



Land Cover Change Detection Using Deep Learning

A Thesis Presented

by

Abiy Getaneh Tibebu

to

The Faculty of Informatics

of

St. Mary's University

**In Partial Fulfillment of the Requirements
for the Degree of Master of Science**

in

Computer Science

August 03, 2024

ACCEPTANCE

Land Cover Change Detection Using Deep Learning

By

Abiy Getaneh Tibebu

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

Internal Examiner

Minale Ashagrie Abebe (PhD)



7/10/2024

External Examiner

Dean, Faculty of Informatics

June 28, 2020

Acknowledgment

First and foremost, I would like to express my heartfelt gratitude to my mother, sister, late grandmother, and my dad. The past couple of years have been challenging, but the unwavering strength and support from my mom and sister have been instrumental in helping me reach this milestone. Thank you for everything, and I will strive to make you proud.

I am deeply thankful to Dr. Mulugeta Adibaru, my thesis adviser, for his patience, constant guidance, and support throughout my thesis work. His insightful and constructive feedback has been invaluable during my research.

Lastly, I extend my deepest gratitude to my dear family and co-workers for their continuous support and the significant roles they have played on my behalf.

Table of Content

Acknowledgment	i
List of Figures	v
List of Table	vi
List of Equation	vii
List of Acronyms.....	viii
Abstract.....	ix
Chapter One	1
1. Introduction.....	1
1.1. Background of the study	1
1.2. Statement of the Problem	3
1.3. Research Question.....	5
1.4. Objective	5
1.4.1. General Objective	5
1.4.2. Specific Objective.....	5
1.5. Scope of the study	6
1.6. Limitations of the study.....	6
1.7. Significance of the study	7
1.8. Methodology of the study	7
1.8.1. Research design	7
1.8.2. Data collocation	8
1.8.3. Study Area	8
1.8.4. Development tools	9
1.8.5. Evaluation metric.....	11
Chapter Two.....	13
2. Literature Review.....	13
2.1. Change Detection	13
2.2. Remote Sensing.....	16
2.3. Unsupervised Learning	17
2.4. Deep Learning	19
2.5. Related Works	20
Chapter Three.....	24
3. Modeling for Land Cover Change Detection Using Deep learning	24

3.1.	Proposed Model.....	24
3.2.	Image Preprocessing	26
3.3.	Image Acquisition	26
3.4.	Co-register Image.....	29
3.5.	Resample Image	35
3.6.	Image Patch/Tiles.....	35
3.7.	Neural Network	39
3.8.	Feature Extraction	39
3.8.1.	Autoencoder (AE).....	39
3.9.	Encoder.....	41
3.9.1.	Convolutional Neural network (CNN)	41
3.9.2.	Convolutional Layer	42
3.9.3.	Pooling Layer	43
3.9.10.	Activation Function	43
3.10.	Decoder.....	44
3.10.10.	Deconvolution layer.....	44
3.11.	Loss function	45
3.12.	Euclidean distance	45
3.13.	Binary Change Map.....	46
Chapter Four	47
4.	Result	47
4.1	Overview	47
4.2.	Summary of Dataset Description	47
4.3.	Result.....	48
4.3.1.	Precision Analysis	50
4.3.2.	Recall Analysis.....	52
4.3.3.	F1-Score Analysis.....	54
4.3.4.	Accuracy Analysis.....	55
4.3.5.	MSE Analysis.....	57
4.4.	Discussion on the results	58
Chapter Five	60
5.	Conclusion and Recommendation	60

5.2. Conclusion..... 60
5.3. Recommendation..... 62
Reference..... 63
APPENDIX..... 70

List of Figures

<i>Figure 1: Study area</i>	8
<i>Figure 2: Electromagnetic spectrum</i>	17
<i>Figure 3: The Proposed Architecture</i>	25
<i>Figure 4: Vertical (A) Oblique (B)</i>	27
<i>Figure 5: Example of vertical and Oblique aerial photograph</i>	28
<i>Figure 6: Vertical and Oblique aerial photograph of Imperial Square Adey Abeba stadium taken in 2018</i>	28
<i>Figure 7: Vertical aerial photograph of Imperial Square Adey Abeba stadium taken in 2021</i> ...	29
<i>Figure 8 : Aerial photo of Addis Abeba Coverage 2018</i>	31
<i>Figure 9: Aerial photo of Addis Abeba Coverage 2021</i>	32
<i>Figure 10: Figure 0 12: Cropped image based on administrative boundaries yeka subcity worda 10</i>	36
<i>Figure 11: An illustration of image tile</i>	37
<i>Figure 12: Sample of co-registered multitemporal aerial photographs</i>	38
<i>Figure 13: Contrasting an artificial neuron with a biological neuron</i>	39
<i>Figure 15: The architecture of the conventional autoencoder and reconstructed image</i>	40
<i>Figure 16: CAE architecture showing convolutional and deconvolutional layers</i>	40
<i>Figure 17: An illustration of a convolutional neural network with several pooling and convolution layers</i>	42
<i>Figure 18: The convolution and subsampling process</i>	43
<i>Figure 19: ReLU activation function graph</i>	44
<i>Figure 20: The performance of the CAE</i>	48
<i>Figure 21: Graph represent of the result</i>	49

List of Table

Table 1. Summary of Reviewed Literature..... 22

Table 2:Description of the total number of image tiles 37

Table 3:Labels for the figures..... 49

List of Equation

Equation 1: Accuracy equation.....	12
Equation 2: Precision equation	12
Equation 3: Recall equation	12
Equation 4: F1 equation	12
Equation 5 : MSE Equation	45
Equation 6: Euclidean Distance Equation.....	45

List of Acronyms

AE	Autoencoder
API	Application Programming Interface
CAE	Convolutional autoencoders
CNN	Convolutional Neural Networks
CRGE	Climate Resilient Green Economy
DL	Deep learning
EF	Early Fusion
GAN	Generative Adversarial Network
Gdal	Geospatial Data Abstraction Library
GHG	Greenhouse Gases
GLI	Green Legacy Initiative (GLI)
LSTM	Long Short-Term Memory
LULC	Land-use and land-cover
OA	Overall accuracy
RNN	Recurrent Neural Networks
Siam	Siamese
SSGI	Space Science and Geospatial Institute
U.N. FAO	Food and Agriculture Organization of the United Nations

Abstract

Land cover change detection is essential for monitoring environmental changes. This paper aims to address the need for detecting changes in land cover over large areas and datasets. The study conducts a design experiment using aerial photos of Yeka Subcity, Worda 10, in Addis Ababa, taken in 2018 and 2021. Our work represents a significant advancement in the efficient and precise analysis of large-scale data for land cover change detection. We propose an unsupervised learning approach that employs a Convolutional Autoencoder (CAE) to robustly learn features from the input data. Temporal variations are identified using Euclidean distance, and Otsu thresholding is applied to generate binary change maps. The CAE model, optimized with Mean Squared Error (MSE) as the loss function, achieved an 89% accuracy rate in detecting land cover changes. This deep learning-based approach demonstrates considerable promise and effectiveness for large-scale land cover change detection.

Keywords: Land cover, Deep learning, Unsupervised learning, Convolution Autoencoder, Euclidean distance, Otsu thresholding

Chapter One

1. Introduction

1.1. Background of the study

Every year, the world goes through many changes caused by human activity [1, 2] Several studies have shown that land cover change can be triggered by the interplay of socioeconomic and natural environmental factors[3]. Even though a lot of research has been done to try and figure out what causes land cover change in most developing nations, the problem is still not entirely understood. Studies on the changes in land cover have been carried out at several sizes, ranging from local or finer levels to global or coarse ones. In the context of most developing countries, studies at both scales face significant challenges. While global-scale land cover change studies provide insights into the rate and direction of changes from a broad perspective, they often obscure important details characteristic of small-scale settings typical of farming systems in developing countries. Conversely, finer-scale studies focus on isolated and specific areas, which makes it difficult to grasp the overall picture, highlighting the need for more such studies. Given the importance of studying land cover changes in the context of a rapidly growing population, entrenched poverty, an economy heavily reliant on land resources, and severe ongoing land degradation, the role of land cover change studies is essential[1].

Identification of changes through the analysis of remote sensing images proves valuable in both civil and military settings [10, 11]. In military applications, it may be used to obtain information on the whereabouts of enemy military units, new military sites, and damage assessments. [6]. For civil applications, identifying land cover change detection (LCCD) can be helpful for a variety of purposes, including analyzing land use change, tracking shifting cultivation, evaluating deforestation, researching vegetation phenology changes, analyzing damage, identifying crop stress, monitoring snowmelt during disasters, analyzing day-and-night thermal characteristics, city planning, and other environmental changes [7].

The pursuit of precise change detection comes from the extensive land areas and examining their evolution over some time [8]. Detecting changes manually using sequential imagery is a slow, laborious process, time-consuming, tedious, formidable task, and even boring [13, 15, 16]. Thus, there is an evident need for an automatic change detection system that analyzes and correlates two

sets of images of the same region obtained at different times and shows the interpreter the changes and their positions [11]. The proposed solution suggests that a substantial enhancement in image processing speed can be achieved by focusing solely on the changes. This approach aims to spare humans from the laborious task of processing all the information present in both images, thereby streamlining the overall process [12]. The challenge of automating change detection in image pairs or sequences has been a subject of study for several decades [8].

