. *Remote Sensing Letters* 1, 4: 187–194.

Liu JG & Mason PJ 2016. *Image Processing and GIS for Remote Sensing: Techniques and Applications*. Second. John Wiley & Sons.

Liu L & Wang Z 2016. Encoding Temporal Markov Dynamics in Graph for Visualizing and Mining Time Series.

Liu X, Hu G, Chen Y, Li X, Xu X, Li S, Pei F & Wang S 2018. High-resolution multi-temporal mapping of global urban land using Landsat images based on the Google Earth Engine Platform. *Remote Sensing of Environment* 209, January: 227–239.

Lopez JF, Shimoni M & Grippa T 2017. Extraction of African urban and rural structural features using SAR sentinel-1 data. *2017 Joint Urban Remote Sensing Event, JURSE 2017*: 1–4.

Lu D, Mausel P, Brondízio E & Moran E 2004. Change detection techniques. *International Journal of Remote Sensing* 25, 12: 2365–2401.

Lunetta RS, Knight JF, Ediriwickrema J, Lyon JG & Worthy LD 2006. Land-cover change detection using multi-temporal MODIS NDVI data. *Remote Sensing of Environment* 105, 2: 142–154.

Ma B, Pu R, Zhang S & Wu L 2018. Spectral Identification of Stress Types for Maize Seedlings under Single and Combined Stresses. *IEEE Access* 6: 13773–13782.

Ma L, Liu Y, Zhang X, Ye Y, Yin G & Johnson BA 2019. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing* 152, March: 166–177.

Martínez F, Frías MP, Pérez-Godoy MD & Rivera AJ 2018. Dealing with seasonality by narrowing the training set in time series forecasting with kNN. *Expert Systems with Applications* 103: 38–48.

Masek JG 2017. Landsat 7 [online]. Available from: https://landsat.gsfc.nasa.gov/landsat-7

Masek JG 2013. Landsat 8 Overview [online]. Available from: https://landsat.gsfc.nasa.gov/landsat-8/landsat-8-overview

Mather PM & Koch M 2011. *Computer Processing of Remotely-Sensed Images: An Introduction*. Fourth. John Wiley & Sons.

Maxwell AE, Warner TA & Fang F 2018. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing* 39, 9: 2784–2817.

Mboga N, Persello C, Bergado JR & Stein A 2017. Detection of informal settlements from VHR images using convolutional neural networks. *Remote Sensing* 9, 11.

McCarthy. J 2007. What Is Artificial Intelligence?

Mendel J 2017. *Uncertain Rule-Based Fuzzy Systems*. 2. Springer.

Michalski RS, Carbonell JG & Mitchell TM 2013. *Machine Learning: An Artificial Intelligence Approach*. Berline Heidelberg: Springer Science & Business Media.

Mohd Hasmadi I, Pakhriazad H & Shahrin M 2009. Evaluating supervised and unsupervised techniques for land cover mapping using remote sensing data. *Geografia: Malaysian Journal of Society and Space* 5, 1: 1–10.

Moser G, Serpico S & Vernazza G 2007. Unsupervised Change Detection From Multichannel SAR Images. 4, 2: 278–282.

Myburgh G & Niekerk A Van 2013. Effect of feature dimensionality on object-based land cover classification: A comparison of three classifiers. *Effect of feature dimensionality on object-based land cover classification: A comparison of three classifiers* 2, 1: 13–27.

Myint SW, Gober P, Brazel A, Grossman-Clarke S & Weng Q 2011. Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. *Remote Sensing of Environment* 115, 5: 1145–1161.

Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R & Muharemagic E 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1: 1–21.

Nassar DM & Elsayed HG 2018. From Informal Settlements to sustainable communities. *Alexandria Engineering Journal* 57, 4: 2367–2376.

Nelson M, Hill T, Remus W & O'Connor M 1999. Time series forecasting using neural networks: Should the data be deseasonalized first? *Journal of Forecasting* 18, 5: 359–367.

Ng EYK & Acharya RU 2009. Remote-Sensing Infrared Thermography: Reviewing the Applications of Indoor Infrared Fever-Screening Systems. *IEEE Engineering in Medicine and Biology* 28, 1: 76–83.

