# Heterogeneous Change Detection on Remote Sensing Data with Self-Supervised Deep Canonically Correlated Autoencoders

*Candidate:*
**Federico Figari Tomenotti**

Internet and Multimedia Engineering
LM-27

Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni
Università degli studi di Genova
a.a. 2018-19
Marzo 2020

*Supervisors:*
Assoc. Prof. Gabriele Moser
Assoc. Prof. Stian N. Anfinsen, UiT The Arctic University of Norway

*Co-Supervisor:*
Luigi T. Luppino, UiT The Arctic University of Norway

## *Colophon*

This master's thesis was submitted to the University of Genoa. However, the main part of the work was carried out during the author's exchange as an Erasmus student to UiT The Arctic University of Norway, where he was supervised by members of the Machine Learning Group in the Department of Physics and Technology.

The thesis has later resulted in a conference paper presented at the IEEE International Geoscience and Remote Sensing Symposium 2020. It was also awarded the Premio Nazionale di Laurea ”Euginio Zilioli” (”Euginio Zilioli” national thesis award) for 2020 by the Italian Association of Remote Sensing (AIT) and the Institute for Electromagnetic Sensing of the Environment of the Italian National Research Council (CNR-IREA).

The work is published to serve as a reference for further publications fostered through the collaboration between the author and the thesis supervisors at the University of Genoa and UiT The Arctic University of Norway.

**To be cited as:**

Figari Tomenotti, Federico (2021). Heterogeneous Change Detection on Remote Sensing Data with Self-Supervised Deep Canonically Correlated Autoencoders. Master's thesis submitted to the Electrical, Electronics and Telecommunications Engineering and Naval Architecture Department, University of Genoa, Italy, March 2020. Published in Septentrio Reports 2021 (4): 1-72. `https://doi.org/10.7557/7.5763`

Septentrio Reports, ISSN: 2387-4597<br>
Septentrio Academic Publishing<br>
UiT The Arctic University of Norway

# Acknowledgements

I want to thank the Machine Learning Group (MLG) at The Arctic University of Norway (UiT) for the warm welcome and the constant support throughout my stay. In particular I need to thank Stian and Luigi for the great effort in helping me doing this work.

I want to thank my family for always support me, throughout all University and life.

At last, I want to thank the “Movimento Liturgico Giovanile” (MLG) which has helped so much in my personal growth.

# Abstract

Change detection is a well-known topic of remote sensing. The goal is to track and monitor the evolution of changes affecting the Earth surface over time. The recently increased availability in remote sensing data for Earth observation and in computational power has raised the interest in this field of research. In particular, the keywords "multitemporal" and "heterogeneous" play prominent roles. The former refers to the availability and the comparison of two or more satellite images of the same place on the ground, in order to find changes and track the evolution of the observed surface, maybe with different time sensitivities. The latter refers to the capability of performing change detection with images coming from different sources, corresponding to different sensors, wavelengths, polarizations, acquisition geometries, etc.

This thesis addresses the challenging topic of multitemporal change detection with heterogeneous remote sensing images. It proposes a novel approach, taking inspiration from recent developments in the literature. The proposed method is based on deep learning - involving autoencoders of convolutional neural networks - and represents an exapmple of unsupervised change detection. A major novelty of the work consists in including a prior information model, used to make the method unsupervised, within a well-established algorithm such as the canonical correlation analysis, and in combining these with a deep learning framework to give rise to an image translation method able to compare heterogeneous images regardless of their highly different domains.

The theoretical analysis is supported by experimental results, comparing the proposed methodology to the state of the art of this discipline. Two different datasets were used for the experiments, and the results obtained on both of them show the effectiveness of the proposed method.

# Contents

# Chapter 1

# Introduction

Nowadays, data are one of the leading assets in our society, and their analysis is a driver for new researches and new investments. Thanks to the new generations of satellites, and increased capacity in storage and computation, remote sensing is gaining more and more importance in research studies of many fields. The number of satellites is continuously growing, and this leads to more acquisitions which allow for an easier Earth and environmental monitoring. Remote sensing images are catching on in our daily life, ranging from common web mapping and weather forecast services to advanced studies on climate change, environmental monitoring, disaster risk management, etc.

This thesis deals with the topic of multi-temporal change detection with heterogeneous remote sensing images. It is an emerging and highly prominent topic in publications and journals. The attention to this matter is related to the vast availability of images acquired by different missions and sensors. In the past, almost only same-sensor (i.e. homogeneous) acquisitions were used for multitemporal change detection. However, it is becoming of the utmost importance to be able to compare heterogeneous images to take advantage of the variety of satellite observations: multispectral, panchromatic and radar with different wavelength bands, radar frequencies, polarisations, acquisition geometries, etc. Let us give a couple of examples: to track changes in time, it is necessary to assure the backward compatibility with older acquisitions systems; maybe old data were acquired by a retired satellite with outdated technology, and here comes the necessity of heterogeneous change detection. Furthermore, in case of disaster recovery, it is essential to use the first available image to assess the damages to roads and infrastructures, and it may not be possible to wait for the next same-satellite acquisition, which can be a few days later. Other social valuable applications of change detection are land usage and urban monitoring, post-catastrophe assessments, crop

monitoring and surveillance.
This hot topic in research is also challenging; the heterogeneous change detection aims to compare two acquisitions which are semantically different, for example, an optical image and a synthetic aperture radar (SAR) image, but also two optical images acquired by different optical sensors with distinct channels are classified as heterogeneous. The core of the problem is to tackle the complexity in comparing two different physical quantities because different sensors measure different quantities. The problem cannot be solved using visual inspection for many reasons, the first is that very specialised knowledge would be needed and also the quantity of data to be analysed would be extremely time-consuming if addressed through a photo-interpretation effort. An automatic approach is developed in this thesis.
The path chosen in this work falls entirely within the framework of unsupervised techniques of machine learning. More specifically, some concepts of classical learning have been used in pair with deep learning strategies.
The research has been focused on bi-temporal acquisitions. The core idea of the methodology proposed is the image translation across two domains, in order to bring the two heterogeneous acquisition towards a common domain in which they can be compared. For this purpose, A deep neural network is deployed to learn the translation function from one domain to the other and vice-versa. The domain translation is guided by prior information extracted automatically off-line from the images through a graph-theoretic approach based on local affinity matrices. The proposed deep neural network is formed by a pair of autoencoders, coupled together by a processing block performing the canonical correlation analysis (CCA) in the latent space to force code space alignment.

## 1.1 Contribution

The candidate's contribution is both theoretical and practical. Firstly, this study investigates the literature about change detection in general and CCA. Secondly, it proposes a new heterogeneous change detection method based on the integration of the CCA method and its derived techniques, of a deep learning architecture based on two autoencoders, and of a priori knowledge extracted through local affinity matrices. In this respect, the present work extends the approach developed in [Luppino et al., 2020], in which an adversarial approach was used to favor the alignment in a common domain. Moreover, the work conducted within the thesis activity also included the development and integration of the code to carry out the experiments, using different tools: Docker to create a virtual environment for the testing of the project; Python, TensorFlow and Keras to develop and integrate the ma-

chine learning code; experiments settings and testing to run the experiments on a server.
This thesis was carried out within an internship at UiT – the Arctic University of Norway and resulted in the following publication:
[Figari Tomenotti et al., submitted], **F. Figari Tomenotti**; L.T. Luppino; M.A. Hansen; G. Moser, S.N. Anfinsen;*Heterogeneous Change Detection with Self-Supervised Deep Canonically Correlated Autoencoders*, submitted to the 2020 IEEE IGARSS International Geoscience and Remote Sensing Symposium (IGARSS), Kona, HI, July 2020.

## 1.2 Outline

This thesis is organised into four chapters. Chapter 2 provides a general introduction to remote sensing, giving importance to data acquisition systems and providing detailed explanations of different methodologies of change detection. Chapter 3 presents some basic theory concepts and technical background in order to understand the machine learning methodologies used: Canonical Correlation Analysis and some Deep learning frameworks are presented. Chapter 4 explains in detail the proposed methodology. Chapter 5 presents and discusses the experimental results and the comparisons. In the last Chapter, 6, conclusions are drawn.

# Chapter 2

# Introduction to remote sensing and change detection

## 2.1 Remote Sensing

Remote sensing is the scientific and technical discipline whose aim is the information acquisition about a target without accessing directly to it. In other words, without touching or reaching it physically, it is possible to retrieve some parameters which allow determining some physical quantity of the object under analysis (such as shape, chemical composition, speed). All these methods take advantage of different electromagnetic techniques and data processing algorithms.

Despite the very generic name and the broad description given above, in this work, we will refer in particular to remote sensing for Earth observation. Earth is the place where we live, and we extract our resources from it: food, fuels, water; therefore, monitoring our planet is of the utmost importance. The results of remote sensing for Earth observation is also essential in many research studies for climate changes: and the major space agencies of the World play an active role in deploying new instruments and developing novel ways of studying these phenomena [NASA]. Moreover, industrial and agricultural applications of remote sensing studies are popular and already employed. Some of the most interesting and valuable applications in this discipline are briefly presented.

- *Land cover mapping.* It permits to monitor urban development as well as farming lands. For example, searching for building alteration or new construction is vital for authorities in order to collect taxes and monitor the security of the country. Besides, soil usage allows for observation of crop subdivision over territory and for statistical purposes [Moser et al., 2012].

- *Bio- and Geophysical parameters retrieval.* Very useful in environmental monitoring: biomass concentration retrieval in forests, analysis of plant species dispersion in a territory or surveillance of their health status; oceans studies and supervision (such as chlorophyll density, temperature [Minnett et al., 2019], Figure 2.1 shows an example). Mapping soil moisture and type for agricultural planning. Measuring wind speed or air temperature in a wide range of places, also in the middle of the oceans (e.g. allowing feasibility studies for wind farms).

- *Disaster Management.* Remote sensing permits authorities to have a clear idea of the entity of a natural (or anthropic) disaster just after it: comparing images of the same zone before and after the event [Inglada and Giros, 2004]. Of course, at least one post-catastrophe image needs to be acquired.

- *Arctic wildlife monitoring.* Scientists have found interesting to monitor animals, especially white animals who live in the Arctic, easy to spot by satellites [Lavigne, 1976].

- *Weather forecast.* It is of uttermost importance both in the short period:"tomorrow there will be a hurrican"; as in the long one: "temperature will increase of 2.5 K in the next 50 year". [Racah et al., 2016]

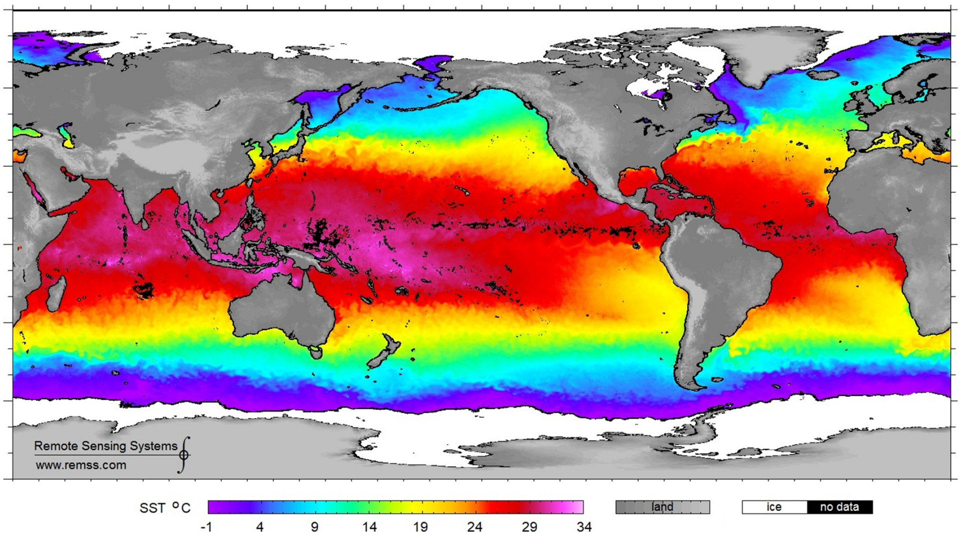


Figure 2.1: Example of remote sensing application: World sea surface temperature, November 2018. Credit: [Minnett et al., 2019]

All the mentioned applications are not new, neither impossible before the use of remote sensing techniques; however, they were too much expensive or time-consuming to be accomplished in an extensive way as nowadays. In the last decades, remote sensing for Earth observations has become increasingly popular due to the new techniques of data processing and the new capabilities in terms of available computational power. Furthermore, the number of satellites for this purpose has hugely increased, and some resources are now available for free. Remote sensing consists of two different operative moments: data acquisition and data processing, summed up in Figure 2.2

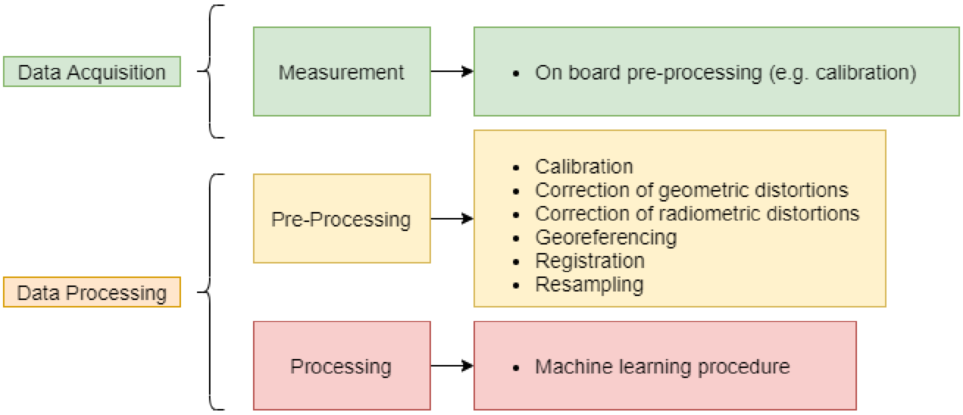

Data Acquisition
Measurement
On board pre-processing (e.g. calibration)
Data Processing
Pre-Processing
Calibration
Correction of geometric distortions
Correction of radiometric distortions
Georeferencing
Registration
Resampling
Processing
Machine learning procedure

Figure 2.2: Summary of the operational moments in an Earth observation processing chain.

## 2.2 Data Acquisition System

Remote sensing can use different means in order to acquire data; however, nowadays, the majority of data are collected from satellites. These satellites are equipped with special sensors which permit to scan the Earth surfaces in many distinct ways and to look for various targets. An example of the working principle is illustrated in Figure 2.3. Technically speaking, passive sensors measure the electromagnetic radiation (related to power density $[Watt/m^2]$) reflected from the Earth surface or spontaneously emitted by the surface itself. Each satellite is equipped to capture the electromagnetic radiation in several bands, where each band is a determined interval of contiguous frequencies. The information carried by each frequency is different, the optical (visible) portion of the spectrum carries information about colours of the target (the same structure we can appreciate with eyes); further on, the thermal infrared frequencies give information about the temperature of the objects. (There are many applications which use this concept to study the temperatures of the oceans or also of the mainland in remote regions of the planet [Handcock et al., 2012]). An example of the working principle is illustrated in Figure 2.3.

An acquisition system is very complex and many challenges need to be overcome in order to have it in place and working. Disregarding the physical structure and the instruments, we would like to spend a few words to explain the difficulties of the information retrieval process. First of all, the atmosphere is formed by many elements in the gas state, and it is hundreds of kilometres thick. The electromagnetic radiation, while passing through it, interact with these elements, and they can lose power due to absorption and distortion. Secondly, the behaviour of the atmosphere is not static, and so it should be modelled as a dynamic system, applying the right corrections to the signal. The atmosphere behaviour respect to the electromagnetic radiation is summed up in Figure 2.4; it is easily understandable that specific frequencies are entirely absorbed, and others remain unchanged crossing the atmosphere.

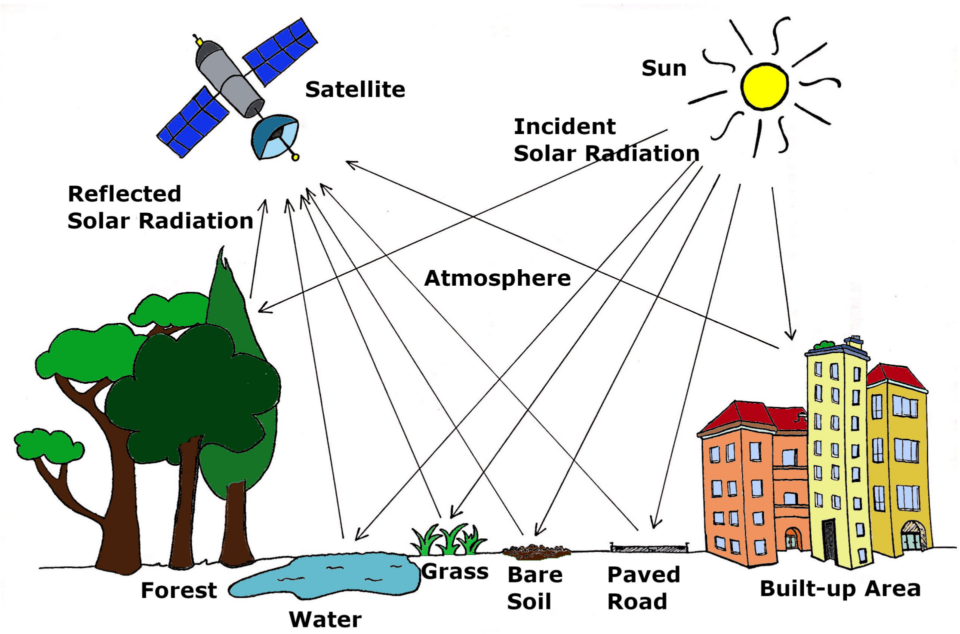


Figure 2.3: Remote sensing for Earth observation. Scheme of the data acquisition step.[1]

[1] Credits to Alessandra Maresca for drawing this beautiful scheme.

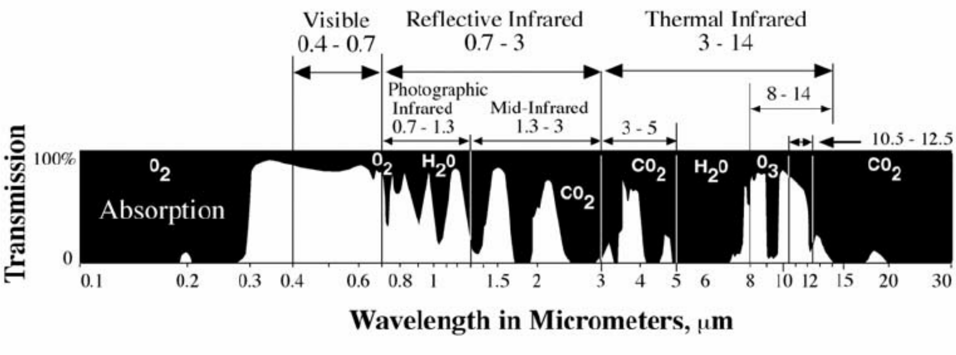


Figure 2.4: Atmospheric windows in the electromagnetic spectrum. White is the percentage of the transmitted power which passes through the atmosphere at that wavelength. Black represents the complementary absorption percentage and emphasizes the absorption bands.

This atmospheric behaviour forces to use only a small portion of the electromagnetic spectrum for our purposes. Indeed, the following Table 2.1 illustrates the wavelengths for Earth observation, which report as an example the bands in use of the Landsat 8 instruments, launched in February 2013.

| Operational Land Imager | | | |
|---|---|---|---|
| band | type | wavelength (μm) | spatial resolution (m) |
| 1 | visible | 0.43-0.45 | 30 |
| 2 | visible | 0.450-0.51 | 30 |
| 3 | visible | 0.53-0.59 | 30 |
| 4 | red | 0.64-0.67 | 30 |
| 5 | near-infrared | 0.85-0.88 | 30 |
| 6 | SWIR1 | 1.75-1.65 | 30 |
| 7 | SWIR2 | 2.11-2.29 | 30 |
| 8 | panchromatic | 0.50-0.68 | 15 |
| 9 | cirrus | 1.36-1.38 | 30 |
| Thermal Infrared Sensor | | | |
| 10 | TIRS1 | 10.6-11.19 | 100 |
| 11 | TIRS2 | 11.5-12.51 | 100 |

Table 2.1: Electromagnetic bands in use by Landsat 8 instruments.