Since 2014, deep learning (DL) has grown in importance within the remote sensing field [13]. DL has numerous applications in remote sensing for instance feature extraction and classification plays a big role. The rapid development, progressing speed, and an expanding variety of remotely sensed data, along with growth and advancements in DL, have elevated it to a prominent investigation topic. Given that the majority of DL architectures, initially designed for multimedia vision and the predominantly digital format of remote sensing data, can be effectively applied to remote sensing optical images[13]. The generated change map can be used as a basic preliminary filter, but it can also yield important details about the number, size, or form of the altered regions. These details can be easily incorporated into higher-level event detectors and object analyzer modules. [14]. Understanding the dynamics of both natural resources and man-made structures is crucial because it offers useful knowledge for making decisions in both public and commercial entities. [4]. However, the usefulness of this data relies on the specific needs of the application as well as the dataset's accessibility.

So coherent utilization of DL methodologies represents a substantial advancement in the field of remote sensing. It highlights DL's potential to enhance the precision and efficiency of various tasks associated with analyzing Earth's surface features. As remote sensing datasets continue to present unique complexities, the adaptability of DL proves instrumental in addressing nuanced problems and extracting meaningful insights. The integration of DL techniques into change detection processes reflects a promising avenue for further advancements in remote sensing research and applications. In this research, we utilized deep learning (DL) and unsupervised learning techniques to identify changes in time series aerial photographs.

1.2.Statement of the Problem

Understanding the dynamic changes in land cover during particular or extended periods is essential for making timely decisions, especially in the public sector. Strategic governance and resource allocation are severely hampered by the inefficiencies, errors, and lack of transparency associated with traditional approaches for detecting and tracking changes in land cover. Land cover change can lead to, forest fragmentation, biodiversity loss, and land degradation[15]. These issues can be seen in developing countries, where rapid population growth results in inconsistent changes. This resulted in deforestation intended to boost agricultural output[22, 23] and fuelwood.

Change detection methods such as visual interpretation and field investigation detection are highly inefficient. Visual interpretation involves the visual of a multitemporal image composite and the on-screen digitization of altered regions. This process requires an experienced analyst who can incorporate texture, shape, size, and patterns into the interpretation to make decisions about temporal changes. Typically, this involves analyzing images from two or three different dates[12]. For instance, visual interpretation has been used to identify changes between arable land and grassland[18], as well as to estimate changes in Amazonia due to selective logging[19]. This method relies heavily on the analyst's ability to incorporate texture, shape, and size, making it a challenging task. Due to these difficulties, visual interpretation is not frequently employed in digital change detection analysis and often requires highly experienced analysts. Additionally, this method cannot provide detailed change information, and the results may vary based on the experience of the analyst. For instance, implementing visual interpretation on data generated by Sentinel satellites, which have produced approximately 25 petabytes of data over the years[20], include significant amount of time required for large-area change detection applications and the challenge of promptly updating the findings. Furthermore, presenting detailed change trajectories is also difficult with this method.

The alternative approach is predicated on traditional field investigation, which is labor-intensive logistically complex and so insufficient on such a large scale. The vastness of the region makes it difficult to get data promptly, which hinders the rapid exchange of crucial information needed for informed decision-making. For instance, Ethiopian government has started the Climate Resilient Green Economy (CRGE) including the Green Legacy Initiative (GLI). This initiative goal aims to create a “*green and climate-resilient Ethiopia*”. Within this GLI initiative, the Ethiopian

government plans to plant 25 billion seeding in the next four years. Effective monitoring of such a large-scale project requires identifying planting sites, tracking plant growth, and identifying drought-prone areas, all of which cover vast territories. This approach is not able to produce concrete proof to support the noted changes in land cover. This shortcoming makes it more difficult to convince a wide range of stakeholders of the merits of comprehensive policies and initiatives.

The combination of labor-intensive on-site field investigations and the tedious nature of manual identification using remote sensing data creates a dual challenge. This not only compromises the speed at which valuable information is obtained but also hampers the overall effectiveness of the decision-making process. To surmount these challenges, there arises a pressing need for a method that is capable of supporting a large amount of territory or multitemporal data by harnessing the power of continuous, multitemporal imagery. Such a system could facilitate the identification of drought-prone areas, monitor the progress of planted seeds, and detect instances of false reporting change that occurred. By leveraging advanced technology, this can streamline monitoring efforts, providing policymakers with accurate and timely information to guide their actions.

In light of these challenges, there is an urgent need to explore advanced methodologies, particularly leveraging emerging technologies like deep learning that can be used to identify temporal variation from large multitemporal data. Such approaches have the potential to revolutionize how we discern, interpret, and communicate land cover changes. The focus of this research is to build a deep learning model that can identify change from unlabeled multitemporal images and visualize the results by generating a binary change map, thereby providing a more efficient and accurate solution for large-scale land cover change detection.

1.3. Research Question

RQ 1: How to identify change from unlabeled multitemporal aerial photographs?

RQ 2: What preprocessing techniques are essential for preparing aerial photograph data for deep learning models in land cover change detection?

RQ 3: What are the key challenges in identifying land cover change detection in a multitemporal image?

RQ 4: How to build a deep learning-based change detection model from aerial photographs?

1.4. Objective

1.4.1. General Objective

The objective of this research is to construct an unsupervised deep learning model designed to effectively identify changes in land cover.

1.4.2. Specific Objective

- To review related state-of-the-art literature to identify the best methods.
- To prepare and preprocess the multi-temporal data.
- To Investigate feature extraction methods for aerial photographs.
- To develop a change detection model that is based on unsupervised learning.
- To evaluate the performance of the proposed model.

1.5.Scope of the study

Through the analysis of aerial imagery taken between 2018 and 2021, this study will concentrate on finding temporal changes in land cover. Using cutting-edge deep learning algorithms, the main goal is to identify changes throughout this time. To do this, we will create a deep learning model that is intended for feature extraction and use to the extracted features to find temporal changes without the need for labeled data. This method will find notable changes in land cover by utilizing the patterns and attributes that are already present in the dataset.

The deep learning model's development and application will be a primary focus of the research. To maximize the model's ability to precisely identify changes, we will investigate a variety of architectural options and preprocessing methods. The effectiveness of the model will be evaluated using appropriate metrics to ensure robust and reliable detection results. We will utilize the model we constructed to find differences between the aerial photographs from 2018 and 2021. A binary change feature map will be used to graphically represent the identified changes, clearly highlighting regions of notable modification. With the help of this visualization, one will be able to comprehend the extent and geographical distribution of changes in land cover, which will provide important insights into the dynamics of the area under study.

In conclusion, by utilizing unsupervised deep learning algorithms on multi-temporal aerial images, this research seeks to enhance the practice of land cover change identification and eventually offer a complete and automated system for detecting land cover changes over time.

1.6.Limitations of the study

One limitation of the research is the limited availability of multi-temporal data. For this research, we were only able to get aerial photos for this study covering three years between 2018 and 2021. Due to the very short period, there could be less noticeable changes, which could make it more difficult to identify significant changes in land cover.

Another limitation is the difficulty of obtaining sufficient hardware for the proposed deep learning model training. Not only for training the data and for the preprocessing nearly 100Gb of multitemporal data has been taken several times. The computational demands of the training process and the preprocessing have led to prolonged training times, which has slowed down the overall progress of the study.

It has also been challenging to locate experts in the field of remote sensing to confirm the alterations that have been seen. The verification procedure takes a long time, and it has proven difficult to get specialists who are prepared to devote the required time to this task.

1.7. Significance of the study

This study can hold significant importance as it applies deep learning methods in the realm of change detection within Ethiopia, addressing a notable void in existing research efforts. Although the discipline of remote sensing has witnessed extensive research on change detection, a prominent gap persists as most studies have not ventured into the realm of deep learning methodologies.

The key to the matter lies in the scarcity of deep learning-based approaches, which has resulted in a limited availability of preprocessed multitemporal data is a crucial element for the success of such methodologies. This study seeks to bridge this gap by introducing the concept that researchers from diverse disciplines, particularly those in computer science specializing in deep learning, can contribute valuable insights to the field of change detection. By extending an invitation to experts in remote sensing, this research aims to foster collaboration and knowledge exchange, emphasizing the significance of their involvement in providing labeled data that can fuel advancements in deep learning-based change detection. Investigating land cover is crucial in the current era to efficiently harness available resources, facilitate proper planning, implement effective management practices, and promote sustainable development [21].

1.8. Methodology of the study

1.8.1. Research design

An experimental research approach will be used in this work, which is especially pertinent given the use of deep learning techniques for change detection. In experimental research, variables are systematically changed in order to see how such changes affect the result variable. In the realm of change detection, this approach allows for the deliberate adjustment of parameters within deep learning models to assess their impact on the accuracy of detecting changes in images over time. The controlled nature of experimental research is particularly valuable in minimizing external factors that could influence the outcomes, providing a robust framework for investigating the effectiveness of deep learning algorithms in change detection. This approach is pivotal for establishing causal relationships between model configurations and detection accuracy, finally enhancing our knowledge and improving how deep learning is applied in change detection settings.

1.8.2. Data collocation

Gathering essential data poses a challenge in the research process, with the requirement for multitemporal images to train a model for change identification. Access to such data can be facilitated through the Space Science and Geospatial Institute (SSGI).

1.8.3. Study Area

The research will be conducted in the Addis Ababa region around Yeka worda 10, utilizing aerial photography captured in the years 2018 and 2021 G.C. These images serve as invaluable tools for our study, allowing us to analyze and compare changes in the region over the specified time frame.

