

https://dspace.org/

Industrial change detection using deep learning applied to DInSAR and optical satellite imagery

Karim, Zainoolabadien

2020

https://hdl.handle.net/10539/35618

Karim, Zainoolabadien (2020) Industrial change detection using deep learning applied to DInSAR and optical satellite imagery, University of the Witwatersrand, Johannesburg, <http://hdl.handle.net/10539/35618>

Downloaded from a DSpace repository. https://dspace.org/

Industrial Change Detection using Deep Learning applied to DInSAR and optical satellite imagery



A Research Report submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science

Zainoolabadien Karim

Supervised by Prof. Terence van Zyl

School of Computer Science and Applied Mathematics,
Faculty of Science,
University of the Witwatersrand, Johannesburg.

12 November 2020

Abstract

In order to detect industrial change in satellite imagery binary classification is investigated using traditional ML with feature extractors (HOG and LBP) with SVM, a simple CNN with only 2 convolutional layers called Simple ConvNet, state-of-the-art Deep Learning (DL) and Transfer Learning (TL) methods. Differential Interferometric Synthetic Aperture Radar (DInSAR) is used to generate interferograms and Terrain Corrected Displacement maps. A Sentinel-1B Synthetic Aperture Radar (SAR) image data stack from 28th Nov. 2017 to 5th Dec. 2018 is used with the images 12 days apart, where imagery was available. Blobs are detected in a displacement map which is generated using DInSAR and the Laplacian of Gaussian algorithm. The blobs are qualitatively verified using optical images from the Sentinel-2 satellite. If subsidence or uplift has indeed occurred then the blob is classified as positive. However, if uplift or subsidence has not occured then the blob is classified as negative which refers to noise. The blob detection algorithm has a high false positive rate. However, true positive blobs are detected corresponding to quarries, mines, construction, etc. A novel dataset is developed comprising of DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced imagery and the corresponding Sentinel-2 RGB optical satellite images of the blobs. A variety of DL architectures that are pretrained on ImageNet, a computer vision performance benchmark dataset, are used for implementing TL with Feature Extraction (FE). For the EfficientNet B4 and ResNet-50 architectures TL with Fine-Tuning (FT) as well as classic DL i.e. training from scratch, Random Initialisation (RI), are also evaluated. Ensemble performance containing certain architectures is also evaluated. The best performing architecture and method (84.34%) is the ResNet-50 with TL via FE applied. It outperformed all the other models and methods including newer, deeper (due to more data being needed to train deeper networks) and ensemble models from an accuracy perspective. Two of the ensembles evaluated using FE have an accuracy of 84.40% and 84.27% respectively with the same F1 score as ResNet50. It is concluded that a correlation between an increase in model size inferring a lower FE accuracy depends on architecture and holds for ResNet and EfficientNet but not for ResNetV2 architectures. The ResNet50 which has slightly more parameters than the EfficientNet B4 performs better with RI and FT respectively. Most models using TL with FE independent of network size outperformed RI and FT with ResNet50 and EfficientNet B4 respectively on this dataset. Most of the DL and TL models outperformed the traditional ML models except for the EfficientNet B4 models for RI and FT. Simple ConvNet (83.37%) outperformed most of the models except for FE with ResNet101, ResNet50 and the ensemble models. FE with the ResNet50 only outperformed Simple ConvNet by 0.97%. Thus, simple CNNs should not be overlooked for small datasets. Displacement Time-series has been developed for all the pixels in the study area. Although there is some noise upward and downward trends can be seen corresponding to change. A velocity map was developed with uplift of 20cm to subsidence of 60cm over the 1 year period noting that the subsidence sometimes occurred in areas of low coherence where it is not accurate.

Declaration

I, Zainoolabadien Karim, hereby declare the contents of this research report to be my own work. This research is submitted in partial fulfilment for the degree of Master of Science in Computer Science by Coursework and Research Report, MSc Comp. Sci (CW/RR), at the University of the Witwatersrand, Johannesburg. This work has not been submitted to another university, or for any other degree.

Z. Karim, 12 Nov. 2020

Acknowledgements

The author wishes to place on record his sincere gratitude to his research supervisor, Prof. Terence Van Zyl, for his supervision of this research. Both his vast background in the field, as well as input on the contents of this document have been of great help and are indeed tremendously appreciated.

Contents

Abstract i

Declaration ii

Acknowledgements iii

Table of Contents iv
Contents . . . iv
List of Figures . . . vi
List of Tables . . . viii

1 Introduction 1
1.1 Background . . . 1
1.2 Problem Statement . . . 2
1.3 Significance and Motivation . . . 2
1.4 Research Aims and Objectives . . . 2
1.5 Research Questions . . . 3
1.6 Delineations, Limitations and Assumptions . . . 3
1.7 Definitions . . . 3
1.8 Outline . . . 4

2 Background and Related Work 5
2.1 Introduction . . . 5
2.2 Theory . . . 5
2.2.1 Change Detection . . . 5
2.2.2 Sentinel-1 and DInSAR . . . 5
2.2.3 Image Processing . . . 8
2.2.4 Machine Learning . . . 8
2.2.5 Deep Learning (DL) . . . 8
2.2.6 Transfer Learning (TL) . . . 12
2.3 Works in Topic . . . 13
2.3.1 Persistent Scatterer Interferometry (PSI) . . . 14
2.3.2 Small Baseline Subset . . . 14
2.3.3 SqueeSAR . . . 16
2.3.4 Applications of Deep Learning and Transfer Learning . . . 16
2.4 Conclusion . . . 17

3 Research Methodology 18
3.1 Introduction . . . 18

3.2 Research Design . . . 18
3.3 Methodology . . . 18
3.3.1 Research Instruments . . . 19
3.3.2 Data . . . 19
3.3.3 DInSAR Analysis . . . 21
3.3.4 Candidate Selection . . . 23
3.3.5 Binary Classification . . . 24
3.3.6 Time Series Analysis . . . 29
3.4 Limitations . . . 29
3.5 Conclusion . . . 29

4 Results **30**
4.1 Introduction . . . 30
4.2 Candidate Selection . . . 30
4.3 Binary Classification . . . 34
4.3.1 Traditional ML: HOG and LBP with SVM . . . 35
4.3.2 Simple Convolutional Neural Network (Simple ConvNet) . . . 36
4.3.3 Transfer Learning (TL) via Feature Extraction (FE) . . . 37
4.3.4 Transfer Learning (TL) via Fine Tuning (FT) and Random Initialisation (RI) . . . 37
4.4 Time Series . . . 40
4.5 Research Questions . . . 43
4.6 Conclusion . . . 45

5 Conclusion **47**

References **48**

List of Figures

2.1 Interferometric SAR geometry showing interferometer and perpendicular baselines. . . . 6

3.1 Areas that the full Sentinel-1 image covers 19
3.2 InSAR processing graph to develop interferograms 21
3.3 DInSAR processing graph to develop differential interferograms 21
3.4 27 May-2 July differential interferogram showing phase 22
3.5 Displacement graph to develop displacement maps 22
3.6 Terrain Corrected Displacement Map 27 May - 2 July, showing uplift (blue) and subsidence (red) 22
3.7 Uplift (blue) and subsidence (red) blobs in Pretoria 23
3.8 Google Maps image showing construction site 1 23
3.9 Google Maps image showing construction site 2 24
3.10 (a) Sentinel-2 optical RGB Image 1, (b) HOG of Sentinel-2 optical RGB Image 1, (c) Displacement Map, (d) HOG of Displacement Map, (e) Coherence Map, (f) HOG of Coherence Map, (g) Phase Map and (h) HOG of Phase Map. 26
3.11 (a) Sentinel-2 optical RGB Image 1, (b) LBP of Sentinel-2 optical RGB Image 1, (c) Displacement Map, (d) LBP of Displacement Map, (e) Coherence Map, (f) LBP of Coherence Map, (g) Phase Map and (h) LBP of Phase Map. 27
3.12 Simple Convolutional Neural Network (Simple ConvNet) architecture. 28
3.13 ResNet50 Fine Tuning final layer (reproduced with permission, ©, 2020, IEEE [8]). . . 28

4.1 Sentinel-2 optical RGB image showing a blob where change in crops is detected, 2 July (left) and 14 July (right), reproduced with permission, ©, 2020, IEEE [8]. 31
4.2 Example of a blob that is false positive caused by crops on a farm: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]). 32
4.3 Example of a blob that is a difficult false positive: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]). 32
4.4 Example of a blob that is an ideal true positive: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]). 33
4.5 Example of a blob that is a true positive but with bad coherence: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]). 33
4.6 Example of a blob that is a true positive at a construction site: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps. 34
4.7 True positive blob, Riverglen construction site, Google Earth image 34

4.8 A difficult true positive blob: (a) Sentinel-2 optical RGB Image 1 (b) Displacement Map (c) Coherence Map and (d) Phase Map. 35
4.9 A difficult positive case, Google Earth image showing construction 35
4.10 Confusion Matrix (EfficientNet abbreviated to EffNet) 38
4.11 Performance of models vs no. of parameters showing trends of different architectures: ResNet (orange), EfficientNet (blue) and ResNet V2 (green). 39
4.12 Displacement time series for quarry, showing a seemingly downward trend despite noise 40
4.13 Displacement Time series for Riverglen construction, showing downward and upward trends 40
4.14 Displacement Time series for Cullinan Diamond Mine, showing downward and upward trends 41
4.15 Displacement Time series for construction in blob 135, showing downward and upward trends 41
4.16 Velocity map showing per pixel velocities, subsidence (blue) and uplift (red) 42
4.17 Terrain Corrected Coherence and phase Differential Interferogram 2 July to 14 July 2018 43

List of Tables

3.1 Perpendicular baselines of Sentinel-1 image acquisitions 20

4.1 Blob Detection Algorithm results at threshold of 1 30
4.2 Performance of the models and methods applied with corresponding no. of parameters . 36
4.3 Ensemble definitions 37

Chapter 1

Introduction

The application of state-of-the-art Deep Learning (DL) [1] and Transfer Learning (TL) [2] methods to detect industrial change in satellite imagery using binary classification are discussed in this report. The Laplacian of Gaussian algorithm is used to detect blobs in a displacement map produced by applying Differential Interferometric Synthetic Aperture Radar (DInSAR) processing to a Sentinel-1 image pair [3]. Depending on whether or not subsidence or uplift has indeed occurred in a blob upon visual verification via optical images acquired from the Sentinel-2 satellite a blob is either classified to have had change occur i.e. positive, or classified as noise having caused the blob i.e. negative. A novel balanced dataset is developed which contains 1440 blobs in total. The dataset comprises of DInSAR processed Sentinel-1 [4] displacement, coherence and phase georeferenced imagery combined with the corresponding optical images of the areas in the blobs from the Sentinel-2 satellite [5]. A variety of DL architectures that have weights which are pretrained on the ImageNet [6] dataset are used to apply TL via Feature Extraction (FE) [2] in order for feature maps to be generated after the final layer which is responsible for predictions has been removed. Thereafter, classification is performed on the features using a Logistic Regression classifier [2]. For some of the architectures considered TL via Fine-Tuning (FT) [7] as well as traditional DL which involves training from scratch i.e. Random Initialisation (RI) are also evaluated. Some of the architectures are also combined to form small ensembles that are also evaluated [8]. The DL and TL techniques are compared to binary classification using traditional Machine Learning with feature descriptors as a baseline and are also compared to using a simple Convolutional Neural Network (CNN) which only has two convolutional layers called Simple ConvNet. This chapter discusses the background to the research, the problem statement, the significance and motivation for conducting this research, the research aims and objectives, the research questions, delineations, limitations, assumptions and definitions of technical terms used in this research field. A paper [8] on certain aspects of this research was presented at the 2020 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA) Conference held in Cape Town, South Africa, 29-31 Jan. 2020: https://ieeexplore.ieee.org/document/9041139

1.1 Background

The European Space Agency's (ESA) Sentinel-1 constellation of satellites use C-band Synthetic Aperture Radar (SAR) in order to capture SAR images both in the day and at night no matter the weather [4]. Displacement Maps are produced by using the Differential Interferometric SAR (DInSAR) processing technique which is applied to pairs of SAR images [9]. Time Series techniques have been developed

to arrive at more accurate displacement maps. These involve Persistent Scatterer Interferometry (PSI) [10], Small Baseline Subset (SBAS) [11] techniques and hybrid methods such as SqueeSAR [12]. Land subsidence detection using the DInSAR processing method has been applied in Mexico [13], UK [14], Spain [15], Vietnam [16] and China [17].

1.2 Problem Statement

This research investigates whether or not DInSAR applied to Sentinel-1 satellite imagery can be used to detect industrial development over Gauteng, South Africa, during a year i.e. 28th Nov. 2017 to 5th Dec. 2018. A Blob Detection algorithm is investigated for this purpose. However, since it has a high false positive rate binary classification machine learning models that would be able to automatically detect whether the blobs are indeed true positives or are false positives are also investigated. This research also investigates whether or not state-of-the-art Deep Learning, Transfer Learning from architectures pretrained on the ImageNet RGB images dataset as well as ensembles of models can be used to perform binary classification on the novel dataset of DInSAR processed georeferenced images with their corresponding Sentinel-2 optical satellite imagery. The performance of these models is compared to traditional Machine Learning methods using feature extractors with a simple linear model as a baseline and also to a simple CNN trained from scratch.

1.3 Significance and Motivation

Sentinel-1 imagery is free and provides an ideal opportunity for Earth Observation data to be exploited for the benefit of the country. An application can be developed in the future to allow for the monitoring of development (uplift) in an area in order to identify new construction developments for zoning and town planning purposes. Another use case is to monitor subsidence in mining areas and to detect large sinkholes. Also if a contractor is allocated funds for a particular project the progress can be monitored to see whether work has started and is ongoing or not.

1.4 Research Aims and Objectives

The aim of this research is to apply a current DInSAR technique to detect uplift or subsidence in Sentinel-1 imagery. This research also aims to investigate the use of Deep Learning and Transfer Learning (using models that are trained on the ImageNet RGB images dataset) state-of-the-art methods alongwith ensembles of some models to perform binary classification on the novel dataset of DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced images with the corresponding Sentinel-2 optical satellite RGB imagery. The abovementioned aims of this research will be achieved via the objectives, goals, contained below:

- perform work similar to Sowter et al. [13] with respect to generating a terrain corrected velocity map using images acquired in 12 day intervals over a 1 year period, using a simpler method,
- develop a dataset of Sentinel-1B imagery,
- develop a time series for selected points of interest where its known that development occurred in order to quantitatively measure uplift,

- qualitatively verify visually using optical imagery whether industrial development indeed occurred or not,
- implement the Laplacian of Gaussian algorithm for blob detection in the images to detect uplift or subsidence and quantitatively measure its performance and
- implement a variety of supervised Deep Learning and Transfer Learning binary classification models including ensembles of some models which will be trained, tested and validated on the labelled balanced dataset of blob images containing both positive and negative cases i.e. blob is visually verified to be uplift/subsidence and blob is not uplift/subsidence respectively.

1.5 Research Questions

The research questions are listed below:

1. Can Sentinel-1 imagery be used for industrial development change detection and can change at the selected pixels where construction is known to have occurred be qualitatively verified visually?
2. What is the accuracy of the blob detection algorithm and is the change increasing at each point in the displacement time series?
3. Can Transfer Learning using a pretrained model with weights trained on the ImageNet RGB images dataset be used to perform binary classification on the novel dataset containing DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced images with the corresponding Sentinel-2 optical satellite RGB imagery which contains an equal quantity of positive and negative blobs? Furthermore, which method and architecture has the best performance and is there any relationship between model size and performance?

1.6 Delineations, Limitations and Assumptions

The research focuses on the Gauteng region of South Africa over the period 28th Nov. 2017 to 5th Dec. 2018. The resolution of the satellite imagery is a limitation that prevents the detection of small sinkholes. The size of the blobs are also a limitation as smaller blobs cannot be detected.

1.7 Definitions

- DL - Deep Learning
- ML - Machine Learning
- SVM -Support Vector Machine
- SAR - Synthetic Aperture Radar
- RGB - Red, Green and Blue
- InSAR - Interferometric Synthetic Aperture Radar
- PSI - Persistent Scatterer Interferometry
- SBAS - Small Baseline Subset

- ANN - Artificial Neural Network
- TL - Transfer Learning
- DNN - Deep Neural Network
- DInSAR - Differential Interferometric Synthetic Aperture Radar
- FE - Feature Extraction
- FT - Fine Tuning
- CNN - Convolutional Neural Network
- RI - Random Initialisation
- HOG - Histogram of Oriented Gradients
- LBP - Local Binary Patternn
- ResNet - Residual Network
- EffNet - Efficient Network

1.8 Outline

First the research background and work related to Sentinel-1 and the Differential Interferometric Synthetic Aperture Radar technique, Machine Learning, Deep Learning and Transfer Learning is discussed. Then the methodology employed to conduct the research is discussed as well as the tools used to conduct the research. Thereafter, the results are presented and the research questions are answered building up to the conclusion of the research.

Chapter 2

Background and Related Work

2.1 Introduction

This chapter reviews the literature related to the Sentinel-1 SAR satellite (which is the source of the data) and the DInSAR techniques which are used world-wide for change detection such as subsidence detection, monitoring earthquakes and other natural disasters. The Machine Learning, Deep Learning and Transfer Learning background as well as related work in these fields are also discussed.

2.2 Theory

The theory related to change detection, the Sentinel-1 satellite and the DInSAR technique, Image Processing, Machine Learning, Deep Learning and Transfer Learning are discussed below as an understanding of these concepts is essential to this research.

2.2.1 Change Detection

Within the Remote Sensing (earth observation using satellites) context change detection refers to the procedure whereby differences that are contained in satellite imagery obtained at different times over the earth's surface are detected. There are pixel-based and also object-based change detection techniques. There are practical applications of knowledge related to change in land-use and land-cover such as quantifying deforestation, monitoring of disasters (such as fire, etc.,), land usage management, etc., [18]. Satellite imagery has become more easily accessible in order to perform remote sensing change detection with the advent of the Sentinel-1 satellite.

2.2.2 Sentinel-1 and DInSAR

The Sentinel-1 satellite mission is the European Space Agency's (ESA) constellation of two polar-orbiting satellites v.i.z. Sentinel-1A and Sentinel-1B equipped with Synthetic Aperture Radar (SAR) sensors (C-band) which take SAR measurements both during the day and at night under all weather conditions. This imagery provides a free alternative to proprietary satellite imagery although it has a lower resolution than X band satellite imagery. Each satellite (Sentinel-1A and Sentinel-1B) has a revisit frequency of 12 days over Africa and 6 days over Europe [4, 8].

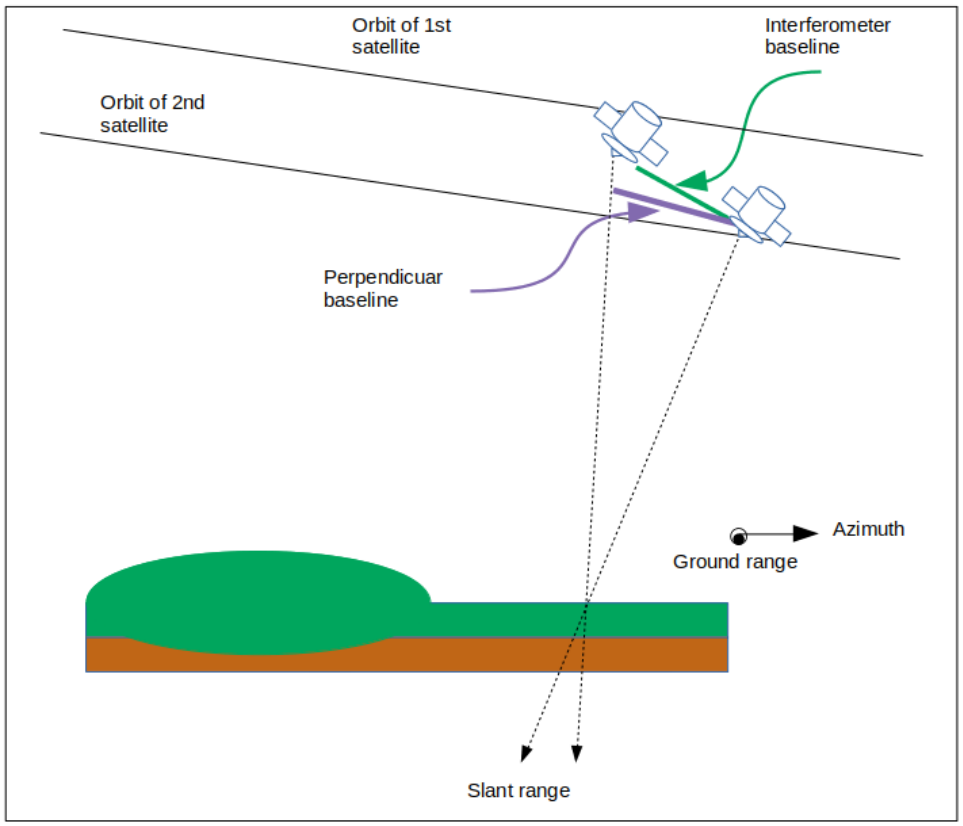

Figure 2.1: Interferometric SAR geometry showing interferometer and perpendicular baselines.