### 2.2.1 Sensor and characteristics

There are many types of sensor for remote sensing, and they can be classified in different ways. The biggest classification is dividing sensors between passive and active:

- *passive*: they measure the spectral signature of the electromagnetic radiation emitted or reflected.
  The electromagnetic profile acts like a signature which allows to identify materials. More clearly, we can state that every material has its property of reflectance, and analysing the behaviour in a collection of frequency it is possible to separate it from all the others. An example is represented in fig 2.5.

- *active*: they illuminate the Earth with an electromagnetic source (usually in the microwaves) and measure the backscattered energy, this is known as radar technique. The most advanced type used is the SAR (synthetic aperture radar) system. Radar is extremely convenient because it uses longer wavelength compared to optical sensors. It ensures the signal to pass easily through clouds, smoke and to work day and night. However, it can not rely on a specific spectral signature for all materials.

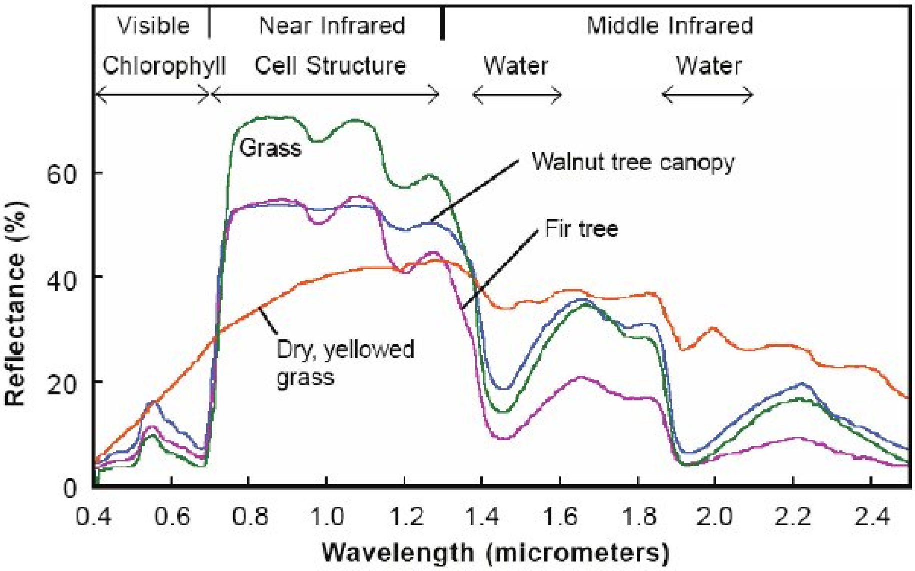


Figure 2.5: Example of spectral signature: vegetation reflectance. Plants are generally different in their reflectance signature, but differences are also appreciable between green and dry plants of the same species. Credit: [Govender et al., 2007]

Optical sensors are also characterised by some quantities which define the quality of the final image: the spatial resolution (size of the smallest distinguishable target on Earth), the spectral resolution (width of the bandpass around each scanned frequency), the radiometric resolution (quantisation of each band), the temporal resolution (revisiting time over the same zone).

It is also possible to classify the passive sensors based on the number of used bands in the electromagnetic spectrum:

- panchromatic sensor: it is a single-channel detector which usually spans all the visible range; the acquired images are black and white pictures from the space. The actual spatial resolution can reach 0.3 meters.

- multispectral sensor: it is a multi-channel detector, with 5-7 bands; usually the visible region is included.

- superspectral sensor: it acquires an image which is a superposition of different intensity measures in many separate and narrow bands of the spectrum. This type of sensor usually has more than 10 bands.

- hyperspectral sensor: it is also known as imaging spectrometer, and it deploys many bands, usually hundreds, with a very narrow bandwidth.

## 2.3 Data processing

Remote sensing is not only data acquisition but above all, data manipulation and processing.
The acquired images undergo two different steps: the pre-processing phase and the processing proper. The pre-processing includes some calibration, correction of geometric or radiometric distortion and georeferencing. Instead, the processing phase aims to extract the useful and desired information also combining them with ancillary information, maybe ground measurements or some a priori information. Data processing for change detection makes use of machine learning algorithms; both supervised and unsupervised settings found their application in remote sensing. For now, let us only say that supervised algorithms need some extra input to reach their goal correctly. On the other hand, unsupervised ones do not need any other input more than the satellite data.
In the data processing framework, there are many possibilities in order to achieve different goals. The target of this thesis is to perform change detection, which is described in the following sections.

### 2.3.1 Data types

Remote sensing is about acquiring data and process them to get useful information. Before entering deeply into the processing part, we shall statistically characterise the data.
First of all, data are always affected by errors; in this application, they are mainly due to noise during the acquisition process. In particular, optical data have two major noise types: additive uniform noise and salt and pepper noise. On the contrary, radar images are affected by speckle, which is a multiplicative noise-like phenomenon. Properly speaking speckle is not noise but an inborn result of the radar process acquisition, however, it makes images look noisy. The argument will be examined more in-depth in the next chapter.

## 2.4 Change detection

This discipline aims at finding differences given a series of images of the same place, taken in different time instants. It is useful to highlight changes on the ground (e.g. new buildings, change of crop). The simplest case is when only two images are present $\boldsymbol{X}^{t_1}$ and $\boldsymbol{Y}^{t_2}$, where $t_1, t_2$ are two generic time instant, with $t_1 < t_2$.

Having a couple of images representing the same place (e.g. an urban area), maybe in RGB colours or in b/w, it does not sound like a hard task spotting differences between them. Even though our brain is capable of distinguishing differences, it performs this operation in a very sophisticated way. For example, it would neglect some features that we know not proper of the terrain or the buildings, for example, the shadows. However, a machine does not know what is a shadow, that it can turn with the Sun movements, and that is not a proper change on the ground. Taking pictures from the satellite implies to count for the differences in illumination, time of the day or angle of view. These are not hard tasks for our brain; however, they are for a computer.

On the contrary, a human can only analyse some $km^2$ of terrain; instead, a machine can analyse entire regions in a small amount of time. This argument is also more persuasive if we think to compare hyperspectral images when the number of channels is quite high, and the information carried in some bands, outside the visible region, can be meaningless to us, or better we are not able to appreciate changes.

On the downside, a computer needs to know what is looking for and what type of difference to neglect. Because, as partially already stated, the Sun elevation, parallax effects, registration error and noise can generate spectrally appreciable changes, but without belonging to a specific or semantic class transition [Volpi, 2013]. In other words, a critical point in change detection is the influence of image changes which do not represent real variations in the structure of the analysed environment; we have mentioned shadows, but further, we can say clouds (in optical images) and clouds shadows on the terrain; atmospheric interaction during different seasons or time of the day. To cope with all these problems some countermeasures have been adopted; the basic one is the assumption to use acquisition where relevant changes are more significant in intensity than signal changes due to other reasons (e.g. atmospheric conditions). The next two sections investigate two different change detection framework: the Homogeneous and the Heterogeneous. The former utilises types of images acquired by the same sensor, and in similar conditions of light, orbit direction and angle. The latter, instead, is more challenging because it concerns images from different sensors and also from different domains, for example, from optical and radar sensors.

The final goal of change detection is a change map, that is a 2-class classification of the original image; in other words, each pixel must be labelled as changed or not changed.

### 2.4.1 Homogeneous Change Detection

Homogeneous change detection means combining and comparing information acquired by the same sensor, or at least the same sensor type. It deals with the comparison of images which lay in the same domain, so acquired with the same frequency, polarisation, geometry, etc. The key point is to have a homogeneous domain where the measurements taken by the instruments represent the same quantity: intensity, reflectance, radiance.
Different methodologies have been developed to obtain the change map in a Homogeneous case. Nevertheless, the most simple way is through mathematical and comparison operators: difference for optical images, and ratio for radar images. The approach is different because the two types of images suffer from different noise patterns. For homogeneous change detection, there are two typical approaches as highlighted in [Bovolo and Bruzzone, 2015]: fusion at the feature level and fusion at the decision level.
*Fusion at feature level* is intended as a comparison in the raw data domain. It is possible to extract the multitemporal information needed, analysing the different signatures in the two time instants. This class of techniques is mainly used with unsupervised algorithms. To cite some of them: differentiation/ratio (also known as Univariate Image Differencing or Change Vector Analysis for optical images) with thresholding and automatic thresholding algorithms [Moser and Serpico, 2006]; non-linear feature extraction is also feasible but more complex; further, the Principal Component Analysis can be applied to the single time image or to the stacked features as in [Fung and LeDrew, 1987].
*Fusion at decision level* is quite different from the previous because it assumes to classify and to segment the two images and then perform change detection on the result of the segmentation. In this case, the segmentation can be done relying on each image separately or exploiting the mutual information between them to construct the segmented images.

It is evident how the two methodologies are prone to errors in different cases; however, when well-tuned and relying on good images (correctly registered, calibrated, etc.), they can achieve good performances. Moreover, there are many areas of interest where the homogeneous change detection framework is easily applicable and very convenient. For example, to monitor some medium-long term changes: because - even if the revisit time of a satellite is long or some acquisitions are useless due to weather condition - it is possible to obtain an excellent final result. The main drawback of the

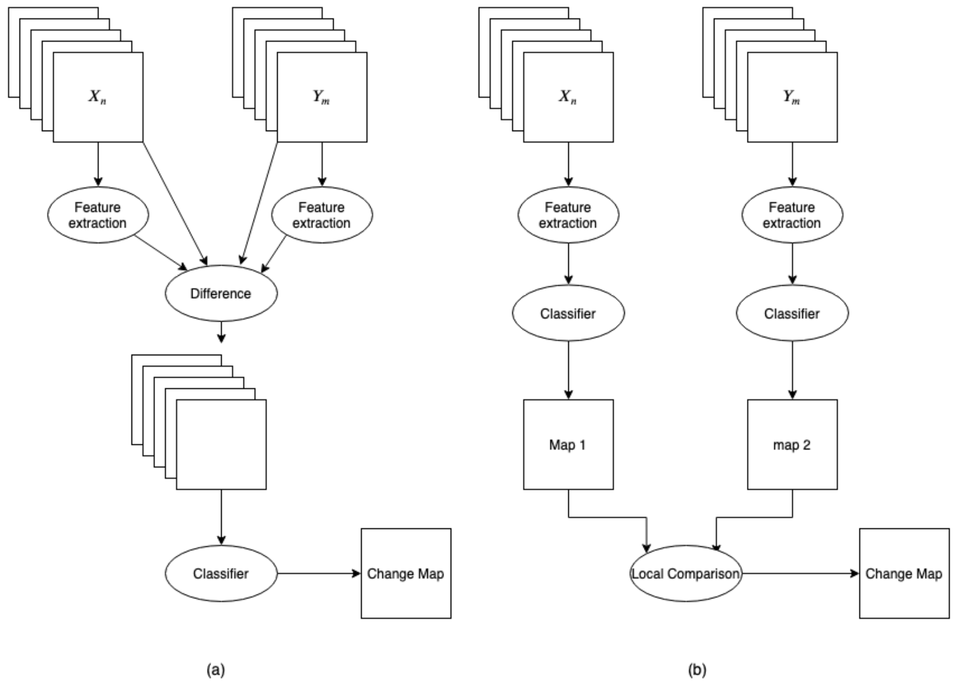


Figure 2.6: Multitemporal data fusion for change detection. (a) Fusion at feature level, (b) Fusion at decision level.

Homogeneous methodology, in the most strict circumstances, it is that the algorithm can be applied only to the measures taken from one sensor in one specific operational modality. In case a satellite (or a family of them) has been retired and substituted with a new one, the compatibility of the old method in order to compare old and new images is not assured.Moreover, the instruments of many recent missions can be operated in a variety of modalities, which differ in their geometry, polarization, or frequency.
Concluding, it is essential to say that all the previous methods do not always fit with very-high-resolution images.

### 2.4.2 Heterogeneous Change Detection

Heterogeneous change detection (HCD) is an emerging topic in earth observation. It answers the increasing availability of remote sensing data by offering methods that allow to combine images of radically different nature and still extract reliable information about changes on the surface. The images could be acquired by multimodal sensors, such as optical instruments and synthetic aperture radar (SAR), or they can be recorded with different sensor parameters or under distinct environmental conditions, cases that would otherwise not be comparable unless possibly through meticulous pre-processing and co-calibration. In the bitemporal setting (two images available), HCD is particularly useful to obtain situational awareness after sudden change events such as a natural disaster. That is when it is important to use the first image source of opportunity to map changes, instead of waiting for the next acquisition that permits a comparison of homogeneous images. Furthermore, for monitoring long-term trends, the joint analysis of heterogeneous sources allows us to extend the time frame of the analysis or to increase the temporal resolution. Lastly, SAR images are available also in case of cloud cover (tropical and sub-tropical areas are very prone to this phenomenon) or smoke cover of the sky, because microwave penetrates in them.

Regardless of the motivation, HCD relies on the fundamental assumption that the changed areas have a distinct signature for all the sensors involved, even though the physical origin of this signal may be different. Moreover, since an absolute reference is lacking when we contrast heterogeneous data, the problem is inherently ill-posed, and the labelling of pixels or segments as changed and unchanged is generally ambiguous. It is necessary to assume some additional prior information in order to discern the change class. A typical prior assumption is that the change concerns small regions or a minority of the pixels in an image or another one is when the characteristic signature of one of the classes involved in the transition is known. The mentioned minority assumption is common in generic methods, while signature assumptions can be advantageous to customise an algorithm for a thematic application.

While the first works on HCD were developed in the supervised setting, focus in recent years has turned to the unsupervised case [Mercier et al., 2008]. This makes the method more suitable for practical cases since ground truth in Earth observation is sparse and costly to collect. Another trend is that deep learning prevails more and more, as in other areas of computer vision and image analysis. Most current HCD approaches adopt transformations between the input domains, or from these to a common latent domain, to bring data to a space where they can be efficiently compared. Convolutional neural network (CNN) architectures such as autoencoders and generative adversarial networks are flexible and powerful tools that can accomplish these image translation tasks, as reviewed in [Luppino et al., 2019, 2020].

# Chapter 3

# Theory and technical background

## 3.1 Machine Learning Introduction

Machine Learning is a discipline at the intersection of computer science and statistics. It is the ability to use data and models to predict some behaviour, or again to use data to create an high predictive model of a phenomenon. The machine learning core is detecting patterns and regularities underneath the raw data. Machine learning is also a part of the big world of artificial intelligence. To be intelligent, a system needs also to have the capability of adapt to the changes in the environment. So we have stated that machine learning is to create models based on statistical and probabilistic rules. This thesis deploys some classical machine learning algorithms and methods as well as some modern deep learning ones. In the following, the basics knowledge to understand our methodology is presented; and because this work is not a systematic dissertation on machine learning, neither on deep learning only the necessary concepts will be illustrated.

## 3.2 Prior computation: the affinity matrix

An affinity matrix is a statistical object used to show similarity between data points. Is is constructed setting a metric and looking for data which have minimum distances, and represent them with a 1 in the matrix (a 0 means different data); so it uses the concept of distance, but however it is quite the opposite, because when the distance between two instances is 0,

the matrix entry is set to 1. The deploy of this concepts let machine to mimic the human action of associating similar things. And this similarity can be every concept, it depend on the metric chosen. Specialising the concept, an affinity matrix can look for repetitive or similar patterns inside pixels and group of pixel.
An extension of a binary affinity matrix is a matrix where each entry is calculated as a result of a multiplication of our data with a kernel, in this case values can range, for example, in the set $[0, 1]$.

## 3.3 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a method for reducing the dimensionality of a couple of sets of samples taking into account their mutual correlation. It projects the samples in a common space where the correlation between them is maximized.

The next section introduces the CCA theory for vectors following [Mardia et al., 1979], the other try to define some operative rules.

### 3.3.1 Theory

Suppose to have two random vectors: $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively q-dimensional and p-dimensional: $\boldsymbol{x} \in \mathbb{R}^q$, $\boldsymbol{y} \in \mathbb{R}^p$. Now suppose further that

$$\begin{aligned} \boldsymbol{\mu} &= E\{\boldsymbol{x}\} \\ \boldsymbol{\nu} &= E\{\boldsymbol{y}\} \end{aligned}$$

are their means, and

$$Cov(\boldsymbol{x}) = \boldsymbol{\Sigma}_{11} = E\{(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\} \qquad \in \mathbb{R}^{q \times q} \tag{3.1}$$

$$Cov(\boldsymbol{y}) = \boldsymbol{\Sigma}_{22} = E\{(\boldsymbol{y} - \boldsymbol{\nu})(\boldsymbol{y} - \boldsymbol{\nu})^T\} \qquad \in \mathbb{R}^{p \times p} \tag{3.2}$$

$$Cov(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^{\boldsymbol{T}} = E\{(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\nu})^T\} \qquad \in \mathbb{R}^{q \times p} \tag{3.3}$$

Now consider two linear combinations $\eta = \boldsymbol{a}^T \boldsymbol{x}$ and $\phi = \boldsymbol{b}^T \boldsymbol{y}$ . They are projections of our vectors along the directions of $\boldsymbol{a}$ and $\boldsymbol{b}$. The correlation between $\eta$ and $\phi$ is

$$\rho(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \boldsymbol{\Sigma}_{12} \boldsymbol{b}}{(\boldsymbol{a}^T \boldsymbol{\Sigma}_{11} \boldsymbol{a} \boldsymbol{b}^T \boldsymbol{\Sigma}_{22} \boldsymbol{b})^{\frac{1}{2}}} \tag{3.4}$$

Now we want to find $\boldsymbol{a}$ and $\boldsymbol{b}$ for which the correlation is maximised. In other words we try to solve the problem

$$\max_{\boldsymbol{a},\boldsymbol{b}} \quad \boldsymbol{a}^T \boldsymbol{\Sigma}_{12} \boldsymbol{b} \qquad \text{s.t.} \quad \boldsymbol{a}^T \boldsymbol{\Sigma}_{11} \boldsymbol{a} = \boldsymbol{b}^T \boldsymbol{\Sigma}_{22} \boldsymbol{b} = 1 \tag{3.5}$$

because equation 3.4 does not depend on the scaling of $\boldsymbol{a}$ and $\boldsymbol{b}$ (both the numerator and the denominator depends linearly on the magnitude of the two), hence it is not restrictive to consider a unit-variance constraint on each projection [Alpaydin, 2014].
It is now possible to write our problem as a Lagrangian problem,

$$L(\lambda, \boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^{\boldsymbol{T}} \boldsymbol{\Sigma}_{12} \boldsymbol{b} - \frac{\lambda_x}{2}(\boldsymbol{a}^{\boldsymbol{T}} \boldsymbol{\Sigma}_{11} \boldsymbol{a} - 1) - \frac{\lambda_y}{2}(\boldsymbol{b}^{\boldsymbol{T}} \boldsymbol{\Sigma}_{22} \boldsymbol{b} - 1) \tag{3.6}$$

and then we take the partial derivatives respect $\boldsymbol{a}$ and $\boldsymbol{b}$ and equal them to zero

$$\frac{\partial f}{\partial \boldsymbol{a}} = \boldsymbol{\Sigma}_{12} \boldsymbol{b} - \lambda_x \boldsymbol{\Sigma}_{11} \boldsymbol{a} = \boldsymbol{0} \tag{3.7}$$

$$\frac{\partial f}{\partial \boldsymbol{b}} = \boldsymbol{\Sigma}_{21} \boldsymbol{a} - \lambda_y \boldsymbol{\Sigma}_{22} \boldsymbol{b} = \boldsymbol{0} \tag{3.8}$$

After some calculation, we end up with an eigenproblem, and in its solution, $\boldsymbol{a}$ and $\boldsymbol{b}$ should be eigenvectors of $\boldsymbol{\Sigma}_{11}{}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}$ and $\boldsymbol{\Sigma}_{22}{}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}{}^{-1}\boldsymbol{\Sigma}_{12}$, respectively [Hardoon et al., 2004]. Because we are interested in maximixing the correlation, we choose the two eigenvectors with the highest eigenvalues; let us define the two eigenvalues as $\boldsymbol{a}_1, \boldsymbol{b}_1$, of dimensions respectively $q$ and $p$; the eigenvalues are actually just one, shared by the two matrices (eigenvalues of $\boldsymbol{AB}$ are the same of $\boldsymbol{BA}$ [Alpaydin, 2014]).
It is however possible to choose how many pairs of eigenvectors $\boldsymbol{a}_i, \boldsymbol{b}_i$ to use. If $k$ pairs of eigenvectors are in use, to project our data we must take the matrix $q \times k$ whose columns are $\boldsymbol{a}_i$, and respectively the matrix $p \times k$ composed by $\boldsymbol{w}_i$ as columns. The new space has constituted by non redundant features: all the $\boldsymbol{a}_i$ are uncorrelated and each $\boldsymbol{a}_i$ is uncorrelated with $\boldsymbol{b}_j, i \neq j$.