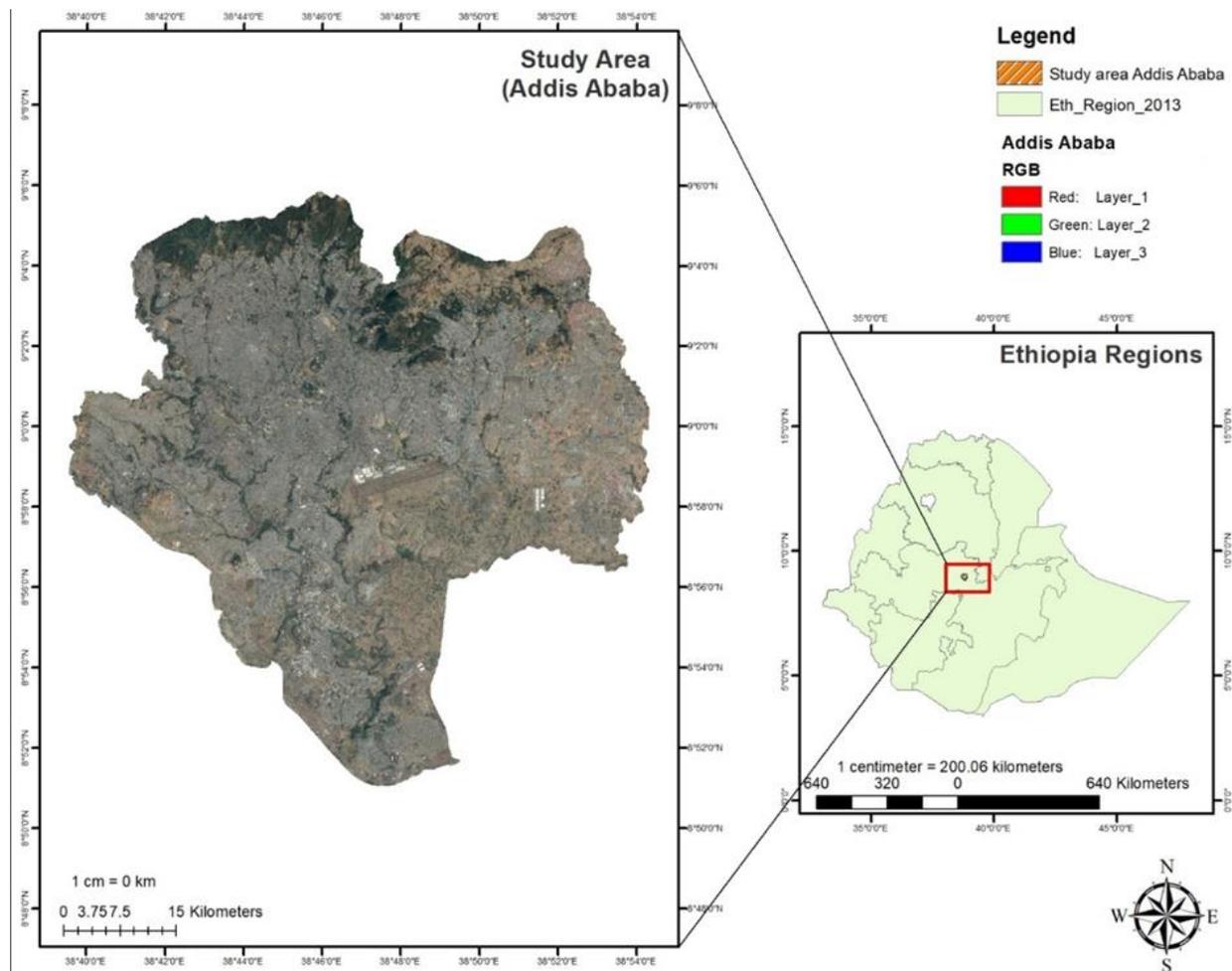


Figure 1: Study area

1.8.4. Development tools

In crafting the proposed model's design, this paper will leverage a diverse set of tools to facilitate its development. The selection of these tools is driven by their ability to contribute distinct functionalities and enhance various aspects of the model. Specifically, the integration of tools will encompass a comprehensive approach to ensure the robustness and effectiveness of the designed model.

- **Python:** stands out as one of the most widely embraced programming languages, gaining significant prominence, especially in fields like image processing, machine and deep learning. Renowned for its versatility and supported by an extensive library, Python has emerged as the preferred language for a diverse range of researchers and developers alike. This popularity is further amplified by a robust ecosystem, fortified by influential libraries like TensorFlow, Scikit-learn, and PyTorch. These libraries offer comprehensive frameworks that streamline the process of constructing and deploying deep-learning architectures with ease.
- **Keras:** stands out as a high-level API within the TensorFlow platform, specifically designed to cater to the requirements of contemporary deep learning. It comprehensively addresses each stage of the deep learning workflow, such as data processing, hyperparameter tuning, and deployment. Which makes Keras a robust and versatile tool for building our deep-learning model.
- **NumPy:** is a very popular Python library developed by Travis Oliphant in 2005. it has a large collection of high-level mathematical functions.
- **Anaconda:** is an open-source platform that enables you to write and execute Python code. It is the most popular choice for learning and utilizing Python in scientific computing, data science, and machine learning, with over thirty million users globally.
- **Global Mapper:** has received a great deal of praise from the remote community for its outstanding capacity to analyze and visualize geographic data, including both vector and raster data formats. The software plays two roles in the context of research projects. First

and foremost, it makes data visualization easier, which in turn promotes a thorough comprehension of complex raster data. Moreover, Global Mapper is essential to the production of co-registered pictures, which makes multitemporal dataset alignment easier.

- **ArcGIS:** one of the most popular applications of geospatial technology. It is used for visualizing and analyzing the multitemporal image. Also, it is used to create a study area map with the standard.
- **Git:** Git is a decentralized version control system created to manage file versions. It has assisted research efforts by effectively managing code and documentation.
- **Geospatial Data Abstraction Library (Gdal):** is a library of tools for working with geographical data. which is a very helpful tool that works with both raster and vector data formats. For our research, it is utilized for reading the raster data which is Geotiff image.

1.8.5. Evaluation metric

In the realm of any research, the significance of evaluation cannot be overstated, as it serves as a pivotal tool for measuring results and facilitating comparisons with related studies. Evaluation metrics play a crucial role by providing a quantitative measure to assess the performance and effectiveness of our model within the specific context of our research. These metrics not only offer a means to showcase the measurable outcomes but also furnish important insights regarding the model's performance. This, in turn, aids in the informed selection of optimization solutions, guiding the refinement of the model for enhanced efficiency and effectiveness. In essence, the process of evaluation becomes an indispensable aspect, steering researchers toward a deeper understanding of their model's performance and paving the way for strategic optimization decisions. For our research, we will be using the following evaluation metrics.

Confusion Matrix

- A confusion matrix is a table that shows the performance of a classification model. It provides a comprehensive examination of the discrepancies between the predicted and actual classifications for a given dataset. To evaluate the performance of the model, look at the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values in the confusion matrix.

Below is an explanation of each component of a confusion matrix: -

- TP: Pixels recognized as having undergone alteration.
- TN: Unchanged pixel found.
- FP: Incorrect identification of unchanged pixels as a change.
- FN: Modified pixels that are mistakenly categorized as unaltered.

- **Accuracy**

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}$$

Equation 1: Accuracy equation

- **Precision**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Equation 2: Precision equation

- **Recall**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Equation 3: Recall equation

- **F1 Score**

$$\text{F1 Score} = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

Equation 4: F1 equation

Chapter Two

2. Literature Review

2.1. Change Detection

Observing an object, system, environment, phenomenon, or dataset at various intervals allows one to recognize and analyze changes in its status. This technique is known as change detection [22]. Its applicability now extends across diverse domains such as medical diagnosis and treatment [29], civil infrastructure [33], driver assistance systems [29], and video surveillance [30]. Specifically, within remote sensing, the process of identifying changes in a given geographic region by comparing a collection of photos taken at various times is known as "change detection" [31]. Essentially, it involves the ability to quantify temporal effects using multitemporal images [7], which are a series of images taken at different points in time that capture the same geographic area or scene. Geographic information systems (GIS), remote sensing, and other disciplines that monitor and analyze environmental changes frequently employ multitemporal imagery. These images allow for the analysis and comparison of changes and patterns over time, proving invaluable for monitoring continuous changes in land cover across vast geographical areas, which would be difficult to detect using conventional field investigations alone [32]. This procedure is an essential technology in the field of earth observation. Its purpose is to differentiate between changed and unchanged pixels in bi-temporal or multi-temporal remote sensing images gathered from the same geographical zone or area, but at different periods [33]. The collection of multitemporal images for this purpose can include satellite imagery like multispectral images and aerial photos, or even a combination of heterogeneous datasets obtained from various satellite sources. With the proliferation of digital remote sensing data, including radar and satellite imagery like SPOT images, and the subsequent improvement in spatial resolution over the past decade, the implementation of change detection in various applications has become increasingly feasible [28].

The core principle of change detection hinges on the premise that alterations in land cover should manifest as variations in radiance values. However, the accurate identification of these changes is susceptible to several influencing factors. These include variations in the air pressure, changes in the Sun's angle, and variations in the moisture content of the soil.

To delve into the impact of atmospheric conditions, the presence of clouds, dust, or other particulates in the atmosphere can scatter or absorb sunlight. This alteration in sunlight can, in turn, modify the radiance values detected by the sensor [35]. Regarding the Sun's position, its angle

can significantly influence the amount of sunlight reaching Earth's surface and subsequently being reflected to the sensor. This variability can result in the same land area appearing differently at different times of the day or year, potentially leading to inaccuracies in change detection [36].