There are four SAR acquisition modes namely Stripmap (SM), Interferometric Wide swath (IW), Wave mode (WV) and Extra-Wide swath (EW). The Interferometric Wide swath (IW) mode facilitates employing a swath width that is large, 250 km, along with a geometric resolution of 5m $\times$ 20m which is moderate. The Terrain Observation with Progressive Scans SAR (TOPSAR) method is used to image three sub-swaths. To acquire a better quality homogeneous image throughout the swath and prevent the TOPSAR defect of scalloping occurring azimuthal steering of the beam from backwards to forwards occurs per burst alongwith the beam being steered in range. This is similar to what is done in the Scanning Synthetic Aperture Radar (SCANSAR) method [4]. In order to facilitate the ability to perform interferometry TOPSAR provides adequate azimuthal overlap in the Doppler spectrum and the wave number spectrum elevationally.

The Sentinel-1 satellite constellation uses SAR which is an active microwave imaging system [9] that records distinct measurements of the amplitude of radar echo signal as well as its corresponding phase which are measured at a specific instant [19]. The process of comparing the perturbations in phase and multiplying the amplitudes of two particular SAR Single Look Complex (SLC) images that are acquired at different times or simultaneously but from different look angles is called Interferometric SAR (InSAR). The interferometer baseline is the distance between the two satellite orbits as shown in Figure 2.1, adapted from Ferretti et al [9]. The perpendicular baseline is the projection which is perpendicular to that of the slant range direction as shown in Figure 2.1. The process of performing the Differential Interferometric SAR (DInSAR) method refers to using a Digital Elevation Model (DEM) in order to subtract the altitude component of the phase from the original interferogram generated by InSAR. This process results in the generation of a differential interferogram [9, 8].

The occurrence of ground deformation is inferred if the differential interferogram contains any

fringes. This is usually as a result of the occurrence of subsidence or uplift. In order to generate a Displacement Map which shows the displacement at each pixel in the map the wrapped differential interferogram is unwrapped via the Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping (SNAPHU) [20] tool. Terrain Correction, which is also referred to as Geocoding, is then applied to the Displacement Map. The Range Doppler orthorectification method is used for Terrain Correction. Distortions in distance occur due to there being differences topographically between the Sentinel-1 sensor's tilt and an image. The procedure used to correct the distortions in distance is referred to as Terrain Correction. Furthermore, the quality of the interferograms generated deteriorates due to artefacts which develop in the interferogram that are caused by effects due to phase noise and the composition of the atmosphere. Thus, coherence estimation is required [21, 3, 20, 8].

Phase noise

The effect due to the manner in which the various scatterers change over the interval between the acquisition of the two SAR images contained within each particular block of resolution is known as phase noise. Phase noise can be caused by temporal variation in scatterers. The orientation of vegetation such as crops, tree leaves and branches, etc. due to wind is an example thereof. There are also scatterers which do not change over time between the acquisitions and are referred to as persistent or stable scatterers. Examples of these scatterers include rocky terrain and infrastructure such as roads, bridges, buildings, etc., in urban areas [9, 8].

There are other ways that phase noise can be caused. These include baseline errors, volumetric scatterers, scatterers that are non-planar and exist in a volume such as tree branches, and the topographical DEM being inadequately subtracted. In areas of high coherence terrain deformation between the image acquisitions implies ground uplift or subsidence [9, 8].

Atmospheric composition effects

The composition of the atmosphere especially with respect to the parameters of humidity, temperature and pressure is unique to the specific acquisition times of the satellite images. Thus, the travel path of the radition in these SAR images is influenced in different manners. Since hot air has a higher water vapour retention capacity than that of cold air images acquired on days that are very hot are prone to an increased amount of atmospheric noise. Other conditions such as rain, snow and strong wind also normally cause phase coherence to be lost. Both altitude measurements and terrain deformation are detrimentally influenced by atmospheric noise. Along the interferogram this is constrained to an interferometric peak-to-peak phase change of 2π [9, 8].

Coherence and Phase Estimation

In order to know where the phase in the Differential Interferogram can be trusted the phase noise needs to be estimated. The process of estimating the phase noise from a set of SAR image acquisitions is called Coherence Estimation. The Coherence is a value which ranges from 0, pure noise, to 1, pure signal. When the altitude component is subtracted from the pair of images the modulus of the coefficient of their cross-correlation function is locally estimated within a small window to be the local coherence. A two-dimensional Fast Fourier Transform is then used to estimate the phase [9, 8].

In order to perform change detection DInSAR is used to generate displacement, phase and coherence maps combined with the corresponding optical images. Before binary classification can be

performed Image Processing will need to be carried out.

2.2.3 Image Processing

Image processing is a subfield of signal processing and uses algorithms to digitally process images [22]. There are a variety of mathematical operations that have been developed to process images to enhance them or extract features from them in order to enable computer vision such as whitening, Gaussian Blur, Histogram Equalisation, Segmentation, Edge Detection, Local Binary Patterns (LBP), SIFT descriptors, Histogram of Oriented Gradients (HOG), Haar-like Filters, etc., [23]. Images need to be preprocessed which may involve normalisation and also feature extraction in the case of traditional Machine Learning before they can have Machine Learning models applied to them for binary classification of the change that was detected in the image.

2.2.4 Machine Learning

Machine learning (ML) refers to the scientific field of study of the algorithms as well as statistical models that computers (machines) rely on in order to carry out certain tasks without having been given specific instructions, instead depending upon their ability to perform pattern recognition and inference. ML exists as an artificial intelligence subdomain [24]. T. Mitchell defined Machine Learning as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [25] and [1]. Binary Classification is a branch of Supervised Machine Learning where a model is trained to predict a label (either positive or negative) based on the labelled training data it was trained on. To ensure that the best model is achieved hyperparameter tuning is required.

Hyperparameter tuning

Due to sampling error and the variance (a measure of how the predictions will change if trained on a different part of the dataset [26]) [27] it is wise to apply cross validation to the dataset when training a machine learning model by first removing a hold out Test dataset that is 20% of the dataset. Then dividing the remaining dataset into Training and Validation sets in a 80:20 ratio k times. A good choice of k is 10. The machine learning model is then trained on the Training dataset and its performance measured on the Validation dataset 10 times. Hyperparameter tuning is the process of selecting the optimal parameters for a particular model during this cross validation of the dataset process. The best parameters are then used and trained on all the Training and Validation data and then the performance is evaluated on the hold out Test dataset that was not trained on at all in order for an accurate reflection of the actual performance of the model to be yielded [28, 24]. In order to improve the performance of the classifier Deep Learning models should also be considered depending on dataset size.

2.2.5 Deep Learning (DL)

A Deep Neural Network (DNN) refers to an Artificial Neural Network (ANN) that has many hidden layers. Deep Learning is the scientific field of study of the design and application of DNNs. An ANN is comprised of layers which have neurons and each neuron maps its inputs to a certain output via an activation function such as a Rectiified Linear Unit (ReLU). An ANN can be used to model any function

[1]. As the size of the neural network increases with depth so too does the number of parameters that need to be trained and the time, computational power and data required to train the DNN.

During training a loss function is being minimised and an optimiser is required in order for convergence of global minima with reference to utilising learning that is based on gradient descent methods. An important parameter to these optimisers is the learning rate which if assigned a value that is inadequately high it can result in the optimisers overshooting the global minima, consequently merely bouncing from wall to wall in functions that are convex and not actually converging. Conversely, the duration to train the network required is dramatically increased should the learning rate parameter be assigned a value which is very small. There are also saddle points which the optimisers can incorrectly take to be the global minima when they are actually local minima [25]. The Stochastic Gradient Descent (SGD) optimiser [25] is more accurate than the Adam [29] optimiser [8]. However, the Adam optimiser can converge faster.

Below more Deep Learning theory related to DL application to object recognition and image classification is discussed [8]. This includes Overfitting, Dataset Augmentation, Back Propagation and the vanishing gradient problem, Convolutional Neural Networks (CNNs) and CNNs for Computer Vision.

Overfitting in Deep Neural Networks

Overfitting to the data is a problem that DNNs are prone to due to their numerous non-linear hidden layers which makes them extremely expressive models able to learn extremely complicated relationships connecting their inputs to their outputs. However, numerous input to output relationships actually occur due to sampling noise and manifest themselves only in the training set and not in the validation or unseen testing dataset even if these sets are drawn from the same distribution. In order to prevent the model overfitting to the biases in training data that would detrimentally affect the DNN's ability to generalise for unseen data regularisation is required. A renowned method of regularisation where a few neurons coupled with the links they have to other neurons are deleted during training is called Dropout. Early Stopping is also used in order to stop training when the network starts overfitting [30], [25], [24], [1] and [8]. To mitigate Overfitting Dataset Augmentation can also be done.

Dataset Augmentation

The method of artifically expanding a small dataset by performing an image processing operation such as rotating the images at an angle, flipping right to left, shifting, cropping, zooming, etc., is referred to as Data Augmentation. It is sometimes required due to DNNs requiring datasets that have images numbering in the tens of thousands due to DNNs having parameters of the order of millions. Data Augmentation assists with the training of DNNs and improves DNN performance. Image Preprocessing is used on the training, validation and unseen testing datasets. However, Data Augmentation must not be used for the testing and validation sets and only used for the training set [31, 8].

More often in the field of Deep Learning Data Augmentation provides regularisation by using random image processing functions with replacement to augment the dataset instead of expanding it. Thus, the generalisability of the DNN model is improved [31]. If Data Augmentation is done for expanding the image dataset it is imperative to consider the problem domain in order to avoid generating images of objects that do not actually exist realistically. However, training a Generative Adversarial Network (GAN) would be an exception to this rule [32]. Tanaka et al. achieved higher performance using Data Augmentation in comparison with using merely the original dataset. Perhaps this is due to the dataset being of small size. Tanaka et al. suggest the use of this method of generating artificial datasets if

sufficient data is not available in order to adequately train a model either as a result of the data being unavailable or because of data privacy issues such as when using medical datasets [32, 8]. Optimal Dataset Augmentation can today even be applied using Machine Learning and is called AutoAugment [31].

AutoAugment

Perez et al. discuss Neural Augmentation proposing a method for a DNN to learn data augmentations that optimise classification performance [33]. AutoAugment was introduced by Google to enable automatic learning of data augmentation policies instead of using a manual approach [31]. The basis of AutoAugment is AutoML (Automatic Machine Learning). AutoML used ML for the design of the ML neural network architecture for NASNet [34] and also used ML to design an optimal optimiser using reinforcement learning [35]. The authors deduced that just like how weights can be transferred in Transfer Learning so too can data augmentation policies be transferred from one dataset to another to improve performance on the new dataset. AutoAugment has an increased performance on smaller datasets [31]. In addition to Overfitting another issue that was encountered by Deep Neural Networks that cannot be mitigated with Dropout and Dataset Augmentation is the vanishing gradient problem.

Back Propagation and vanishing gradient problem

A feedforward artificial neural network is an ANN where the connections between the neurons do not form a cycle like in a Recurrent Neural Network (RNN) which has a feedback loop. The Back Propagation algorithm is used to train feedforward ANNs in a supervised manner by efficiently computing the gradient of the loss function relevant to each weight employing the chain rule going backwards one layer at a time iteratively. DNN algorithms that require the complete gradient to be computed such as the Back-Propagation and Real-Time Recurrent Learning algorithms encounter a problem where the error signals going back through time tend to vanish since gradient based learning methods are used and the chain rule is used to repeatedly multiply the gradients to the weights to optimise a loss function. Eventually the gradient vanishes [1] and performance no longer improves [36]. The vanishing gradient problem is only encountered in deep networks and a solution has been recently introduced by a modern Convolutional Neural Networks (CNN) architecture.

Convolutional Neural Networks (CNNs)

An ANN that is used to perform image classification and object recognition tasks is called a Convolutional Neural Network (CNN) [1]. CNNs have quite an old history that goes as far back as the late 1950's [37]. LeCun et al. showed in the late 1980's that an improvement in classification performance can be obtained if an ANN is directly used with the preprocessed images without the use of feature vectors [38]. However, interest in using CNNs for Computer Vision was sparked when AlexNet (a DNN which uses CNNs) was developed by Alex Krizhevsky who won the 2012 ImageNet Large Scale Visual Recognition Competition. Historically, training DNNs based on CNNs was an impossible task due to computational expense. However, recent technological advances such as the Dropout regularisation technique being invented that dramatically improved DNN regularisation along with Graphical Processing Units (GPUs) being developed which tremendously reduce computational cost enabled AlexNet's success [39, 8].

Convolutional filters are used in a CNN for automatic feature map generation. Thus, traditional manual feature selection as well as feature engineering using classical feature extraction techniques are not needed [1]. The convolutional filters are able to detect all of the relevant details in the images given

enough training data for pattern recognition to occur. Since labelled data is a key component to DL algorithms achieving a high performance a database which comprises of more than fourteen million labelled images which includes hundreds of classes called ImageNet was developed [6]. A golden standard for the quantification of computer vision performance of CNNs was developed. It occurs annually from 2010 and is called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [40, 8].

CNNs for Computer Vision

Below are some CNN-based DL models that are used for Computer Vision [8].

1. **Residual Networks**

   DL was transformed by the advent of Residual Networks (ResNet) [41] which pioneered the concept of explicitly reformulating the layers so that the layers learn residual functions w.r.t. the inputs of the layers as opposed to learning functions that are unreferenced. This laid the foundation for DNNs that are significantly deep to be trained and also solved the dilemma of the vanishing gradient [36]. The ResNet-152 architecture has lower complexity than the VGGNet architecture even though ResNet-152 (comprises of 152 layers) is eight $\times$ deeper than the VGGNet. A 3.57% Top-5 error on ImageNet was obtained by an ensemble of ResNet models winning the 1st place in ILSVRC 2015 [41, 8].

   He et al. advanced ResNet by developing ResNet V2 which is based on a new residual unit that improves generalisation and decreases the complexity of training the model. He et al. discovered that it is possible to propagate the forward and backward signals from a certain block to any of the other blocks directly with the aid of identity mappings by using them as the skip connections and after-addition activation. The discovery was made after He et al. studied the residual blocks' driving force i.e. their propagation formulations. A 4.62% error was obtained with CIFAR-10 by a ResNet with 1001-layers the design whereof was made possible as a result of this important discovery. It is deeper than VGG19 by a factor of 53 [42, 8].

2. **Inception-Residual Networks**

   The Inception architecture was the winner of the 2014 ILSVRC although it only had 22 layers. It also pioneered the notion of using multi-scale processing and keeping computational cost fixed by expanding the width of a DNN architecture [43]. An Inception architecture that also has the benefit of residual connections is called the Inception-ResNet architecture. Training time of the DNN is extremely reduced due to the residual connections having been added. For very wide Inception-ResNets scaling the activation functions properly leads to stabilised training of the DNN. A Top-5 error rate of 3.08 % was achieved on ILSVRC with an ensemble which comprised a single fourth version Inception network as well as three ResNets [44, 8].

3. **ResNeXt Networks**

   The ResNext DNN architecture introduced the notion of expanding a DNN in a different dimension i.e. the cardinality which is the total transformations. It introduced the notion of repeating a basic unit that is aggregating a group of transformations with similar architecture. This idea was termed "network in neuron". In the 2016 ILSVRC ResNext achieved 2nd place [45, 8].

4. **EfficientNet Networks**

   Tan et al. [46] postulated that to increase accuracy one should wisely choose the parameters of depth, width as well as resolution. This was discovered after the scaling effect was investigated.

Tan et al. discussed a method of scaling applied to MobileNets and ResNets that similarly scales each of the parameters using what Tan et al. call "an optimal compound coefficient". The EfficientNet family of architectures which the authors claim increase performance while doing so in a more efficient manner were develeoped via scaling up the new baseline DNN that was discovered via Neural Architecture Search (NAS). The best performing model developed is an EfficientNet-B7 which Tan et al. claim is 6.1 $\times$ faster at predictions yet 8.4 $\times$ smaller versus the top performing network and it achieves state-of-the-art accuracies on ImageNet i.e. 97.1% and 84.4% Top-5 and Top-1 respectively. Tan et al. reported that EfficientNets have good performance when used for Transfer Learning. On the CIFAR-100 dataset a 91.7% accuracy is reported by Tan et al.. On the Flowers dataset an accuracy of 98.8% is achieved. Good performance is also reportedly achieved on another three datasets which are used for Transfer Learning. This was achieved requiring less training time since the parameters are an order of magnitude less than other models [46, 8].

2.2.6 Transfer Learning (TL)

Transfer Learning (TL) refers to using a pre-trained DL architecture with data for a new problem by removing the final predictive layer. The pretrained model is trained on a dataset that is similar to but much larger than the dataset at hand, usually in object recognition and classification tasks an architecture with weights trained on ImageNet is used. This is possible since the model has learnt rich discriminative filters from training on a large dataset (millions of images in the case of ImageNet) and can be applied to classes it was not trained on before to transfer knowledge. This either removes the need for the training process totally or extremely simplifies it since DL has a significant computational cost needing hardware such as multiple GPUs which are expensive and for a sufficiently deep DNN with a dataset that is large this training process can actually require days or weeks. DNNs usually have millions of parameters and good performance is not achieved if a sufficiently large dataset is not available. However, it is still possible to use TL to derive the DL benefits of improved accuracy much faster than and sometimes generating even higher performances as compared to traditional DL but without the time and computational power required to train from scratch by simply transferring the knowledge [2, 8]. There are three types of TL, namely Feature Extraction, Fine Tuning and Initialising with Weights.

Feature Extraction Transfer Learning

When performing the Feature Extraction TL discriminative filters that are rich extractors of features are transferred from a pre-trained model and can be used on classes that they were not even trained on in order to generate feature maps when applied to different datasets. Then a basic ML model such as linear Support Vector Machines, linear Logistic Regression, Random Forest, etc., is trained on these feature maps (after standardisation of the dataset by scaling the features to unit variance and zero mean). This transfer of knowledge enables datasets of significantly smaller size to immensely derive benefit. The task is, thus, transferred to the simple ML domain where the model can be trained quickly on CPUs because when doing TL via FE a DL architecture is not actually trained at all. When applying TL it is necessary to use the same preprocessing that was used on the data which the pretrained model was trained on used [2]. RGB mean subtraction preprocessing is used in the case of ResNet [42, 8].

Fine Tuning Transfer Learning

When performing Transfer Learning using Fine Tuning the final layer that is responsible for predictions is removed. In order to make relevant predictions a customised Softmax final predictive layer is added

suited in terms of the number of classes to the task at hand. Then all the layers must be frozen except for the custom Softmax layer and the network is trained using a small learning rate. Thereafter, the final convolutional layer is also unfrozen. It is trained along with the final layer dedicated to predictions [7, 8].