## 3.4 Deep learning

Deep learning is quite a new approach to learning, however is based on some rather consolidated ideas, for example artificial neural networks. The term *deep* broadly indicates a huge neural network, and more precisely refers to a neural network with a high depth, i.e., many hidden layers. It has begun to attract attention since some years now, because the computational power of our machines has become capable to cope with the complexity in managing very big artificial neural networks. Their fame is due to the optimal results obtained by deep nets in many applications in the most different fields.

### 3.4.1 Artificial Neural Networks

An artificial neural network is a collection of simple units called neurons. Each neuron is composed by a summing unit and an activation function. Suppose to have some inputs $x_j \in \mathbb{R}, j = 1, ..., d$ and for each of them, a connection weight $w_i \in \mathbb{R}$. The output in the simplest case is a weighted sum of the inputs:

$$o = \sum_{j=1}^{d} w_j x_i + w_o$$

where $w_0$ is a bias. It is possible to write in a more compact notation using the dot product $y = \boldsymbol{w}^T \boldsymbol{x}$, where $\boldsymbol{w} = [w_0, w1, ...w_d]^T$ and $\boldsymbol{x} = [1, x_1, ..., x_d]^T$ include the bias. The learning is performed looking for the correct vector $\boldsymbol{w}$. Let us introduce the activation function $\phi$, which can be, for example:

$$y = \varphi(o) = \begin{cases} > 0 & a & \in \mathbf{R} \\ < 0 & b & \in \mathbf{R} \end{cases}$$

This is usually a non linear function, i.e. a sigmoid or a ReLU (rectified linear unit) and outputs just a scalar result. To visually understand the concept we can use Figure 3.1. A single layer of weights can approximate a linear function; instead, a connection between many neurons can also learn some non-linear relations and is called a network.

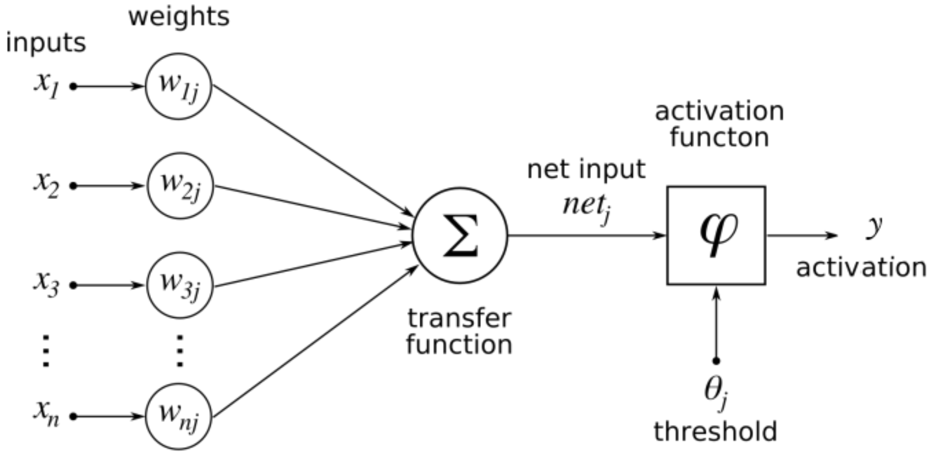


Figure 3.1: Structure of an artificial neuron.

The most simple network is a feedforward neural network which is built up using layers, and each layer is composed by many neurons; there are the input layer, the hidden layers and the output layer. The number of these

layers represents the depth of the network: here comes the term *deep* learning.
The training of the network, as of the single neuron, is performed feeding the network with some data instances, one by one, and defining an error (loss) function to guide the procedure. When a datum transverses the net, the activation propagates in the forward direction, the output is calculated and the error function is evaluated. The goal is to minimise the error, which is done by calculating the derivatives of the loss respect all the parameters $\theta$ (weights and biases) of the network. Based on this gradient, the parameters are changed according to the result of the operation. This operation is performed applying the derivatives and the chain rule, and its result is backward propagated along the chain till the input layer. In this way the error is propagated from output to input and that is why it is called backpropagation. The error is minimised iteration after iteration (some optimisation algorithm is used). This iterative behaviour suggests that training a network can be long and time consuming, but assures that the learning is continuous in time and the machine adapts to changes. The method presented is called stochastic gradient descent and starts initialising the weights randomly [Allen-Zhu et al., 2019]. However, there are advanced methods for learning, for example batch learning procedure, where the parameters are updated after some input data and not every sample; when all the dataset has passed inside the network an epoch is passed.

Motivation for interest in neural networks is also based on theorems which we are going to state and for whose proofs it is possible to read [Cybenko, 1989], [Csáji, 2001].

**Theorem 1** *Universal Approximation Theorem.*
*A feed-forward network with a single hidden layer containing a finite number of neurons can approximate any continuous function uniformly on a compact subset of $\mathbb{R}^n$, under mild assumptions on the activation function.*

Even if it is possible, this may not be practically feasible for whatever function, due to the dimension of the network. Because, the theorem does not say anything about constraints on the number of neurons respect the function complexity. However, under the assumption of a ReLU activation function it has been demonstrated in [Lu et al., 2017] that any Lebesgue-integrable function $f$ from $\mathbb{R}^n$ to $\mathbb{R}$ can be approximated by a fully connected width $(n + 4)$ ReLU network to arbitrary accuracy.

### 3.4.2 Convolutional Neural Networks

Convolutional neural network are a specialised type of networks for processing data that has a known grid-like topology [Goodfellow et al., 2016]. Examples are time-series data (1-D dimension) and images (2-D grid of pixels). The name arises from the specific type of calculation that is performed in the CNN: a convolution. It is possible to imagine this type of network as a series of different stacks of finite impulse-response filters, each disposed in a different layer. In other words, each filter can be thought as a convolution of the image with a kernel of smaller dimension. This type of network is extremely powerful in the images domain because it succeeds to capture the intrinsic representation of images, as sets of edges, patterns and figures.

### 3.4.3 Autoencoders

An autoencoder is a specific type of network whose goal is to copy the input on the output while favouring some specific properties. It can be seen as a network composed by two parts: an encoder function $e(\cdot)$ and a decoder $d(\cdot)$. If an input $\boldsymbol{x}$ is provided, it tries to reproduce it in the output $\boldsymbol{y}$, $d(e(\boldsymbol{x})) = \boldsymbol{x}$. In between the two parts, a code layer is present, which provides a transformed (often compressed) representation of the input data. However, this is not a complete description, because a network that has this behaviour is useless. Actually, the aim of the network is to recover $\tilde{\boldsymbol{x}} \simeq \boldsymbol{x}$, trying to reconstruct only interesting part of the inputs, and discarding some feature we want to get rid of. The topology of the network (Figure 3.2) is similar to a feedforward network, and also the training uses the same techniques, typically minibatch gradient descent following the gradients computed by backpropagation. Unlike the previous networks, also re-circulation may be used, that is an output is re-used as input. The usual architecture has a code layer where the representation of the input information is compressed; this allows the network to extract only the useful features and prevent the network to learn the identity transformation. If the encoder and decoder have too many parameters respect the problem dimension, it can occur that the autoencoder learns the useless identity transformation. To prevent this unwanted situation, some precautions have been adopted and some more properties have been added to the autoencoder: sparsity of the representation, robustness to noise, smallness of the derivative. To implement the first feature, in each cycle of learning not all the samples are fed to the algorithm but only a randomly chosen subset is used. For the second, some noise is added to input samples during the learning and at the output compared with the original de-noised ones. The last aforementioned case, instead, is simply the trick to maintain derivatives small enough, in order to learn better the features which are constant respect to $\boldsymbol{x}$.

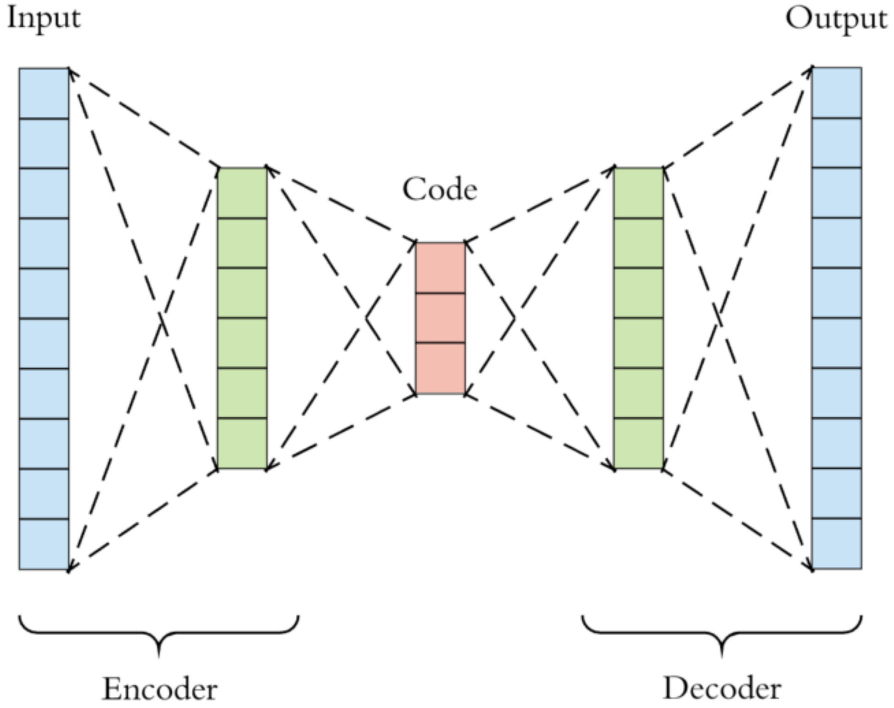


Figure 3.2: Sketch of an autoencoder network. The encoder and decoder are composed of neural networks.

### 3.4.4 Deep Canonical Correlation Analysis

In the world of deep networks many different architectures have been proposed, among them, we recall here [Andrew et al., 2013] and [Wang et al., 2015]. The common ancestor of most of them is reported in Figure 3.3. Moreover, all the cited works deployed a supervised framework for all of them.

In the Deep Canonical Correlation Analysis (DCCA) the metric used to measure the extracted information and the performances is the "quantity of correlation": the sum of the correlation for the top most correlated directions. Indeed, [Andrew et al., 2013] focused on the quantity of correlation which can be extracted with different methodologies and demonstrated that a DCCA extracts much more correlated feature as compared to a CCA and a Kernel CCA (KCCA) [Andrew et al., 2013].

The training for this network, using a gradient descent algorithm, need a custom gradient function which [Andrew et al., 2013] has provided. Moreover, because the correlation objective is a function of the entire training set and cannot be decomposed into a sum over the data points, a stochastic approach is not feasible; the article instead proposes mini-batch descent or full-batch optimisation.

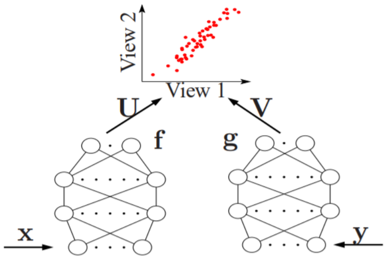


Figure 3.3: DCCA framework proposed in [Wang et al., 2015]. It includes two NN (encoders), and at their output a transformation in a maximally correlated domain through the $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices.

### 3.4.5 Deep Canonically Correlated Autoencoders

Inspired by DCCA [Wang et al., 2015] proposed the Deep Canonically Correlated AutoEncoder (DCCAE), see Figure 3.4. This network implements a trade off: the autoencoder maximises the learning of information between inputs and learned features, instead the CCA maximises the information of the two different views. From this perspective, it can represent more sophisticated interactions between data as compared to a simple DCCA, and also autoencoders alone.

The DCCAE network has been adopted by [Zhou et al., 2019] to perform an heterogeneous change detection task and showed good results and low variance. It especially over-performed CCA and DCCA. The training have been carried on with mini-batch gradient descent, but assuring to use a batch size big enough to be representative in the calculation of the sample covariances.

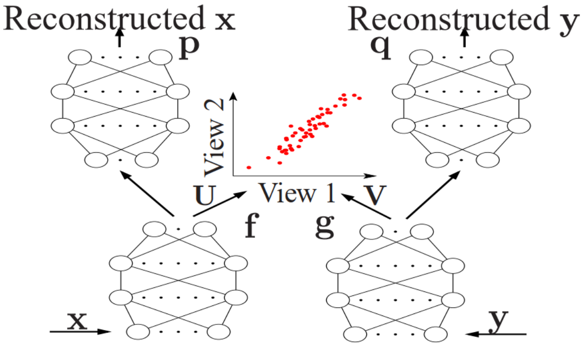


Figure 3.4: DCCAE framework proposed in [Wang et al., 2015]. Two autoencoders are tied together by a CCA performed in the latent space.

# Chapter 4

# The Proposed Change Detection Method

## 4.1 General idea of the methodology

This chapter is devolved to explain the proposed heterogeneous change detection method.
The methodology proposed in this work aims to compare two different types of data, which, on their own, could not be compared. Indeed, it is not possible to use a traditional method to do change detection in this environment; for instance, only subtracting the two images does not have any meaning. As mentioned above the two images lay in two different domains; hence we need to transform them into a common domain and then compare them. Denoting the two images to be compared as $\boldsymbol{X}$ and $\boldsymbol{Y}$ and using the same names for their respective domains, we can summarise as in Figure 4.1. The figure shows that not only it is possible to convert the two images in a common domain, but it is also feasible to convert one image in the domain of the other. Theoretically, it allows to compare the two images in the domain of $\boldsymbol{X}$, or $\boldsymbol{Y}$, or in the latent space $\boldsymbol{Z}$.
In this framework (Figure 4.1) the arrows represent regression functions. In particular, each of these is a neural network properly trained for the purpose.

At this stage, we have brought the two images in a shared (or common) space, where it makes sense to use some elementary change detection method (e.g. image differencing). However, we introduced some neural networks, which need training to be used. In remote sensing, some labelled samples are needed to train a network, and they are difficult to retrieve and expen-

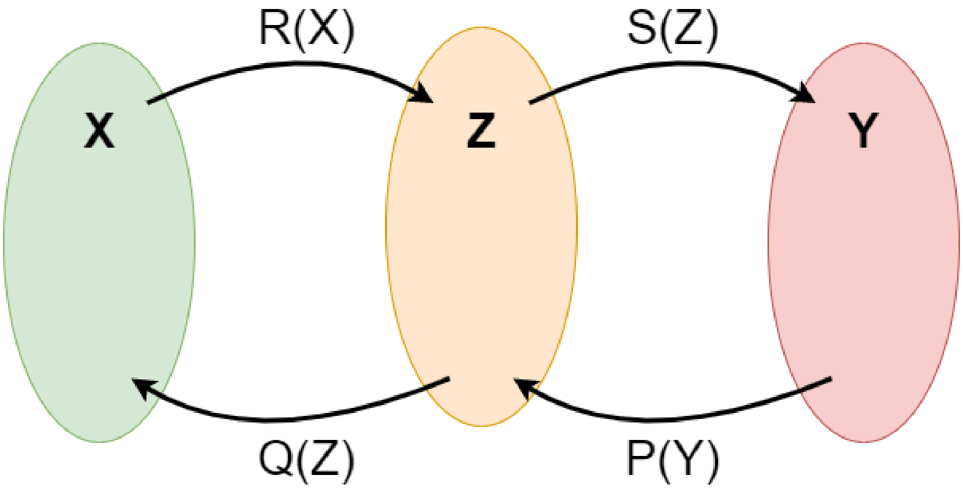


Figure 4.1: The proposed framework with three different domains represented as sets and four regression functions represented as arrows.

sive. Therefore, in our case, we want to train the networks to transform one image into the other one, but using nothing more than the input data.
What would happen if we trained the network with our two images? We could have done because they provide examples of the two distributions we would like the network to learn. However, we must recall that the two images are taken at different times and generally exhibit changes, and this is a big issue for our learning. We want the network to learn to individuate changes as abnormal patterns, and not as a rule. Thus, an innovative technique is used to automatically retrieve some training samples located in likely unchanged areas from our data [Luppino et al., 2019], turning our procedure to a completely unsupervised method. This stage is conceived as a method to extract information and to return a probability-like score that expresses the chance that each pixel is changed from one acquisition to the other; this is explained in [Luppino et al., 2020].

The second big issue to solve is the problem of being sure that the latent space $\boldsymbol{Z}$ is unique and is a common transformed space for both mappings $\boldsymbol{R}(\boldsymbol{X})$ and $\boldsymbol{P}(\boldsymbol{Y})$ (see Figure 4.1). To assure this consistency, a technique involving Canonical Correlation Analysis is proposed [Figari Tomenotti et al., submitted]. A very attractive feature of CCA is that ”if there is noise in either view that is uncorrelated with the other view, the learned representation should not contain the noise in the uncorrelated dimension” [Andrew et al., 2013].

The change detection scheme applied in the rest of the chapter is ex-

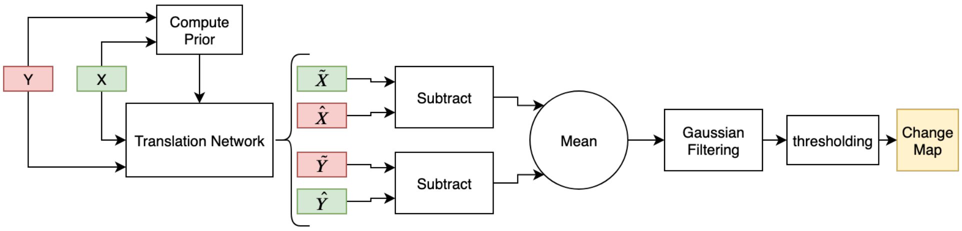


Figure 4.2: Block diagram of the change detection scheme used in this work. $X$ is the image before the change, $Y$ after it.

plained by Figure 4.2. After the problem setting description, the following sections will present the different functional blocks needed to build up the complete proposed system.

## 4.2 Problem setting

Two different sensors scan a geographical area in two different moments in time. We denote the two sensors (and also their respective domains) as $\mathcal{X}$ and $\mathcal{Y}$ and the respective acquisition times as $t_1$ and $t_2$. The two sensors generate two images with the same height $H$ and width $W$ (up to possible re-sampling and co-registration). The two images generally include different numbers of channels identified as $C_1$ and $C_2$. Thus, the two images are respectively $\boldsymbol{X} \in \mathbb{R}^{H \times W \times C_1}$ and $\boldsymbol{Y} \in \mathbb{R}^{H \times W \times C_2}$.

In the following, it is also assumed that a limited part of the image contains changes; this is crucial because we need a reliable non-changed part to train our networks (regression functions).

## 4.3 Affinity-based Change Prior

Our prior information is an affinity-based cross-domain pixel distance proposed in [Luppino et al., 2020], which is interpreted as a probability of change of that pixel.

The following procedure is applied to an image patch of dimension $k \times k$, and, when computed, the patch position is shifted in order to progressively reach all pixels in the entire image; it is applied to the images from both modalities.