Moreover, the water content in the soil plays a pivotal role in altering its reflective characteristics. Moist soil, for example, reflects less sunlight compared to dry soil. This discrepancy in reflective properties can create the illusion of a change in land cover, even when no actual change has occurred [37]. These considerations underscore the need for a nuanced understanding of the environmental factors that can introduce variations in radiance values, guaranteeing the precision of change detection techniques in the context of remote sensing.

Two essential characteristics of a successful change detection system are its accuracy in detecting changes and its capacity to show the geographical distribution of altered types. There are a few typical measures that should be taken into account before starting change detection research. First and foremost, image processing is crucial. To guarantee accuracy and consistency across several temporal images, procedures like image registration and geometry correction are involved. Second, in order to effectively identify changes, appropriate procedures that match the features of the multi-temporal data must be chosen. Finally, it is critical to evaluate the change detection algorithms' accuracy.

The accuracy of change detection hinges on numerous factors, including: -

1. Accurate geometric alignment of multi-temporal images.
2. Calibration or normalization processes for multi-temporal images.
3. Availability to high-quality ground truth data.
4. The complexity of the study area's landscape and environment.
5. The change detection techniques or algorithms implemented.
6. Classification and change detection frameworks.
7. Considerations of time and cost [22].

Before tackling implementing change detection, it's crucial to grasp various aspects. MacLeod and Congalton [38] provide an insightful instance of change detection in monitoring natural resources. This involves detecting the occurrence of change, quantifying its extent, and evaluating its spatial pattern. Furthermore, understanding the factors influencing land cover change is imperative. These factors can be categorized into five main groups: Ecological and geomorphological factors, such as soil erosion and vegetation succession; long-term natural changes, including climate change; human-induced factors, such as alterations in vegetation cover, deforestation, and land degradation; climate change variability, such as the greenhouse effect resulting from human activities[39]. Comprehending these influences aids in selecting the study area and the necessary time series data. By considering these elements, the approach becomes more comprehensive and coherent, laying a solid foundation for effective change detection. In addition to understanding environmental characteristics, implementing a robust change detection methodology using remote sensing data necessitates proficiency in image processing techniques and familiarity with remote sensor systems. Several considerations are crucial in this regard:

Firstly, meticulous preprocessing is indispensable. This involves careful inspection of environmental elements include atmosphere changes, soil moisture content, and phenological properties within the study area before undertaking change detection endeavors. This preliminary analysis ensures a thorough understanding of the data's quality and potential limitations.

Moreover, image preprocessing for change detection is pivotal, distinguishing it from other image-processing tasks. Central to this process is image registration, which entails aligning multiple temporal images spatially. Spatial misregistration can significantly compromise the accuracy of change detection models [49, 50]. Therefore, before implementing change detection algorithms, it is imperative to ensure: -

1. Accurate multi-temporal image alignment, or registration.
2. Radiometric correction to rectify inconsistencies in brightness values across temporal images.
3. Identification of any seasonal variations between multi-temporal images.
4. Selecting imagery that shares the same spatial and spectral properties if possible.

By adhering to these preprocessing prerequisites, researchers can establish a solid foundation for accurate and reliable change detection analyses based on remote sensing data. This meticulous approach enhances the coherence and effectiveness of the change detection process, ensuring its applicability in diverse environmental monitoring and management contexts.

2.2.Remote Sensing

Remote sensing is the discipline focused on gathering and analyzing data from a distance using sensors that don't necessitate physical contact with the target object. While its reach extends to studying planetary surfaces and atmospheres across our solar system, the primary emphasis is on Earth. Generally, remote sensing techniques revolve around utilizing sensors to detect and measure electromagnetic radiation, notably visible light, that has interacted with surface characteristics and the atmosphere[34].

Remote sensing is a fundamental technique for acquiring multitemporal data and a crucial instrument for learning about the Earth's surface and its characteristics. It involves measuring the electromagnetic reaction that interacts with the atmosphere or an object [43, 51]. There are two types of interactions that occur between a sensor and the surface of the Earth: active and passive. Our research focuses on the latter, in which passive sensors use solar energy to illuminate the surface of the Earth and then look for reflected light.

A material's ability to transmit certain kinds of electromagnetic waves depends on its physical characteristics. For example, clear water, lush flora, and dry, barren dirt all have different reflection spectra. Dry, exposed earth becomes more reflective as the wavelength gets longer, from 400 to 1800 nm. Green vegetation, on the other hand, has strong reflectivity in the near-infrared and red-light spectrums. These distinctive qualities are utilized to set green plants apart from other items [34].

In our research, we will employ aerial photography images, utilizing multiple channels, each sensitive to a specific range of wavelengths. This approach results in a layered image that contains not only the brightness of the observed targets but also their spectral characteristics (color).

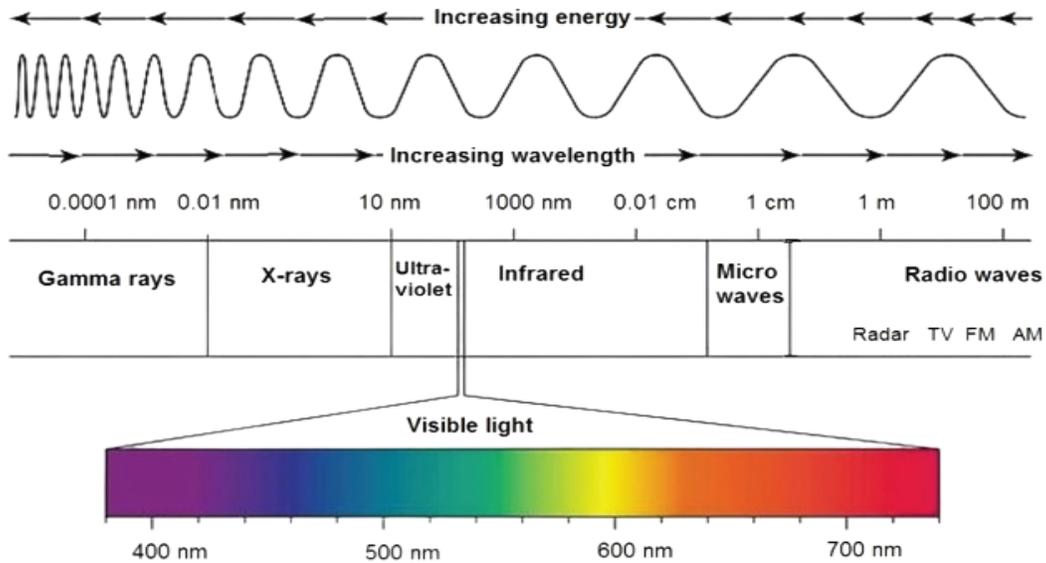


Figure 2: Electromagnetic spectrum [34]

2.3. Unsupervised Learning

Creating methods to solve issues computationally is one of the main objectives in the subject of computer science. Even while human creativity is crucial to the process of creating algorithms, there are some problems whose answers are very hard to come up with only human creativity. These issues frequently fall within the "AI set," or the group of issues that systems purporting to be intelligent should be able to resolve. This covers issues including locating and identifying items in an image, comprehending natural language, identifying human voice, and having the ability to move, open doors, and control objects. Finding patterns in the incoming data appears to be necessary for solving these issues, but figuring out which patterns to look for and how to represent information internally has proven to be a very difficult and time-consuming effort if done by hand.

The technique of utilizing machine learning (ML) algorithms to analyze and classify unlabeled data sets is known as unsupervised learning, or unsupervised machine learning. So, the concept of unsupervised learning is finding structure from uncategorized data. These ML algorithms find hidden relationships or patterns in the data without requiring human assistance. Over the years supervised learning has achieved a lot by solving a prediction problem and the model can generalize well to the dataset that is often trained on [43]. Even while supervised learning has proven to be effective, it is limited in its ability to teach general-purpose representations that can be used for a range of activities and easily adjusted to new ones without requiring human

supervision. Furthermore, it appears quite artificial that we need extensive sets of meticulously labeled data to comprehend the organization in fields like the visual realm, where the arrangement appears readily observable. If we aim to create a representation learning system that mimics biological processes, then relying solely on supervised learning is inadequate. Hence, there are compelling reasons to pursue unsupervised methods for learning the structure within data [44]. Numerous attempts have been made to enable AI-driven change detection without the need for extensive labeled datasets, often through unsupervised or semi-supervised approaches. Transfer learning, for instance, has emerged as a promising method to mitigate the requirement for training samples, although it necessitates data from other domains, thereby straying from pure unsupervised methodologies[45].

Machine learning is an algorithm design methodology that places a strong emphasis on acquiring and using experience. Examples of correct or incorrect input-output behavior, merely monitoring the inputs or a more active engagement between an agent and its surroundings might all be considered forms of this experience. This method's basic tenet is that statistical modeling of experience, often known as data, offers a scalable means of identifying the representations and patterns required to solve AI-related issues. Finding models, or at least design concepts for models, that apply to a broad range of problems is the goal. Because of this, human creativity is detachable from the machine learning method in that it is focused on determining which statistical models and assumptions should be applied to interpret or understand the data, rather than being directed to solving specific issues.

One specific design idea has shown to be incredibly effective over the past decade years in resolving a wide range of issues. Models should be taught several tiers of distributed representations, according to this theory. The family of models known as deep neural networks, which are made up of linked neuronal units arranged in a deep hierarchy, is a popular example of this idea in action. Different functions may be represented by a deep neural network by tuning the strength of the connections between these units. In fact, given an incredibly enormous number of units, it can represent any function arbitrarily well. Other methods also have this virtue of universal approximation. The intriguing thing about these models, though, is that, when placed in a deep computing network, a large number of practical functions related to AI problem-solving may be represented effectively with very few neurons. Deep learning is a discipline that was established

as a result of this important discovery, which was motivated by biological neural networks and verified on several real-world AI challenges. Interest in these models has skyrocketed[44].