The Stochastic Gradient Descent (SGD) optimiser with a very slow learning rate of 0.0001 and momentum of 0.9 is used instead of using an optimiser with an adaptive learning rate so that the magnitude of the weight updates remains very small, thus, avoiding adversely affecting the learned filters [47]. It is possible for Transfer Learning using Fine Tuning to deliver a higher performance than when applying Transfer Learning using Feature Extraction [7].

$$v_{t+1} = \mu v_t - \epsilon \nabla f(\theta_t) \tag{2.1}$$

$$\theta_{t+1} = \theta_t + v_{t+1} \tag{2.2}$$

Momentum speeds up SGD and dampens oscillations by accumulating a velocity vector in directions where reduction of the loss function is sustained across iterations. Equations 2.1 [48] and 2.2 [48] specify the update criteria where v_t is velocity, θ is the set of weight parameters, $f(\theta_t)$ is the objective loss function to be minimised, ϵ is the learning rate (must be > 0), μ is the momentum coefficient which must be in the range 0 to 1 and $\nabla f(\theta_t)$ is the gradient of $\theta(t)$ [48, 49].

Initialising with weights

When training all the layers of a model in order to reduce the convergence time as compared to Random Initialisation (traditional DL) it is possible for a model to be initialised using weights from a model pretrained on a similar dataset. However, this method takes more time to train than TL using FE and TL using FT [50], [51], [52] and [53, 8].

Kornblith et al. concluded that the ImageNet Top-1 accuracy obtained by a specific architecture is a strong predictor of the accuracy obtained when using that architecture for Transfer Learning using Fine Tuning. However, this relationship does not hold true when using Transfer Learning using Feature Extraction. From all the architectures evaluated ResNets with Logistic Regression classifiers performed the best at Feature Extraction for 75% of the twelve data sets evaluated. Kornblith et al. also noted that for data sets that are small which are not present in the ImageNet dataset traditional DL can produce better performance than Transfer Learning. Furthermore, Kornblith et al. also noted that TL using Feature Extraction performs better with smaller datasets than TL using Fine Tuning [7, 8].

2.3 Works in Topic

Below the works in topic are discussed. The following DInSAR techniques are discussed below: Persistent Scatterer Interferometry (PSI) [10] techniques, Small Baseline Subset (SBAS) [11] techniques and hybrid techniques such as SqueeSAR [12]. Thereafter, Deep Learning and Transfer Learning applications are also discussed.

2.3.1 Persistent Scatterer Interferometry (PSI)

PSI involves identifying and exploiting stable natural reflectors which are called Persistent Scatterers (PS) in a series of InSAR images that are temporally far apart [10]. The analysis is done at the highest resolution possible since no multilooking is performed which decreases the resolution. PSI is computationally intensive and is generally used for small region analysis and monitoring of infrastructure [19]. Often the dimension of the PS is smaller than that of the resolution cell and when this happens then even in the case of interferograms which have baselines that are larger than the decorrelation baseline the coherence is good. The full set of ESA ERS satellite imagery were successfully used. Submeter Digital Elevation Model (DEM) accuracy and terrain motion detection at a millimetric scale are also achievable with PSI. This is due to the ability of PSI to facilitate the estimation and removal atmospheric phase screen (APS) contributions with a sufficient spatial resolution at least in rocky and urban regions in the study area as sufficient point scatterers existed even if the two SAR images were many years apart which allowed for the extraction of important phase information on a sparse grid of targets. In vegetated areas artificial PS can be made using corner reflectors with a cross section of $1500m^2$ to improve the PS density. PSI is applied to Ankona, Italy, using ERS data [10]. PSI is used for deformation monitoring using Sentinel-1 IW data in Catalonya, Spain [15]. A 6750 km^2 area deformation map was developed. A good PS density of 620 PS/km^2 for urban regions that had good coverage as well as interconnection between the urban areas was achieved [15].

2.3.2 Small Baseline Subset

The Small Baseline Subset (SBAS) algorithm is a DInSAR algorithm for analysing the temporal development of terrain deformations based on using pairs of differential interferograms that have a small perpendicular baseline to limit the noise due to spatial decorrelation. The temporal sampling rate is increased by applying the singular value decomposition method which facilitates the joining of independent datasets that have large perpendicular baselines. The algorithm uses regions that have an acceptably high level of coherence and are robust to errors contained in the DEM that was used in producing the differential interferograms. Atmospheric phase noise is filtered out by using the temporal and spatial information that are contained in the processed dataset [11]. SBAS analysis is done at a lower resolution than that of PSI analysis since multi-looking is performed to remove noise. SBAS analysis is usually used for large area analysis [19]. SBAS or variations of it have been applied to land subsidence detection widely. The areas that will be discussed are Italy [11], Mexico [13], UK [14], Vietnam [16] and China [17].

Italy

Berardino et al. obtained a deformation time series that correlates with the dynamics of the deformations detected from a dataset over the period 1992-2000 from the European Remote Sensing satellites over the Campi Flegrei caldera (volcanic area) and Naples, an Italian city [11].

Mexico

Sowter et al. use the Intermittent Small Baseline Subset (ISBAS) time-series analysis algorithm [13]. It was produced based on the SBAS algorithm to improve results in agricultural and rural regions since PSI and SBAS are excellent in urban and rocky regions but are generally inadequate in rural and agricultural regions where the coherence of the pixels intermittently fluctuate between adequate and inadequate. This

is specifically applicable for computing the linear components of subsidence in areas where large scale deformation occurs [14]. From an SAR image stack many low-resolution two-pass differential interferograms are produced using the small baseline criteria. For pixels which have a sustained coherence that is high in the whole image stack regression of the phase is used to estimate the linear velocity as well as a height correction for the DEM which is used in the two-pass DInSAR [13].

Sowter et al. present the results of an ISBAS analysis of 18 Sentinel-1 IW products covering a 39,000 km^2 region in Mexico for the period 3rd Oct. 2014 - 7th May 2015 using TOPSAR are presented. A total of 143 small baseline differential interferograms were used in the ISBAS processing. Other C-band and L-band DInSAR investigations which were performed in the 1990s and 2000s were used to validate the results of the ISBAS algorithm applied to Sentinel-1 relating to the magnitude and extent of terrain deformation that has occurred in Mexico City, Chalco, Ciudad Nezahualcyotl and Iztapalapa. Sentinel-1 ISBAS subsidence velocities in certain regions are more than -24 cm/year along the Sentinel-1 line-of-sight. This is equivalent to over approximately -40 cm/year vertical rates of subsidence [13].

UK

Sowter et al. apply ISBAS to a stack of ENVISAT images (32 ESA ENVISAT Advanced Synthetic Aperture Radar (ASAR) sensor images over the duration June 2003 - Feb. 2009. A total of 124 multi-looked differential interferograms were produced with perpendicular baselines that are less than 200 m and temporally separated by 4 years or less. The window size that was used for multilooking was 4×20 SLC pixels. This resulted in pixels of approximately 100m $\times$ 100m with reference to ground range. The Coherence threshold was taken to be 0.25. The area of study was the coalfields in Leicestershire and South Derbyshire where there was correlation between the observed linear displacements and coal mining in the area. The authors also mentioned that the displacements seemed to be bounded by fault structures. The results of ISBAS were much more dense, 35032 points, and vastly distributed, almost encompassing the entire region of interest as opposed to the SBAS results where the velocities' distribution was sparse in general, 5162 points, and mainly situated in regions that are urban. Thus, ISBAS was shown to significantly improve upon the SBAS algorithm [14].

Vietnam

Nguyen et al. use SBAS to derive line-of-sight displacement velocities over the period 3rd April 2016 - 13th Sept. 2017 using Sentinel-1A images to monitor subsidence in Hanoi, Vietnam. Two regions of subsidence were found to have a rate of displacement of 18-27 mm/year. A new region of subsidence was also discovered with an estimated rate of subsidence to be greater than 40 mm/year. Geology together with excessive groundwater extraction was cited as a possible cause of the subsidence [16].

China

Huang et al. produce time series and deformation maps of displacement for Nanjing, China using the SBAS algorithm. There were 13 differential interferograms produced using eight ERS2 SAR acquisitions over Nanjing during the period Aug. 1996 to April 2000. The main subsidence was found to be along the Yangtze River and the region to the west of Qinhuai River with subsidence peaking at around 12.0 cm and the mean velocity around 3.3cm/year [17].

2.3.3 SqueeSAR

A newer and more robust approach is SqueeSAR which is an extension of PSI and combines both PSI and SBAS concepts [19]. SqueeSAR aims to exceed the limitations of PSI by the extraction of geophysical parameters that are not merely acquired from deterministic point scatterer objects but are also from distributed scatterers (DS) in DInSAR data stacks. SqueeSAR simultaneously processes both PS and DS whilst considering their respective statistical characteristics instead of using hybrid processing chains that apply a minimum of two algorithms sequentially to the stack of images and then combine the results. When the coherence matrix of each DS is optimally *squeezed* to produce a vector of optimal (wrapped) phase values then both PS and DS can be processed together without requiring the unwrapping of hundreds of interferograms and without requiring major alterations to the standard PSI processing chain [12].

In order to gauge if SqueeSAR made any improvement to PSI an Alpine area in Italy with vegetation and rough topography were selected since this sort of terrain is relatively challenging for InSAR analysis. The image stack that was used comprised of sixty five RADARSAT images over the period April 2003 - March 2008. A few DS are also identified as PS by the PSI processing chain enabling the comparison of the deformation time-series estimated by SqueeSAR and the deformation time-series estimated by PSI. SqueeSAR produces a higher quality displacement time-series of the DS since it significantly enhances the Signal to Noise Ratio (SNR) decreasing the noise in the time-series. The study region is mostly mountainous and around 6000 targets are identified as PS by PSI. SqueeSAR identifies 4000 of the said 6000 targets as DS. A mean increase in DS coherence of approximately 0.1 was observed. SqueeSAR increases the spatial density of measurement points (MP) significantly from 85 PS/km^2 to 450 MP/km^2. This also improves the quality of the time-series since it allows for more accurate estimation and removal of atmospheric effects. This has enabled the enhancement of defining the footprints of sliding areas and has even lead to identifying new sliding areas which were not previously detected by PSI. However, a major drawback is that to achieve the improvements of SqueeSAR a computational processing time of around four times longer than PSI is required beginning from the focused and co-registered interferograms [12].

2.3.4 Applications of Deep Learning and Transfer Learning

Deep Learning has been applied to document classification using text datasets, computer vision i.e. object detection and image classification in both video and image datasets, speech recognition in audio datasets, computational biology, etc. [30]. Deep Learning has also been applied to time series data for various applications such as voice recognition, facial recognition, forecasting, etc., [1]. Deep Learning is applied to optical images from Sentinel-2 to detect changes in urban areas with CNNs for feature representation and Recurrent Neural Networks (RNNs) for temporal modelling by Papadomanolaki et al., an accuracy greater than 95% was achieved [54]. The DL CNNs GoogLeNet and ResNet50 are also applied by Helber et al. on optical images from Sentinel-2 for land use and cover image classification. Performance accuracies of 98.18% and 98.57% respectively are realised [55]. Wang et al. apply DL to Sentinel-1 SAR images in order to automatically geophysically classify the images [56]. Wang et al. apply TL to Sentinel-1 imagery to detect ships in the ocean [57]. Saha et al. use an unsupervised DL-based method in order to detect change in Sentinel-2 optical imagery [58]. Qiu et al. use Deep Learning with multi-seasonal Sentinel-2 optical imagery with a ResNeXt DNN to map human settlements [59, 8].

2.4 Conclusion

DInSAR is used worldwide for subsidence monitoring via developing a deformation time series via a number of techniques such as PSI, SBAS and SqueeSAR to name some of the techniques. Researchers have successfully used DInSAR for developing deformation time-series which correlates with actual deformation that occurred. Deep Learning has been applied to document classification using text datasets, computer vision i.e. object detection and image classification in both video and image datasets, speech recognition in audio datasets, computational biology, etc. [30]. Deep Learning has also been applied to time series data for various applications such as voice recognition, facial recognition, forecasting, etc., [1]. Deep Learning and Transfer Learning have also been applied to satellite imagery. However, Deep Learning and Transfer Learning from models trained on the ImageNet dataset of RGB images has not been applied to a novel dataset containing DInSAR processed SAR imagery coupled with Sentinel-2 optical imagery and this is, thus, novel.

Chapter 3

Research Methodology

3.1 Introduction

This chapter discusses the research design and the methodology that was employed to carry out the research. Thereafter, Candidate Selection is discussed explaining how the blobs are detected. Binary Classification using traditional Machine Learning models which employ feature descriptors, a simple Convolutional Neural Network model, Deep Learning models, Transfer Learning models and ensembles of these models are subsequently also discussed. Furthermore, the limitations of the research are also discussed.

3.2 Research Design

The research design incorporates a mixture of qualitative, quantitative and experimental aspects. The qualitative aspect relates to visually verifying that indusrial development indeed occurred in a blob identified by the Blob Detection Algorithm as a positive blob. The quantitative aspect relates to the performance of the Blob Detection algorithm and the performance of the Binary Classification models. The experimental aspect relates to investigating whether or not Transfer Learning using models trained on the ImageNet RGB images dataset can be used on the novel dataset of DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced images with the corresponding Sentinel-2 optical RGB satellite imagery. The experimental aspect also relates to finding the best Deep Learning model, Transfer Learning model or ensemble of models with the highest performance at the Binary Classification task.

3.3 Methodology

Figure 3.1 illustrates the region that the full Sentinel-1B Interferometric Wide (IW) swath mode Single Look Complex (SLC) Vertical-Vertical (VV, both send and receive in vertical polarisation) image covers. The spatial resolution of the SLC images is 5m (ground range) $\times$ 20m (azimuth) and the swath width is 250km. The image is split vertically into 3 swaths [4], Gauteng falls into IW1 (swath 1) which is the right hand side one-third of the full image. Only bursts 3 to 7 are used to avoid extremely long processing times. A dataset is used which comprises of images from 28th Nov. 2017 to 5th Dec. 2018 that are 12 days apart, where imagery was available. This is to ensure a small temporal baseline so as to limit noise in the interferogram. A time series of uplift is developed for all the pixels over that particular time period.



Figure 3.1: Areas that the full Sentinel-1 image covers

Optical imagery was also downloaded using Sentinel-2 [5] and processed to qualitatively visually verify that construction was indeed occurring. The Laplacian of Gaussian [60] algorithm is applied to the 2nd July to 14th July 2018 Terrain Corrected Displacement Map (Line of Sight) to detect uplift/subsidence blobs. Blob displacement maps, phase maps, coherence maps and optical images from Sentinel-2 on 1st July 2018 and 16th July 2018 are both subsetted per blob. Numerous supervised Machine Learning, Deep Learning and Transfer Learning models for Binary Classification are implemented, trained and tested on a dataset of these images containing both positive and negative cases i.e. image contains a blob and image does not contain a blob respectively.

The research instruments, data and analysis are discussed below.

3.3.1 Research Instruments

The Sentinel-1B SAR Interferometric Wide (IW) swath mode image dataset and the Sentinel-2 optical image dataset are available from the European Space Agency's Copernicus Open Access Data Hub [61]. The Sentinel Application Platform (SNAP) Sentinel-1 toolbox [62] software is used to process the Sentinel-1 images and generate differential interferograms and displacement maps. The tutorials in [63, 21, 3] are the basis of the DInSAR processing used to generate differential interferograms over the said period. Each differential interferogram is unwrapped with the Statistical-Cost Network-Flow Algorithm for Phase Unwrapping (SNAPHU) [20] tool. The *Sen2Cor* processor is used to process the Sentinel-2 optical imagery from Top of Atmosphere, Level 1C, products to Bottom of Atmosphere atmospherically corrected, Level 2A, products at the 10m resolution [64]. Google Earth [65] imagery is also used to visually qualitatively perform verification for many blobs that are used in the binary classification models since it was sometimes not deducible from the Sentinel-2 optical RGB imagery alone as to whether or not a particular blob was indeed a true positive blob or actually a false positive.

3.3.2 Data

The Sentinel-1 SAR data and Sentinel-2 optical imagery datasets are available from the European Space Agency's Copernicus Open Access Data Hub [61] for free. A dataset comprising of 28 differential

interferograms, phase maps, coherence maps and terrain corrected displacement maps was generated.

The following were generated for the Sentinel-1 dataset in 12 day intervals over the one year period:

- terrain corrected line of sight displacement maps,
- differential interferograms before phase unwrapping,
- terrain corrected displacement maps and
- coherence maps.

The perpendicular baselines of the interferograms generated are shown in Table 3.1. The optimal perpendicular baseline is between an absolute value of 150 and 300 respectively. For small baselines the interferogram does not contain a significant amount of useful information [9].

Table 3.1: Perpendicular baselines of Sentinel-1 image acquisitions

From	**To**	**$\perp$ baselines**
28 Nov 2017	10 Dec 2017	7
10 Dec 2017	22 Dec 2017	35
22 Dec 2017	3 Jan 2018	56
3 Jan 2018	15 Jan 2018	23
15 Jan 2018	27 Jan 2018	10
27 Jan 2018	8 Feb 2018	42
8 Feb 2018	20 Feb 2018	-6
20 Feb 2018	4 Mar 2018	8
4 Mar 2018	16 Mar 2018	-49
16 Mar 2018	28 Mar 2018	-31
28 Mar 2018	9 Apr 2018	108
9 Apr 2018	21 Apr 2018	-87
21 Apr 2018	3 May 2018	30
3 May 2018	15 May 2018	-17
15 May 2018	27 May 2018	45
27 May 2018	8 June 2018	-13
8 June 2018	2 July 2018	39
2 July 2018	14 July 2018	-156
14 July 2018	26 July 2018	97
26 July 2018	7 Aug 2018	-83
7 Aug 2018	19 Aug 2018	171
19 Aug 2018	31 Aug 2018	-141
31 Aug 2018	24 Sept 2018	-7
24 Sept 2018	6 Oct 2018	142
6 Oct 2018	30 Oct 2018	43
30 Oct 2018	11 Nov 2018	-104
11 Nov 2018	23 Nov 2018	-43
23 Nov 2018	5 Dec 2018	27

Sentinel-2 optical RGB imagery was also processed to Level 2A [64]. For candidate selection (blob detection) a dataset of blobs was generated by the blob detection algorithm. Each blob contains a subsetted image showing the blob's phase maps, coherence maps, terrain corrected displacement maps and Sentinel-2 optical imagery on 1st July 2018 and on 16th July 2018. For binary classification a dataset of 1440 blobs was labelled as either positive or negative by storing the respective five images per

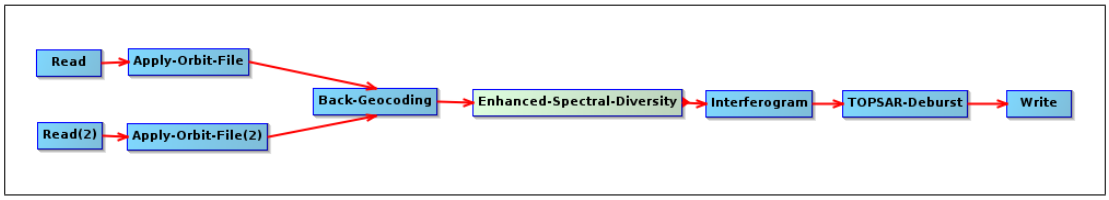

Figure 3.2: InSAR processing graph to develop interferograms

blob in folders with names indicating whether the blob is positive or negative. Time series for all the pixels in the Sentinel-1 image was also developed.

3.3.3 DInSAR Analysis

Figure 3.2 shows the InSAR processing chain used to produce interferograms after 2 $\times$ Single Look Complex (SLC) products have been split so as to only use bursts 3-7 in IW1 respectively. After the interferogram is produced debursting is done to obtain a smooth interferogram [21].

The DInSAR processing chain to produce differential interferograms which use the interferogram as an input is shown in Figure 3.3. The topographical phase removal step autodownloads a Digital Elevation Model (DEM) which is removed from the interferogram. Thereafter, 8 $\times$ 2 Multilooking is performed resulting in a spatial resolution of 28m $\times$ 34m. Then Goldstein Phase Filtering is performed [21] and a Statistical-Cost Network-Flow Algorithm for Phase Unwrapping (SNAPHU) [20] export is carried out [21].