Firstly, we compute the domain-specific affinity matrices $\mathbf{A}^{\mathcal{X}}$ and $\mathbf{A}^{\mathcal{Y}}$, whose elements $A_{ij}^{\mathcal{X}}$ and $A_{ij}^{\mathcal{Y}}$ are pairwise affinities between pixels $i$ and $j$ belonging to the patch. These are computed from pairwise distance measures $d_{ij}^{\mathcal{X}}$ and

$d_{ij}^{\mathcal{Y}}$ as

$$\mathbf{A}_{ij}^{\mathcal{X}} = \exp(-(d_{ij}^{\mathcal{X}})^2/h_{\mathcal{X}}) \tag{4.1}$$

and

$$\mathbf{A}_{ij}^{\mathcal{Y}} = \exp(-(d_{ij}^{\mathcal{Y}})^2/h_{\mathcal{Y}}) \tag{4.2}$$

by use of the common Gaussian kernel function with kernel widths $h_{\mathcal{X}}$ and $h_{\mathcal{Y}}$. The two kernel widths are domain specific, and set equal to the average distance of the $K^{th}$ nearest neighbour, with $K = \frac{3}{4}k^2$. This method allow to capture an intrinsic distance inside the patch [Luppino et al., 2020]. Moreover, the distance measures $d$ are computed as Euclidean distances. This choice is understandable considering the domain and the data distribution: optical images have a Gaussian behaviour (in intensity), whereas SAR images can be transformed applying a logarithm bringing them to near-Gaussianity [Zhan et al., 2018].

Highlighting the fact that the matrices $\boldsymbol{A}$ are symmetric, the cross-domain pixel distance for pixel $i$ is obtained as

$$\alpha_i = \frac{1}{n}\sum_{j=1}^{n} |\mathbf{A}_{ij}^{\mathcal{X}} - \mathbf{A}_{ij}^{\mathcal{Y}}|, \tag{4.3}$$

which is the average absolute affinity difference between pixel $i$ and $n$ other pixels. This assures that $\alpha_i \in [0,1]$, providing small values when pixel relations, within the size $n$ image patch or neighbourhood, remains similar across image domains, and large values otherwise. This is reasonable because only changes between images should present larger values in the difference matrix. This method is very powerful to look for changed patterns inside images and assign to every single pixel a probability of being changed. Even if the method is not too heavy for modern computational power, it can be sped up using a sliding window which moves faster than a pixel per time. Of course, this comes with a resolution degradation. To examine in depth the prior retrieval discussed, it is possible to refer to [Luppino et al., 2020]] where a useful toy-example is presented.

We will utilise $\alpha_i$ to suppress the influence of pixels with a high probability of change, and therefore we must define a weighting function $\Pi(\alpha)$ : $[0,1] \rightarrow [0,1]$ that is monotonically decreasing. Hence, the higher is $\Pi(\alpha)$, the lower is the probability of that pixel to be changed from one acquisition to the other, and the higher is the confidence to use it as a learning sample. We use the simple function

$$\Pi(\alpha_i) = 1 - \alpha_i \tag{4.4}$$

however other decreasing functions can be adapted and used.
The computation of this matrix is meant to be performed offline: the $\Pi(\alpha_i)$ values can be calculated, stored and used when needed.

## 4.4 CCA formulation

The Canonical Correlation Analysis has been formulated as in [Wang et al., 2015] but adding the prior information in it. It is clear from the Section 3.3 that the CCA is a linear method and extract the covariances $\Sigma_{11}, \Sigma_{22}$ and the cross-covariance $\Sigma_{12}$. The approach we choose is to insert here the result of 4.4. So modifying the equations 3.1, using

$$H_1 = \boldsymbol{x} - \boldsymbol{\mu} \quad , \quad H_2 = \boldsymbol{y} - \boldsymbol{\nu}$$

and also using $N$ as the numbers of samples (pixels) taken into account, we obtain

$$\widehat{\boldsymbol{\Sigma_{11}}} = \frac{E\{H_1(H_1 \odot \Pi)^T\}}{N-1} \qquad \in \mathbb{R}^{q \times q} \tag{4.5}$$

$$\widehat{\boldsymbol{\Sigma_{22}}} = \frac{E\{H_2(H_2 \odot \Pi)^T\}}{N-1} \qquad \in \mathbb{R}^{p \times p} \tag{4.6}$$

$$\widehat{\boldsymbol{\Sigma_{12}}} = \widehat{\boldsymbol{\Sigma_{21}^T}} = \frac{E\{H_1(H_2 \odot \Pi)^T\}}{N-1} \qquad \in \mathbb{R}^{q \times p} \tag{4.7}$$

where the $\odot$ stands for the Hadamard product (or element-wise multiplication), which does not change the dimensions of the matrices nor the order of the main eigenvalues, provided that the two matrices are positive-definite. $\boldsymbol{\Sigma}$ are positive semi-definite for construction, but to avoid zeroes in the computations of inverses, a small $\delta$ has been substituted when needed. The result of the CCA block are the optimal matrix projection, $\boldsymbol{U} = [\boldsymbol{u_1}, ..., \boldsymbol{u_L}]$ and $\boldsymbol{V} = [\boldsymbol{v_1}, ..., \boldsymbol{v_L}]$.

## 4.5 Deep Canonical Correlation Analysis with Autoencoders

### 4.5.1 The network topology

The chosen topology is similar to the one in [Luppino et al., 2020], and it is inspired by [Wang et al., 2015] for what concerns the CCA block. In our methodology we are interested in taking advantage of our prior information inside the just mentioned framework. As far as we know we are the first to deploy a DCCAE methodology in an unsupervised fashion. The architecture is composed of two autoencoders, coupled in a novel fashion through different losses computation. The four networks are Deep Convolutional Neural Networks, and they implement image regression functions. The encoders take images as input and they transform them in a common domain called $Z$; the functions are $R(\cdot) : \mathbb{R}^{H\times W\times C_1} \to \mathbb{R}^{H\times W\times C_z}$ and $P(\cdot) : \mathbb{R}^{H\times W\times C_2} \to \mathbb{R}^{H\times W\times C_z}$ so the image dimensions are preserved also in the latent space and the feature dimension is a common parameter. The decoders $S(\cdot)$ and $Q(\cdot)$ perform the inverse transformation, taking the images from the $\mathcal{Z}$ domain to the two original domains. Additionally, the CCA block performs a linear Correlation between the output of the two encoders, thus highlighting the most canonical correlated features and calculating the correlation itself for each feature. Figure 4.3 presents the network topology.

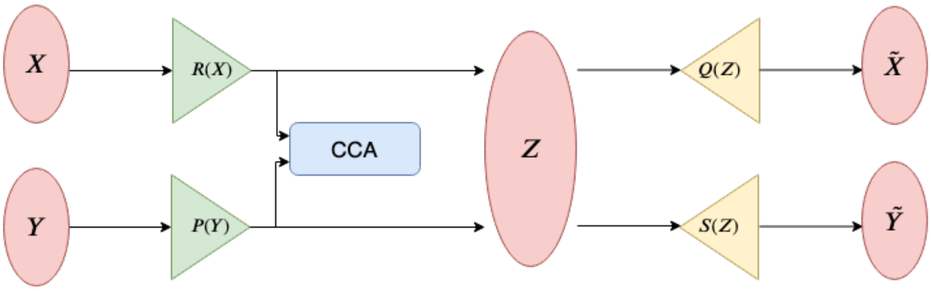


Figure 4.3: Network topology divided by colours: encoders in green, decoders in yellow, sets in red and the CCA block in blue.

### 4.5.2 Training and losses definition

The training phase of the network is crucial for the system itself, and has been studied in depth, in order to assure a fast and robust training. The training parameters are the network weights, defined in a vector called $\boldsymbol{\vartheta}$. The overall loss function has been designed ad hoc, and it consists of four loss terms with respective weights.

$$\mathcal{L}_{\text{tot}} = \lambda_{CCA} \cdot \mathcal{L}_{\text{CCA}} + \lambda_{\text{Recon}} \cdot \mathcal{L}_{\text{Recon}} + \lambda_{\alpha} \cdot \mathcal{L}_{\alpha} + \lambda_{\text{Cross}} \cdot \mathcal{L}_{\text{Cross}} \tag{4.8}$$

*Canonical Correlation.* The canonical correlation loss is computed on the output of the encoder, and the loss term is defined as follows (analogous but not identical to representation as 3.3).

$$\mathcal{L}_{\text{CCA}} = -\frac{1}{n} tr(\boldsymbol{U}^T R(\boldsymbol{x}) P(\boldsymbol{y})^T \boldsymbol{V}) \tag{4.9}$$

where $n$ is the total number of pixels in a patch, $U, V$ are the optimal transformation matrices, $\boldsymbol{x}, \boldsymbol{y}$ represents co-located patches of the respective images and $tr$ is the matrix trace. $\boldsymbol{U}, \boldsymbol{V}$ are now matrices, and no more vectors as explained in 3.3, because now $\boldsymbol{x}, \boldsymbol{y}$ are multi-channel images.

This term forces the two autoencoders to converge to the same latent space, which is the space where the correlation between the retrieved representations is maximised. It is possible to set the latent space dimension $C_z$ (feature dimensions) as big as desired, respecting the constraint

$$C_z \leq \max\left(C_1, C_2\right)$$

*Reconstruction of the input.* It is obvious, having autoencoders, we want to have our outputs as much similar to the inputs as possible; in other words, our reconstruction from the latent space should be as faithful as possible. Stating we would like to have

$$\boldsymbol{X} \simeq \tilde{\boldsymbol{X}} = Q\left(R\left(\boldsymbol{X}\right)\right)$$

and analogously for $\boldsymbol{Y}$. Recalling that $\boldsymbol{\vartheta}$ is the weight vector of the entire network, and calling $\boldsymbol{x}$ and $\boldsymbol{y}$ the vectors collecting the data of an image patch centered on the same pixel in the two image domains, the loss term is defined as

$$\begin{aligned} \mathcal{L}_{\text{Recon}}\left(\boldsymbol{\vartheta}\right) =& \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[\|Q\left(R\left(\boldsymbol{x}\right)\right) - \boldsymbol{x}\|_2^2\right] + \\ & \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[\|S\left(P\left(\boldsymbol{y}\right)\right) - \boldsymbol{y}\|_2^2\right] \end{aligned} \tag{4.10}$$

It is clear from 4.10 that the raw difference between input and output should be minimised. In Figure 4.4 it is illustrated the operation to obtain the parameters of the loss.

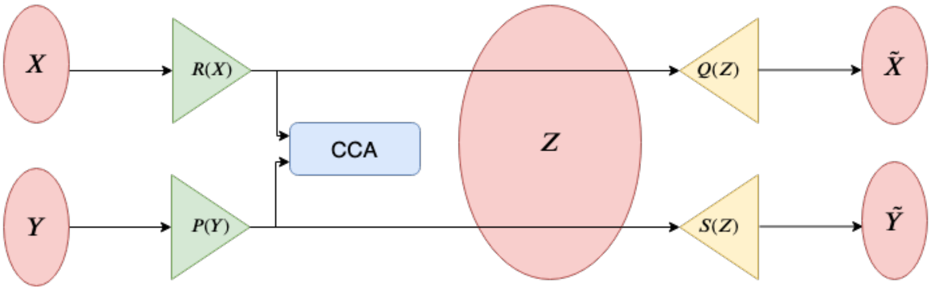


Figure 4.4: Reconstruction of the input. The terms used to compute the loss are at the right and left ends of the diagram. The contribution in the $\boldsymbol{Z}$ domain are not mixed, codes from $\boldsymbol{X}$ and $\boldsymbol{Y}$ are maintained separated.

*Prior weighted similarity.* This is one of the novelties that were recently proposed in [Luppino et al., 2020], it encapsulates the use of the prior information about the probability of each pixel of being changed in the translation of the images. In other words, we would like our network to learn the transformation from one domain to the other, so

$$\hat{\boldsymbol{X}} \simeq Q\left(P\left(\boldsymbol{Y}\right)\right)$$

must hold true. However, it is necessary that our network learns only from unchanged pixels, and so during the learning phase a correction term should be used, as stated in Equation 4.11. In order to define this loss, it is necessary to define the following notation:

$$\|\boldsymbol{a}\|_{\Pi}^{2} = \sum_i \Pi_i \, \|\boldsymbol{a}_i\|_2^2$$

where $\boldsymbol{a}_i$ is a generic feature vector representing the $i$-th pixel in a patch, its modulus is the sum squared of all the features (Euclidean metric). The weighting of $\Pi$ is applied pixel-wise on the pixel plane within the patch represented by a vector $\boldsymbol{a}$. In other words, $\|\boldsymbol{a}\|_{\Pi}^{2}$ is the modulus of $\boldsymbol{a}$, weighted on $\Pi$ pixel-wise.

$$\begin{aligned} \mathcal{L}_{\alpha}(\boldsymbol{\vartheta}) = & \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}} \left[ \|F(\boldsymbol{x}) - \boldsymbol{y}\|_{\Pi}^{2} \right] + \\ & \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}} \left[ \|G(\boldsymbol{y}) - \boldsymbol{x}\|_{\Pi}^{2} \right] \end{aligned} \tag{4.11}$$

where $F(\cdot) \triangleq S(R(\cdot))$ and $G(\cdot) \triangleq Q(P(\cdot))$. Figure 4.5 illustrates the network operations to obtain the loss parameters.

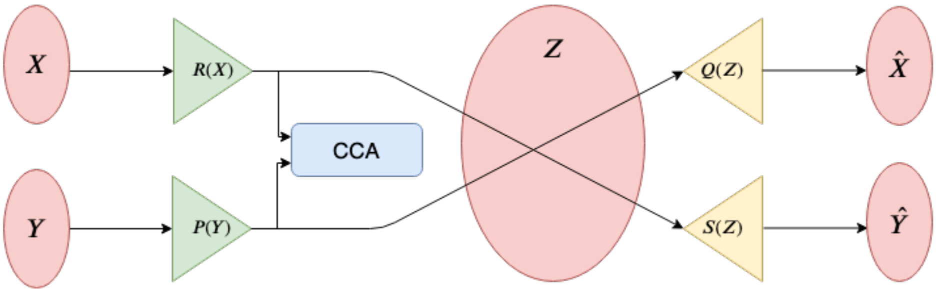


Figure 4.5: Prior weighted similarity. Contributions in the $\boldsymbol{Z}$ domain are cross-connected and weighted by $\Pi(\alpha)$ on the pixel plane.

*Consistency cycle.* As pointed out in [Zhu et al., 2017], domain translations should maintain consistency cyclically; it means that after the data have been transformed once, they can be re-transformed to their original domain without becoming meaningless or losing properties. If the regression functions are rightly tuned the following must hold

$$\boldsymbol{X} \simeq Q(P(\hat{\boldsymbol{Y}})) = Q(P(S(R(\boldsymbol{X}))))$$

and to force our network to maintain this alignment we introduced 4.12

$$\begin{aligned} \mathcal{L}_{\text{Cycle}}(\boldsymbol{\vartheta}) = & \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}} \left[ \|G(F(\boldsymbol{x})) - \boldsymbol{x}\|_{2}^{2} \right] + \\ & \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}} \left[ \|F(G(\boldsymbol{y})) - \boldsymbol{y}\|_{2}^{2} \right] \end{aligned} \tag{4.12}$$

Figure 4.6 illustrates the just mentioned concept

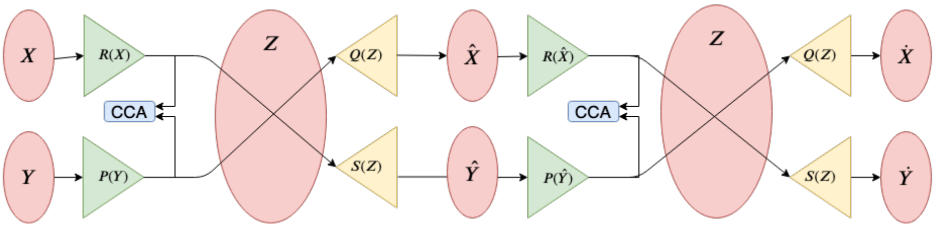


Figure 4.6: Consistency cycle. The cycle is like two prior-weighted similarities in cascade: it performs a double transformation on $\boldsymbol{X}$ and $\boldsymbol{Y}$.

The training procedure, as described in this paragraph, minimising the total loss follows the formula 4.13

$$\min_{\boldsymbol{\vartheta},\boldsymbol{U},\boldsymbol{V}} \quad \mathcal{L}_{\text{tot}} \tag{4.13}$$

s.t.

$$\begin{aligned}
&\boldsymbol{U}^T \left(\frac{1}{n} R(\boldsymbol{x})^T R(\boldsymbol{x}) + r_1 \boldsymbol{I}\right) \boldsymbol{U} = \boldsymbol{I} \\
&\boldsymbol{V}^T \left(\frac{1}{n} P(\boldsymbol{y})^T P(\boldsymbol{y}) + r_2 \boldsymbol{I}\right) \boldsymbol{V} = \boldsymbol{I} \\
&\boldsymbol{u_i}^T R(\boldsymbol{x}) P(\boldsymbol{y})^T + r_1 \boldsymbol{u_j} = 0, \quad \text{for} \quad i \neq j
\end{aligned}$$

where $r_1, r_2$ are regularisation parameters of the CCA.
The constraints have been taken into account into the CCA evaluation, and they assure to have uncorrelated directions inside each matrix projection; this leads to maximise the information kept in the transformed space. The expectations in the loss contributions 4.10, 4.11, and 4.12 are estimated as sample means on a random ensemble of fixed-size image patches drawn from the two image domains $\boldsymbol{X}$ and $\boldsymbol{Y}$.

### 4.5.3 The back-propagation

Backpropagation of the network is obvious, for what it concerns the Neural Networks strictly speaking, however, to minimise also with respect to $\alpha$, a manually written procedure have been added. It was required in order to use a gradient-based optimisation, as we have done in this thesis. Indeed, the gradient of $corr(H_1, H_2)$ is required, and the paper by [Andrew et al., 2013] has been followed. Demonstration of the 4.14 formula can be found in that paper.

$$\frac{\delta corr(H_1, H_2)}{\delta H_1} = \frac{1}{m-1}(2\nabla_{11}\bar{H}_1 + \nabla_{12}\bar{H}_2) \tag{4.14}$$

where

$$\nabla_{12} = {\Sigma_{11}}^{-\frac{1}{2}} U V^T {\Sigma_{22}}^{-\frac{1}{2}}$$
$$\nabla_{11} = -\frac{1}{2} {\Sigma_{11}}^{-\frac{1}{2}} U D V^T {\Sigma_{11}}^{-\frac{1}{2}}$$

where $H_1$ and $H_2$ are the transformed $X$ and $Y$ in the $Z$ domain, and $\bar{H_1} = H_1 - \frac{1}{m} H_1 \mathbf{1}$ is the centered data matrix (and respectively $\bar{H_2}$ for $H_2$). Instead, $UDV^T$ is the singular value decomposition of the matrix $T$, which is the solution of the canonical correlation.

### 4.5.4 Thresholding

The final result is obtained differentiating the transformed images and applying a suitable thresholding algorithm to them, in order to retrieve the two classes: changed, not-changed. The difference image is obtained as a weighted mean (on the number of channels) of $|\tilde{X} - \hat{X}|$ and $|\tilde{Y} - \hat{Y}|$. The algorithm used to detect the optimal threshold in an unsupervised way is the Otsu threshold algorithm [Otsu, 1979]. It is an iterative method which works on minimising intra-class variance, defined as a weighted sum of variances of the two classes, where the weights are the probability of fixing the threshold on a level. This exploits the fact that minimising intra-class variance is equal to maximising the inter-class variance, as stated in [Otsu, 1979].