In unsupervised learning, direct application to classification or regression problems isn't feasible due to the absence of corresponding output data. Instead, the focus is on unraveling the underlying structure of the dataset, grouping data based on similarities, and representing the dataset in a compressed format.

2.4. Deep Learning

The science of Earth is enduring a massive transformation, evolving under the impact of a revolution of data. The generation of this data has been driven by Earth-observing satellites combined with the ongoing discovery and creation of physics-based earth system computing models[46]. This integrated information offers an unprecedented chance for monitoring the interpretations of the complex dynamics of the ecosystems and climate system of earth and related societal issues such as food, water, energy security, and climate change that affect the future of our planet. As regards DL, a key technology that has set off far-reaching changes in various disciplines ranging from computer vision to natural language interpretation, it also has the potential for an environmental turnaround by bringing about nothing less than a revolution in Earth and environmental sciences.

Earth science has experienced these last years a dramatic development of DL applications which helped to gain remarkable results. However, it is faced with quite prominent difficulties when it comes to applications of this technology to Earth science data even despite that. The problems are multimodality; a high degree of heterogeneity in time and space as well as the characteristics of Earth science data which is deficient and noisy. Dealing with the mentioned obstacles requires improvement of our understanding and making new approaches that can integrate existing knowledge of Earth Science disciplines into the algorithms successfully. The solutions that will resolve this key issue of deep learning can bring to not only making deep learning better but also fast-tracking the discoveries from a variety of scientific subjects[47].

2.5.Related Works

The research undertaken in this study focuses on change detection using Landsat Enhanced Thematic Mapper Plus (ETM+) data with a 30-meter spatial resolution, covering the city of Taizhou, China, during the period from 2000 to February 2003. Four different classes are included in the research: unaltered areas, city expansion (including moves from bare soils, grasslands, or cultivated fields to buildings or roads), soil change (especially moves from cultivated fields to bare soil), and water change (transformations from non-water regions to water regions). The research aims to display binary and multi-class change detection maps [49]. The objective of the research is to develop a unified RNN architecture capable of seamlessly integrating spectral, spatial, and temporal aspects into a single framework. This holistic approach enables the model to extract features from multispectral imagery, taking into account spectral, spatial, and temporal dimensions. In order to do this, the research combines recurrent neural networks (RNNs) with convolutional neural networks (CNNs), breaking down the approach into three essential architectural parts. Firstly, the initial phase centers on spectral-spatial feature extraction, employing two identical convolutional sub-networks. These sub-networks are adept at processing sequences of 5x5 multispectral patches of images, enabling the discovery of spectral features within the spectral data [49]. The second stage uses a Recurrent Sub-Network, namely Long Short-Term Memory (LSTM), to handle temporal dependencies. This network leverages the features learned in the previous phase and is proficient at handling sequential input data. It analyzes the bi-temporal dependencies within the images, allowing the model to capture how changes evolve and make predictions based on this historical context. The final stage encompasses two fully connected layers dedicated to classification. These layers transform the extracted features into actionable change detection maps.

Among the proposed methods, while the previously discussed ones are grounded in supervised learning, an alternative approach involves unsupervised learning. This specific approach consists of two layers: a U-Net segmentation layer and a feature-level subtraction block layer. A two-path network with shared weight parameters is included into the feature-level subtraction block layer. These paths are instrumental in extracting dynamic difference images (DI) at both low- and high-level features. The extraction process involves subtracting the output images from both branches at the corresponding layer [50]. This method uses information from both low- and high-level characteristics to generate subtle difference pictures. The feature-level subtraction block layer and

the U-Net segmentation layer are connected in a way that enables the integration of low- and high-level features. This layer is designed to accept both low- and high-level features, establishing a comprehensive integration with the feature-level subtraction process. The primary objective of the U-Net segmentation layer is to effectively segment changes within temporal images. By incorporating the extracted dynamic difference images, the U-Net segmentation layer contributes to the unsupervised learning approach by identifying and delineating temporal changes within the imagery. The objective of this paper is to develop a deep learning architecture capable of mitigating spectral distortion in multispectral images and effectively identifying changes within these images. The proposed method's strengths lie in its ability to harness both geometrical and spectral resolution enhancements, allowing for the accurate detection of small changes. To substantiate the effectiveness of the proposed method, the authors employ key performance metrics, specifically Percentage Correct Classification (PCC). A PCC score of 98.78% underscores the accuracy of the proposed method in correctly classifying changes within the multispectral images.

A method to unsupervised learning-based change detection in multitemporal pictures is presented by the other suggested End-to-End CD Network, which is built on CNNs. The architecture is structured around three parallel channels, each contributing uniquely to the overall task. In the first and third channels, the emphasis is on extracting deep features from the multi-temporal images, capturing both spectral and temporal information. The second channel is dedicated to change detection, leveraging differencing and stacking operations on deep features from the other channels to highlight significant changes over time[51]. The primary objective of this paper is to tackle a critical limitation in existing methods for change detection, specifically the neglect of spatial features and underutilization of spectral information. Many current approaches focus solely on one type of remote sensing data, rendering them unsuitable for diverse datasets. Moreover, prevalent CNN-based change detection methods typically rely on 2D convolution layers, overlooking the interdependencies among spectral bands. This oversight hampers their ability to fully leverage the rich spectral information available. To overcome these shortcomings, the proposed method advocates for a novel approach. By integrating both 3D and 2D deep features, the model aims to synergistically harness spatial and spectral information, transcending the limitations of existing techniques. This strategic combination is anticipated to yield improved performance results, addressing the inherent challenges associated with change detection across diverse RS datasets and enhancing the overall effectiveness of the CNN-based approach[51].

Table 1. Summary of Reviewed Literature

Author	Techniques	Mode	Conclusion	
			Accuracy	
[49]	CNN and RNN	supervised	0.9873	The paper's strength lies in its end-to-end approach, eliminating the need for manual feature engineering, and its learn a composite representation of spectral, spatial, and temporal features, which is essential for detecting changes in multispectral images. However, a potential weakness of the paper is its reliance on dilated convolution, which may lead to the omission of deep features from the multispectral image.
[50]		unsupervised	0.9878	This paper's goal is to create a deep learning architecture that can mitigating spectral distortion in multispectral images and effectively identifying changes within these images. The proposed method's strengths lie in its ability to harness both geometrical and spectral resolution enhancements, allowing for the accurate detection of small changes.
[51]	CNN-based CD	unsupervised	0.9889	The primary objective of this paper is to tackle a critical limitation in existing methods for change detection, specifically the neglect of spatial features and underutilization of spectral information. Many current approaches focus solely on one type of remote sensing (RS) data, rendering them unsuitable for diverse datasets. Moreover, prevalent CNN-based change detection methods typically rely on 2D convolution layers, overlooking the interdependencies among spectral bands. This oversight hampers their ability to fully leverage the rich spectral information available. To overcome these shortcomings, the proposed method advocates for a novel approach. By integrating both 3D and 2D deep features, the model aims to synergistically harness spatial and spectral information, transcending the limitations of existing

				techniques. This strategic combination is anticipated to yield improved performance results, addressing the inherent challenges associated.
--	--	--	--	---

Chapter Three

3. Modeling for Land Cover Change Detection Using Deep learning

3.1. Proposed Model

The proposed method for identifying changes leverages unsupervised learning due to the scarcity of labeled multitemporal data for change detection. The model employs an encoder-decoder architecture based on Convolutional Neural Networks (CNNs). The encoder consists of three convolutional layers that capture essential features from the input data, systematically extracting hierarchical representations of the input images.

Conversely, the decoder also consists of convolutional layers designed to reconstruct the spatial information from the encoded features. These decoder layers are meticulously constructed to decode the abstract representations learned by the encoder, ultimately reconstructing the original spatial structure of the input data.

Together, the encoder and decoder components form a cohesive architecture that effectively captures and reconstructs the features. During training, backpropagation and optimization algorithms are employed to minimize the loss function. This process allows the network to adjust the weights of the convolutional filters, enhancing reconstruction accuracy.

Once the model is trained, it processes multitemporal images to extract features. These features are concatenated from multiple layers to create a rich representation with enhanced discriminative power. A change map is then generated using pixel-wise Euclidean distance based on these features.

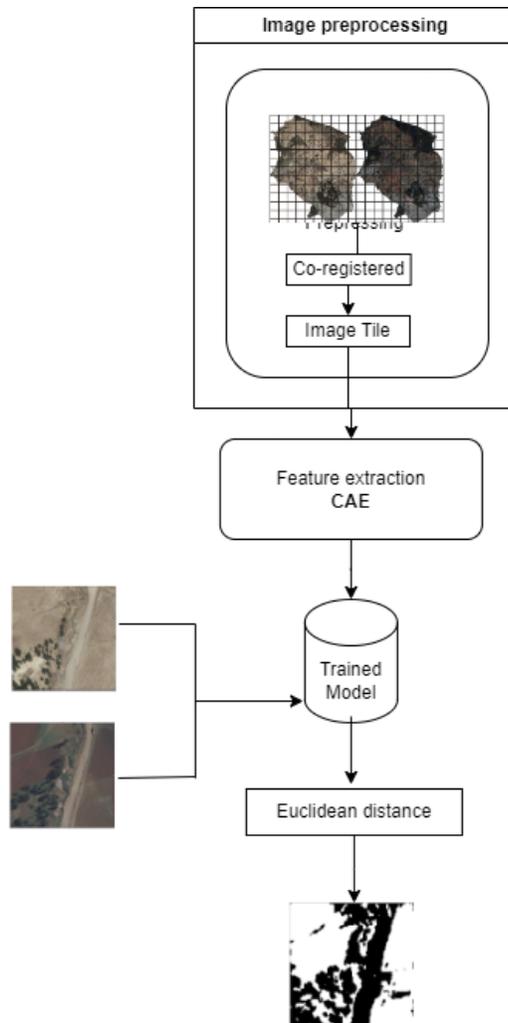


Figure 3: The Proposed Architecture

To construct an unsupervised model capable of identifying changes in remote sensing data. The steps that the research will take are as follows: -

- Image preprocessing.
- Feature extraction
- Based on the feature extraction, identify change.
- Evaluate the model performance.