Nijhawan R, Srivastava I & Shukla P 2017. Land Cover Classification Using Supervised and Unsupervised Learning Techniques. *IEEE*.

Nilsson NJ 1982. *Principles of Artificial Intelligence*. Berlin Heidelberg New York: Springer.

Novack T, Esch T, Kux H & Stilla U 2011. Machine learning comparison between WorldView-2 and QuickBird-2-simulated imagery regarding object-based urban land cover classification. *Remote Sensing* 3, 10: 2263–2282.

Olthof I, Butson C & Fraser R 2005. Signature extension through space for northern landcover classification: A comparison of radiometric correction methods. *Remote Sensing of Environment* 95, 3: 290–302.

Ongsulee P 2018. Artificial intelligence, machine learning and deep learning. *International Conference on ICT and Knowledge Engineering*: 1–6.

Owolawi PA, Afullo TJ & Malinga SB 2009. Effect of rainfall on millimeter wavelength radio in Gough and Marion Islands. *Progress in Electromagnetics Research Symposium* 1, 1: 81–88.

Pal M 2008. Ensemble of support vector machines for land cover classification. *International Journal of Remote Sensing* 29, 10: 3043–3049.

Pal M 2005. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing* 26, 1: 217–222.

Pal M & Mather PM 2005. Support vector machines for classification in remote sensing. *International Journal of Remote Sensing* 26, 5: 1007–1011.

Palframan A 2005. A syntactical analysis of settlement form – an investigation of Socio-spatial characteristics in low-income housing settlements in.

Pan Z, Xu J, Guo Y, Hu Y & Wang G 2020. Deep learning segmentation and classification for urban village using a worldview satellite image based on U-net. *Remote Sensing* 12, 10: 1–17.

Pang B, Zha K, Cao H, Shi C & Lu C 2019. Deep RNN Framework for Visual Sequential Applications. *IEEE/CVF Conference*: 423–432.

Pascanu R, Gulcehre C, Cho K & Bengio Y 2014. How to construct deep recurrent neural networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*: 1–13.

Patra DP 2010. Remote Sensing and Geographical Information System (GIS). *The Association for Geographical Studies*.

Pelletier C, Webb GI & Petitjean F 2019. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing* 11, 5: 1–25.

Petropoulos GP, Kalaitzidis C & Prasad Vadrevu K 2012. Support vector machines and object-based classification for obtaining land-use/cover cartography from Hyperion hyperspectral imagery. *Computers and Geosciences* 41: 99–107.

Phalke AR & Özdoğan M 2018. Large area cropland extent mapping with Landsat data and a generalized classifier. *Remote Sensing of Environment* 219, October: 180–195.

Polykretis C, Grillakis MG & Alexakis DD 2020. Exploring the impact of various spectral indices on land cover change detection using change vector analysis: A case study of Crete Island, Greece. *Remote Sensing* 12, 2.

Pontius RG & Millones M 2011. Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing* 32, 15: 4407–4429.

Poona NK & Ismail R 2014. Using Boruta-selected spectroscopic wavebands for the asymptomatic detection of fusarium circinatum stress. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7, 9: 3764–3772.

Poona NK, Van Niekerk A, Nadel RL & Ismail R 2016. Random Forest (RF) Wrappers for Waveband Selection and Classification of Hyperspectral Data. *Applied Spectroscopy* 70, 2: 322–333.

Priyam A, Gupta R, Rathee A & Srivastava S 2013. Comparative Analysis of Decision Tree Classification Algorithms. : 334–337.

Qian Y, Zhou W, Yan J, Li W & Han L 2015. Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Remote Sensing* 7, 1: 153–168.

Qin Y, Niu Z, Chen F, Li B & Ban Y 2013. Object-based land cover change detection for cross-sensor images. *International Journal of Remote Sensing* 34, 19: 6723–6737.

Qin Zhen, Zhang Y, Meng S, Qin Zhiguang & Choo KKR 2020. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion* 53, May 2019: 80–87.

Quinlan JR 1996. Learning decision tree classifiers. *ACM Computing Surveys* 28, 1: 71–72.