The SNAPHU export is used to unwrap the phase using Terminal software and create an unwrapped phase header file [21]. Two-dimensional Phase unwrapping is a method to decipher valid phase signal from a phase image where the values of phase are known only modulo 2π radians. SNAPHU handles phase unwrapping as a Maximum A Posteriori (MAP) probability estimation task. The aim thereof considering the observable inputs is for the calculaton of the unwrapped solution which is most likely. Due to the fact that the statistics which govern the relationship between the input data and the unwrapped solution are reliant on the quantity which is measured SNAPHU has three statistical models within it which are for topography, smooth generic and deformation data. Network-flow methods are used to approximate a solution to the optimization task. Complete unwrapped solutions are said to be always developed by SNAPHU [20].

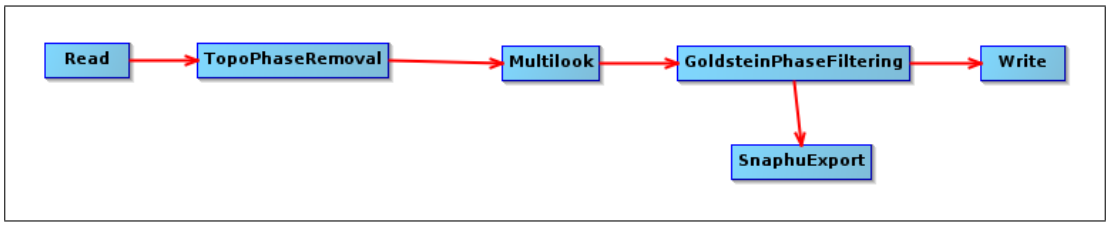

Figure 3.3: DInSAR processing graph to develop differential interferograms

An example of a differential interferogram (27th May - 2nd July) is illustrated in Figure 3.4. Figure 3.5 shows the processing chain to produce displacement maps which uses the differential interferogram and unwrapped phase header file as inputs. The output is the displacement map [21]. Thereafter, terrain correction is applied to generate a terrain corrected displacement map as shown in Figure 3.6.

Figure 3.7 shows an example of blobs of uplift and subsidence in Pretoria due to construction at

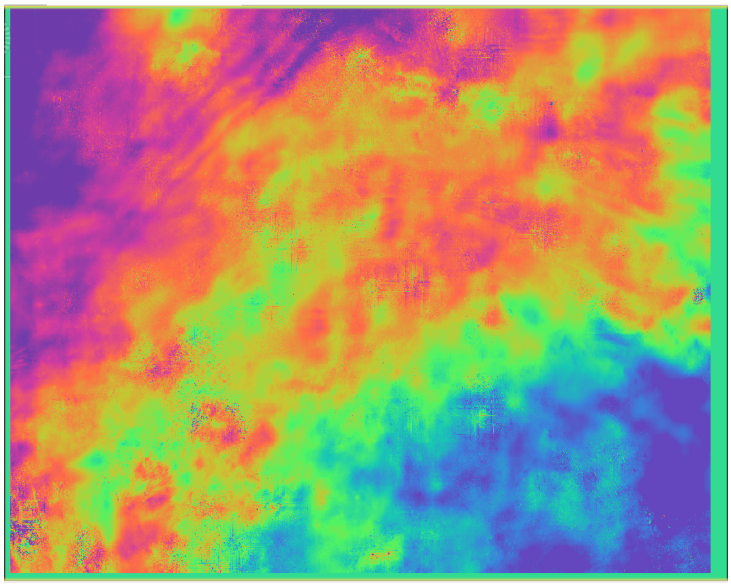

Figure 3.4: 27 May-2 July differential interferogram showing phase

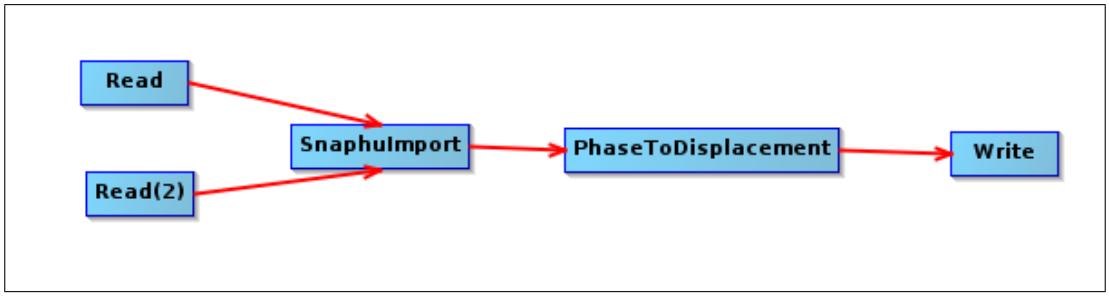

Figure 3.5: Displacement graph to develop displacement maps

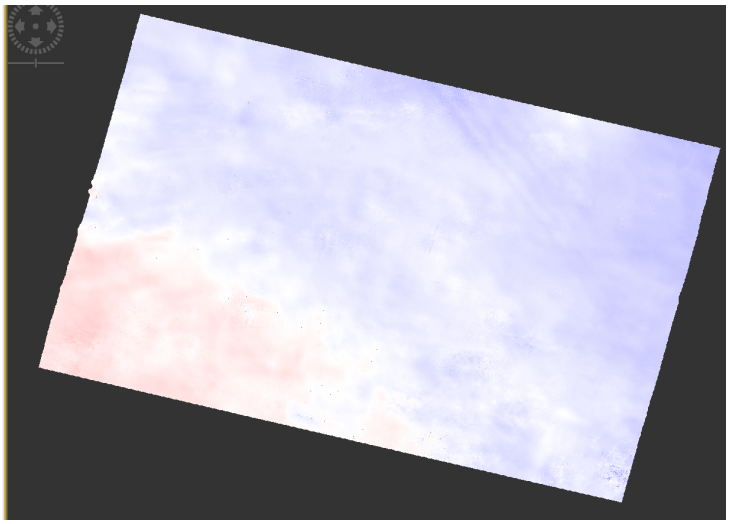

Figure 3.6: Terrain Corrected Displacement Map 27 May - 2 July, showing uplift (blue) and subsidence (red)

the locations -25.8301, 28.3275 and -25.8324, 28.3225 respectively. Distinct blobs are seen referring to uplift (blue) and subsidence (red). Figure 3.8 shows the Google Maps image of the first location which shows that it is indeed a construction site. Figure 3.9 shows the Google Maps image of the second location which shows that it is also indeed a construction site.

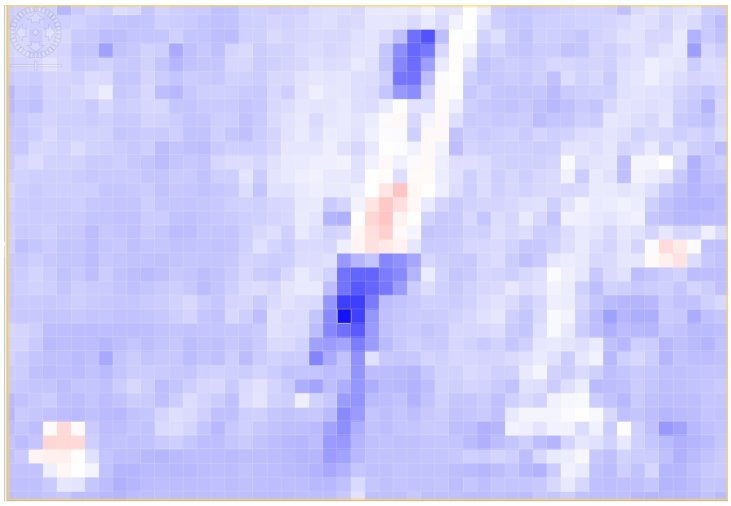

Figure 3.7: Uplift (blue) and subsidence (red) blobs in Pretoria

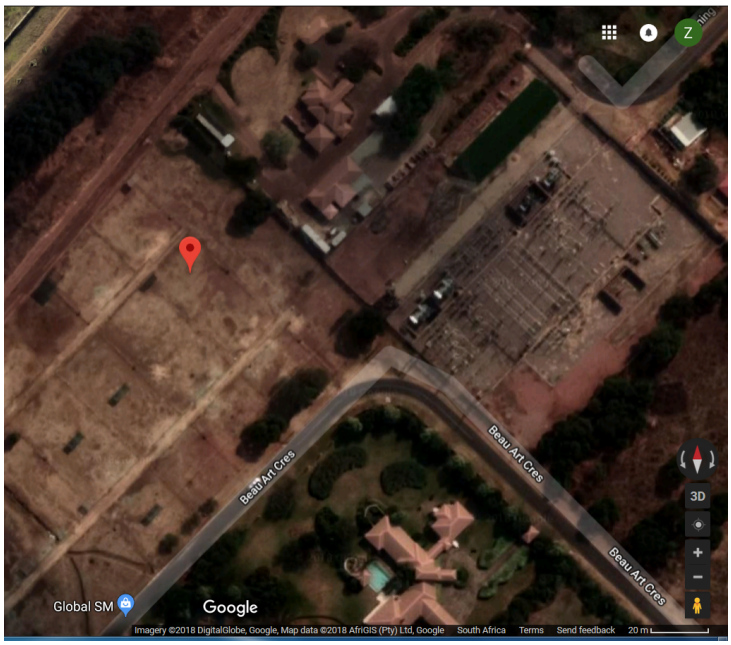

Figure 3.8: Google Maps image showing construction site 1

3.3.4 Candidate Selection

For Candidate Selection the most accurate Blob Detection algorithm is used i.e. the Laplacian of Gaussian (LoG) algorithm [60]. The 2nd July to 14th July 2018 Terrain Corrected Displacement Map that had a perpendicular baseline of 156 which is within the optimal range [9] is used to perform blob detection. This is post-reprojection to the Coordinate Reference System (CRS) of the optical images from



Figure 3.9: Google Maps image showing construction site 2

Sentinel-2. The optical images were then processed to a Level 2A product i.e. atmospherically corrected to the Bottom of the Atmosphere [64]. In order to subset the Terrain Corrected Displacement Map to the blob the Geospatial Data Abstraction Library (GDAL) [66] was used. A quantity of pixels that are equal to half of the blob radius is used to zoom out from the blobs. The optical images from Sentinel-2 on 1st July and 16th July 2018 are first normalised. Thereafter they are also similarly subsetted. Both the Coherence and Phase images are processed in a similar manner to how the Displacement Map was processed with respect to Terrain Correction and then reprojection to the Sentinel-2 CRS. This is also so that the coherence and phase imagery can be similarly subsetted using the coordinates of the respective blobs as well as GDAL [66]. Thereafter, all of the *.TIF* files which have undergone being subsetted are all resized to the resolution of the blob with the max. blob radius and are also multiplied by a factor of 3.4 in order to avoid resolution loss in the optical images from Sentinel-2 that is at a resolution of 10m. The reason behind this is that the Displacement map has a 28×34 m resolution as a result of multilooking (avg. pooling) of 8×2 having been applied in an attempt to reduce the noise contained in the interferogram. This results in square images at a resolution of 570×570 being produced. Before the different DL models and methods are applied these images are again resized to the required image size of the respective model. The labelled dataset contains 720 positive blobs and 720 negative blobs [8].

3.3.5 Binary Classification

A balanced dataset of 1440 blobs in total is developed using the Blob Detection algorithm. The dataset contains subsetted displacement, coherence and phase maps as well as two optical images from Sentinel-2 satellite for each of the blobs in the dataset. However, only one RGB image is used to avoid biasing the model towards the Sentinel-2 optical imagery. Thus, each of the image types contributed as equally as possible when combining an RGB image which is actually 3 bands with other bands of imagery that are single bands such as the displacement map, phase and coherence maps. Thus, there are effectively 2×3 band images for each blob [8], a RGB image and the three bands of the DINSAR processed Sentinel-1 displacement, coherence and phase bands. A visualisation tool was developed which allowed one to view each of the images for a blob alongwith the HSV (Hue, Saturation and Value) versions of the Sentinel 2 RGB images after the blob subsetted dataset has been generated. The coordinates of the blob being queried can also be viewed by just typing in the *blob number*. A *CSV* file was also automatically generated containing the radius, coordinates and blob number of each blob in the dataset. This facilitated the process of manually labelling the binary classification dataset.

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

To measure the performance of binary classification the following metrics are used in addition to accuracy. Precision (also called Positive Predictive Value) is given by Equation 3.1, Recall (also called True Positive Rate or Sensitivity) is given by Equation 3.2 and F1 score which is the harmonic mean of Precision and Recall is given by Equations 3.3 and 3.4, where TP, FN and FP refer to the amount of True Positives, False Negatives and False Positives respectively [67]. A lower value of Precision can infer that there is a significant quantity of False Positives. Similarly, a lower value of Recall can infer that there is a significant quantity of False Negatives. In medical data a False Negative may potentially be worse than a False Positive since if a certain medical condition was being tested for if it was not picked up the person may die from it. In that case a higher Recall may be desired. However, in general an F1 score is a good metric that indicates the harmony between these metrics.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.3}$$

$$F1\ score = \frac{TP}{TP + \frac{(FP+FN)}{2}} \tag{3.4}$$

Traditional ML approaches and a simple CNN are applied as a baseline to gauge the improvement obtained, if any, when using Deep Learning (DL) and Transfer Learning (TL) models.

Traditional ML: HOG and LBP with SVM

The Histogram of Oriented Gradients (HOG) [23] and Local Binary Pattern (LBP) [23] feature descriptors are used with a Linear Support Vector Machine (Linear SVM) model [24]. Each of the feature descriptors are used on their own with Linear SVM as well as the combination of both HOG and LBP with Linear SVM. For the HOG feature descriptor the images are resized to an image size of 112 $\times$ 224 to obtain the required 1:2 image aspect ratio. The HOG feature descriptor is implemented using 9 orientations, an 8 $\times$ 8 pixel cell window and 2 $\times$ 2 cells per block. The HOG features are normalised. Figure 3.10 and 3.11 illustrate an example of the images in the dataset with their corresponding HOG and LBP images respectively for a positive blob.

The Local Binary Pattern is implemented using images which are the same size as required for ResNet i.e. 224 $\times$ 224. A radius of three and a number of points equal to eight $\times$ the radius (twenty four points) are used to obtain the uniform LBP which is also two-dimensional. Thereafter, a histogram of the LBP image is taken which is then normalised. Both the HOG and LBP feature descriptors are applied to the RGB image, displacement, phase and coherence maps band images separately. Thereafter, the features are concatenated into a single vector for each descriptor respectively. In all three models ten-fold cross validation is used with the Linear SVM's *C* hyperparameter (the regularisation inverse) tuned with values in the set [0.001, 0.01, 0.1, 1.0, 10.0, 100.0].

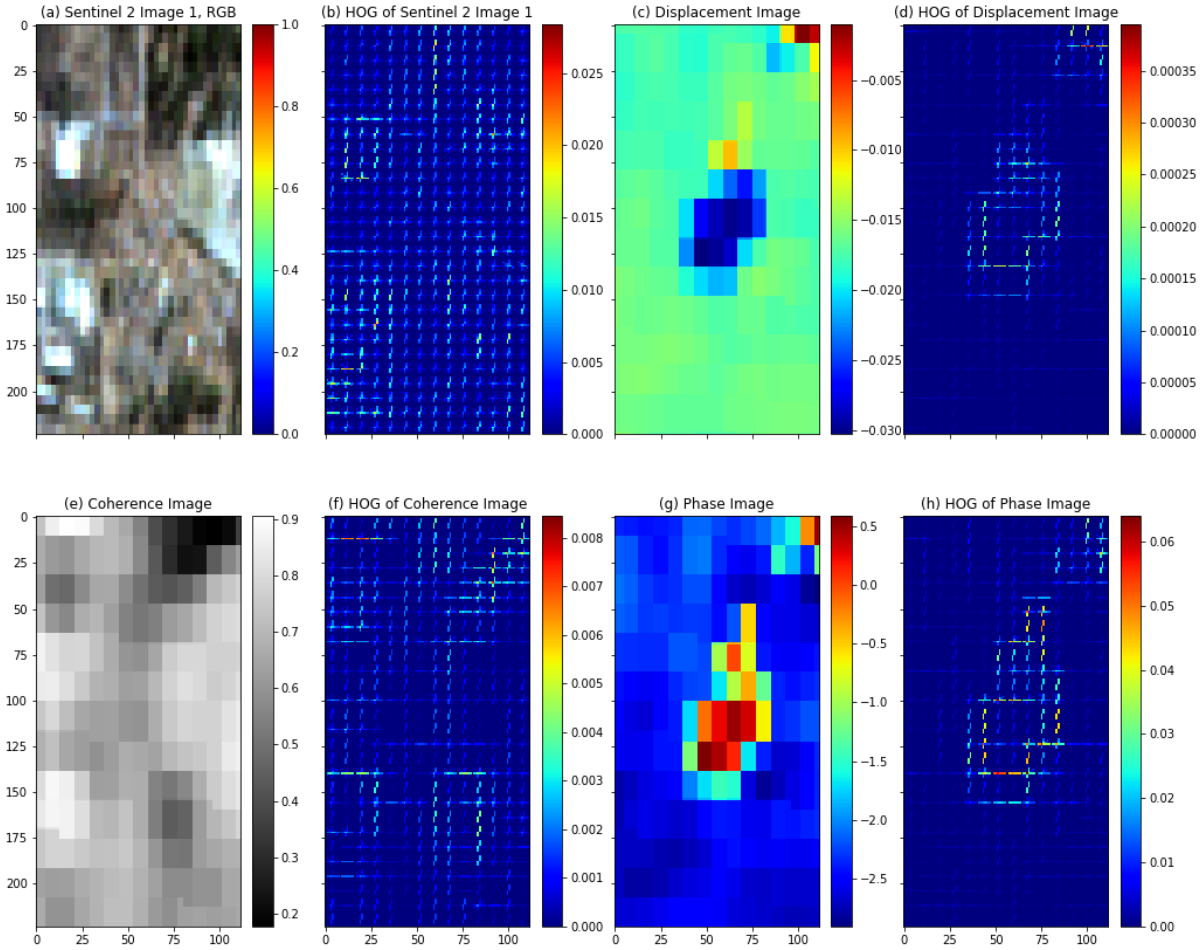

Figure 3.10: (a) Sentinel-2 optical RGB Image 1, (b) HOG of Sentinel-2 optical RGB Image 1, (c) Displacement Map, (d) HOG of Displacement Map, (e) Coherence Map, (f) HOG of Coherence Map, (g) Phase Map and (h) HOG of Phase Map.

Simple Convolutional Neural Network (Simple ConvNet)

Figure 3.12 illustrates the architecture of the simple CNN (Simple ConvNet) which consists of two networks in parallel, one for the RGB images and one for the corresponding displacement, phase and coherence maps bands. Each CNN uses 2 convolutional layers with Batch Normalisation [1] and Maximum Pooling [1] like in the ResNet convolutional blocks. The convolutional layers use 3×3 filters. The first convolutional layer has 32 filters and the second layer has 64 filters. Maximum pooling is carried out using a 2×2 filter kernel. Thereafter, there is a Global Average Pooling layer [68] followed by a Dropout layer [30] to improve generalisation and avoid overfitting. The Dropout hyperparameter is tuned with values in the set [0.2, 0.3, 0.4]. The output from the Softmax layers are averaged to form the final output of the Simple ConvNet.