3.2. Image Preprocessing

Developing projects in computer vision and machine learning invariably requires data, especially image data. Unfortunately, complexity, correctness, and sufficiency are a few issues with image data. This is why preprocessing (cleaning and formatting the data) is necessary before building a computer vision model in order to get the desired outcomes. Preprocessing images is an essential aspect of the machine learning model workflow, especially for tasks involving images. Preprocessing is the process of modifying and fine-tuning an image's pixel values to make the data more suitable for further operations within a standard model[52]. This approach is noteworthy since it improves the model's prediction speed while also cutting down on training time. Additionally, image preprocessing is essential for identifying and enhancing the important characteristics in the dataset, which makes the model work more accurately and effectively.

3.3. Image Acquisition

The utilization of aerial photography for land cover change assessment stands as a cornerstone within the realm of earth sciences. This methodological approach assumes paramount significance, particularly in regions such as Ethiopia, where access to longitudinal satellite imagery remains constrained, accentuating the pivotal role of aerial photography as a robust and enduring tool in elucidating temporal shifts in land cover dynamics[53].

The historical trajectory of aerial photography underscores its unparalleled advantage, boasting an extensive and continuous temporal coverage dating back to the early 1930s, a period preceding the advent of satellite technology[53]. This demonstrates that it was the first remote-sensing technique. This enduring legacy renders aerial photography the foremost repository of spatially comprehensive records about land cover alterations, especially in regions characterized by limited satellite data accessibility.

Aerial photography epitomizes a paradigm of remote sensing that is not only ubiquitous but also distinguished by its versatility and cost-effectiveness when compared to satellite-based methodologies. It represents the pioneering modality of remote sensing, steadfastly enduring into the contemporary era despite the ascendancy of satellite technologies.

The process of using cameras installed on airplanes to take pictures of the Earth's surface, its atmosphere, or its hydrosphere using aerial photography equipment, which includes both outdated film-based devices and more recent digital alternatives. These cameras serve as conduits for

capturing images that delineate the reflective properties of diverse land features, with reflectance being transcribed onto photographic emulsions comprised of light-sensitive silver halide crystals, a configuration emblematic of both monochromatic and color photographic compositions[54][55].For the digital-based aerial photography digital camera with a high-resolution sensor is mounted on an aircraft or drone the capture land features.

Crucially, aerial photography encompasses two principal geometrical configurations: vertical and oblique. In the context of mapping endeavors, vertical aerial photography assumes primacy, facilitated by cameras affixed to the floor of survey aircraft, thereby producing images characterized by consistent scale properties across their entire vertical axis. Conversely, oblique photography entails a departure from verticality, wherein the camera's axis deviates from the perpendicular, yielding images that capture terrain features from a distinctively angled perspective[56].

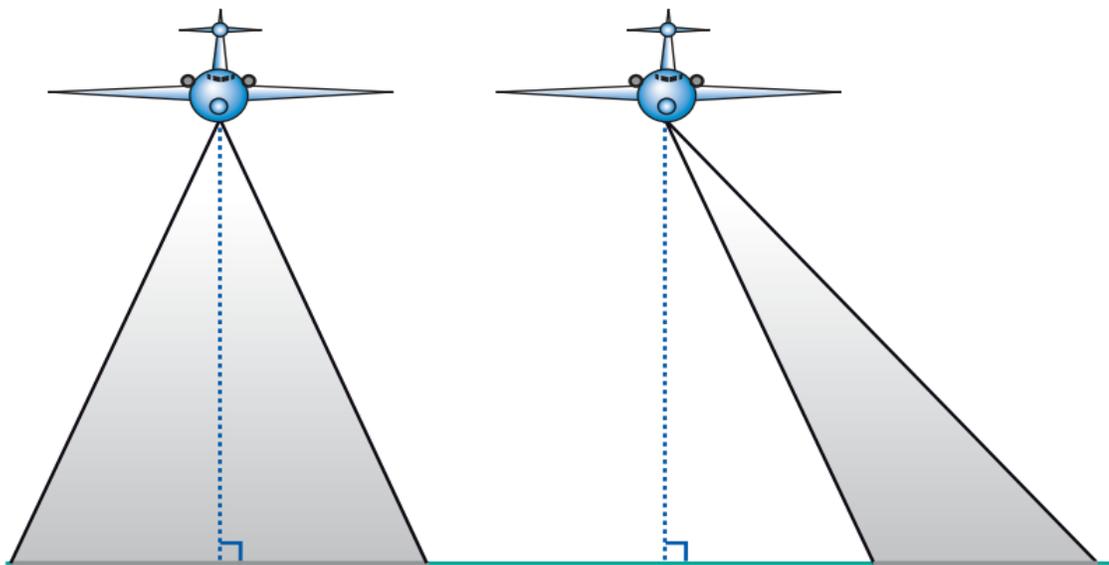


Figure 4: Vertical (A) Oblique (B) [56]



Figure 5: Example of vertical and Oblique aerial photograph[56]

Vertical aerial photograph (a) and (b) Oblique aerial photograph



Figure 6: Vertical and Oblique aerial photograph of Imperial Square Adey Abeba stadium taken in 2018



Figure 7: Vertical aerial photograph of Imperial Square Adey Abeba stadium taken in 2021

The Space Science and Geospatial Institute (SSGI) has developed a comprehensive repository of remote sensing data through its collection efforts, international donations, and the purchase of satellite imagery from various countries. This repository includes multitemporal images essential for various applications. For example, SSGI has acquired SPOT images taken in 2006 and 2016 G.C. for census-related purposes.

For this research, an aerial photograph of Addis Ababa taken in 2018 and 2021 G.C will be utilized, ensuring a coherent and comprehensive approach to studying land cover dynamics in the region.

3.4.Co-register Image

When dealing with multitemporal data, the process of image registration becomes pivotal. Its primary goal is to harmonize different images so that their information can be effectively combined or compared. We can achieve smooth integration or fusion by geometrically aligning two or more photos to guarantee that related pixels, which indicate identical objects, match perfectly [57]. Accurate registration is crucial for analyzing temporal variations, as any misalignment between images could significantly impact the reliability of the results.

In the realm of multitemporal aerial photos, achieving co-registration poses challenges due to disparities in coverage areas across images captured at different times. Proposed methods for co-registration involve leveraging either image coordinates or vector data, such as shapefiles.

However, the variability in coverage areas, particularly pronounced in the northern and western parts of Addis Ababa across different years, presents a notable obstacle.