### 4.5.5 Filtering

Lastly, to obtain a clear and meaningful result, a filtering algorithm is applied. It helps against spurious results, especially to filter out isolated pixels, i.e. pixels classified as changed in a contiguous and extended non-changed region or vice versa. Of course, this is done because it is very unlikely to have a single-pixel classified differently from all its neighbours.
The method proposed in [Krähenbühl et al., 2011] is used. It exploits spatial context to filter with a fully connected conditional random field model. It defines the pairwise edge potentials between all pairs of pixels in the image by a linear combination of Gaussian kernels in an arbitrary feature space. Iterative optimisation of the random field has one main downside: it requires the propagation of all the potentials across the image. However, this highly efficient algorithm reduces the computational complexity from quadratic to linear in the number of pixels by approximating the random field with a mean-field whose iterative update can be computed using Gaussian filtering in the feature space. The number of iterations and the kernel width of the

Gaussian kernels are the only hyper-parameters manually set, and we opted to tune them according to [Luppino et al., 2019]: 5 iterations and a kernel width of 0.1.

## 4.6 Extensions of the proposed approach

### 4.6.1 Linear CCA

The most simple alternative that has been investigated is the use of a simple linear Canonical Correlation Analysis, as it is. The procedure is: a) feed the images to the CCA; 2)CCA finds a representation where the correlation is maximised; 3)project the two images in the new space; 4)subtract one image from the other to obtain a difference image. This alternative is supposed to work if the two images under examination have some common or contiguous bands; this method is unlikely to work in a completely heterogeneous environment. This peculiarity is because the transformation applied by CCA is linear, so it can compensate for differences in certain data types, or extract the more similar feature from two images, but it is not able to perform a non-linear transformation.

### 4.6.2 Variations on Deep Canonical Correlation Analysis

Another interesting network explored is the Deep Canonical Correlation Analysis proposed by [Andrew et al., 2013]. We have of course modified the CCA to use our prior information as described above, and the network does not include any feedback, and the only loss considered is the maximisation of the canonical correlation of the two views. So, it is possible to say that the two networks have learned a non-linear type of CCA because the non-linearity shows up from the Neural network itself.

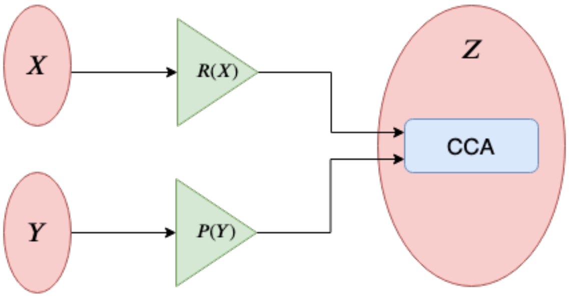


Figure 4.7: DCCA sketch: two encoders whose output are used to calculate the canonical correlation in the output/$\boldsymbol{Z}$ space.

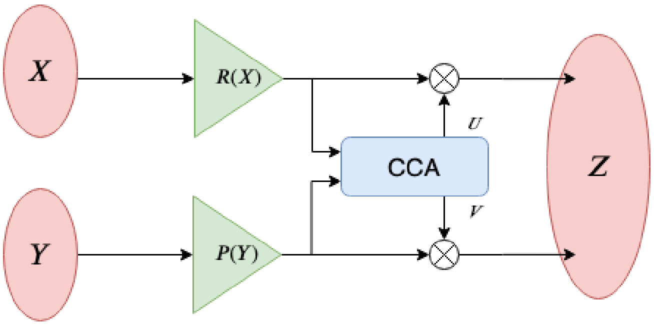


Figure 4.8: variation of DCCA: two encoders whose output are CCA-projected in a common space $\boldsymbol{Z}$.

### 4.6.3 Variation of DCCAE

A variation of DCCAE has been implemented for testing, its main architectures is proposed below.

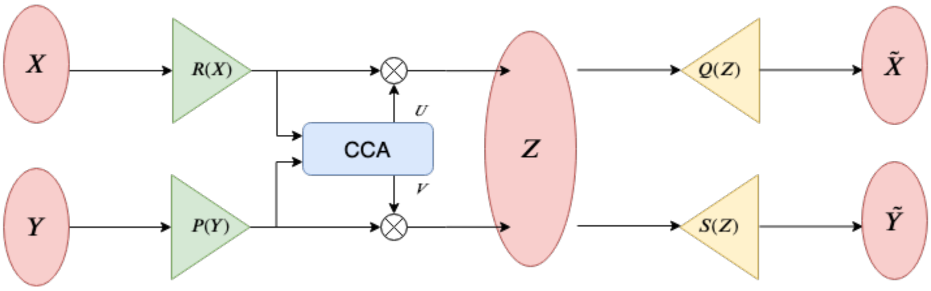


Figure 4.9: variation of DCCAE: the output of the encoders is CCA-projected in a common space with maximised correlation and then re-transformed back to the $\boldsymbol{X}$ and $\boldsymbol{Y}$ domains.

Figure 4.9 represents a very similar network to the previous, but the core difference here is that the projection of the encoder results along with the directions of maximum correlations. The operation is performed multiplying the encoder's output by $\boldsymbol{U}$ and $\boldsymbol{V}$, which are the optimal matrices for the projection. In theory, it is also possible to try this configuration with different dimensions for the latent space $\boldsymbol{Z}$.
Nevertheless, this alternative implementation has a drawback: the projection along the most correlated direction can change abruptly from one iteration to another, leaving encoders and decoders largely unpaired. This shortcoming happens because CCA is just a linear method looking for correlation; indeed, during the training of the network, the direction which maximises the CCA functional can change and does not let the network adapt to this change. This behaviour was straightforward to appreciate, because during the training, when the network was learning and losses were decreasing, suddenly a peak occurred. The peaks were related to this change in the CCA transformation.
Another approach which has been tried is to train encoders and decoders separately, or performing the CCA only once per epoch or similar; however, the limitation persists. So this network has been abandoned; however, it leaves an open question, i.e., finding a CCA-similar algorithm which changes slowly from one representation to the other.

### 4.6.4 DCCAE with latent space differentiation

The last architecture we would like to mention is a DCCAE (as in Section 4.5.2) with the differencing procedure for the change detection performed in the latent space $\mathcal{Z}$, where the information should be maximally correlated. However, even if it was built as in Figure 4.3 or 4.9, it has some limitations, as shown in preliminary tests: the code space is not assured to be always perfectly aligned. This deficiency is probably due to the weak constraints of the network in that point, which does not allow to have a robust and persistent representation in the $\mathcal{Z}$ domain.

# Chapter 5

# Experimental Results

## 5.1 Environment

All the experiments were run on a graphics processing units (GPU) server in the domain of the Machine Learning Group of the UiT. The code was implemented in Python and Tensorflow 2.0, and will be available on line soon.

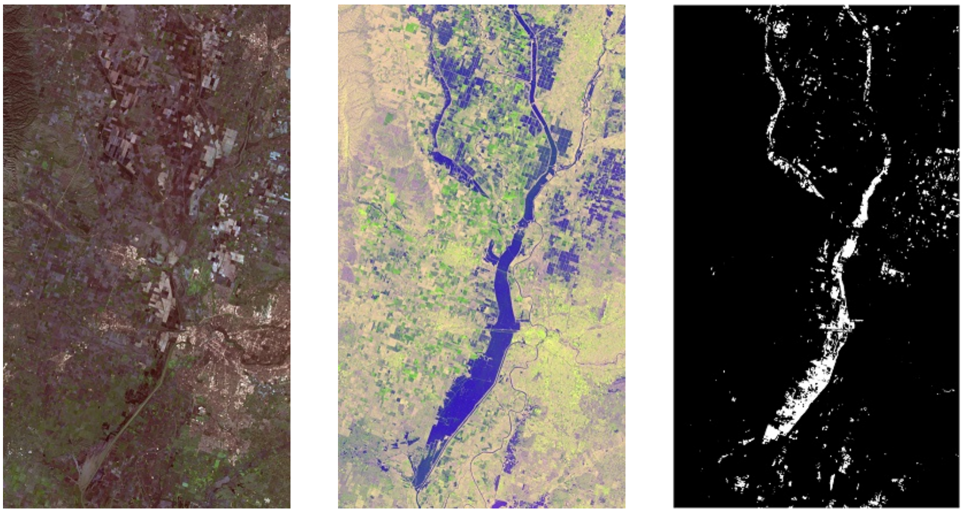
## 5.2 Datasets description

### Flood in California

Figure 5.1a displays the RGB channels of a Landsat 8 acquisition[1] of Sacramento County, Yuba County and Sutter County, California on 5 January 2017. The Operational Land Imager (OLI) and TIRS (Thermal InfraRed Sensor) sensors on Landsat 8 together acquire data in 11 channels, from deep blue up to thermal infrared.
This area was affected by a flood in Febraury of the same year, and the second acquisition, in Figure 5.1b, has been taken by Sentinel-1A[1] and recorded in polarisations VV and VH on 18 February 2017 by a single C-band SAR. The ratio between the two intensities is included both as the blue component of the false colour composite in 5.1b and as the third channel provided as input to the networks. All these SAR channels were log-transformed. The ground truth in Figure 5.1c has been provided by [Luppino et al., 2019] and

[1] Data processed by ESA, http://www.copernicus.eu/

was manually annotated. Originally these images were of 3500×2000 pixels, but they have been resampled to 850 × 500 pixels because of computational time constraints.



(a) Landsat 8 (b) Sentinel-1A (c) Ground truth

Figure 5.1: Flood in California. (a) Pre-event image taken by Landsat 8, RGB channels displayed. (b) Post-event image taken by Sentinel-1A, SAR channels in false colours. (c) Ground truth.

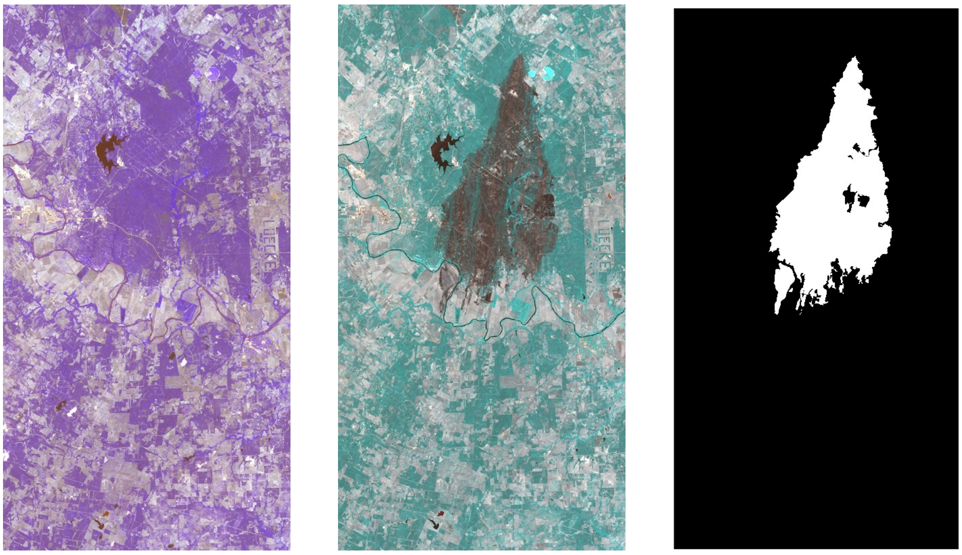
### Forest fire in Texas

Bastrop County in Texas was struck by a forest fire during September-October, 2011. The Landsat 5 Thematic Mapper (TM) acquired the image pre-event, a multispectral optical image with 7 bands. The Earth Observing-1 Advanced Land Imager (EO-1 ALI) acquired the post-event multispectral optical image with 10 bands. The resulting co-registered and cropped images of size 1520×800 are displayed in false colour in Figure 5.2a and Figure 5.2b[2]. Some of the spectral bands of the instruments (7 and 10 in total, respectively) overlap, so the signatures of the land covers involved are partly similar. Volpi *et al.* [Volpi et al., 2015] provided the ground truth shown in Figure 5.2c. Table 5.1 shows a comparison in the bands of the two instruments.

[2] Distributed by LP DAAC, http://lpdaac.usgs.gov

| Landsat-5 TM | | | Earth-Observing-1 ALI | | |
|---|---|---|---|---|---|
| band | $\lambda$ ($\mu$m) | GSD (m) | band | $\lambda$ ($\mu$m) | GSD (m) |
| - | - | - | MS-1’ | 0.433-0.453 | 30 |
| 1 | 0.45-0.52 | 30 | MS-1 | 0.450-0.515 | 30 |
| 2 | 0.52-0.60 | 30 | Ms-2 | 0.525-0.605 | 30 |
| - | - | - | PAN | 0.480-0.690 | 10 |
| 3 | 0.63-0.69 | 30 | MS-3 | 0.630-0.690 | 30 |
| 4 | 0.0.76-0.90 | 30 | MS-4 | 0.775-0.805 | 30 |
| - | - | - | MS4’ | 0.845-.890 | 30 |
| - | - | - | MS-5’ | 1.200-1.300 | 30 |
| 5 | 1.55-1.75 | 30 | MS-5 | 1.550-1.750 | 30 |
| 6 | 10.40-12.50 | 120 | - | - | - |
| 7 | 2.08-2.35 | 30 | MS-7 | 2.080-2.350 | 30 |

Table 5.1: bands of Landsat-5 TM and Earth-Observer 1 OLI. The bands partially overlap. $\lambda$ stands for the wavelength, and GSD for the Ground Sample Distance, which is related to the resolution on the ground.



(a) Landsat 5 ($t_1$)

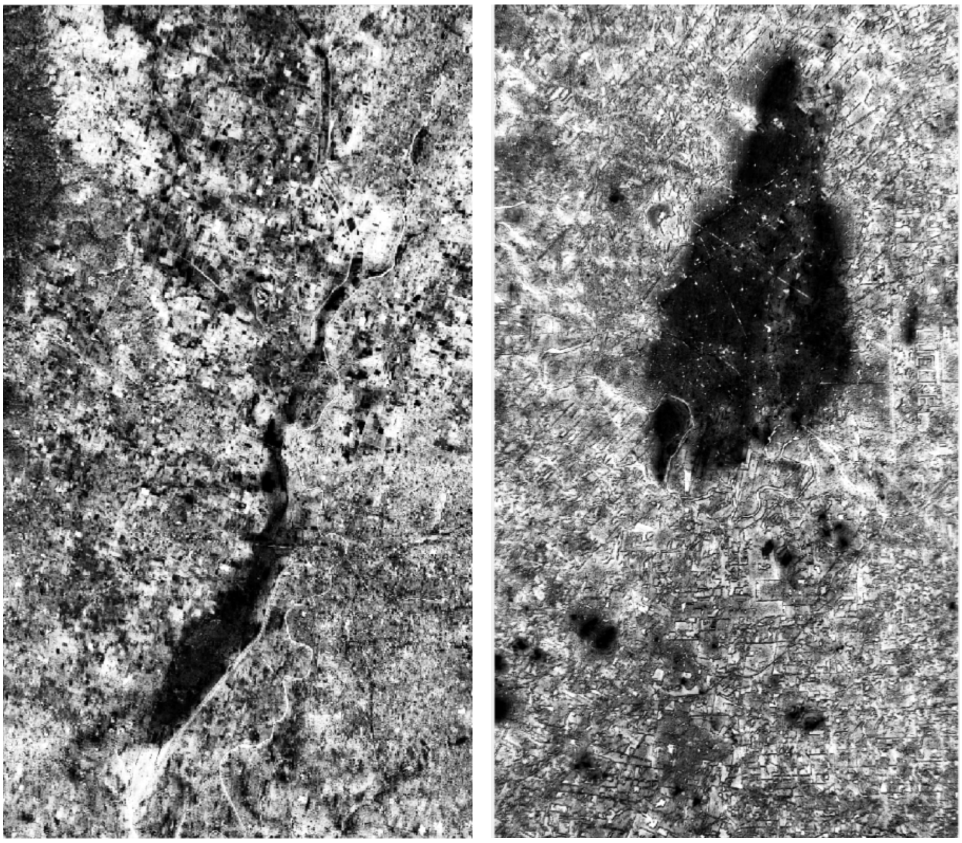
(b) EO-1 ALI ($t_2$)

(c) Ground truth

Figure 5.2: Forest fire in Texas. (a) Pre-event image taken by Landsat 5. (b) Post-event image taken by Earth Observing-1 ALI. (c) Ground truth.

## 5.3 Affinity-based Prior

The prior was computed off line, even if it is not a heavy task. The procedure has been explained in the previous chapter, here the results are presented in Figure 5.3: the greyscale (black to white) indicates increasing probability of not been changed. Having in mind the ground-truths above (Figures 5.2c,5.1c), it is possible to compare those to these images which represent this prior information: how much to trust each pixel in learning the transformation. However, the visual inspection may be misleading, because the histogram of the images have been stretched for visualisation; so the real prior has values which are far more similar (refer to Figures 5.4 and 5.5) than the illustrated ones. Nevertheless it is worth noting how -especially in the Texas one- the change is very well localised, and precisely identified. This leads to say that our prior information is very reliable, remembering the fact that it has been extracted only from the images.
The drawback of this prior is the big number of false alarms, considered as pixels not changed, and represented here in black. Considering the fact that the prior is used to learn a transformation, a more correct consideration may be to use only highly trustworthy pixels, even if they are a minority. However not every class, present in the image, can be represented by highly reliable pixels. Expressing the same concept in a more statistical sense, the prior procedure struggles to find a bi-modal representation of the data, were highly reliable pixels are far away from highly unreliable ones. And the fact that not every class is surely represented in one of the two modalities can be a weakness of the algorithm. It is noteworthy, from the visual inspection of the real histograms (not-stretched) in Figures 5.4 and 5.5 that the Texas data set shows a slightly bimodal distribution; it means two classes seem to be separable, changed and unchanged, and of course the changed are represented in black, which means values towards zero, and it is well shown in the histogram how this class is a minority. On the contrary, in the California prior histogram, from a visual inspection is not possible to infer the two classes, there is only a small asymmetry in the distribution towards the unchanged side.
However, this is a really useful and powerful prior information, and it is out of its scope to be used as a stand-alone procedure for change detection, nevertheless in some cases it can be useful for an immediate and visual feedback to locate and identify the changes.



(a) California dataset.

(b) Texas dataset.

Figure 5.3: Pixelwise prior for both datasets. Brighter pixels are less prone to be changed from the pre-event to the post-event image. The images were subject to histogram stretching for displaying purposes. (a) California dataset. (b) Texas dataset.