TL using Feature Extraction

TL using Feature Extraction is used with a variety of architectures that have weights pretrained on ImageNet for the generation of feature maps after the final layer responsible for predictions has been

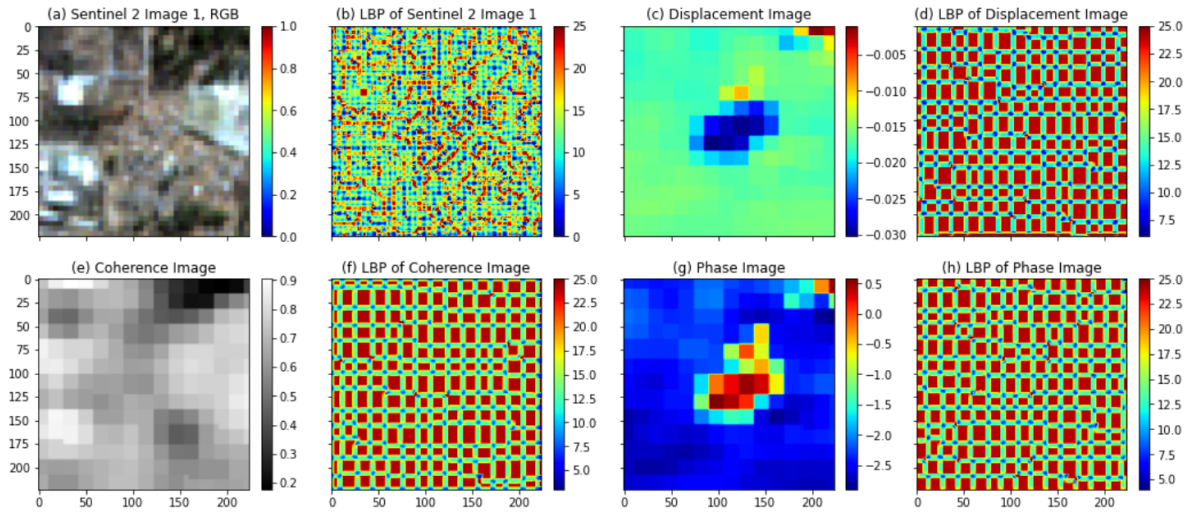

Figure 3.11: (a) Sentinel-2 optical RGB Image 1, (b) LBP of Sentinel-2 optical RGB Image 1, (c) Displacement Map, (d) LBP of Displacement Map, (e) Coherence Map, (f) LBP of Coherence Map, (g) Phase Map and (h) LBP of Phase Map.

removed. Thereafter, a simple Logistic Regression classifier is used on the generated feature maps. Ten-fold cross validation is used with the C parameter which is the regularisation inverse, hypertuned in the region of best performance with the values in the set [0.001, 0.0005, 0.0001]. From the simple machine learning models that were evaluated using FE the Logistic Regression model performed better than the others. Thus, it was selected for use with FE [8].

Transfer Learning using Fine Tuning and Random Initialisation

Figure 3.13 [8] shows the last layer of the architecture developed by simply using two built-in ResNet50 models in the Keras library, one for the RGB image (three band image) and one for the three Sentinel-1 DInSAR processed bands of displacement, coherence and phase bands stacked together. Two models were actually required in Keras for Fine Tuning and classic DL i.e. Random Initialisation since the built-in models can only be applied to three band imagery and not six band imagery. Thus, the results of the two models are averaged in the last layer. Early Stopping, stopping training when the validation loss starts to continually increase over a defined number of epochs (four) inferring that overfitting to the training data is beginning, and Dropout were used to prevent overfitting [8]. Data Augmentation was also done using image data generators where random images in the dataset were sheared, zoomed, width shifted, height shifted, rotated or horizontally flipped [1]. The models have the Dropout hyperparameter tuned in the region of best performance with the values in the set [0.2, 0.3, 0.4] with 10-fold cross validation. Training is done for 50 epochs. In Fine Tuning first all the layers are frozen and only the predictive layer is trained for 50 epochs (with Early Stopping at a patience of four epochs). Then just the last convolutional layer is unfrozen and also trained with the final predictive layer for 50 epochs (also with Early Stopping at a patience of four epochs). The SGD optimiser with a very slow learning rate of 0.0001 and momentum of 0.9 is used for both Random Initialisation and Fine Tuning.

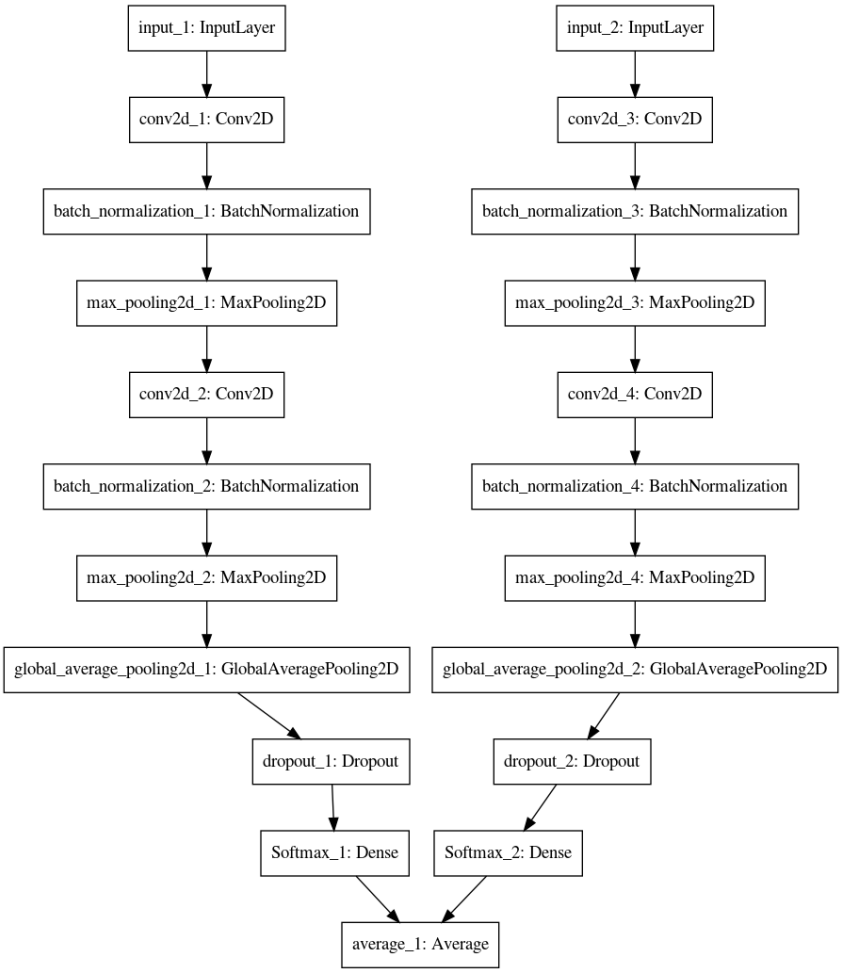

Figure 3.12: Simple Convolutional Neural Network (Simple ConvNet) architecture.

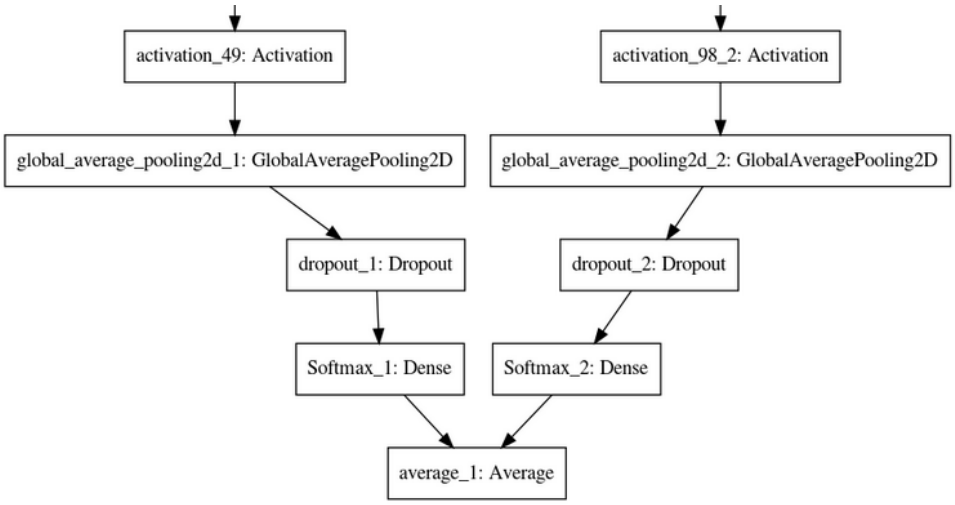

Figure 3.13: ResNet50 Fine Tuning final layer (reproduced with permission, ©, 2020, IEEE [8]).

Ensemble Models

Ensembles containing a few architectures were also evaluated with TL using Feature Extraction and with the Logistic Regression model applied to classify the features. The average performances obtained with

10-fold cross validation was measured [8].

Preprocessing of images

For all of the models except for the traditional ML models which use feature descriptors the coherence, phase and displacement map bands are scaled to the RGB range from 0 to 255. The ResNet50 architecture uses mean image subtraction image preprocessing. The best performance can only be achieved if the same preprocessing and image sizes that the pretrained model used is used when doing TL [42, 8]. For the Logistic Regression models (used in TL FE) and Support Vector Machine (traditional ML) models the features are scaled to unit variance and zero mean. This is to cater for features which may have a larger scale or variance than the rest of the features which will skew the model. Caution needs to be exercised when selecting scaler parameters. If there is a parameter to store the result in-place then although this can save memory and reduce computation time it can lead to data corruption occurring when running the model for the rest of the cross-validation folds.

3.3.6 Time Series Analysis

The Terrain Corrected Displacement Maps were used to develop a 28 data point time series for all the pixels in the Sentinel-1 Displacement map by stacking the displacement maps together using the GDAL library [66]. The velocity is given by the total displacement over a year divided by the time duration i.e. 1 year, resulting in velocity in m/year dimensions.

3.4 Limitations

Resolution of the multilooked differential interferogram if using 8 $\times$ 2 multilooking is at best 28m $\times$ 34m in ground range. So any uplift or subsidence smaller than this will not be picked up.

3.5 Conclusion

A Sentinel-1B SAR IW SLC VV image data stack over the period 28 Nov. 2017 to 5 Dec. 2018 is used for the research with the images 12 days apart where available. DInSAR is used to generate interferograms and terrain corrected displacement maps. For selected pixels where construction is known to have occurred time-series is developed. A blob detection algorithm is used to detect blobs in the 2 June to 14 June 2018 displacement map. The algorithm performance is quantified. The research methodology describing how the interferograms and displacement maps are generated is discussed as well as Candidate Selection. The Binary Classification algorithms are also discussed which use various traditional Machine Learning models, Simple ConvNet, Deep Learning architectures, Transfer Learning methods and ensembles of these models. The development of Time Series for each pixel as well as the limitations of the research are discussed.

Chapter 4

Results

4.1 Introduction

This chapter discusses the results achieved. First, Candidate Selection results are discussed. Thereafter, the results obtained from the various Binary Classification algorithms are discussed which include traditional Machine Learning models using feature descriptors, a simple CNN, Deep Learning architectures and Transfer Learning methods using both Feature Extraction and Fine Tuning. Some ensembles of models were also developed. Lastly the time series developed is discussed. Thereafter, the research questions are analysed.

4.2 Candidate Selection

The blob detection algorithm performance was measured by varying the parameter settings. The algorithm could only be used on the 2nd to 14th July Displacement Map after reprojection to the optical images from Sentinel-2 Coordinate Reference System (CRS). This is done to subset the Sentinel-2 optical RGB imagery with coordinates corresponding to the respective Displacement Map pixels. The 2nd to 14th July displacement map has an optimal perpendicular baseline and the images were acquired in winter when vegetation which detrimentally affects coherence is reduced. The modulus of the normalised Displacement Map image was used as the input to the blob detection algorithm. The same method of subsetting was applied to the coherence and phase imagery. [8].

Table 4.1: Blob Detection Algorithm results at threshold of 1

min sigma	Blobs	Mean Radius	Min Radius	Max Radius	True positive	False positive
10	120	15	14	56	3	117
20	30	30	28	53	1	29
30	14	44	42	55	0	14

The *min sigma* parameter (minimum standard deviation for the Gaussian kernel) of the Laplacian of Gaussian algorithm was varied from 10 to 30 whilst keeping the *max sigma* (maximum standard deviation for the Gaussian kernel) and *threshold* (the absolute lower bound for scale space maxima) parameters equal to 40 and 1 respectively. As the *min sigma* parameter was increased less blobs were detected and the radius of the blobs detected also increased in size as expected [60]. Furthermore, the

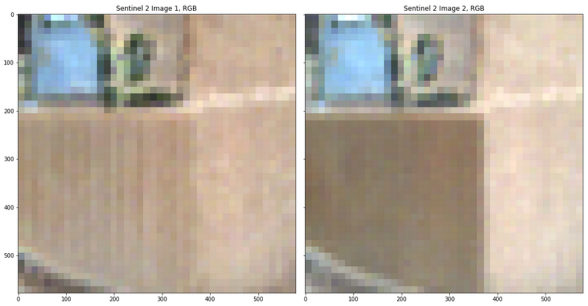

Figure 4.1: Sentinel-2 optical RGB image showing a blob where change in crops is detected, 2 July (left) and 14 July (right), reproduced with permission, ©, 2020, IEEE [8].

number of false positives also increased. In fact at a *min sigma* of 10 only 3 blobs were true positives and 117 were false positives. At a *min sigma* of 20 only 1 blob was a true positive and 29 were false positives. At a *min sigma* of 30 no blobs were true positive and 14 were false positives. The Blob Detection algorithm performance results are tabulated in Table 4.1. The reason there are more false positives than true positives is due to the underlying distribution of blobs. There are more false positive blobs in the displacement map than true positives. The blob detection algorithm is sensitive to changes in vegetation which is more widespread than changes that correspond to true positives. This is since the coherence is detrimentally affected by vegetation which causes a false positive blob to appear in the image and there are more vegetated areas in the study area. The resolution of the imagery also contributes to the high false positive rate since only changes that are at least 28m $\times$ 34m can be picked up. This is why Sentinel-1 is typically used for large area monitoring for earthquakes, etc. The decision boundary for this bnary classification task is often quite blurred and is even difficult for a human to deduce the correct classification.

In order to generate enough blobs to develop a sufficiently sized binary classification dataset the threshold parameter was reduced to 0.01. This resulted in 3742 blobs being detected at a *min sigma* of 10 and *max sigma* of 40. A dataset of 1440 blobs were labelled. Imagery from Google Earth was also used for certain cases when just the Sentinel-2 optical RGB imagery could not be used to decide as to whether or not the respective blob is indeed a true positive blob or actually a false positive blob.

In Figure 4.1 [8] a blob detected by the algorithm illustrates a change that has been detected on farm land (due to crops). Both true positive blobs and false positive blobs have been detected by the algorithm. True positives which correspond to change were detected at mines, quarries, construction developments, highways, factories and parking lots in complexes, etc. False positive blobs which correspond to noise were detected as a result of trees, crops, grass, etc. False positives due to ponds, lakes, water collecting at the top of mine dumps mixing with the mine dump contents making a green liquid, etc., were also detected [8]. Whilst the algorithm sometimes detected many blobs in one area like a mine for example the blobs were usually sufficiently far apart enough to contain a different scene within the image.

False positive blobs

In Figure 4.2 [8] a false positive is illustrated which is caused by crops causing noise. The optical image from Sentinel-2, Displacement Map, Coherence Map and Phase Map images are shown. Out of the full set of blobs detected a large quantity detected by the algorithm are actually false positives such as this example due to farm land, and other land with vegetation contained in the images. This is due to the fact that the coherence is extremely low in vegetated regions and at water bodies such as lakes, rivers, dams, etc. Hence, the need for this classifier. A difficult false positive scenario is illustrated in Figure 4.3 [8]

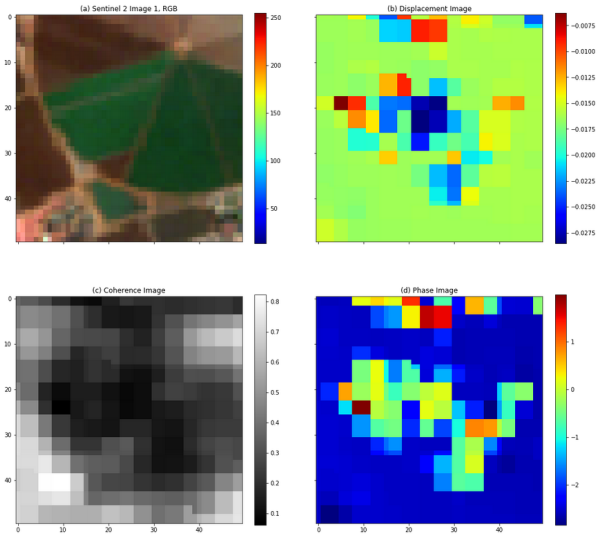

Figure 4.2: Example of a blob that is false positive caused by crops on a farm: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]).

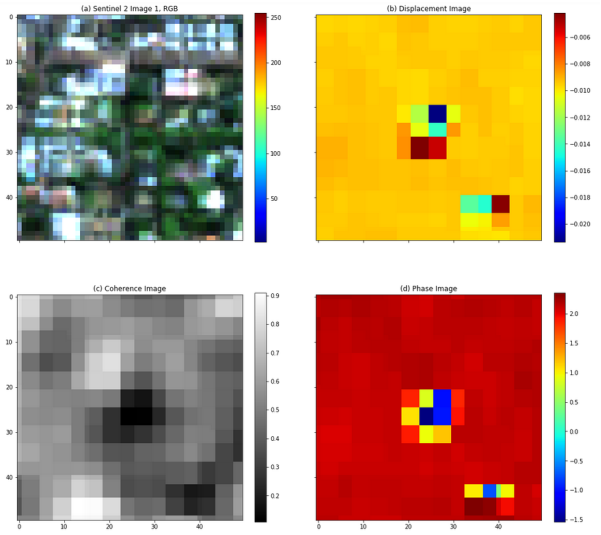

Figure 4.3: Example of a blob that is a difficult false positive: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]).

wherein bad coherence is caused by the trees which result in the errorneous detection of a blob. Between the true positive blobs and false positive blobs the decision boundary is not always clearly defined [8].

True positive blobs

At a mine in Cullinan, Tshwane an ideal true positive case is detected at $(-25.6577269, 28.5055817)$ as a result of mining activity. This is illustrated in figure 4.4 [8]. At Klipfintein Quarry a true positive blob is also detected due to activity at the quarry, the blob is shown in Figure 4.5 [8]. This blob also illustrates the principle that a true positive blob is also capable of displaying coherence that is bad and looks like noise and its not just a characteristic of false positive blobs [8]. Construction is also detected

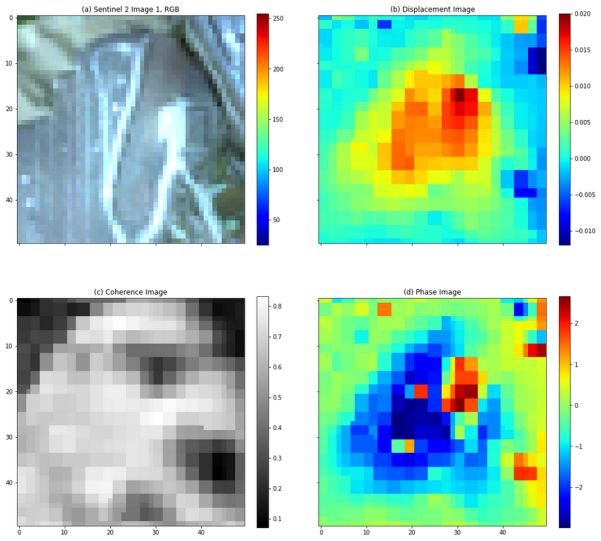

Figure 4.4: Example of a blob that is an ideal true positive: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]).

at Riverglen $(-25.97339905, 28.0122530)$ in figure 4.6 [8]. The corresponding Google Earth image is shown in Figure 4.7 [8].

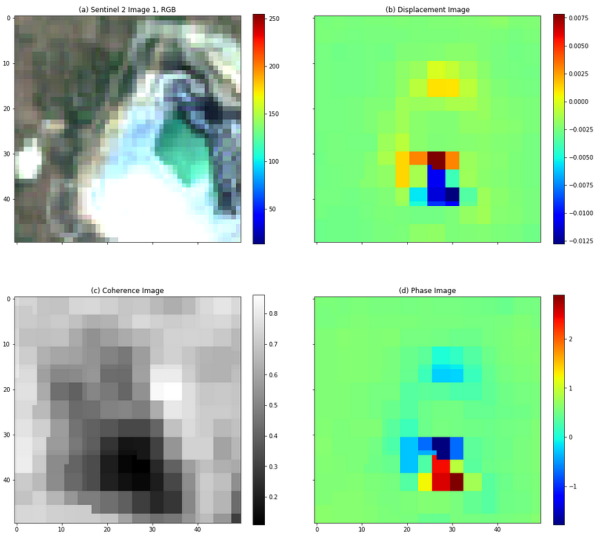

Figure 4.5: Example of a blob that is a true positive but with bad coherence: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps (reproduced with permission, ©, 2020, IEEE [8]).