Radke RJ, Andra S, Al-Kofahi O & Roysam B 2005. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing* 14, 3: 294–307.

Riggan ND & Weih RC 2009. A Comparison of Pixel-based versus Object-based Land Use / Land Cover Classification Methodologies. *Journal of the Arkansas Academy of Science* 63: 145–152.

Rispens SM, Pijnappels M, van Dieën JH, van Schooten KS, Beek PJ & Daffertshofer A 2014. A benchmark test of accuracy and precision in estimating dynamical systems characteristics from a time series. *Journal of Biomechanics* 47, 2: 470–475.

Rodriguez-Galiano VF, Ghimire B, Rogan J, Chica-Olmo M & Rigol-Sanchez JP 2012. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing* 67, 1: 93–104.

Rodríguez JD, Pérez A & Lozano JA 2010. Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 3: 569–575.

Rouse JW, Haas RH, Shell JA & Deering DW 1974. *Third Earth Resources Technology Satellite-1 Symposium*. Washington, D.C.: Goddard Space Flight Center.

Ruiz P 2018a. DenseNets.

Ruiz P 2018b. *ResNets*.

Sagheer A & Kotb M 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 323: 203–213.

Sahu KC 2006. *Textbook of Remote Sensing and Geographical Information Systems*. Atlantic.

Salmon BP, Kleynhans W, Schwegmann CP & Olivier JC 2015. Proper comparison among methods using a confusion matrix.

Salmon BP, Kleynhans W, Van Den Bergh F, Olivier JC, Grobler TL & Wessels KJ 2013. Land cover change detection using the internal covariance matrix of the extended kalman filter over multiple spectral bands. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 6, 3: 1079–1085.

Samala RK, Chan HP, Hadjiiski L, Helvie MA, Richter CD & Cha KH 2019. Breast cancer diagnosis in digital breast tomosynthesis: Effects of training sample size on multi-stage transfer learning using deep neural nets. *IEEE Transactions on Medical Imaging* 38, 3: 686–696.

Segal-Rozenhaimer M, Li A, Das K & Chirayath V 2020. Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN). *Remote Sensing of Environment* 237, November 2018: 111446.

Seydi ST, Hasanlou M & Amani M 2020. A new end-to-end multi-dimensional CNN framework for land cover/land use change detection in multi-source remote sensing datasets. *Remote Sensing* 12, 12.

Shang X & Chisholm LA 2014. Classification of Australian native forest species using hyperspectral remote sensing and machine-learning classification algorithms. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7, 6: 2481–2489.

Shao YH, Chen WJ & Deng NY 2014. Nonparallel hyperplane support vector machine for binary classification problems. *Information Sciences* 263: 22–35. [online]. Available from: http://dx.doi.org/10.1016/j.ins.2013.11.003

Sharma Siddharth, Sharma Simone & Anidhya A 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology* 4, 12: 310–316.

Shi W, Zhang M, Ke H, Fang X, Zhan Z & Chen S 2021. Landslide Recognition by Deep Convolutional Neural Network and Change Detection. *IEEE Transactions on Geoscience and Remote Sensing* 59, 6: 4654–4672.

Shrestha A & Mahmood A 2019. Review of deep learning algorithms and architectures. *IEEE Access* 7: 53040–53065.

Simonyan K & Zisserman A 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*: 1–14.

Sinha S, Santra A & Mitra SS 2018. A method for built-up area extraction using dual polarimetric alos palsar. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4, 5: 455–458.

Sinha S, Sharma LK & Nathawat MS 2015. Improved Land-use/Land-cover classification of semi-arid deciduous forest landscape using thermal remote sensing. *Egyptian Journal of Remote Sensing and Space Science* 18, 2: 217–233.

Soltau H, Liao H & Sak H 2017. Neural speech recognizer: Acoustic-To-word LSTM model for largevocabulary speech recognition. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2017-Augus: 3707–3711.

Song J, Gao S, Zhu Y & Ma C 2019. A survey of remote sensing image classification based on CNNs. *Big Earth Data* 3, 3: 232–254.