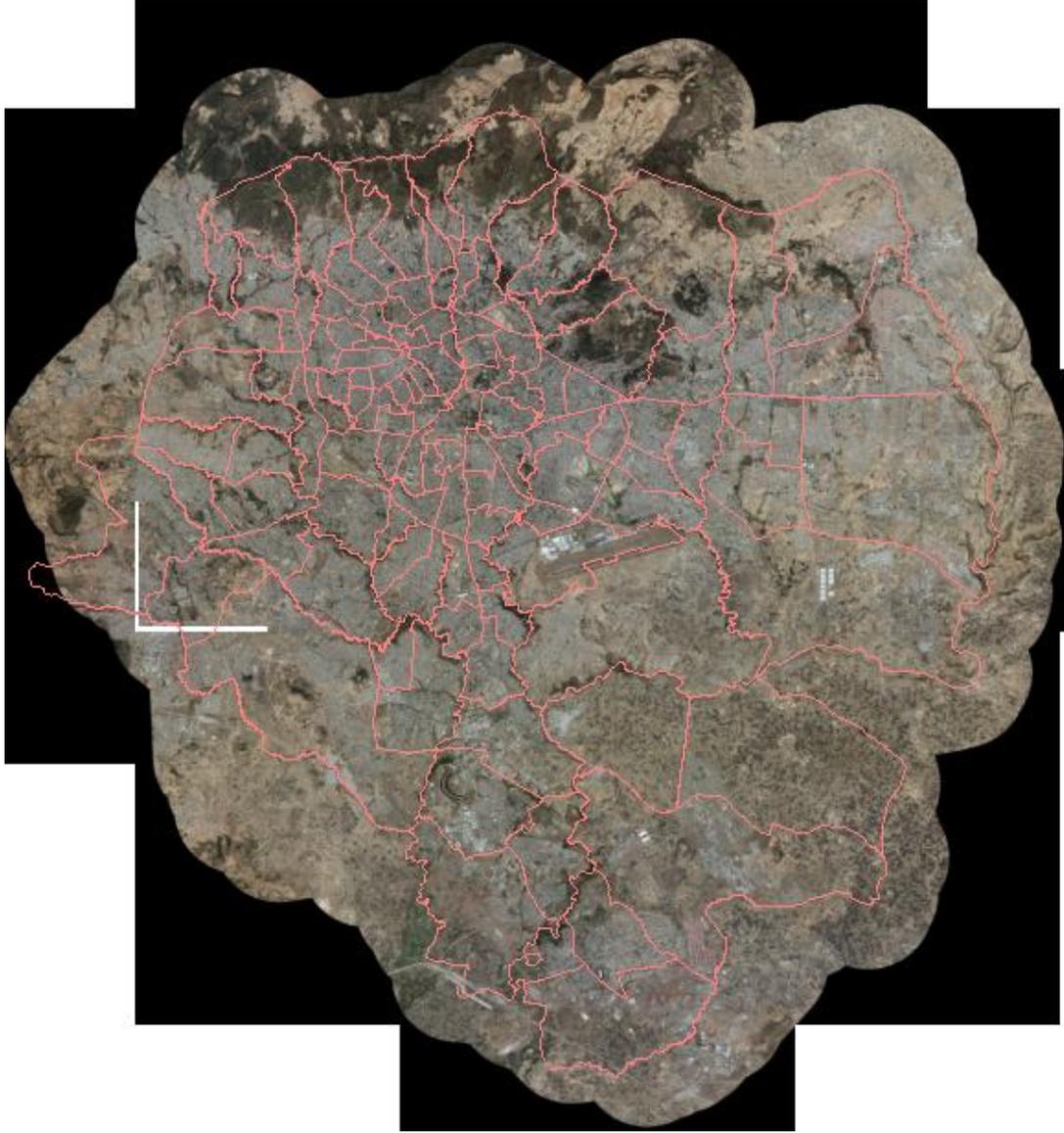


Figure 8 : Aerial photo of Addis Abeba Coverage 2018



Figure 9: Aerial photo of Addis Abeba Coverage 2021

The analysis of multitemporal images of Addis Ababa reveals a notable inconsistency in coverage between different periods. Specifically, when comparing the shapefile projected onto the images from 2018 and 2021, it becomes evident that there is not a consistent coverage area.

This inconsistency poses a challenge for co-registration efforts, as the varying coverage areas between the two years hinder the alignment process. Without a consistent coverage area, achieving accurate geometric alignment between images becomes more difficult.

Addressing this issue is crucial for ensuring reliable analysis and interpretation of the multitemporal data. It underscores the importance of finding effective solutions to standardize coverage areas and facilitate the seamless integration of information from different periods.

To tackle this challenge, one approach entails identifying matching coordinates across multitemporal images. The "rasterio" package in Python proves valuable here, allowing us to read image data and metadata to determine the bounding box of each image tile. A bounding box serves to define the spatial extent of a dataset, describing its geographic coverage. However, pinpointing specific coordinates within large image tiles with 16,000x16,000 for 2018 and 40,000x40,000 for 2021 can be computationally demanding and time-consuming. This involves iterating through all georeferenced pixels to find matches, which becomes increasingly daunting with larger datasets, such as a 95GB multitemporal image of Addis Ababa.

Alternatively, employing shapefiles offers a solution to standardize coverage areas. Shapefiles, a simple format for storing geometric and attribute data of geographic features, prove instrumental in this endeavor. By utilizing shapefiles containing the boundaries of Addis Ababa and its administrative divisions ("wordas"), we can project these boundaries onto the images to crop out the respective areas. This process yields co-registered image tiles or patches. Tools like Global Mapper, adept at visualizing both multitemporal data and shapefiles, streamline this process, ensuring consistency across images and creating the co-registered image.



Figure 10: Boundaries of Addis Ababa and its administrative divisions ("wordas")

Aligning images based on shapefile boundaries mitigates the challenges posed by varying coverage areas, facilitating reliable co-registration. This approach enables the integration of data from different periods for analysis or visualization, ultimately enhancing the utility of multitemporal aerial photos.

3.5. Resample Image

The aerial images taken at different points in time exhibit some differences. The 2018 image has a spatial resolution of 30cm, while the 2021 image has a spatial resolution of 10cm. This difference can significantly affect change detection. In an image, the distance that a pixel represents is referred to as spatial resolution. For example, NASA's Landsat collects imagery at a 15-meter resolution, meaning each pixel represents a 15m-by-15m area on the ground.

A 10cm resolution photo captures finer details compared to a 30cm resolution photo. Smaller features, such as individual plants or small objects, will be visible in the 10cm aerial photo but may be indistinguishable in the 30cm aerial image. Changes that are small in scale may be detectable in the 10cm image but not in the 30cm image. Consequently, relying on the 30cm image for areas where fine details matter can result in incomplete change detection.

To address the resolution differences, nearest neighbor resampling is performed on the 10cm image to match the 30cm resolution by Arcgis. Nearest neighbor resampling is a technique used to resize images by assigning the value of the nearest pixel to each point in the resized image. This method is simple and preserves the original values of the pixels, which is important for maintaining the integrity of the data.

3.6. Image Patch/Tiles

To create the image tiles the research employed a rigorous strategy to produce co-registered image tiles. First trimmed the multitemporal images according to the administrative boundaries of the area of interest. This step ensured that each tile captured a specific geographic region within Addis Ababa, thereby facilitating focused analysis and it allows us to reduce the computational resource required to train the proposed model.



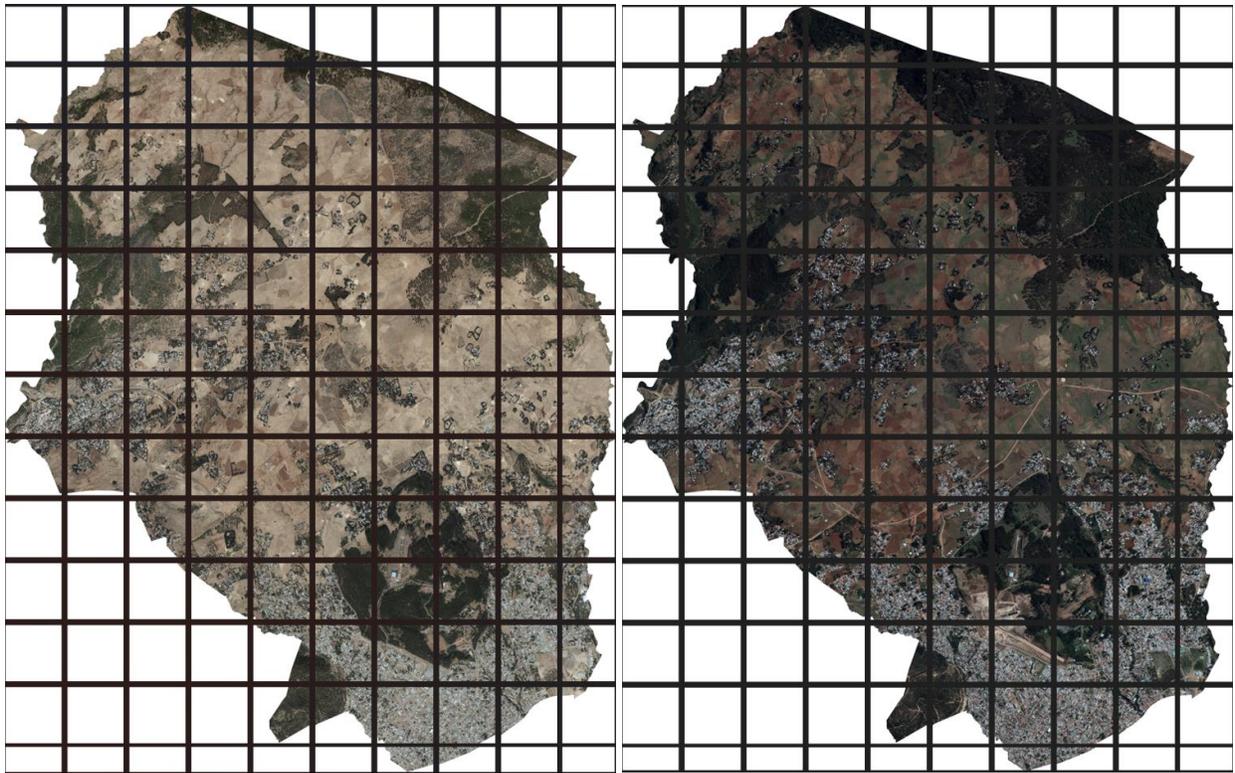
2018 aerial photography



2021 aerial photography

Figure 10: Figure 0 12: Cropped image based on administrative boundaries yeka subcity worda 10

Once the images were cropped to conform to the administrative boundaries, we transitioned to the next phase of the process with a cohesive set of co-registered images. Building upon this foundation, I opted for a systematic approach by uniformly cropping the images at regular intervals of 440 pixels. This deliberate choice served to standardize the procedure, ensuring a harmonized structure across all tiles.



2018 aerial photography

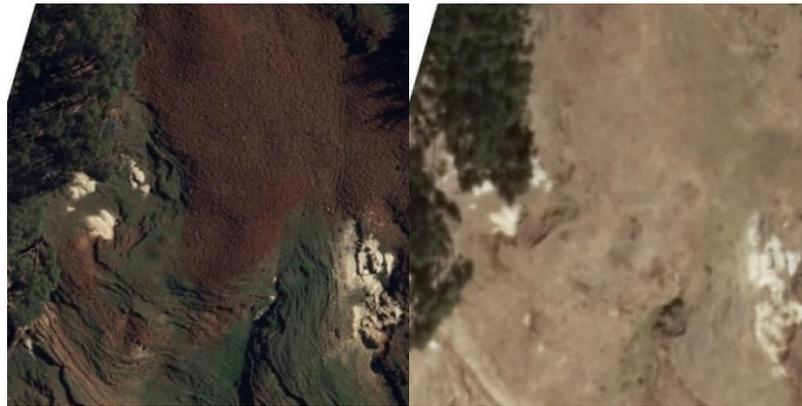
2021 aerial photography

Figure 11: An illustration of image tile of the image in every 440 pixels

Following these guidelines allowed me to produce a collection of co-registered image tiles that accurately depicted different areas of yeka subcity worda 10 at various points in time. This methodical process improves the multitemporal data's dependability and interpretability for change detection.