Figure 4.8 shows a difficult positive case where construction is detected in a field. In Figure 4.9, the Google Earth image, trucks can be seen laying the foundation of what may be a shopping mall. A dataset of 1440 blobs in total was generated which has an equal quantity of both positive and negative blobs. The 720 positive blobs comprised of approx. 270 blobs where construction was detected, 141 blobs where mining was detected, 222 blobs where storage (freights, equipment, stock, rail carriages, planes, etc., stored outside) was detected, 9 blobs where roads were detected, 38 blobs where parking lots were detected and 40 blobs containing parts of urban areas so as to teach the classifier to differentiate

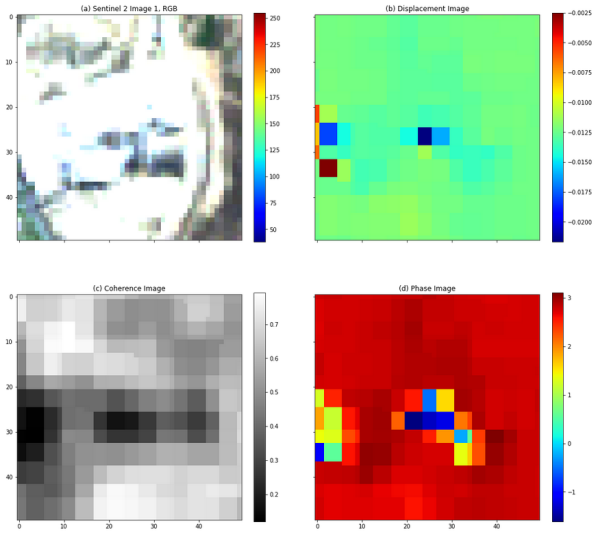

Figure 4.6: Example of a blob that is a true positive at a construction site: (a) optical image from Sentinel-2 (b) Displacement (c) Coherence and (d) Phase Maps.



Figure 4.7: True positive blob, Riverglen construction site, Google Earth image

between farmland and urban areas. There are estimated to be 650 unique sites contained in the positive data. In large areas such as mines the blob detection algorithm would detect part of the mine in one blob and another part of the mine in another blob.

4.3 Binary Classification

The Binary Classification results using the various techniques i.e. traditional Machine Learnng using Feature Extractors, a simple CNN, Transfer Learning via Feature Extraction (FE), Transfer Learning via Fine Tuning (FT) as well as using traditional DL i.e. Random Initialisation (RI) with various models and some ensembles of these models are discussed below. The average (μ) accuracy results obtained using 10-fold cross validation and a held out unseen test dataset are in Table 4.2 which also includes the variances (σ^2) of the accuracies and the number of parameters for the models considered [41, 45, 46, 44, 42]. The F1 score, Precision and Recall results are also included in Table 4.2. Small ensembles of some

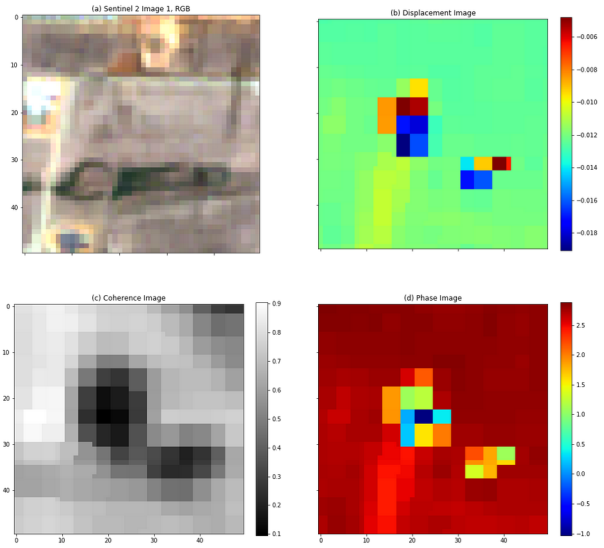

Figure 4.8: A difficult true positive blob: (a) Sentinel-2 optical RGB Image 1 (b) Displacement Map (c) Coherence Map and (d) Phase Map.



Figure 4.9: A difficult positive case, Google Earth image showing construction

of the models defined in Table 4.3 were also considered and their results are also included in Table 4.2.

4.3.1 Traditional ML: HOG and LBP with SVM

The traditional Machine Learning approaches using the HOG and LBP feature extractors both individually and their combination in conjunction with a Linear Support Vector Machine (SVM) model have performances of 60.63%, 61.15% and 67.95% respectively. The model with LBP and Linear SVM also had a smaller variance and larger F1 score than the other two traditional models. It is interesting to note that the model with uniform LBP and Linear SVM even outperformed the Deep Learning (Random Initialisation/training from scratch) and Transfer Learning via Fine Tuning with the EfficientNet B4 ar-

Table 4.2: Performance of the models and methods applied with corresponding no. of parameters

Model	Method	Acc. μ [%]	Acc. σ^2	F1 score	Precision	Recall	# params
ResNet50	**FE**	**84.341**	**0.949**	**0.843**	**0.843**	**0.843**	**26m**
Ensemble 1	FE	84.339	1.004	0.843	0.844	0.843	95m
Ensemble 2	FE	84.272	0.404	0.843	0.844	0.843	70m
Ensemble 3	FE	84.133	0.913	0.840	0.840	0.840	51m
ResNet101	FE	83.611	1.793	0.835	0.835	0.835	44m
Simple ConvNet	ML	83.368	1.646	0.835	0.836	0.835	<0.1m
Inception-ResnetV2	FE	82.811	1.748	0.828	0.828	0.828	56m
EfficientNet B0	FE	82.291	0.831	0.824	0.826	0.824	5m
ResNeXt50	FE	81.320	1.305	0.813	0.813	0.813	25m
ResNet50	RI	80.939	6.714	0.809	0.810	0.809	26m
EfficientNet B4	FE	80.590	0.920	0.806	0.807	0.806	19m
ResNet152	FE	80.069	2.743	0.799	0.801	0.800	60m
EfficientNet B7	FE	78.645	3.226	0.784	0.792	0.785	66m
ResNet50V2	FE	78.506	1.189	0.783	0.787	0.785	26m
ResNet152V2	FE	78.472	1.872	0.784	0.787	0.784	60m
ResNet50	FT	70.244	37.677	0.693	0.729	0.702	26m
LBP + SVM	ML	67.951	1.285	0.680	0.680	0.680	<0.1m
EfficientNet B4	FT	65.416	25.048	0.634	0.689	0.655	19m
EfficientNet B4	RI	64.410	55.479	0.593	0.747	0.643	19m
HOG + LBP + SVM	ML	61.146	3.791	0.612	0.612	0.612	<0.1m
HOG + SVM	ML	60.625	2.365	0.606	0.608	0.607	<0.1m

chitecture by 3.54% and 2.54% respectively. This is due to the small size of the dataset and the large amount of parameters that DL models have to train. Nevertheless most of the models using DL and TL outperformed the traditional ML models.

4.3.2 Simple Convolutional Neural Network (Simple ConvNet)

The Simple ConvNet with only two convolutional layers achieved an impressive performance of 83.37% outperforming most of the models except for TL via FE with ResNet101, ResNet50 and the ensemble models. The Simple ConvNet was only 0.97% behind the best performing model (TL via Fe with ResNet50). This is also attributable to the small size of the dataset and the large amount of parameters required to train a DL model. It is imperative to note that the Simple ConvNet was specially trained on this particular dataset which included band images of displacement, phase and coherence which are different from the RGB images contained in ImageNet. In fact even the Sentinel-1 optical RGB image is vastly different to the images contained in the ImageNet dataset. Nevertheless, TL with the ResNet50, ResNet101 and the ensemble models still had better performance compared to the baseline. Furthermore, Simple ConvNet also makes use of Batch Normalisation, Dropout and Global Average Pooling which improves its regularisation and its generalisability. So it is actually a modern simple CNN rather than a traditional basic CNN or ANN with fully connected layers and less regularisation resulting in severe overfitting.

Table 4.3: Ensemble definitions

Ensemble	Models	Method
Ensemble 1	ResNet50 + ResNeXt50 + ResNet101	FE
Ensemble 2	ResNet50 + ResNeXt50 + EfficientNetB4	FE
Ensemble 3	ResNet50 + ResNeXt50	FE

4.3.3 Transfer Learning (TL) via Feature Extraction (FE)

Ensemble 2 had the smallest accuracy variance with the 3rd highest accuracy (only 0.07% lower than that of the model with the highest accuracy) and the same F1 score as the best performing model. Feature Extraction using Microsoft's ResNet50 architecture published in Dec. 2015 [41] used for the generation of feature maps which then have the Logistic Regression classifier applied to them results in the best accuracy performance i.e. 84.34%. This is 3% better than Feature Extraction with the newer ResNeXt50 architecture, 0.73% better than Feature Extraction with the ResNet101 which has more than double the hidden layers than ResNet50 has and is a much deeper network (deeper networks have more parameters and require more training data) and 3.75% better than Google's EfficientNet B4 architecture published in 2019 [46]. FE with ResNet50 also vastly outperformed FE with InceptionResNetV2, EfficientNet B0, ResNet152, EfficientNet B7, ResNet50V2 and ResNet152V2. TL via FE with the ResNet50 also outperformed the traditional ML models using the LBP, HOG and their combination feature extractors with Linear SVM by 16.39%, 23.20% and 23.72% respectively. However, it only outperformed Simple ConvNet by 0.97%.

TL via FE with the ResNet50, Ensemble 1 and Ensemble 2 all have the same F1 score of 0.843 closely followed by Ensemble 3 with an F1 score of 0.840. Thereafter followed by ResNet101 with an F1 score of 0.835. TL via FE with the ResNet50 also performed with an accuracy of 4.27% better than the ResNet152 architecture. So the TL via FE accuracy decreases with an increase in depth of the ResNetV1 architecture. TL via FE with ensembles came in 2nd, 3rd and 4th place considering average accuracy outperforming all of the models except for TL via FE with the ResNet50. This demonstrates the power of ensembling. The ensemble model with the best performance was Ensemble 1 which contained ResNet50, ResNeXt50 and ResNet101 and achieved an accuracy of 84.34%. The ensemble models do not outperform Transfer Learning using the Feature Extraction method with the ResNet50 which has an average accuracy of 84.34% as per Table 4.2. All the ensembles evaluated using FE have an accuracy of at least 84% and an F1 score of at least 0.84. TL via FE for the EfficientNet architecture performs best with the EfficientNet B0 model (82.29%). EfficientNet B0 performs 1.7% better than the EfficientNet B4 model (80.59%) despite EfficientNet B4 having more parameters. EfficientNet B0 has the smallest amount of parameters (5m) and yet achieves a decent performance (82.29%) which is better than EfficientNet B7 (78.65%) and ResNet152's performance (80.07%) despite these models having higher ImageNet accuracies and having $13\times$ and $12\times$ more parameters than EfficientNet B0 respectively.

4.3.4 Transfer Learning (TL) via Fine Tuning (FT) and Random Initialisation (RI)

Table 4.2 also contains the average accuracy results obtained using 10-fold cross validation on a held out unseen test dataset when using Transfer Learning via Fine Tuning and using Random Initialisation i.e. full training with the ResNet50 and EfficientNet B4 architectures. For the ResNet50 architecture RI accuracy was 10.7% higher than the Fine Tuning accuracy. ResNet50 Feature Extraction performed better than both of these methods by 3.4% and 14.1% respectively. This may be due to RI and TL via FT requiring two models that are then aggregated in the final layer unlike with TL via FE where

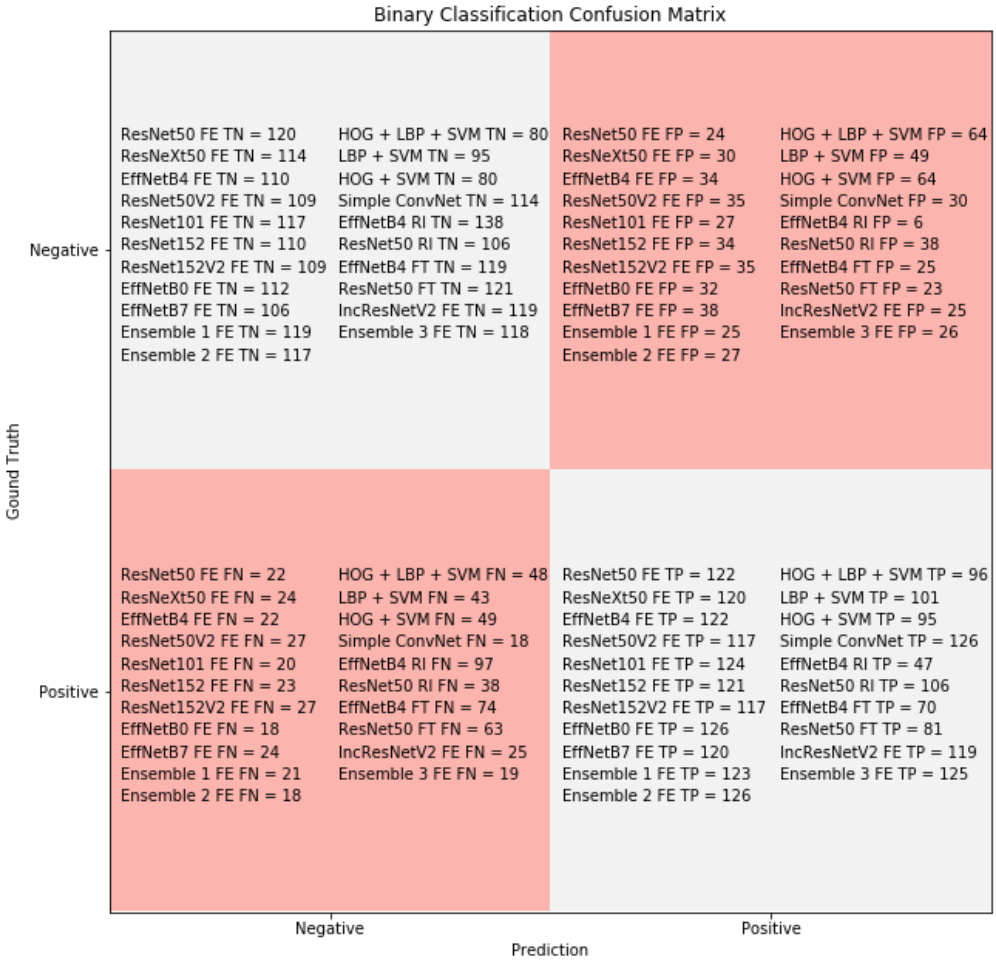

Figure 4.10: Confusion Matrix (EfficientNet abbreviated to EffNet)

no actual training of the network is done. EfficientNet B4 performed better when used for Feature Extraction (80.59%) than when used for training from scratch (64.41%) and using Fine Tuning (65.42%) respectively. Transfer Learning via Feature Extraction with many of the models performed better than Transfer Learning using Fine Tuning and Random Initialisation with ResNet50 and EfficientNet B4. The dataset size was 1440 blobs in total including positive and negative. A much bigger dataset may have resulted in Random Initialisation performing better than Transfer Learning. It is imperative to note that all of these results are particular to this novel dataset and the size of the dataset is a crucial factor in defining whether TL or traditional DL i.e. Random Initialisation performs better. The similarity of one's dataset with the dataset that one is attempting to transfer Learn from can also be a factor affecting performance.

Random Initialisation with the ResNet50 (80.94%) performed 16.5% better than with the EfficientNet B4 (64.41%) on this small dataset. Fine Tuning with a ResNet50 obtained a 5% higher accuracy than Fine Tuning with the EfficientNet B4. This is despite the EfficientNet architecture having been developed using Neural Architecture Search i.e. where Machine Learning was used to find the optimal architecture to optimise performance on the ImageNet RGB images dataset [46]. The smallest model, EfficientNet B0, outperforms both ResNet50 and EfficientNet B4 models with TL via FT and RI respectively despite the fact that ResNet50 and EfficientNet B4 models have better ImageNet accuracies and have $5\times$ and $4\times$ more parameters than EfficientNet B0 respectively.

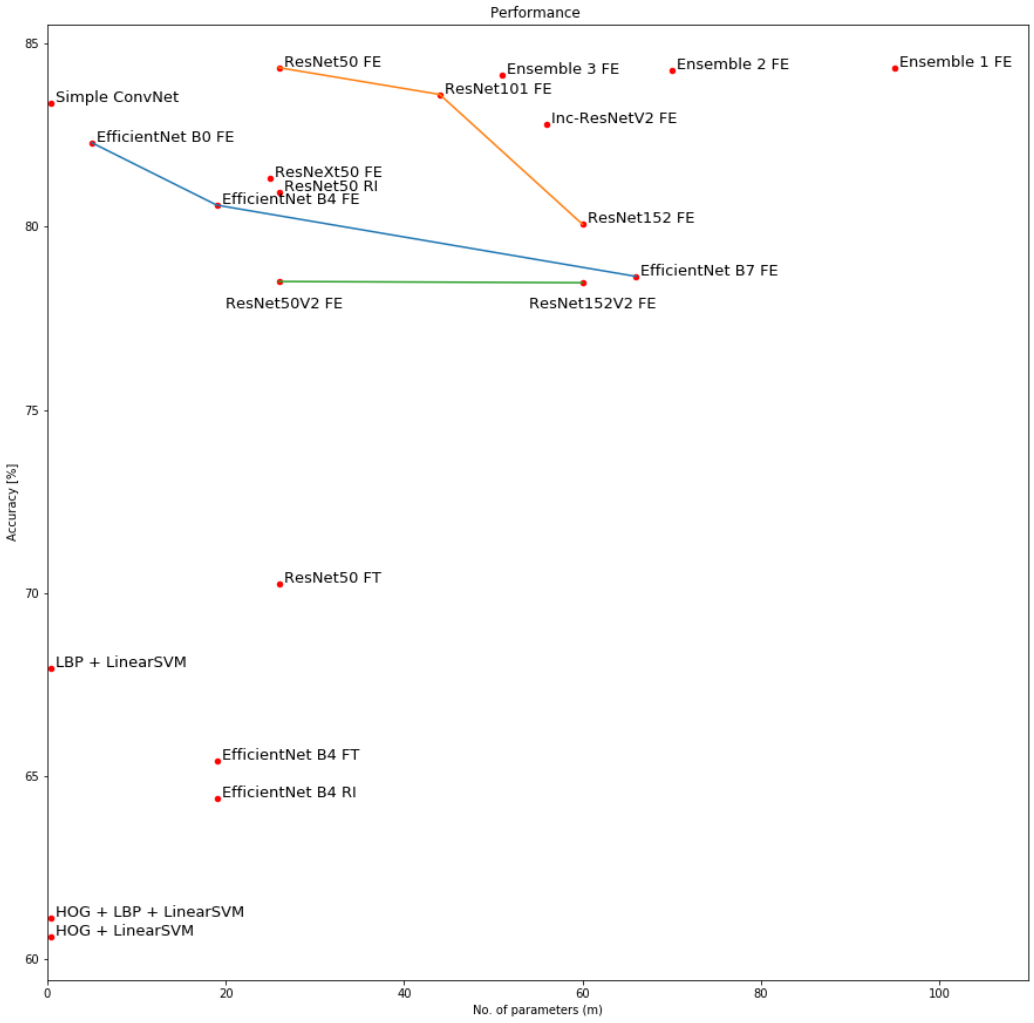

Figure 4.11: Performance of models vs no. of parameters showing trends of different architectures: ResNet (orange), EfficientNet (blue) and ResNet V2 (green).