Stehman S V. 1996. Estimating the kappa coefficient and its variance under stratified random sampling. *Photogrammetric Engineering and Remote Sensing* 62, 4: 401–407.

Stoian A, Poulain V, Inglada J, Poughon V & Derksen D 2019. Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *Remote Sensing* 11, 17: 1–26.

Strahler AH 1980. The use of prior probabilities in maximum likelihood classification of remotely sensed data. *Remote Sensing of Environment* 10, 2: 135–163.

Sundarakumar K, Harika M, Aspiya begum S, Yamini S & Balakrishna K 2016. Land Use and Land Cover Change Detection for Urban Sprawl Analysis of Ahmedabad City using Multitemporal Landsat Data. *International Journal of Advanced Remote Sensing and GIS* 5, 1: 1670–1677.

Suzuki S & Matsui T 2012. Remote Sensing for Medical and Health Care Applications. In Escalante B (ed) *Remote Sensing: Applications*, 479–489. InTech.

Swain PH & Hauska H 1977. Decision Tree Classifier: Design and Potential. *IEEE Trans Geosci Electron* GE-15, 3: 142–147.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V & Rabinovich A 2015. *Going Deeper with Convolutions Christian*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J & Wojna Z 2016. Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem: 2818–2826.

Szeliski R 2011. *Computer Vision: Algorithms and Applications*. London: Springer.

Tan HX, Aung NN, Tian J, Chua MCH & Yang YO 2019. Time series classification using a

modified LSTM approach from accelerometer-based data: A comparative study for gait cycle detection. *Gait and Posture* 74, August: 128–134.

Tan J, Yang J, Wu S, Chen G & Zhao J 2021. A critical look at the current train/test split in machine learning. [online]. Available from: http://arxiv.org/abs/2106.04525

Tang J, Alelyani S & Liu H 2014. Feature Selection for Classification: A Review. *Data classification: Algorithms and applications*,: 571–605.

TensorFlow 2021. TensorFLow Core v2.6.0

Tewkesbury AP, Comber AJ, Tate NJ, Lamb A & Fisher PF 2015. A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment* 160: 1–14.

Tran BN, Tanase MA, Bennett LT & Aponte C 2018. Evaluation of spectral indices for assessing fire severity in Australian temperate forests. *Remote Sensing* 10, 11: 1–18.

Tzotsos A & Argialas D 2008. *Support Vector Machine Classification for Object-Based Image Analysis*. Springer.

UN 2018. United Nations Department of Economic and Social Affairs Population Division. *World urbanization prospects 2018: Highlights.ST/ESA/SER.A/421*.

USGS 2018. MODIS Overview [online]. Available from: https://lpdaac.usgs.gov/data/get-started-data/collection-overview/missions/modis-overview/

Usman M, Liedl R, Shahid MA & Abbas A 2015. *Land use/land cover classification and its change detection using multi-temporal MODIS NDVI data*.

Vasan D, Alazab M, Wassan S, Safaei B & Zheng Q 2020. Image-Based malware classification using ensemble of CNN architectures (IMCEC). *Computers and Security* 92: 101748.

Verbesselt J, Hyndman R, Newnham G & Culvenor D 2010. Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment* 114, 1: 106–115.

Verbesselt J, Hyndman R, Zeileis A & Culvenor D 2010. Phenological change detection while accounting for abrupt and gradual trends in satellite image time series. *Remote Sensing of Environment* 114, 12: 2970–2980.

Verhulp J & Van Niekerk A 2017. Transferability of decision trees for land cover classification in a heterogeneous area. *South African Journal of Geomatics* 6, 1: 30.

Viana J, Santos JV, Neiva RM, Souza J, Duarte L, Teodoro AC & Freitas A 2017. Remote sensing in human health: A 10-year bibliometric analysis. *Remote Sensing* 9, 12: 1–12.

Wang F & Tax DMJ 2016. Survey on the attention based RNN model and its applications in computer vision.

Wang Z & Oates T 2015. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. *AAAI Workshop - Technical Report* WS-15-14: 40–46.

Wei W & Mendel JM 2000. Maximum-likelihood classification for digital amplitude-phase modulations. *IEEE Transactions on Communications* 48, 2: 189–193.