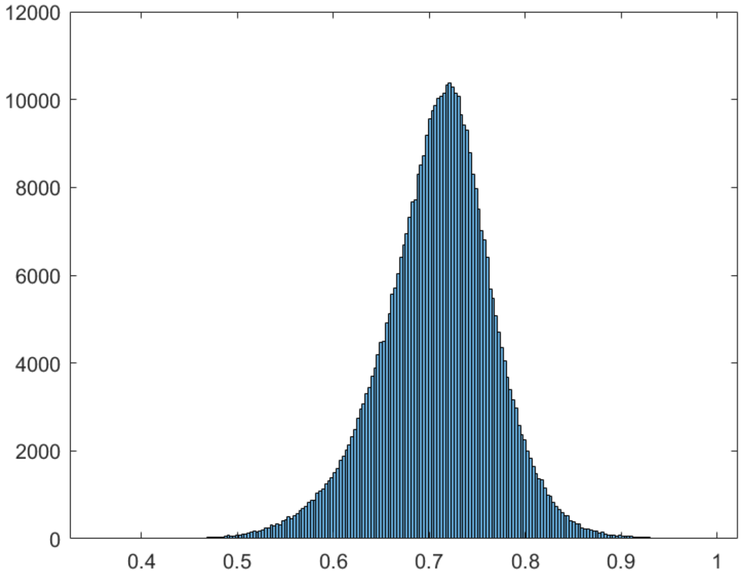


Figure 5.4: Histogram of the California Prior. Histogram (not-stretched) of pixels intensity of Figure 5.3a

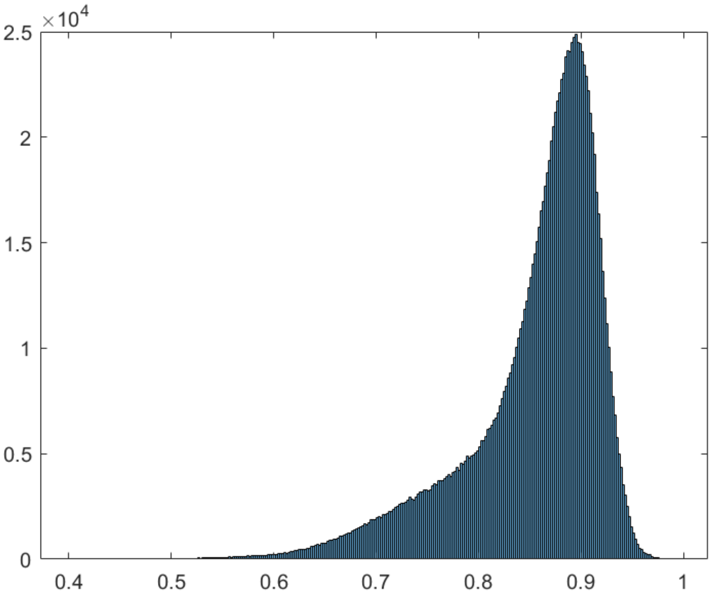
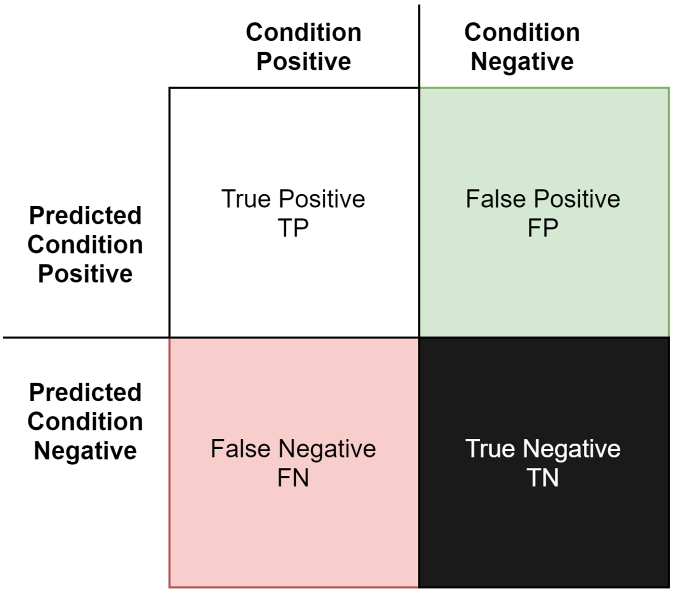


Figure 5.5: Histogram of the Texas Prior. Histogram (not-stretched) of pixels intensity of Figure 5.3b

## 5.4 Evaluation metric

To evaluate the performance of the proposed methods, the Cohen's Kappa ($\kappa$) was adopted along with other standard coefficients. The Cohen's Kappa $\kappa$ - first used by [Cohen, 1960] -is used to measure inter-rater reliability and it is considered more robust than a simple percent agreement calculation. However, there are some critics about its use; the controversy is due to the fact that is not symmetrical, and also about the difficulty of interpreting its value in some situations.
However, the $\kappa$ were adopted to measure the similarity between the retrieved solution of the change detection process and the ground-truth. They were considered as two distributions and compared. The comparison is done considering four classes: true positive, true negative, false positive and false negative. The correctly classified pixels belong to the first two classes, whereas the wrongly classified fall in the two others. The confusion matrix is used for the calculation and is constructed as shown in Figure 5.6.



| | **Condition Positive** | **Condition Negative** |
|---|---|---|
| **Predicted Condition Positive** | True Positive<br>TP | False Positive<br>FP |
| **Predicted Condition Negative** | False Negative<br>FN | True Negative<br>TN |

Figure 5.6: Confusion matrix used to compute Cohen's Kappa $\kappa$: the correctly classified pixels are indicated in white and black, the wrongly classified ones in green and red. The colors show the convention used in the confusion maps in the result section.

The $\kappa$ is calculated in the following way

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{5.1}$$

where

$$Pr(a) = \frac{TP + TN}{N} \qquad Pr(e) = \frac{(TP + FP)}{N}\frac{(FN + TN)}{N}$$

and $N$ is the total number of observations (or pixels). Thus $-1 \leq \kappa \leq 1$. This coefficient tries to remove from percent agreement the agreement by chance, assuring a more balanced judgement.

## 5.5 Results of the proposed DCCAE method

### 5.5.1 Settings

The developed architecture was composed of encoders and decoders, each made of two hidden layers, with 100 filters each. The chosen activation function was a Leaky ReLU with a negative slope chosen to be $\gamma = 0.3$.[Maas et al., 2013], except for the last layer of NN which was a fully convolutional layer with a $tanh(\cdot)$ activation function. Furthermore, the optimisation method chosen was batch gradient descent, with a decreasing learning rate with exponential decay. During the training, data augmentation (patch rotations and flipping) was applied; moreover a dropout procedure was applied as well with a dropout rate of $\beta = 0.2$, to increase generalisation capacity. During the optimisation, gradient clipping was set to 1, in order to have always a gradient value under a certain threshold, even if our it was quite huge.

The overall network was trained for 100 epochs, where each epoch was composed of 5 batches and each batch of 20 patches. Each square patch, in turn, was composed of 100 pixels per side, so in total $10^4$ pixels. All these dimensions were fixed in order to have a right amount of pixel to compute meaningful (and representative) sample covariances and having in mind the machine memory size constraints. The latent space dimension was fixed to 3, in order to be suitable for both datasets: the California data set has an image with only 3 features; and for the Texas data set it seemed a fair code space dimension, especially because of the Figure 5.18, which will be explained later on.
All the others setting were manually set with a trial and error procedure, according to both the datasets at our disposal. Recalling the loss function 4.8, the $\boldsymbol{\lambda}$ was defined as in Table 5.2: Regarding the last three parameters, $\lambda_{regularisation_1}$ and $\lambda_{regularisation_2}$ are the CCA regularisation coefficients to avoid zeroes in the matrices and are set as found in the literature [Wang et al., 2015]. Furthermore, $\lambda_{l2}$ is the regularisation parameter to prevent overfitting in the gradient descent method.

| Loss weights | |
|---|---|
| $\lambda_{CCA}$ | 0.01 |
| $\lambda_{Recon}$ | 1 |
| $\lambda_{\alpha}$ | 1 |
| $\lambda_{Cycle}$ | 1 |
| Regularisation terms | |
| $\lambda_{regularisation_1}$ | $10^{-4}$ |
| $\lambda_{regularisation_2}$ | $10^{-2}$ |
| $\lambda_{l2}$ | $0.5 \cdot 10^{-4}$ |

Table 5.2: Network parameters set through a trial and error procedure.

### 5.5.2 Results

The results of our experiments are shown in the following figures and tables. Data were taken running each experiment 100 times, it is not a huge sample, but enough to be representative. Results for the California dataset are presented in Figures 5.8, 5.9, 5.10; whereas for the Texas dataset there are Figures 5.11, 5.12, 5.13. More specifically, Tables 5.7a and 5.7b describe the $\kappa$ coefficient of our DCCAE compared to other networks which have been considered as the state of the art (our implementation of the SCCN [Liu et al., 2018] and cGAN [Niu et al., 2019]) and the two networks developed in [Luppino et al., 2020]. The graph is in form of boxes, which contain the 25 to 75 percentiles, whiskers extend to the 5 and 95 percentiles, and remaining data points are plotted as red +.

The two network taken as reference are the SCCN and the cGAN, a brief description of them is here provided. The SCCN works with two CNNs (pretrained with a deep belief network) which learn a representation for optical and SAR images in terms of common pixel-wise features, and directly comparing them obtains the change map. The cGAN, instead, is a method based on finding a common representation for SAR and optical images with a generative adversarial framework.

It is evident how the ACE-net reaches state-of-the-art performance, whereas X-Net performs slightly worse, but more stably in terms of the $\kappa$ variance. The proposed DCCAE network with the affinity prior, is labeled DCCAE and combines the best features from both the ACE-Net and the X-net, reaching a very high $\kappa$ as well as exhibiting small variance, which indicates a stable performance. It is worth noting that the seemingly accurate performance of the SCCN algorithm on the California dataset is a side-effect of degenerate behaviour, as explained in [Luppino et al., 2020]. Summing it up, the SCCN network - due to its simplicity (few parameters) - learns to recognise only the background in the first image, so presenting the image

with a big flood it detects the changes in that area.
The aim of the DCCAE network was, indeed, to achieve accurate peak performances as the ACE-net, but with a smaller variance, as the X-Net. From these two boxplots which help to summarise the behaviour of the network it is possible to state that the self-supervised DCCAE network outperforms the state-of-the art, achieving very accurate performances both for peak values in classification and for the stable behaviour highlighted by the small variance.

Moving on to another evaluation method adopted to assess the performances of the DCCAE network, confusion maps were built for both datasets as an immediate tool to visually evaluate the performances of the method. Recalling the legend of the confusion maps: true positives (white), true negatives (black), false positives (green) and false negatives (red).
Figure 5.8b shows the result obtained on the California dataset, which presents some false positives, especially around the contours, thus suggesting that a 'thinner' filter may lead to a further improvement in the results. The other confusion map, relative to the Texas dataset 5.11b, highlights an excellent localisation of the changes, and very little impact of false negatives, which are only on the contours of the changed area. An explanation takes into account the fact that the prior is calculated with a kernel which extrapolates a spatial context: on an edge, it is not obvious its response and its behaviour cannot be as thin as a pixel. Especially because natural damages and effects are not bounded by sharp line on the ground which divides burnt vegetation not-burnt.
It is also interesting to note how here the algorithm found two preserved areas inside the fire scar, and a line (bottom left of the scar) which can be a street or a small river. The linear CCA was not able to detect these unchanged zones, but the nonlinear method succeed to extract them.

For what concerns the false positives, we also recall that the ground truth of each data set is focused on the ground changes due the corresponding major event (a flood and a forest fire) that happened in between the two acquisitions, but does not acknowledge the possible presence of further changes (maybe a farmer mowed the lawn or did some agricultural job). These further changes, if present, could be detected by the considered methods and, if so, they would erroneously be considered false positives as compared to the ground truth map

In order to explore the results more in depth, Table 5.3 is proposed.It is worth mentioning the difference between the overall accuracy (OA) calculated as the number of pixels correctly classified over their total number and the $\kappa$ coefficient. This highlights also the fact that in the California dataset

| California | | | |
|---|---|---|---|
| | $\kappa$ | OA | Time |
| mean | 0.45 | 0.91 | 9 $[min]$ |
| standard deviation | 0.04 | 0.01 | 3 $[s]$ |
| max | 0.51 | 0.93 | 9.17 $[min]$ |
| Texas | | | |
| | $\kappa$ | OA | Time |
| mean | 0.83 | 0.97 | 18 $[min]$ |
| standard deviation | 0.12 | 0.02 | 3 $[s]$ |
| max | 0.91 | 0.98 | 18.4 $[min]$ |

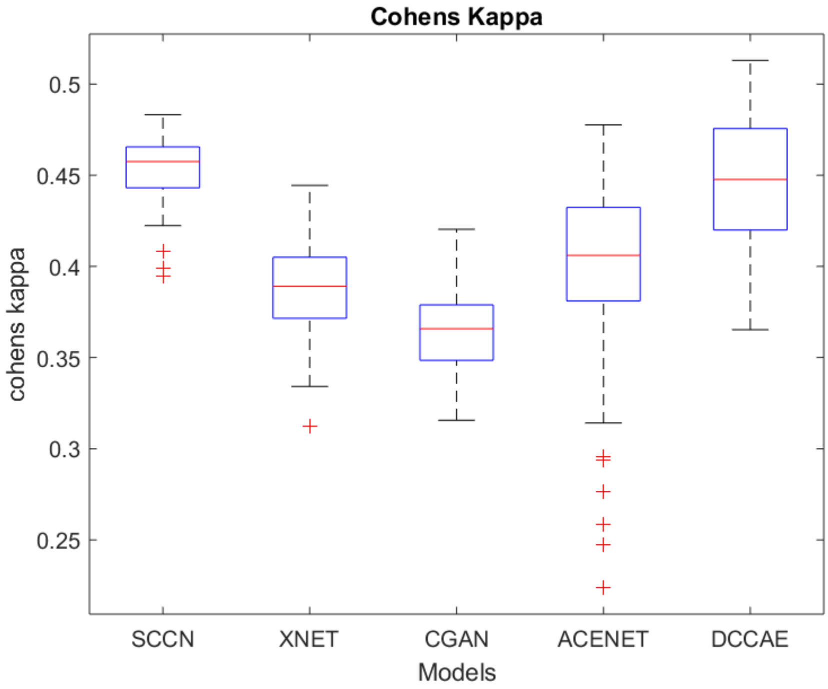
Table 5.3: Performances of the proposed method with both datasets. Each column represents a performance parameter, Cohen's Kappa $\kappa$, Overall Accuracy (OA), and the time elapsed to obtain a result on the dataset are listed.

we have a medium $\kappa$, however a big percentage of pixels are correctly classified.

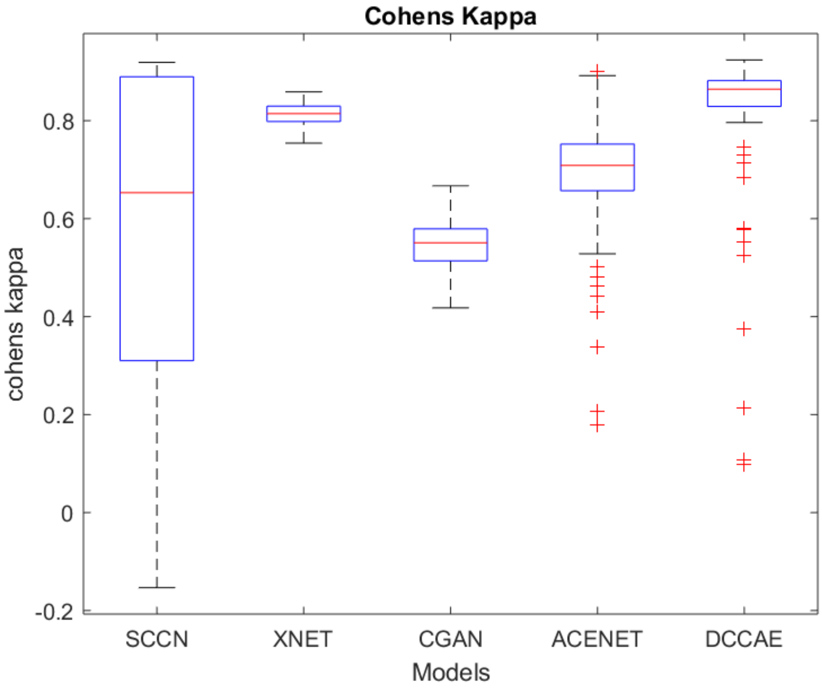
Let us move on discussing the intermediate results, that are the representations in the latent space and after the CCA correlation. The images are at the end of this section, and are shown in false colour because pixel intensities have no physical meanings in this domain.
It is interesting to note how images in the latent space $\mathcal{Z}$ - labelled as $X_{CODE}$ and $Y_{CODE}$ - are very well aligned and differences between images are significant also here. The alignment information is inferred looking at the colours, equal colour palette suggest the same domain of the images. These images were fed to the decoder to retrieve the final result.
Moving towards the CCA domain, we can appreciate the $X_{CCA}$ and $Y_{CCA}$ representations. They are not used inside the process, but only retrieved to check the behaviour of the network. Surely, they are more similar to each other than the $\mathcal{Z}$ representation. Maybe it is not significant here because we have taken the last iteration of the network; however, during the training, the evolution of the CCA space is far more aligned than the $\mathcal{Z}$ space, especially in the first $10, 20$ epochs. Even if, as already mentioned, the CCA space can change abruptly its representation, this is not a drawback because the network takes into account only the maximisation of the correlation information, which increases despite this nonlinear behaviour.

We consider now the last representation used in the network, which is also the core of its image differencing part: $\tilde{X}, \hat{X}, \tilde{Y}, \hat{Y}$. These representations

are explained in the previous chapter lay in the $\mathcal{X}$ or $\mathcal{Y}$ domain.
The interesting thing of these representations is that the changes are clearly visible in $\tilde{X}$ and $\hat{Y}$. This false asymmetry is easily understandable using the schemes provided in Section 4.5.2; $\tilde{X}$ is the reconstruction of $X$ , which is the image before the event, so it is the pre-event reconstruction, while $\hat{X}$ is the post-event image coming from the $\mathcal{Y}$ domain and transformed in the $\mathcal{X}$ domain. In a parallel way, the method works for $Y$, the only difference is that $\tilde{Y}, \hat{Y}$ exchanges their roles, because the $\mathcal{Y}$ domain contains the post-event image.

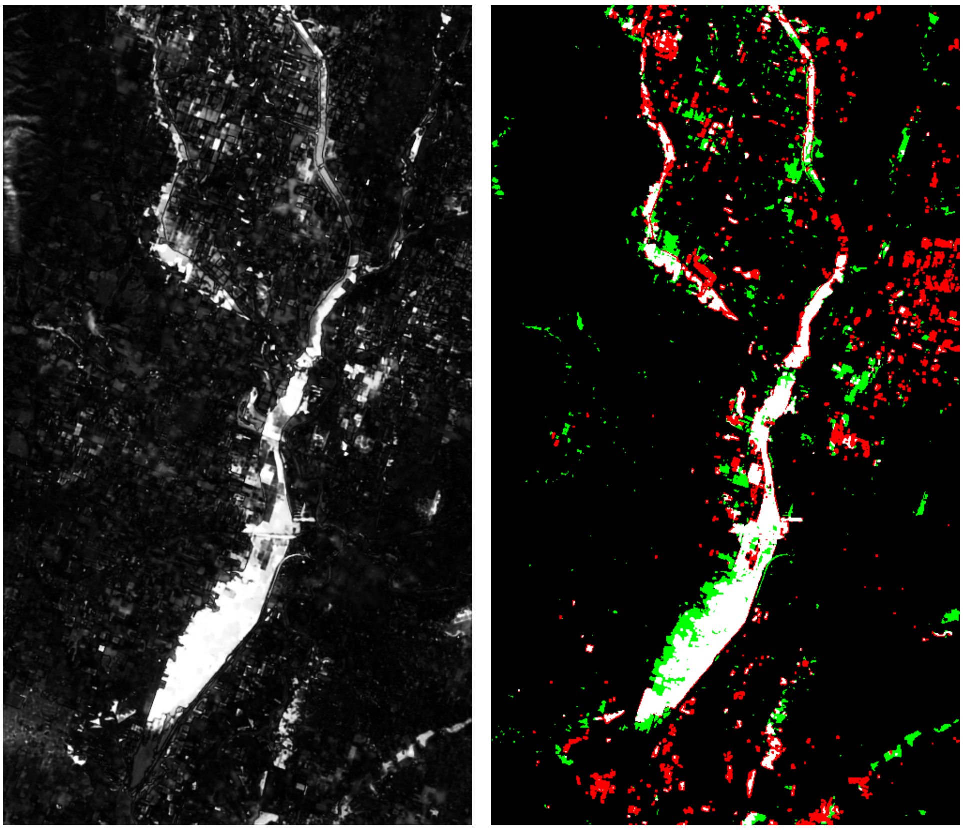


(a) California dataset.



(b) Texas dataset.

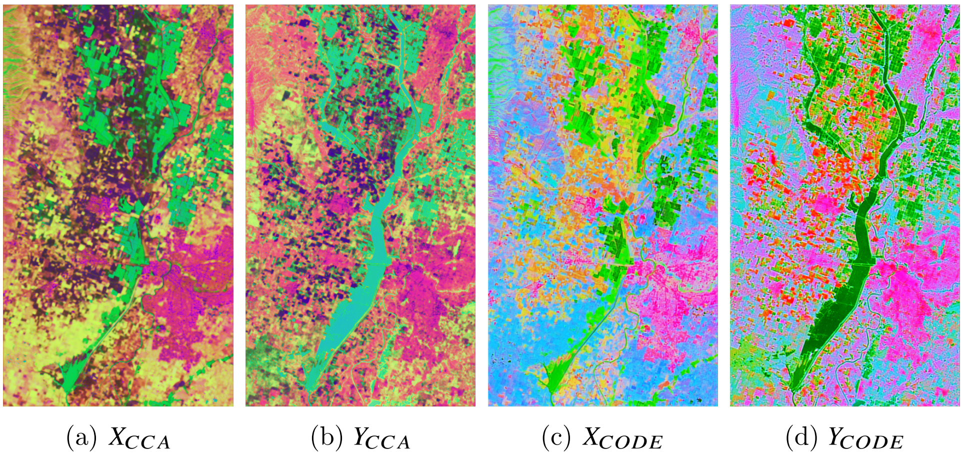
Figure 5.7: Comparison of Cohen's $\kappa$ for the proposed method (DCCAE, on the right) and the state of the art (SCCN, cGan and the [Luppino et al., 2020] models XNET, ACENET). The boxes contain the 25th to 75th percentiles, whiskers extend to the 5th and 95th percentiles, and the remaining data points are plotted as red +. (a) California dataset. (b) Texas dataset.