Figure 4.10 illustrates the confusion matrix of the models evaluated. ResNet50 with TL via FE has 24 False Positives (FP) and 22 False Negatives (FN). Most of the models tend to have more FP than FN except for the InceptionResNetV2 FE, EfficientNetB4 RI, ResNet50 RI, EfficientNetB4 FT and ResNet50 FT. InceptionResNetV2 FE and ResNet50 RI each had equal numbers of FP and FN respectively. Figure 4.11 illustrates the average accuracy performance of the models versus the number of parameters. Models that have more than 30m parameters all had accuracies above 78%. So there does seem to be a non-linear correlation between model size and accuracy. However, this is due to Fine Tuning and Random Initialisation only having been evaluated on only two architectures: EfficientNet B4 (19m parameters) and ResNet50 (26m parameters). From Figure 4.10 it is evident that most models using Transfer Learning with Feature Extraction independent of the no. of parameters outperformed Random Initialisation and Transfer Learning with Fine Tuning with ResNet50 and EfficientNet B4 respectively on this dataset. The inverse correlation between accuracy and network size can be seen on a per architecture basis (when keeping architecture constant) in the ResNet V1 (orange graph) and EfficientNet (blue graph) architectures. This effect is less pronounced in the ResNetV2 architecture (green graph) where the accuracy is almost the same. It is also evident that when using the ensembles evaluated an increase in number of parameters results in a slight increase in average accuracy.

4.4 Time Series

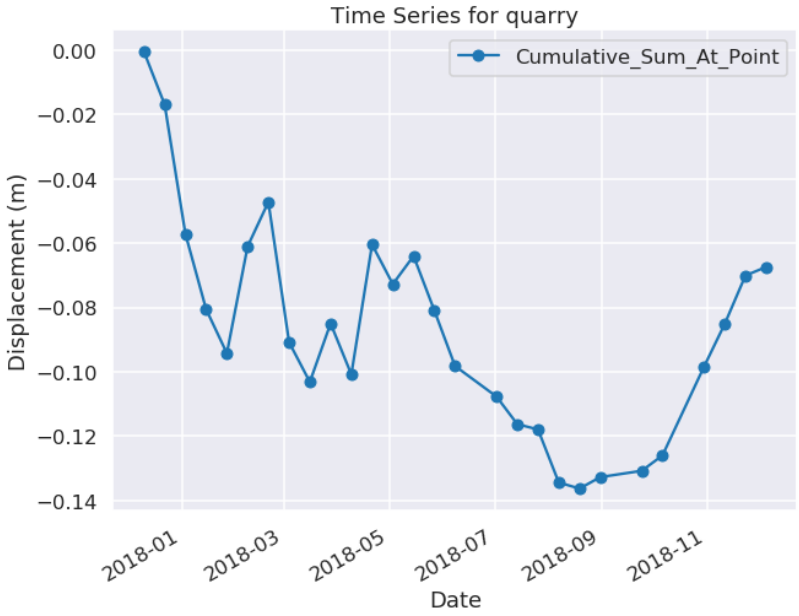

Figure 4.12: Displacement time series for quarry, showing a seemingly downward trend despite noise

The Terrain Corrected Displacement Maps were used to develop a 28 data point time series for all the pixels in the Sentinel-1 Displacement map by stacking the displacement maps. However, given the noise in the images due to some of the images not being in the optimum range of perpendicular baseline i.e. between 150 and 300 [9] an upward trend cannot be clearly seen in each section of the time series. Nevertheless, there is an upward trend when considering the cumulative sum for many pixels at the last data point which corresponds to the last interferogram.

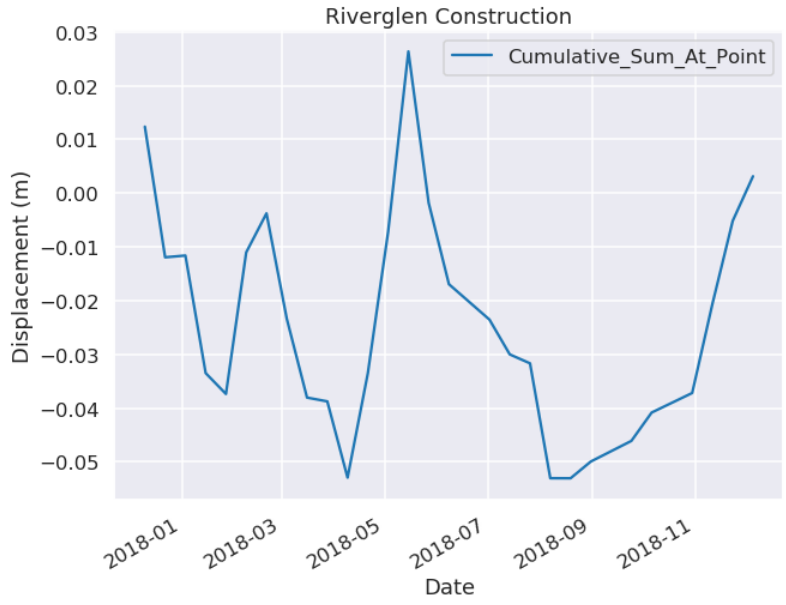

Figure 4.13: Displacement Time series for Riverglen construction, showing downward and upward trends

Figure 4.12 shows the time series developed for a quarry. There does seem to be a downward trend between July and September 2018 despite the noise in the signal. Noting that this could also be due to accumulation of material on site before it being moved out. Figure 4.13 shows the time series developed for construction in Riverglen. There does seem to be downward and upward trends in the signal. During construction first there is excavation to lay the foundation and this would infer subsidence and then uplift. So downward and upward trends can be expected.

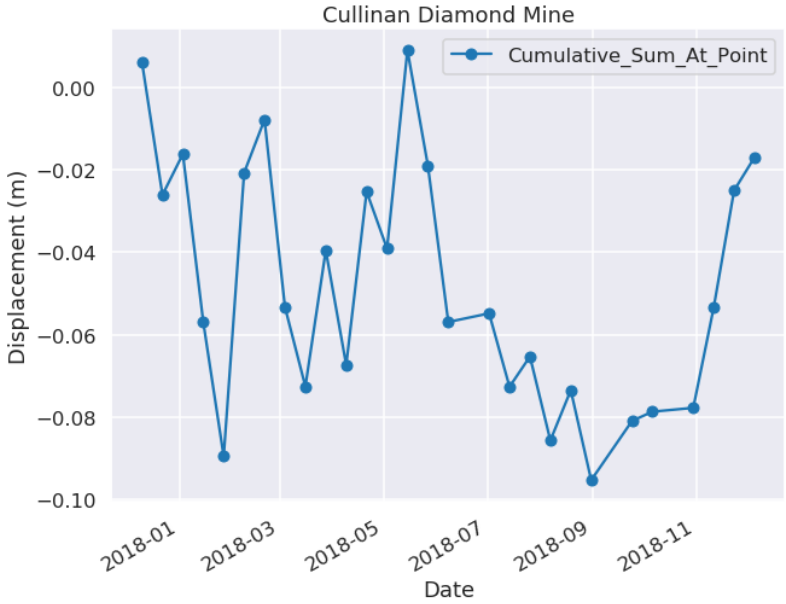

Figure 4.14: Displacement Time series for Cullinan Diamond Mine, showing downward and upward trends

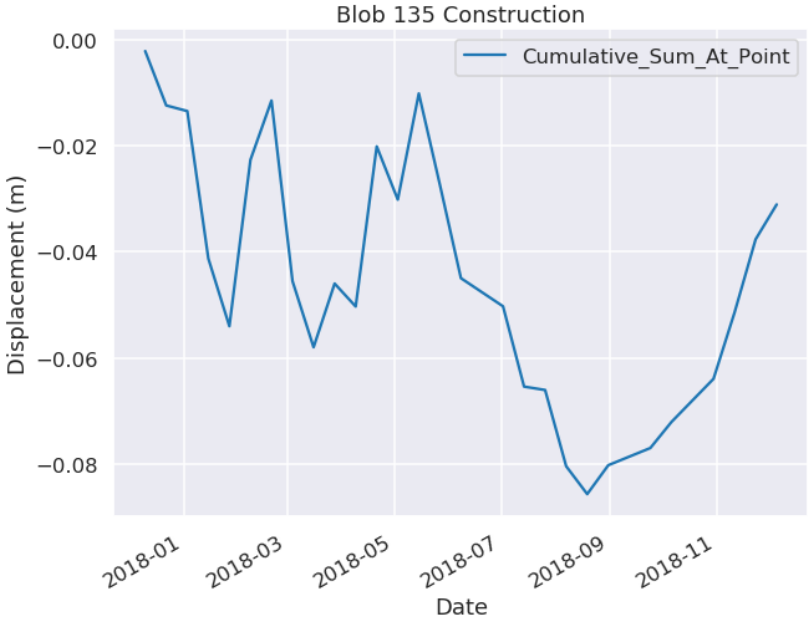

Figure 4.15: Displacement Time series for construction in blob 135, showing downward and upward trends

Figure 4.14 shows the time series developed for Cullinan Diamond mine. There does seem to be a downward trend between June and November 2018 despite the noise in the signal. At a mine there is accumulation of material before it is moved out which could be the cause of the upward and then downward trends seen. Figure 4.15 shows the time series developed for another blob which contained construction. There does seem to be downward and upward trends in the signal which are expected during construction.

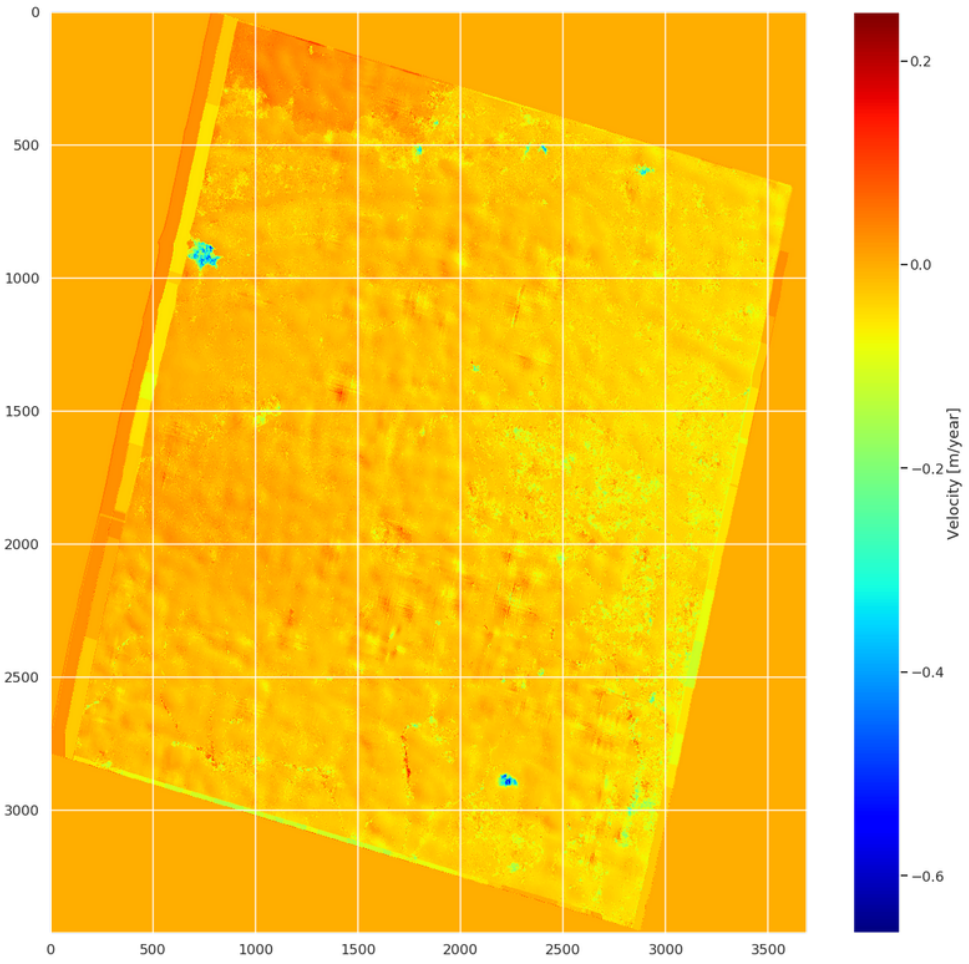

Figure 4.16: Velocity map showing per pixel velocities, subsidence (blue) and uplift (red)

Figure 4.16 illustrates the velocity map developed. The velocity map infers that there was 20 cm uplift and 60 cm subsidence over the 1 year period under study. However, it must be noted that areas of bad coherence where the data cannot be trusted are not masked out. Figure 4.17 shows the Terrain Corrected Coherence and Differential Interferogram (wrapped phase) maps 2 July to 14 July 2018. The areas in Figure 4.16 at around (750, 800) at the top left and the area at the bottom right at around (2250, 2850) that are dark blue inferring upto 60 cm subsidence in a 1 year period actually had bad coherence with reference to the coherence map in Figure 4.17. Nevertheless, it could still have been a quarry since true positive blobs can also have bad coherence. Other methods such as the Small Baseline Subset technique (SBAS) should rather be considered for developing time series as they may be more accurate than the approach employed.

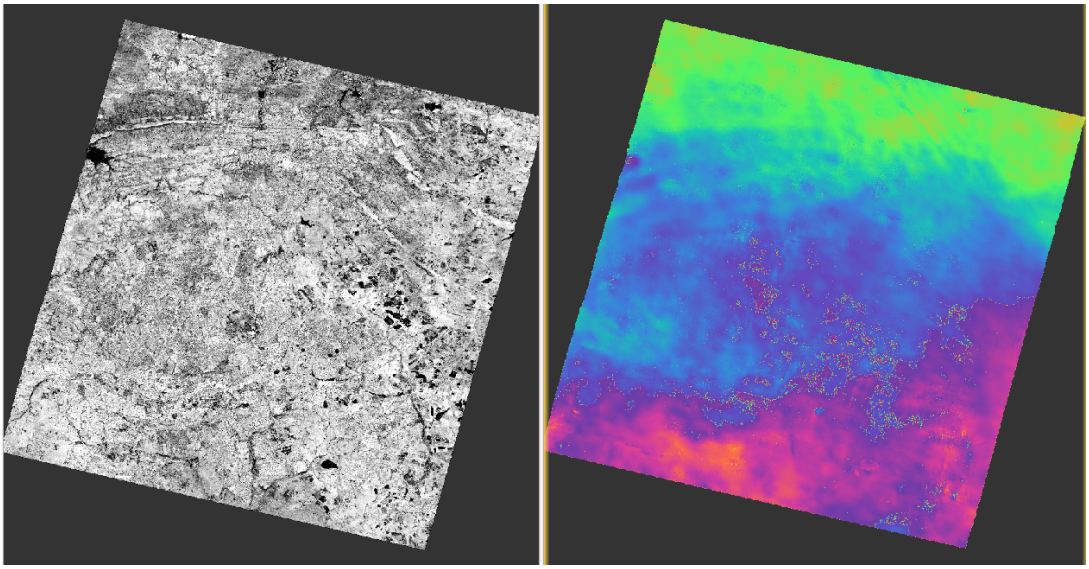

Figure 4.17: Terrain Corrected Coherence and phase Differential Interferogram 2 July to 14 July 2018

4.5 Research Questions

The following research questions can now be answered.

1. **Can Sentinel-1 imagery be used for industrial development change detection and can change at the selected pixels where construction is known to have occurred be qualitatively verified visually?**

   Yes, it picks up mining and the harvesting of crops. It also picks up when there is construction and the ground is excavated to lay the foundation for buildings. It also detects areas where roads are being developed in upcoming towns. The Blob Detection algorithm also detects blobs at highways and parking lots inferring that there were cars driving past or parked there respectively in the first Sentinel-1 image but in the second Sentinel-1 image those vehicles were not there. The blob infers that subsidence was detected. Its also possible that the converse is true i.e. there were no cars or trucks driving past or parked there respectively when the first image was taken but these vehicles were there when the second image was taken. The system detected uplift in this case. The same argument applies to industrial customers who store products or equipment outside.

   Change at the selected pixels where construction is known to have occurred is qualitatively verified. This is confirmed by satellite imagery of numerous construction sites where blobs were detected. These construction sites range from large construction like in Riverglen to pockets of smaller construction in the same area which are cumulatively detected to be a blob.

2. **What is the accuracy of the blob detection algorithm and is the change increasing at each point in the displacement time series?**

   The blob detection algorithm when evaluated at a *threshold* of 1, *max sigma* of 40 and *min sigma* of 10 is 2.5% accurate. This accuracy decreases with an increase in *min sigma* and increases with a decrease in *threshold* since decreasing the *threshold* parameter allows for more blobs to be detected. There is a high false positive rate due to the underlying distribution of false positive blobs being much higher than true positive blobs in the displacement map. This since there are many more areas of bad coherence which cause false positive blobs. Thus, the need for binary

classification.

The change is not clearly consistently increasing in each point in the time series due to the noise in the dataset due to many of the images not being in the optimal perpendicular baseline of 150-300 m and also due to atmospheric effects. However, in some of the time series that were developed both upward and downward trends can be seen for some of the data points.

3. **Can Transfer Learning using a pretrained model with weights trained on the ImageNet RGB images dataset be used to perform binary classification on the novel dataset containing DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced images with the corresponding Sentinel-2 optical satellite RGB imagery which contains an equal quantity of positive and negative blobs? Furthermore, which method and architecture has the best performance and is there any relationship between model size and performance?**

   Yes, Transfer Learning has been successfully implemented using a variety of Deep Learning architectures using both TL using Feature Extraction and TL using Fine Tuning methods. Thus, the benefits of Deep Learning accuracy has been transferred from a broad spectrum of different state-of-the-art Deep Learning architectures trained on the ImageNet RGB images dataset to binary classification on the novel dataset containing DInSAR processed Sentinel-1 displacement, coherence and phase georeferenced images with the corresponding Sentinel-2 RGB satellite imagery.

   The highest accuracy, 84.34%, is obtained by Transfer Learning using Feature Extraction with a ResNet50 version 1 architecture developed by Microsoft in 2015 [41] used for feature maps generation with a basic Logistic Regression classifier. Using the ResNet50 architecture classic Deep Learning i.e. Random Initialisation had a 10.7% higher accuracy than using the ResNet50 with Fine Tuning. Transfer Learning using Feature Extraction with the ResNet50 had an accuracy higher than these models by 3.4% and 14.1% respectively. This may be due to RI and TL via FT requiring two models that are then aggregated in the final layer unlike with TL via FE where no actual training of the network is done. EfficientNet B4 performed better when used for Feature Extraction (80.59%) than when used for training from scratch (64.41%) and using Fine Tuning (65.42%) respectively. Transfer Learning using Feature Extraction with many of the models performed better than TL with Fine Tuning and Random Initialisation with ResNet50 and EfficientNet B4 respectively since the size of the balanced binary classification dataset is 1440 blobs which is relatively small for Deep Learning. A much bigger dataset may have resulted in Random Initialisation performing better than Transfer Learning.

   TL via FE with the ResNet50 also outperformed the traditional ML models using the LBP, HOG and their combination feature extractors with Linear SVM by 16.39%, 23.20% and 23.72% respectively. However, it only outperformed Simple ConvNet (83.73%) which has two convolutional layers and uses Batch Normalisation, Dropout and Global Average Pooling by 0.97%. Most of the models using DL and TL outperformed the traditional ML models which use the HOG, LBP and their combination feature extractors with Linear SVM except for the EfficientNet B4 models for RI and TL via FT. The Simple ConvNet outperformed most of the models except for TL via FE with ResNet101, ResNet50 and the ensemble models. It is imperative to note that all of these results are particular to this novel dataset and the size of the dataset is a crucial factor in defining whether a simple CNN, TL or traditional DL i.e. Random Initialisation performs better. The similarity of one's dataset with the dataset that one is attempting to transfer Learn from can also be a factor affecting performance.