Wong MS, Nichol JE, Lee KH & Emerson N 2008. Modeling water quality using terra/modis 500m satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*: 679–684.

Wong TT & Yeh PY 2020. Reliable Accuracy Estimates from k-Fold Cross Validation. *IEEE Transactions on Knowledge and Data Engineering* 32, 8: 1586–1594.

Woodcock CE, Macomber SA, Pax-Lenney M & Cohen WB 2001. Monitoring large areas for forest change using Landsat: Generalization across space, time and Landsat sensors. *Remote Sensing of Environment* 78, 1–2: 194–203.

Xu Y, Yu L, Cai Z, Zhao J, Peng D, Li C, Lu H, Yu C & Gong P 2019. Exploring intra-annual variation in cropland classification accuracy using monthly, seasonal, and yearly sample set. *International Journal of Remote Sensing* 40, 23: 8748–8763.

Xue J & Su B 2017. Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors* 2017.

Yan W 2012. Toward automatic time-series forecasting using neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 23, 7: 1028–1039.

Yang C, Chen Z & Yang C 2020. Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images. , 1.

Yang CL, Yang CY, Chen ZX & Lo NW 2019. Multivariate Time Series Data Transformation for Convolutional Neural Network. *Proceedings of the 2019 IEEE/SICE International Symposium on System Integration, SII 2019*: 188–192.

Yildirim O, Baloglu UB, Tan RS, Ciaccio EJ & Acharya UR 2019. A new approach for arrhythmia classification using deep coded features and LSTM networks. *Computer Methods and Programs in Biomedicine* 176: 121–133.

Yin W, Kann K, Yu M & Schütze H 2017. Comparative Study of CNN and RNN for Natural Language Processing.

Yuan X, Tanksley D, Jiao P, Li L, Chen G & Wunsch D 2021. Encoding Time-Series Ground Motions as Images for Convolutional Neural Networks-Based Seismic Damage Evaluation. *Frontiers in Built Environment* 7, April.

Zaremba W, Sutskever I & Vinyals O 2014. Recurrent Neural Network Regularization. , 2013: 1–8.

Zha Y, Gao J & Ni S 2003. Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. *International Journal of Remote Sensing* 24, 3: 583–594.

Zhan Y, Fu K, Yan M, Sun X, Wang H & Qiu X 2017. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. *IEEE Geoscience and Remote Sensing Letters* 14, 10: 1845–1849.

Zhang C, Benz P, Argaw DM, Lee S, Kim J, Rameau F, Bazin J-C & Kweon IS 2021. ResNet or DenseNet? Introducing Dense Shortcuts to ResNet. : 3549–3558.

Zhang GP & Qi M 2005. Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research* 160, 2: 501–514.

Zhang L, Zhang L & Kumar V 2016. Deep learning for Remote Sensing Data. *IEEE Geoscience and Remote Sensing Magazine* 4, 2: 22–40.

Zhang L, Xia GS, Wu T, Lin L & Tai XC 2016. Deep Learning for Remote Sensing Image Understanding. *Journal of Sensors* 2016.

Zhang L, Tan J, Han D & Zhu H 2017. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today* 22, 11: 1680–1685.

Zhang X 2020. Machine Learning. In *A Matrix Algebra Approach to Artificial Intelligence*, Singapore: Springer.

Zhang Z, Vosselman G, Gerke M, Tuia D & Yang MY 2018. Change Detection between Multimodal Remote Sensing Data Using Siamese CNN. : 1–17.

Zhong Z, Li J, Luo Z & Chapman M 2018. Spectral-Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Transactions on Geoscience and Remote Sensing* 56, 2: 847–858.

Zhou T, Li Z & Pan J 2018. Multi-feature classification of multi-sensor satellite imagery based on dual-polarimetric sentinel-1A, landsat-8 OLI, and hyperion images for urban land-cover classification. *Sensors (Switzerland)* 18, 2: 1–20.

Zhu XX, Tuia D, Mou L, Xia G-S, Zhang L, Xu F & Fraundorfer F 2017. Deep learning in remote sensing: a review.