(a) Difference image

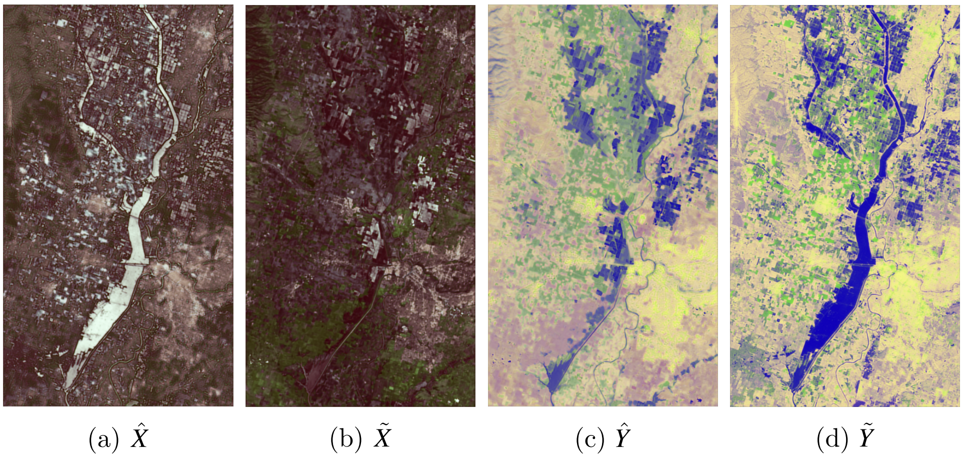
(b) Confusion map

Figure 5.8: California datatset. $\kappa = 0.51$. (a) Gaussian filtered difference image: black/darker pixels are expected to be unchanged; white(ish) pixels are likely changed. (b) Confusion Map: black and white are correctly classified not-changed and changed pixels; green are not-changed pixels wrongly classified as changed; red are changed pixels wrongly classified as not-changed.



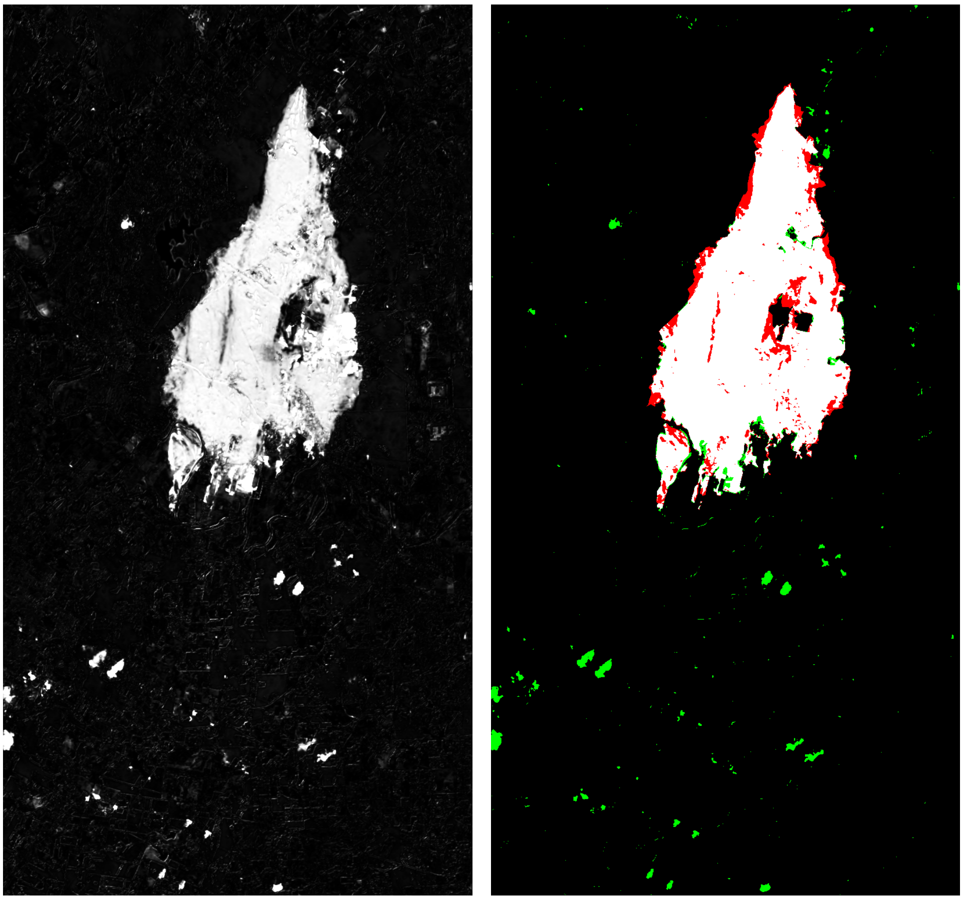
(a) $X_{CCA}$ (b) $Y_{CCA}$ (c) $X_{CODE}$ (d) $Y_{CODE}$

Figure 5.9: California dataset. (a)(b) representation of $X$ and $Y$ in the CCA aligned space, internal representation of the CCA, displayed only for investigation purposes. (c)(d)Code space representation of $X$ and $Y$, representation of the latent space of the network.



(a) $\hat{X}$ (b) $\tilde{X}$ (c) $\hat{Y}$ (d) $\tilde{Y}$

Figure 5.10: California dataset. Representation of the various output of the network. The difference image comes out from the mean of $\hat{X} - \tilde{X}$ with $\hat{Y} - \tilde{Y}$. (a)$\hat{X}$ output of the *prior weighted similarity*, derives from $Y$. (b)$\tilde{X}$ output of the *reconstruction of the input*, derives from $X$. (c)$\hat{Y}$ output of the *prior weighted similarity*, derives from $X$. (d)$\tilde{Y}$ output of the *reconstruction of the input*, derives from $Y$.



(a) Filtered difference image

(b) Confusion Map

Figure 5.11: Texas dataset. $\kappa = 0.90$. (a) Gaussian filtered difference image: black/darker pixels are expected to be unchanged; white(ish) pixels are likely changed. (b) Confusion Map: black and white are correctly classified not-changed and changed pixels; green are not-changed pixels wrongly classified as changed; red are changed pixels wrongly classified as not-changed.

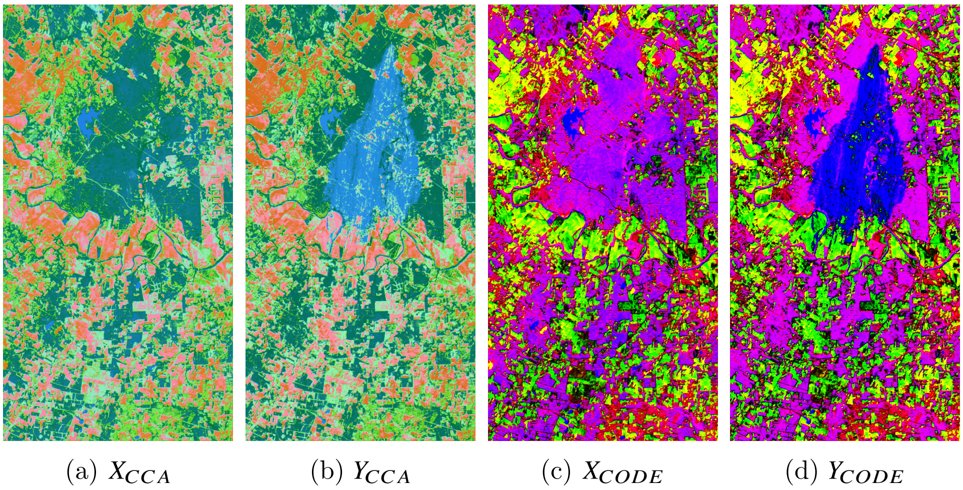

(a) $X_{CCA}$ (b) $Y_{CCA}$ (c) $X_{CODE}$ (d) $Y_{CODE}$

Figure 5.12: Texas dataset. (a)(b) representation of $X$ and $Y$ in the CCA aligned space, internal representation of the CCA, displayed only for investigation purposes. (c)(d)Code space representation of $X$ and $Y$, representation of the latent space of the network.

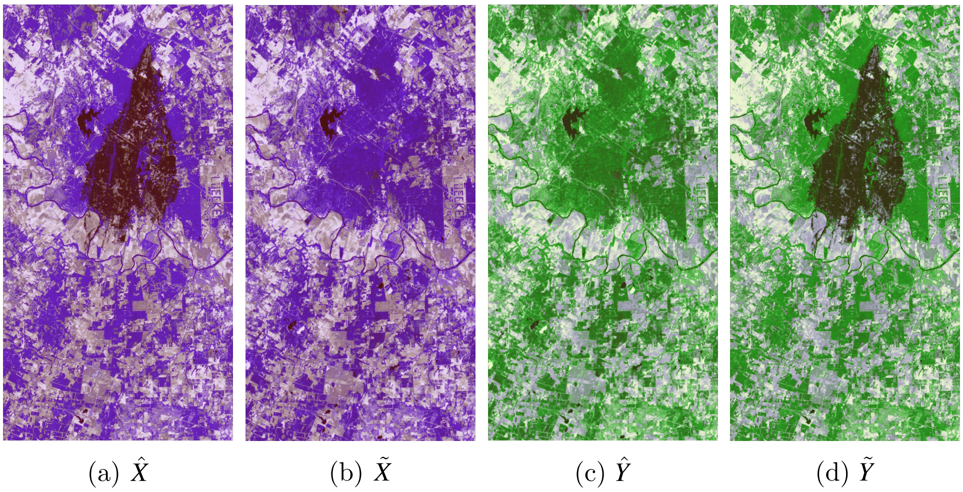
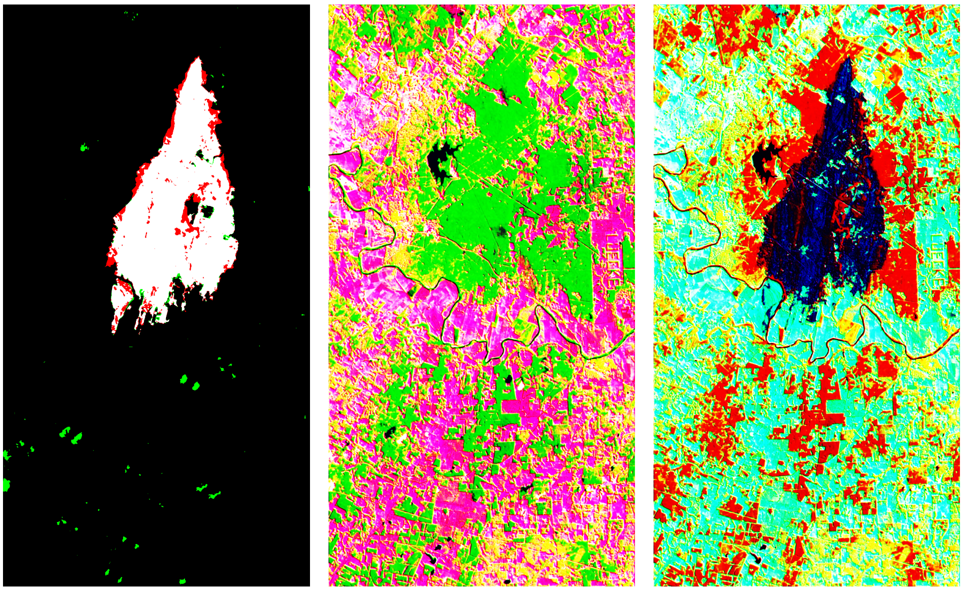

(a) $\hat{X}$ (b) $\tilde{X}$ (c) $\hat{Y}$ (d) $\tilde{Y}$

Figure 5.13: Texas dataset. Representation of the various output of the network. The difference image comes out from the mean of $\hat{X} - \tilde{X}$ with $\hat{Y} - \tilde{Y}$. (a)$\hat{X}$ output of the *prior weighted similarity*, derives from $Y$. (b)$\tilde{X}$ output of the *reconstruction of the input*, derives from $X$. (c)$\hat{Y}$ output of the *prior weighted similarity*, derives from $X$. (d)$\tilde{Y}$ output of the *reconstruction of the input*, derives from $Y$.

### 5.5.3 Comments

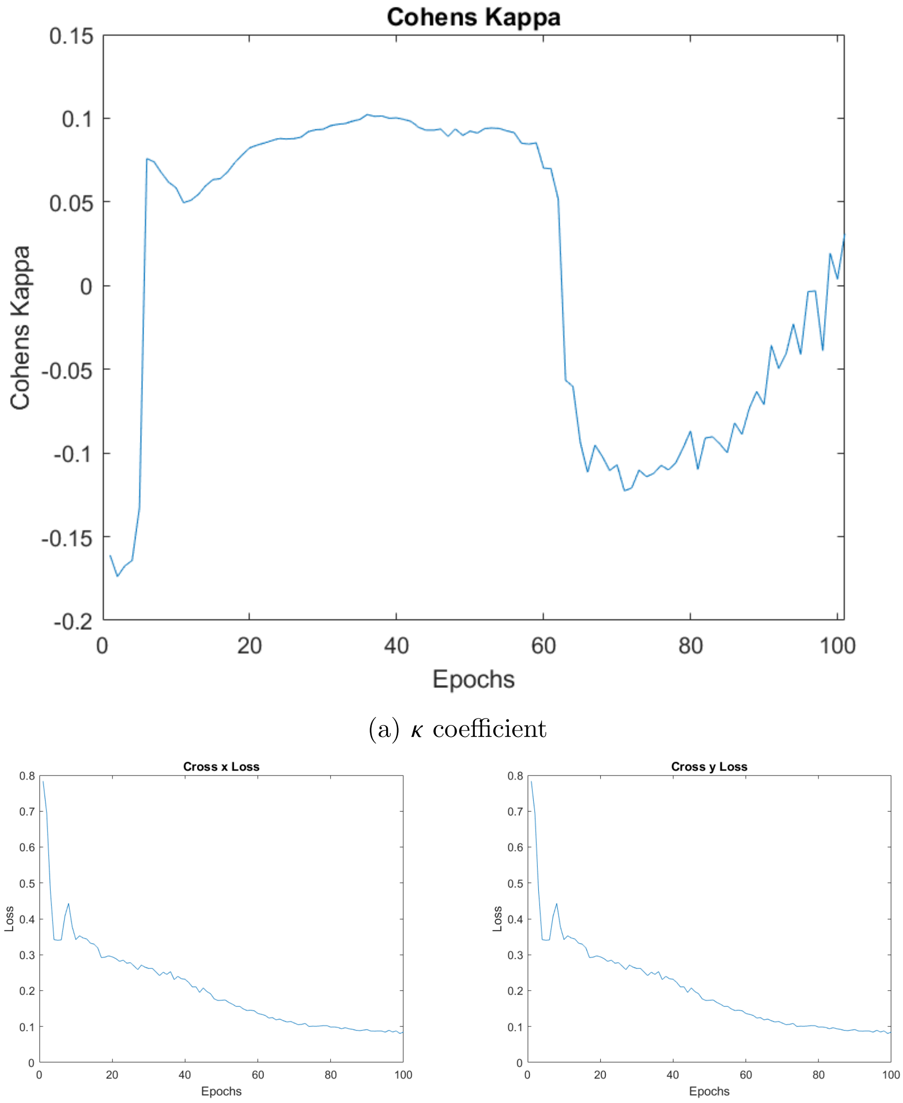
Some comments on our results and some proposal of improvement are here presented.
First of all the alternative implementation called *DCCAE with latent space differentiation* is not feasible due to the non-perfect alignment of the $\mathcal{Z}$ and CCA domains. This can take the reader to doubt also about the alignment needed to achieve good results in the DCCAE network, however the misalignment is not big enough to be an obstacle for the training of the couple of autoencoders. To demonstrate this statement, Figure 5.14, which illustrates a DCCAE result on the Texas dataset, is taken as an example; the change detection objective is reached, indeed, the confusion map is almost perfect. But, if it had been computed in the latent space, it would not have been as good; and it is possible to see it by a visual inspection of the images extracted from the latent space $\mathcal{Z}$. It is clear how Figures 5.14b and 5.14c, representing the exit of the encoders, are not aligned at all (the colour palette is different), and a subtraction between the two can not lead to a good result.



(a) Confusion map (b) $X_{CODE}$ (c) $Y_{CODE}$

Figure 5.14: California dataset. $\kappa = 0.91$. (a) Confusion map with $\kappa = 0.91$. (b)(c) are an example of failure of perfect alignment in the $\mathcal{Z}$ or $CODE$ domain, it is evident from the different palette of colour in use.

The second point to be highlighted of the method results is the presence, in Figure 5.7b, of many outliers, one which has a $\kappa \simeq 0$. In these cases, it is common that the algorithms does not learn how to reconstruct $\hat{X}$ from the change image $Y$, even if it is able to retrieve $\hat{Y}$ from $X$. This fact influences the cycle reconstruction and the network cannot learn a proper reconstruction. It is possible to see a small anomalies in the loss $\mathcal{L}_{\alpha}$, and then the network fatigue to recover or cannot recover at all. To better understand the problem it is possible to refer to the Figure 5.15, where 5.15b and 5.15c are the two components of the $\mathcal{L}_{\alpha}$ loss (see 4.11). It is possible to see the mentioned anomaly, which happens at epoch number 7. Moreover, looking for a similar pattern in the $\mathcal{L}_{Cycle}$, it has been found in epoch number 8 and 9, proving the fact that this small misalignment transverses the network invalidating the training process.



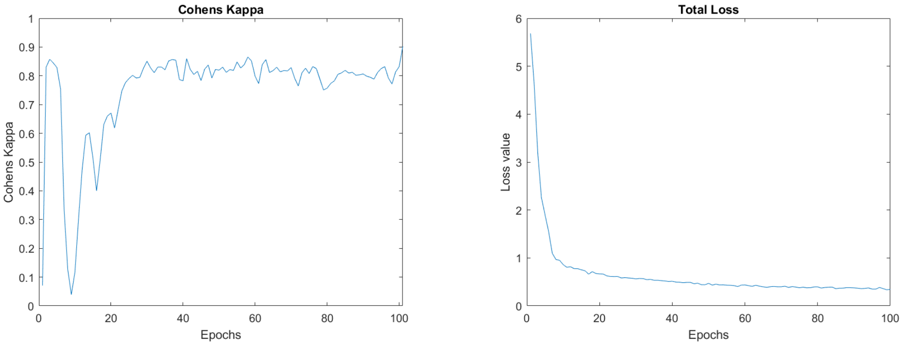
(a) $\kappa$ coefficient

(b) Cross-x loss function

(c) Cross-y loss function

Figure 5.15: Texas dataset. Behavior of the proposed DCCAE method as a function of the number of epochs. The three plots illustrate the training of the network along the epochs (on the x-axis). The abnormality peak in plots (b) and (c) is well evident. (a)$\kappa$ coefficient along the epochs. (b) Cross-x loss function, which is the *prior weighted similarity* from $X$ to $Y$. (c) Cross-x loss function, which is the *prior weighted similarity* from $Y$ to $X$.

At last, a comment on a good property of the DCCAE: even if the training goes on for many epochs, more than necessary, it does not need any early stopping criterion. Thus, highlighting the optimal stability gained through the cross balancing of the losses. When it reaches good performances, it is very unlikely it can decrease. Figure 5.16 provides an example, which shows the good learning capability in the first twenty-thirty epochs, and them it remain stable, without overfitting.