   Ensembles containing a small number of some models defined in Table 4.3 were also considered. The average results of the ensembles using 10-fold cross validation and a held out unseen

test dataset are also included in Table 4.2. The ensemble model with the highest accuracy was Ensemble 1 (ResNet50, ResNeXt50 and ResNet101) which had an accuracy of 84.339%. The ensemble models do not outperform TL with FE with the ResNet50 which has an average accuracy of 84.341% as per Table 4.2. All the ensembles evaluated using FE have an accuracy of at least 84% and an F1 score of at least 0.84. Ensemble 2 has the lowest variance and a slight increase in performance is observed in the ensembles evaluated that is proportional to an increase in parameter size.

The ResNet50 version 1 [41] outperformed all the other architectures in Transfer Learning via Feature Extraction that were evaluated outperforming some models with newer architectures (ResNeXt [45], ResNet version 2 [42] and InceptionResNetV2 [44]). It even outperformed some models with architectures released in 2019 (EfficientNet B0, EfficientNet B4 and EfficientNet B7 [46]). It also outperformed deeper models even with 3 $\times$ more the amount of hidden layers due to more data being needed to train deeper networks. ResNet50 (26m parameters) outperformed the state of the art EfficientNet B4 (19m parameters) by 5% and 16.5% in Transfer Learning via Fine Tuning and training from scratch (Random Initialisation) respectively despite the latter having a reported ImageNet accuracy of 4% higher than the ResNet50.

For the ResNet version 1 architecture using Transfer Learning via Feature Extraction the ResNet50 has better accuracy and F1 score performance (84.34%) than the ResNet 101 (83.61%) and the ResNet152 (80.07%) respectively. The performance is inversely correlated with an increase in hidden layers (depth) or network size. For the EfficientNet architecture models B0 (82.29%), B4 (80.59%) and B7 (78.65%) using Transfer Learning with Feature Extraction a similar decrease in accuracy and F1 score is found with an increase in network size (no. of parameters). However, for the ResNetV2 architecture for the two models evaluated (ResNet50V2 and ResNet152V2) the accuracy and F1 score are almost the same. So whether there is a correlation between an increase in model size inferring a lower Transfer Learning via Feature Extraction accuracy or not depends on architecture. The ResNet50 which has slightly more parameters than the EfficientNet B4 performs better than it for Random Initialisation and TL via Fine Tuning respectively. From Figure 4.10 it is evident that most models using Transfer Learning with Feature Extraction independent of the no. of parameters outperformed Random Initialisation and Transfer Learning with Fine Tuning with ResNet50 and EfficientNet B4 respectively on this dataset. This correlates with what is reported in the literature as Kornblith et al. also noted that TL using Feature Extraction performs better with smaller datasets than TL using Fine Tuning [7, 8].

4.6 Conclusion

The Blob Detection algorithm has a high false positive rate due to the underlying distribution of false positive blobs being much higher than true positive blobs in the displacement map. The algorithm detected true positive blobs which were caused by the occurrence of change at mines, quarries, construction developments, highways, factories and residenial complexes which have parking lots, etc. Vegetation such as trees, crops, grass, etc., resulted in false positive blobs occurring. False positives due to ponds, lakes, water collecting at the top of mine dumps mixing with the mine dump contents making a green liquid, etc., [8] were also detected by the algorithm.

A novel dataset for binary classification was developed which has four images per blob that correspond to the DInSAR bands of displacement, coherence and phase combined with the corresponding Sentinel-2 optical RGB image. Traditional ML using HOG and LBP feature extractors, a simple CNN, Deep Learning and Transfer Learning methods such as Feature Extraction TL, Fine Tuning TL, classic

Deep Learning i.e. Random Initialisation and Ensemble Learning are applied using a variety of architectures. ResNet50 with Transfer Learning using Feature Extraction has the highest accuracy. However, two of the ensemble models have the same F1 score as ResNet50 with Transfer Learning Feature Exraction and come very close to its performance in terms of accuracy. Transfer Learning via Feature Extraction accuracy decreases in some architectures with an increase in model size but this is dependant on the architecture. Most models using Transfer Learning using Feature Extraction outperform Random Initialisation and Transfer Learning using Fine Tuning with the ResNet50 and EfficientNet B4 respectively. Most of the models using DL and TL outperformed the traditional ML models which use the HOG, LBP and their combination feature extractors with Linear SVM except for the EfficientNet B4 models for RI and TL via FT. The Simple ConvNet outperformed most of the models except for TL via FE with ResNet101, ResNet50 and the ensemble models. There are both upward and downward trends found in the displacement time series developed. A velocity map was developed [8].

Chapter 5

Conclusion

DInSAR is used worldwide for deformation monitoring by developing a deformation time series via a number of techniques such as PSI, SBAS and SqueeSAR. Researchers have successfully used DInSAR for developing deformation time-series which correlates with actual deformation that occurred. In addition to subsidence monitoring DInSAR is also used to monitor earthquakes and volcanic eruptions.

A Sentinel-1B Interferometric Wide (IW) swath mode Single Look Complex (SLC) Vertical-Vertical (VV, both send and receive in vertical polarisation) SAR image data stack over the period 28th Nov. 2017 to 5th Dec. 2018 was used for the research with the images 12 days apart, where imagery was available. DInSAR was used to generate interferograms and terrain corrected displacement maps. Time series was developed for all of the pixels. However, since many of the interferograms have a perpendicular baseline below the optimal baseline of between 150 and 300 and due to phase and atmospheric noise the interferograms are noisy which affects the time series. Nevertheless in some of the time series developed some trends could be seen. A Laplacian of Gaussian algorithm is used to detect blobs in a DInSAR processed displacement map. The blob detection algorithm performs poorly with a high false positive rate due to DInSAR being vulnerable to low coherence in vegetated areas and there being a higher distribution of false positive blobs than true positive blobs in the displacement map. The resolution of the imagery also contributes to the high false positive rate since only changes that are at least 28m $\times$ 34m can be picked up. However, true positive blobs are detected which correspond to quarries, mines, construction (uplift and subsidence), etc. Due to the high false positive rate there is a need for a binary classifier.

A novel dataset was developed for binary classification containing four images per blob which correspond to the displacement, coherence and phase images and the corresponding optical images from Sentinel-2. Numerous architectures are used to apply Transfer Learning methods such as TL with FE, TL with FT along with traditional Deep Learning as well as Ensemble Learning. TL with FE is done with models that have weights which are pretrained on ImageNet for the generation of feature maps after the final layer that is responsible for predictions is removed. Thereafter, the linear Logistic Regression binary classifier is applied. Ten-fold cross validation is carried out with the parameter *C* which is the regularisation inverse hypertuned. The classifier classifies blobs as either positive corresponding to industrial change or as negative corresponding to noise since false positives can form in the displacement map due to bad coherence.

Transfer Learning using a pretrained model with weights trained on the ImageNet RGB images dataset can be used to perform binary classification on the novel dataset containing DInSAR Sentinel-1 bands georeferenced images with the corresponding Sentinel-2 optical RGB satellite imagery containing the same quantity of positives and negative blobs. This avoids having to train computationally expensive Deep Neural networks that have parameters numbering in the tens of millions and require millions of

images whilst gaining the benefit of Deep Learning accuracy by simply transferring the knowledge learnt from the ImageNet RGB images dataset to the novel Sentinel-1 DInSAR processed phase, coherence and displacement images with their corresponding Sentinel-2 optical imagery dataset.

ResNet50 [41] with Transfer Learning via Feature Extraction (84.34%) outperformed all the other models and methods that were evaluated including newer models, deeper models even with 3 $\times$ more the amount of hidden layers (due to more data being needed to train deeper networks) and ensemble models from an accuracy perspective. The existence of a correlation between an increase in model size inferring a lower Transfer Learning via Feature Extraction accuracy depends on architecture. The ResNet50 which has more parameters than the EfficientNet B4 architecture performs better with Random Initialisation and TL via Fine Tuning respectively. Most models using Transfer Learning with Feature Extraction independent of the no. of parameters outperformed Random Initialisation and Transfer Learning with Fine Tuning with ResNet50 and EfficientNet B4 respectively on this dataset.

TL via FE with the ResNet50 also outperformed the traditional ML models using the LBP, HOG and their combination feature extractors with Linear SVM by 16.39%, 23.20% and 23.72% respectively. However, it only outperformed a simple CNN, Simple ConvNet (83.73%), which has two convolutional layers and uses Batch Normalisation, Dropout and Global Average Pooling by 0.97%. Most of the models using DL and TL outperformed the traditional ML models which use the HOG, LBP and their combination feature extractors with Linear SVM except for the EfficientNet B4 models for RI and TL via FT. The Simple ConvNet outperformed most of the models except for TL via FE with ResNet101, ResNet50 and the ensemble models. It is imperative to note that all of these results are particular to this novel dataset and the size of the dataset is a crucial factor in defining whether a simple CNN, TL or traditional DL i.e. Random Initialisation performs better. The similarity of one's dataset with the dataset that one is attempting to transfer Learn from can also be a factor affecting performance.

Transfer Learning using Feature Extraction is the easiest method of transferring the knowledge and power of a Deep Learning model that has learnt rich discriminative filters on a very big dataset containing millions of images to a dataset for a new task which is normally much smaller in size than required in order to sufficiently train a DL architecture. However, a simple modern CNN such as Simple ConvNet should not be overlooked as it has the potential of outperforming DL and even TL with many architectures. Expanding the developed ensemble models and researching other methods of boosting the performance of the classifier can comprise future work. Perhaps a higher score could be obtained if one class learning was done [69, 70]. Deep Neural Networks are usually trained on datasets of thousands or tens of thousands and more images. Thus, more training data may also improve the results.

The Terrain Corrected Displacement Maps were used to develop a 28 data point time series for all the pixels in the Sentinel-1 DInSAR processed Displacement maps by stacking the displacement maps. However, given the noise in the images due to some of the images not being in the optimum range of perpendicular baseline and due to atmospheric and phase noise the time series is possibly noisy. Nevertheless, upward and downward trends can be seen corresponding to change. A velocity map was developed with uplift of 20cm to subsidence of 60cm over the 1 year period noting that the subsidence sometimes occurred in areas of low coherence where it is not accurate.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014. [Online]. Available: http://arxiv.org/abs/1403.6382

[3] RUSCopernicus Training, "RUS Webinar: Land Subsidence mapping with Sentinel-1 - HAZA03," https://rus-copernicus.eu/portal/wp-content/uploads/library/education/training/HAZA03_LandSubsidence_MexicoCity_Tutorial.pdf , Last accessed: Nov. 2018.

[4] Sentinel-1 Team, *Sentinel-1 User Handbook*. The Netherlands: ESA Publications, 2013.

[5] European Space Agency Sentinel Online, "Sentinel-2," https://sentinel.esa.int/web/sentinel/missions/sentinel-2 , Last accessed: Nov. 2018.

[6] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[7] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" *CoRR*, vol. abs/1805.08974, 2018. [Online]. Available: http://arxiv.org/abs/1805.08974

[8] Z. Karim and T. van Zyl, "Deep Learning and Transfer Learning applied to Sentinel-1 DInSAR and Sentinel-2 optical satellite imagery for change detection," *Proceedings of the 2020 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA) Conference, ©, 2020, IEEE*, pp. 579–585. [Online]. Available: https://ieeexplore.ieee.org/document/9041139

[9] A. Ferretti, A. Monti-Guarnieri, C. Prati, and F. Rocca, *InSAR Principles: Guidelines for SAR Interferometry Processing and Interpretation, TM–19*. The Netherlands: ESA Publications, Feb. 2007.

[10] A. Ferretti, C. Prati, and F. Rocca, "Permanent Scatterers in SAR Interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 1, Jan. 2001.

[11] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 11, pp. 2375–2383, Nov 2002.

[12] A. Ferretti, A. Fumagalli, F. Novali, C. Prati, F. Rocca, and A. Rucci, "A New Algorithm for Processing Interferometric Data-Stacks: SqueeSAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 9, pp. 3460–3470, Sept. 2011.

[13] A. Sowter, M. C. Amat, F. Cigna, S. Marsh, A. Athab, and L. Alshammari, "Mexico city land subsidence in 2014–2015 with Sentinel–1 IW TOPS: results using the Intermittent SBAS (ISBAS) technique," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 230–242, 2016.

[14] A. Sowter, L. Bateson, P. Strange, K. Ambrose, and M. F. Syafiudin, "DInSAR estimation of land motion using intermittent coherence with application to the South Derbyshire and Leicestershire coalfields," *Remote Sensing Letters*, vol. 4, no. 10, pp. 979–987, 2013. [Online]. Available: https://doi.org/10.1080/2150704X.2013.823673

[15] N. Devanthry *et al.*, "Deformation monitoring using Persistent Scatterer Interferometry and Sentinel-1 SAR data," *Procedia Computer Science*, vol. 100, pp. 1121–1126, 2016.

[16] M. Nguyen, C. Chang, and K. Tseng, "Using Sentinel-1 TOPS SAR and SBAS for Land Subsidence Monitoring in Hanoi, Vietnam," *International Symposium on Remote Sensing 2018, At Korea*, May 2018.

[17] Q. Huang and X. He, "SAR interferometry for long term deformation mapping using SBAS method: A case study in Nanjing area," *2009 Joint Urban Remote Sensing Event*, pp. 1–5, May 2009.

[18] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *International Journal of Photogrammetry and Remote Sensing*, vol. 80, pp. 91–106, 06 2013.

[19] P. Agram, "InSAR time-series A quick introduction," *UNAVCO Jet Propulsion Laboratory*, 2015.

[20] Stanford Radar Interferometry Research Group, "SNAPHU: Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping," https://web.stanford.edu/group/radar/softwareandlinks/sw/snaphu/ , Last accessed: Nov. 2018.

[21] RUS Copernicus Training, "RUS Webinar: Land Subsidence mapping with Sentinel-1 - HAZA03, video," https://www.youtube.com/watch?v=w6ilV74r2RQ , Last accessed: Nov. 2018.

[22] R. Schalkoff, *Digital image processing and computer vision*. John Wiley & Sons Australia, Limited, 1989. [Online]. Available: https://books.google.co.za/books?id=-vtRAAAAMAAJ

[23] S. Price, *Computer Vision, Models, Learning and Inference*. Cambridge, UK: Cambridge University Press, June 2012.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[25] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[26] Jason Brownlee, "Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning," https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/ , Last accessed: Oct. 2019.

[27] A. B. Downey, *Think Stats*. O'Reilly Media, Inc., 2011.

[28] M. Kuhn and K. Johnson. (2013) Applied predictive modeling. New York, NY. [Online]. Available: http://www.amazon.com/Applied-Predictive-Modeling-Max-Kuhn/dp/1461468485/

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[31] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *CoRR*, vol. abs/1805.09501, 2018. [Online]. Available: http://arxiv.org/abs/1805.09501

[32] F. H. K. dos Santos Tanaka and C. Aranha, "Data augmentation using gans," *CoRR*, vol. abs/1904.09135, 2019. [Online]. Available: http://arxiv.org/abs/1904.09135

[33] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017. [Online]. Available: http://arxiv.org/abs/1712.04621

[34] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *CoRR*, vol. abs/1707.07012, 2017. [Online]. Available: http://arxiv.org/abs/1707.07012

[35] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 459–468. [Online]. Available: http://proceedings.mlr.press/v70/bello17a.html

[36] J. F. Kolen and S. C. Kremer, *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. IEEE, 2001. [Online]. Available: https://ieeexplore.ieee.org/document/5264952

[37] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology vol. 148,3*, pp. 574–591, 1959.

[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[42] S. R. Kaiming He, Xiangyu Zhang and J. Sun, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016. [Online]. Available: http://arxiv.org/abs/1603.05027

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[44] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[45] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016. [Online]. Available: http://arxiv.org/abs/1611.05431

[46] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[47] Franois Chollet, "Building powerful image classification models using very little data," https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html , Last accessed: Mar. 2020.

[48] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147. [Online]. Available: http://proceedings.mlr.press/v28/sutskever13.html

[49] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, 12 1964.

[50] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[51] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014. [Online]. Available: http://arxiv.org/abs/1411.1792

[52] L. Torrey and J. W. Shavlik, "Chapter 11 transfer learning," 2009.

[53] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, *A Survey on Deep Transfer Learning: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 47, 2018, Proceedings, Part III*, 10 2018, pp. 270–279.

[54] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzalos, "Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data," 2019.

[55] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *CoRR*, vol. abs/1709.00029, 2017. [Online]. Available: http://arxiv.org/abs/1709.00029

[56] C. Wang, A. Mouche, P. Tandeo, J. Stopa, B. Chapron, R. Foster, and D. Vandemark, "Automated geophysical classification of sentinel-1 wave mode sar images through deep-learning," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 1776–1779.

[57] Y. Wang, C. Wang, and H. Zhang, "Combining single shot multibox detector with transfer learning for ship detection using sentinel-1 images," in *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, Nov 2017, pp. 1–4.

[58] S. Saha, Y. T. Solano-Correa, F. Bovolo, and L. Bruzzone, "Unsupervised deep learning based change detection in sentinel-2 images," in *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, Aug 2019, pp. 1–4.

[59] C. Qiu, M. Schmitt, H. Taubenbck, and X. X. Zhu, "Mapping human settlements with multi-seasonal sentinel-2 imagery and attention-based resnext," in *2019 Joint Urban Remote Sensing Event (JURSE)*, May 2019, pp. 1–4.

[60] Scikit-image development team, "Blob Detection," http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html , Last accessed: Nov. 2018.

[61] European Space Agency, "Copernicus Open Access Hub," https://scihub.copernicus.eu/dhus/#/home , Last accessed: Nov. 2018.

[62] European Space Agency Science Toolbox Exploitation Platform, "SNAP," http://step.esa.int/main/toolboxes/snap/ , Last accessed: Nov. 2018.

[63] L. Veci, *Sentinel-1 Toolbox TOPS Interferometry Tutorial*. Canada: Array Systems Computing Inc, 2015.

[64] European Space Agency, "Sen2Cor," https://step.esa.int/main/third-party-plugins-2/sen2cor/ , Last accessed: Feb. 2019.

[65] Google, "Google Earth," https://earth.google.com , Last accessed: Mar. 2019.

[66] OSGeo Project, "Geospatial Data Abstraction Library," https://www.gdal.org , Last accessed: Feb. 2019.

[67] P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 838–846. [Online]. Available: http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf

[68] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.

[69] M. Richter, T. Lngle, and J. Beyerer, "Gaussian mixture trees for one class classification in automated visual inspection," 07 2017, pp. 341–351.

[70] M. Kemmler, E. Rodner, and J. Denzler, "One-class classification with gaussian processes," in *ACCV*, 2010.

[71] T. K. Ho, "Random Decision Forests," *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282, 14-16 Aug. 1995.

[72] E. M. Kleinberg, "Stochastic discrimination," *Annals of Mathematics and Artificial Intelligence*, vol. 1, pp. 207–239, 1990.

[73] Z. Duan, "Characters recognition of binary image using knn," in *Proceedings of the 4th International Conference on Virtual Reality*, ser. ICVR 2018. New York, NY, USA: ACM, 2018, pp. 116–118. [Online]. Available: http://doi.acm.org/10.1145/3198910.3234651

[74] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[75] J. K. Kim, J. M. Park, K. S. Song, and H. W. Park, "Detection of clustered microcalcifications on mammograms using surrounding region dependence method and artificial neural network," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 18, no. 3, pp. 251–262, Apr 1998. [Online]. Available: https://doi.org/10.1023/A:1007989231205

[76] W. Zhang, K. Doi, M. L. Giger, Y. WU, R. M. Nishikawa, and R. A. Schmidt, "Computerized detection of clustered microcalcifications in digital mammograms using a shiftinvariant artificial neural network," *The International Journal of Medical Physics Research and Practice*, vol. 21, no. 4, pp. 517–524, Apr 1994.

[77] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," *CoRR*, vol. abs/1902.09843, 2019. [Online]. Available: http://arxiv.org/abs/1902.09843

[78] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[79] European Space Agency Sentinel Online, "Sentinel-2," https://sentinel.esa.int/web/sentinel/missions/sentinel-2 , Last accessed: Nov. 2018.

[80] Franois Chollet, "Keras: The Python Deep Learning library," https://keras.io , Last accessed: Mar. 2020.