(a) $\kappa$ coefficient

(b) Loss function

Figure 5.16: Texas dataset. Plots of the behavior of $\kappa$ and the overall loss function of the proposed DCCAE method against the number of epochs. (a)$\kappa$ coefficient. (b)Total Loss function.

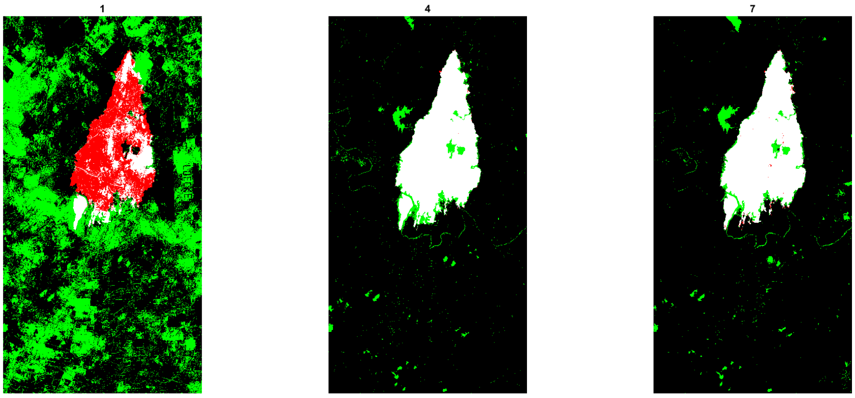
## 5.6 Linear CCA

This section provides result for the problem applying only the linear Canonical Correlation Analysis on the datasets as a benchmark result. It is obvious to foresee better result for the Texas dataset. This is expected because the Texas has data acquired with the same modality (optical), even if in different frequencies. This results in a heterogeneous change detection problem, but in a relatively simple version. The results for this dataset are shown in figures below, Figure 5.17.

The transformation is performed calculating all the correlations between

features, and taking the more correlated feature for each choice, as also displayed in the graph of Figure 5.19. It highlights that the first feature alone contain 50% of the total correlation, and the subsequent three the remaining part; the last three instead, does not contain any correlation information.
The different experiments shown in Figure 5.17 are performed giving as input the dimension of the transformed space. Thus, canonical correlation analysis was performed, and the two input images were projected in the space defined by the most correlated features. At last, the change map is retrieved by subtracting the two images (as in standard procedures) and thresholding the result.

It is worth noting how the performances increase by increasing the dimension of the space of projection. In theory, a feature space of dimension 1, as illustrated in the Figure 5.17a is enough to represent two classes, but not to extract the information for the classification, even if it is the one presenting the more correlation. In fact, from the table in Figure 5.18 we can see the $\kappa \simeq 0$, which indicates a nearly chance-equivalent. Instead, after using more than 3 features (the more correlated of the 7), the $\kappa$ value becomes $\simeq 0.9$.



(a) Dimension 1 (b) Dimension 4 (c) Dimension 7

Figure 5.17: Texas dataset. Confusion maps result of linear CCA performed with a different number of eigenvectors, so with transformed spaces of different dimensions. (a) Only one eigenvector is used for the correlation calculation and for the transformation, i.e., one transformed feature. (b) Four eigenvectors are used to compute the correlation and for the projection, i.e., four transformed features. (c) All the seven feature are used to compute the correlation and project into the space with seven features.

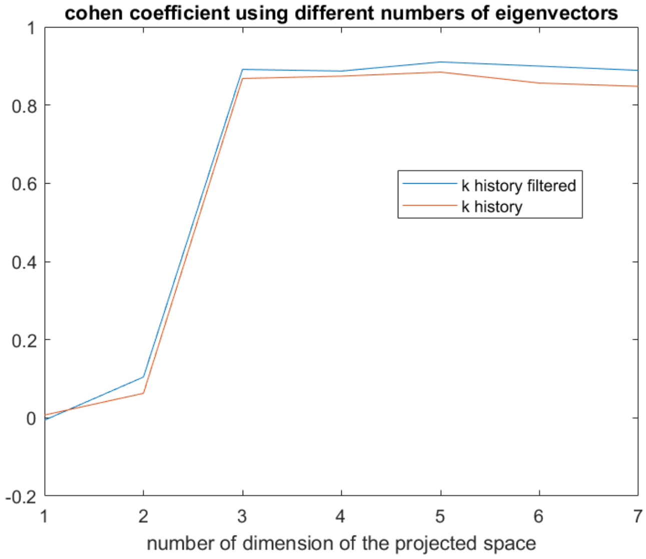


Figure 5.18: Texas dataset. Curve indicating the $K$ values against CCA methods applied with different dimensions of the transformed space (on the x-axis). With a projection to one or two transformed features the results are poor, with three features or more the results are quite accurate and stable.

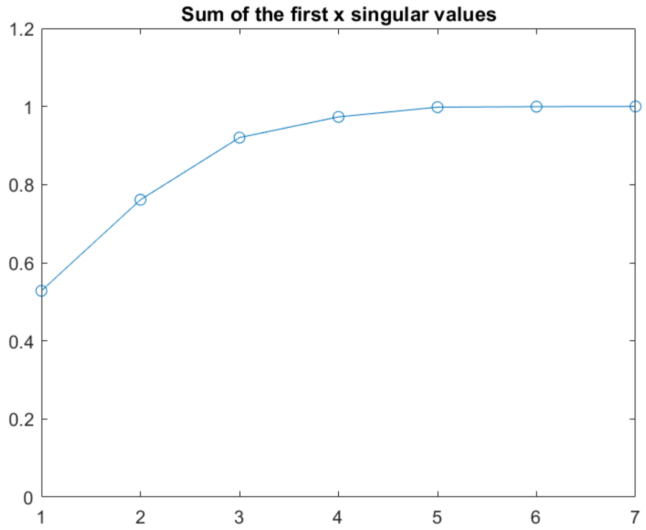
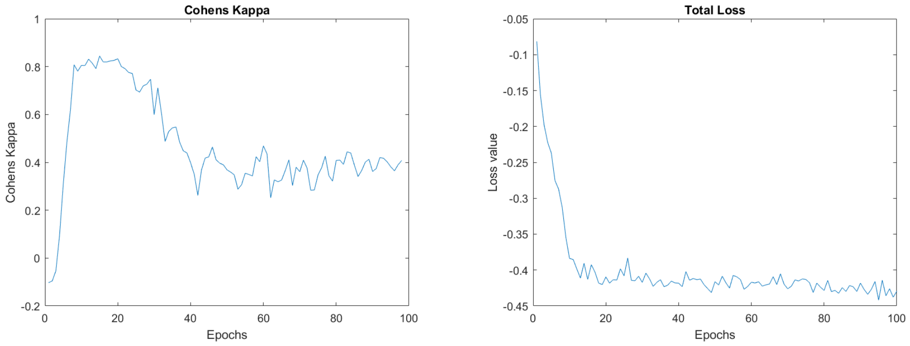


Figure 5.19: Texas dataset. CCA performed in a 7-dimensional space. On the y-axis and x-axis the cumulative sum of the eigenvalues and the number of features used are shown. The Texas dataset can be projected at most to seven transformed features (maximally correlated space), in fact the correlation saturates to 1 with seven features.

On the California dataset, which is more ”difficult”, and expected to be not solvable with a linear transformation, the linear CCA does not obtain anything more than $\kappa \simeq 0$, so we preferred to omit figures for brevity. In the following, where CCA will be mentioned, its behaviour is the same as explained here, because the same functional block is nested in the DCCAE architecture.

## 5.7 DCCA

The DCCA framework was explored and obtained rather accurate results up to applying an appropriate mechanism of early stopping. Only for completeness, the result on the Texas dataset in term of $\kappa$ is reported in Figure 5.20a. As it is possible to see, after 20 epochs the performances start to decrease. Furthermore, looking at the loss, the moment when the network is learning is easily visible, but going on with the learning leads to overfitting and can be identified quite precisely in the Cohen's Kappa graph.



(a) $\kappa$ coefficient

(b) Loss function

Figure 5.20: Texas dataset. DCCA method. The plots show the behavior of $\kappa$ and the overall loss function against the number of epochs. Large values of both $\kappa$ and the loss are obtained after few epochs. Then, around epoch 30, both abruptly decrease due to overfitting. (a) $\kappa$ coefficient with a peak of 0.84 at epoch 15. (b) Loss function.

# Chapter 6

# Conclusion

In the present thesis, the challenging problem of unsupervised change detection with heterogenous remotes sensing images has been addressed. After a first phase of detailed study of the related scientific literature, a novel approach rooted in deep learning has been developed. The proposed DCCAE network has been successfully implemented, and alternative configurations have been studied as well. Experimental results with two heterogeneous data sets involving three optical and one radar sensor have pointed out the capability of the method to achieve accurate detection of the changes regardless of the fundamentally different nature of the considered multitemporal acquisitions.

In particular, the goal to achieve accurate peak performances and small accuracy variance within the random ensemble of the output results, was achieved thanks to the proposed network topology in which the CCA acts as an alignment block for the training of a pair of autoencoders. The prior information, extracted through local affinity matrices and incorporated into the method, has revealed itself vital in order to formulate an unsupervised technique.

The results of the experiments are fully satisfactory, and the proposed method outperforms the state-of-the-art, which is based on deep architectures using generative adversarial and stacked autoencoder components, in the illustrated cases. It is worth recalling that these results were obtained consistently within the validation with two very different dataset, with different features and complexity, thus suggesting the flexibility of the proposed approach.

Some points of weakness of the developed methodological solution are the always present objections to deep neural networks: they require time to be fine-tuned, their behaviour generally depends on several hyper-parameters; we do not completely know how they work on a purely methodological standpoint, and this partial knowledge limits the protection capacity against a possible misbehaving. Furthermore, the computational power needed to train a deep network is quite big, yet nowadays easy affordable.

Further works should address an improvement of the prior information, in order to assure a bimodal distribution also with difficult datasets, as the California dataset proposed.
Moreover, another problem to address is the the presence of many outliers in the presented experiment on the Texas dataset, whose cause is not fully understood.

One final comment it about the training policy of the proposed method: for its formulation, there is the necessity to train the network every time a pair of images need to be used; it is not meant as a a network you can pretrain and use again and again. On one hand, this allows to have a network trained only for the specific task at hand and consistently optimized for it. On the other hand, the need for training on every input multitemporal pair of images may be inconvenient. Yet, the possible integration with incremental learning or with further domain adaptation concepts may reduce this training requirement in future extensions of the method.

# Bibliography

[Allen-Zhu et al. 2019] Allen-Zhu, Zeyuan ; Li, Yuanzhi ; Song, Zhao: A Convergence Theory for Deep Learning via Over-Parameterization. In: Chaudhuri, Kamalika (Hrsg.) ; Salakhutdinov, Ruslan (Hrsg.): *Proceedings of the 36th International Conference on Machine Learning* Bd. 97. Long Beach, California, USA : PMLR, 09–15 Jun 2019, S. 242–252

[Alpaydin 2014] Alpaydin, Ethem: *Introduction to machine learning.* MIT press, 2014

[Andrew et al. 2013] Andrew, Galen ; Arora, Raman ; Bilmes, Jeff ; Livescu, Karen: Deep canonical correlation analysis. In: *International conference on machine learning*, 2013, S. 1247–1255

[Bovolo and Bruzzone 2015] Bovolo, Francesca ; Bruzzone, Lorenzo: The Time Variable in Data Fusion: A Change Detection Perspective. In: *IEEE Geoscience and Remote Sensing Magazine* 3 (2015), 09, S. 8–26

[Cohen 1960] Cohen, J.: A coefficient of agreement for nominal scales. In: *Educational and Psychological Measurement* 20 (1960), S. 37–46

[Csáji 2001] Csáji, Balázs C.: *Approximation with Artificial Neural Networks*, Faculty of Sciences; Eötvös Loránd University, Hungary, Diplomarbeit, 2001

[Cybenko 1989] Cybenko, G: Approximation by superpositions of a sigmoidal function. In: *Mathematics of Control, Signals and Systems* 4 (1989), December, S. 303, 314. – URL `https://doi.org/10.1007/BF02551274`

[Figari Tomenotti et al. submitted] Figari Tomenotti, Federico ; Luppino, Luigi T. ; Hansen, Mads A. ; Moser, Gabriele ; Anfinsen, Stian N.: Heterogeneous Change Detection with Self-Supervised Deep Canonically Correlated Autoencoders. (submitted), January

[Fung and LeDrew 1987] FUNG, Tung ; LEDREW, Ellsworth: Application of principal components analysis to change detection. In: *Photogrammetric engineering and remote sensing* 53 (1987), Nr. 12, S. 1649–1658

[Goodfellow et al. 2016] GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron: *Deep Learning*. MIT Press, 2016. – `http://www.deeplearningbook.org`

[Govender et al. 2007] GOVENDER, Megandhren ; CHETTY, Kershani ; BULCOCK, Hartley: A review of hyperspectral remote sensing and its application in vegetation and water resource studies. In: *Water S.A* 33 (2007), 05

[Handcock et al. 2012] HANDCOCK, Rebecca ; TORGERSEN, Christian ; CHERKAUER, Keith ; GILLESPIE, Alan ; TOCKNER, Klement ; FAUX, Russel ; TAN, Jing: Thermal Infrared Remote Sensing of Water Temperature in Riverine Landscapes. In: *Fluvial Remote Sensing for Science and Management* (2012), 08, S. 85–113. ISBN 9781119940791

[Hardoon et al. 2004] HARDOON, David R. ; SZEDMAK, Sandor ; SHAWE-TAYLOR, John: Canonical correlation analysis: An overview with application to learning methods. In: *Neural computation* 16 (2004), Nr. 12, S. 2639–2664

[Inglada and Giros 2004] INGLADA, Jordi ; GIROS, Alain: On the real capabilities of remote sensing for disaster management-feedback from real cases. In: *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium* Bd. 2 IEEE (Veranst.), 2004, S. 1110–1112

[Krähenbühl et al. 2011] KRÄHENBÜHL, Philipp ; KOLTUN ; VLADLEN: Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In: SHAWE-TAYLOR, J. (Hrsg.) ; ZEMEL, R. S. (Hrsg.) ; BARTLETT, P. L. (Hrsg.) ; PEREIRA, F. (Hrsg.) ; WEINBERGER, K. Q. (Hrsg.): *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., 2011, S. 109–117

[Lavigne 1976] LAVIGNE, D. M.: Counting Harp Seals with ultra-violet photography. In: *Polar Record* 18 (1976), Nr. 114, S. 269–277

[Liu et al. 2018] LIU, J. ; GONG, M. ; QIN, K. ; ZHANG, P.: A Deep Convolutional Coupling Network for Change Detection Based on Heterogeneous Optical and Radar Images. In: *IEEE Trans. Neural Netw. Learn. Syst.* 29 (2018), March, Nr. 3, S. 545–559

[Lu et al. 2017] LU, Zhou ; PU, Hongming ; WANG, Feicheng ; HU, Zhiqiang ; WANG, Liwei: The Expressive Power of Neural Networks: A View from the Width. In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ;

Bengio, S. (Hrsg.) ; Wallach, H. (Hrsg.) ; Fergus, R. (Hrsg.) ; Vishwanathan, S. (Hrsg.) ; Garnett, R. (Hrsg.): *Advances in Neural Information Processing Systems 30.* Curran Associates, Inc., 2017, S. 6231–6239

[Luppino et al. 2019] Luppino, Luigi T. ; Bianchi, Filippo M. ; Moser, Gabriele ; Anfinsen, Stian N.: Unsupervised Image Regression for Heterogeneous Change Detection. In: *IEEE Trans. Geosci. Remote Sens.* 57 (2019), Nr. 12, S. 9960–9975

[Luppino et al. 2020] Luppino, Luigi T. ; Kampffmeyer, Michael C. ; Bianchi, Filippo M. ; Jenssen, Robert ; Moser, Gabriele ; Serpico, Sebastiano B. ; Anfinsen, Stian N.: *Deep image translation with an affinity-based change prior for unsupervised multimodal change detection.* Oct 2020. – arXiv:2001.04271

[Maas et al. 2013] Maas, Andrew L. ; Hannun, Awni Y. ; Ng, Andrew Y.: Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml* Bd. 30, 2013, S. 3

[Mardia et al. 1979] Mardia, Kantilal Varichand ; Kent, John T. ; Bibby, John M.: *Multivariate analysis.* London [u.a.] : Acad. Press, 1979 (Probability and mathematical statistics). – ISBN 0124712509

[Mercier et al. 2008] Mercier, G. ; Moser, G. ; Serpico, S. B.: Conditional Copulas for Change Detection in Heterogeneous Remote Sensing Images. In: *IEEE Transactions on Geoscience and Remote Sensing* 46 (2008), May, Nr. 5, S. 1428–1441. – ISSN 1558-0644

[Minnett et al. 2019] Minnett, P.J. ; Alvera-Azcárate, A. ; Chin, T.M. ; Corlett, G.K. ; Gentemann, C.L. ; Karagali, I. ; Li, X. ; Marsouin, A. ; Marullo, S. ; Maturi, E. ; Santoleri, R. ; Picart, S. S. ; Steele, M. ; Vazquez-Cuervo, J.: Half a century of satellite remote sensing of sea-surface temperature. In: *Remote Sensing of Environment* 233 (2019), S. 111366. – URL `http://www.sciencedirect.com/science/article/pii/S0034425719303852`. – ISSN 0034-4257

[Moser and Serpico 2006] Moser, G. ; Serpico, S. B.: Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery. In: *IEEE Transactions on Geoscience and Remote Sensing* 44 (2006), Oct, Nr. 10, S. 2972–2982. – ISSN 1558-0644

[Moser et al. 2012] Moser, Gabriele ; Serpico, Sebastiano B. ; Benediktsson, Jon A.: Land-cover mapping by Markov modeling of spatial–contextual information in very-high-resolution remote sensing images. In: *Proceedings of the IEEE* 101 (2012), Nr. 3, S. 631–651

[NASA ] NASA: *Missions - Climate Observation.* – URL `https://climate.nasa.gov/nasa_science/missions`. – website visited 26/02/2020

[Niu et al. 2019] Niu, X. ; Gong, M. ; Zhan, T. ; Yang, Y.: A Conditional Adversarial Network for Change Detection in Heterogeneous Images. In: *IEEE Geosci. Remote Sens. Lett.* 16 (2019), Jan, Nr. 1, S. 45–49

[Otsu 1979] Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1979), Jan, Nr. 1, S. 62–66. – ISSN 2168-2909

[Racah et al. 2016] Racah, Evan ; Beckham, Christopher ; Maharaj, Tegan ; Kahou, Samira E. ; Prabhat ; Pal, Christopher J.: ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In: *NIPS*, 2016, S. –

[Volpi 2013] Volpi, Michele: *Kernel-based methods for change detection in remote sensing images*, Faculte des Geosciences et de l'Environnement, Dissertation, 2013

[Volpi et al. 2015] Volpi, Michele ; Camps-Valls, Gustau ; Tuia, Devis: Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 107 (2015), 03

[Wang et al. 2015] Wang, Weiran ; Arora, Raman ; Livescu, Karen ; Bilmes, Jeff: On deep multi-view representation learning. In: *International Conference on Machine Learning*, 2015, S. 1083–1092

[Zhan et al. 2018] Zhan, Tao ; Gong, Maoguo ; Jiang, Xiangming ; Li, Shuwei: Log-Based Transformation Feature Learning for Change Detection in Heterogeneous Images. In: *IEEE Geosci. Remote Sens. Lett.* 15 (2018), Nr. 9, S. 1352–1356

[Zhou et al. 2019] Zhou, Yuan ; Liu, Hui ; Li, Dan ; Cao, Hai ; Yang, Jing ; Li, Zizi: *Cross-Sensor Image Change Detection Based on Deep Canonically Correlated Autoencoders.* S. 251–257. In: *Artificial Intelligence for Communications and Networks*, 07 2019. – ISBN 978-3-030-22967-2

[Zhu et al. 2017] Zhu, Jun-Yan ; Park, Taesung ; Isola, Phillip ; Efros, Alexei A.: Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In: *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, S. 2223–2232