Table 2: Description of the total number of image tiles

Date	Total number of Image tiles	For 2018	For 2021
2018	6000 image tiles	3000	3000
2021			



A

B



A

B



A

B

Figure 12: Sample of co-registered multitemporal aerial photographs

A) 2018 aerial photographs, B) 2021 aerial photographs

3.7. Neural Network

Before going in-depth in feature extraction, what is neural Artificial neural networks, or ANNs, draw inspiration from the structure and function of biological brains, albeit in a simplified form adapted for computer processing. While not a perfect emulation, ANNs feature interconnected units called neurons and activation functions, mimicking the behavior of biological neurons. While a single neuron may not achieve much on its own, the collective power of hundreds, thousands, or even more neurons, with their intricate interconnections, often yields superior results that surpass those of other machine learning techniques[58].

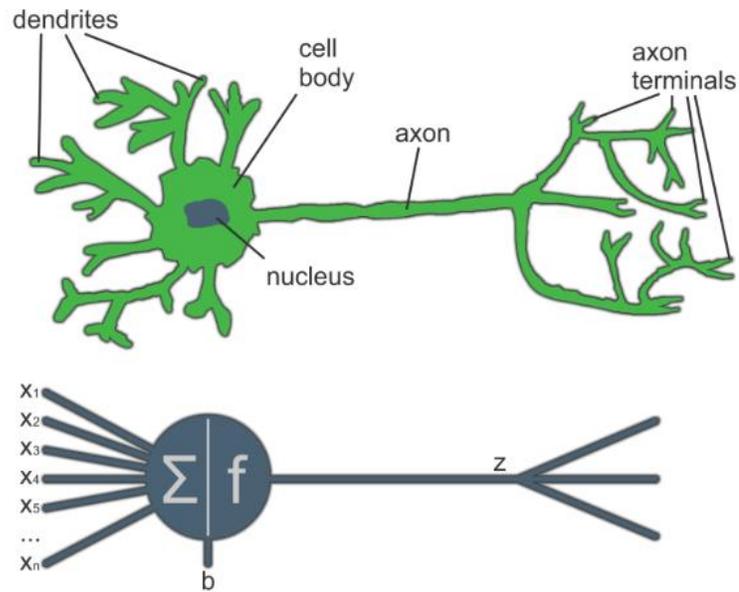


Figure 13: Contrasting an artificial neuron with a biological neuron[58]

3.8. Feature Extraction

3.8.1. Autoencoder (AE)

An Autoencoder trains the neural network in an unsupervised manner by utilizing the encoding and decoding process[59], particularly beneficial when dealing with unlabeled data in a given problem domain. One of its primary advantages lies in its ability to effectively reduce dimensionality, a crucial aspect in numerous remote sensing applications[20]. By compressing the input data into a lower-dimensional representation, AE facilitates the extraction of essential features while minimizing the computational burden associated with high-dimensional data.

Moreover, in remote sensing applications, AE offers another significant advantage by obviating the need for labeled data. This means that instead of relying on meticulously labeled datasets, which typically require input from domain experts, AE can autonomously learn meaningful representations directly from the raw data. This not only streamlines the development process but also alleviates the challenges associated with labeling vast amounts of data, making it more feasible to tackle large-scale remote sensing tasks efficiently.

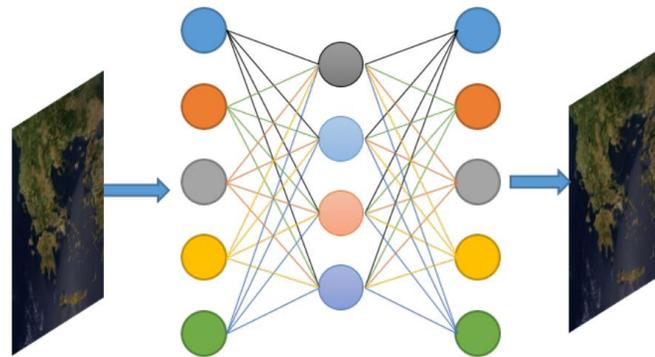


Figure 14: The architecture of the conventional autoencoder and reconstructed image

Typical single-layer autoencoder structure: In order to provide an output that is as near to the input as feasible, this one-hidden layer structure learns the optimal compressed form [60]. For the change this research will utilize a Convolutional Autoencoder (CAE) to transform input data into a compact, semantically rich representation before reverting it to its original form during decoding, ensuring faithful reconstruction of the initial input. Unlike traditional autoencoders that rely on fully connected layers, CAEs employ the encoder and decoder to comprise convolutional and deconvolutional layers, respectively, in place of completely linked layers[61].

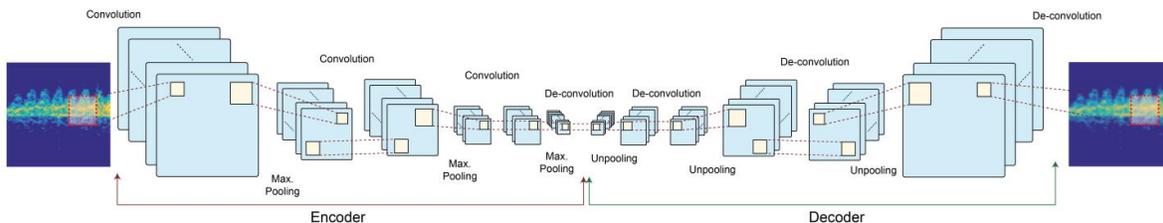


Figure 15: CAE architecture showing convolutional and deconvolutional layers [61]

3.9.Encoder

3.9.1. Convolutional Neural network (CNN)

The human visual brain served as a major source of inspiration for the CNN network. CNNs are specialized Deep Neural Network (DNN) designs that employ convolutions in several layers rather than the more conventional matrix multiplications. They are a great tool for processing data that is sampled often, including 2D and 3D photography. Convolutional layers, pooling, nonlinear activations, and fully linked layers are the four primary layer types found in CNN architectures for image processing applications [86, 88]. Without requiring human input, this type of primary layer may automatically extract important information from the input. Among the most well-known CNN variants are Residual Neural Network (ResNet), AlexNet, and Visual Geometry Group (VGG) [63]. Together, these variations demonstrate CNNs' exceptional accuracy and efficiency in interpreting complicated visual data.

Before CNNs were developed, combining the image pixels into single large vector was the standard procedure method for processing image inputs. However, using this method meant losing a lot of spatial information about how close pixels were to each

other. Although dense networks can theoretically detect proximity of pixels during training, using an architecture specifically designed to capture spatial connectivity could significantly improve image processing capabilities

The fundamental concept behind convolutional networks involves connecting neurons exclusively to pixels within a small region of the image. By confining a neuron's inputs to a specific sub-region, the computations it performs during gradient descent and weight updates inherently become a function of that particular image segment. This localized connectivity allows for the systematic traversal of input windows across the entire image, enabling distinct sets of neurons to process various regions independently. Analogous to dense layers, where multiple neurons process identical inputs, convolutional layers facilitate the deployment of numerous neurons for each window of connections, thereby accommodating the diverse processing requirements of individual spatial regions. For example, it would be fantastic to use the edge detection skills trained by these neurons across the entire input image. Although we are unable to directly regulate the function that a particular neuron learns, but by using weight tying we can apply the same function to the entire image.

Convolutional networks are inherently well-suited for image processing assignments because of their capacity to maintain spatial information and efficiently leverage shared functions across the image domain. Unlike the conventional approach of treating images as single large vectors, CNNs capitalize on spatial proximity and shared functions to achieve superior performance in various image processing tasks.

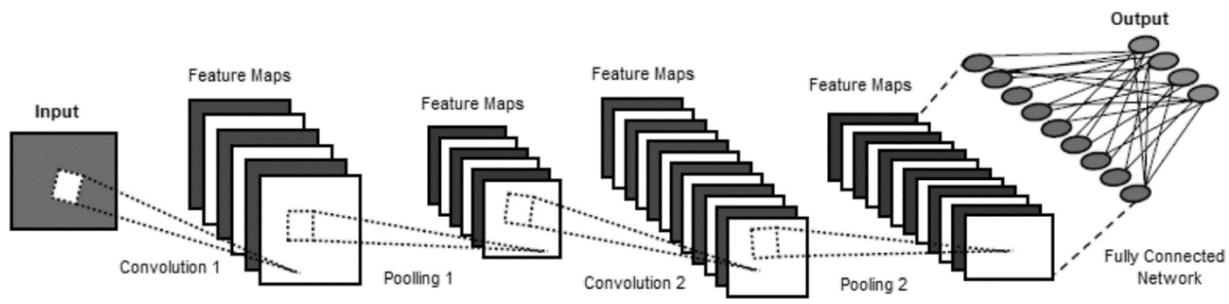


Figure 16: An illustration of a convolutional neural network with several pooling and convolution layers[63]

3.9.2. Convolutional Layer

The convolution layer serves as the fundamental element of CNNs, bearing the primary computational burden. Here, a kernel moves across the input, starting from the left and progressing to subsequent layers, performing a dot product with the input at each step. This process generates an output feature map where convolutional neural layers can identify specific features within an image. Filters in the convolutional layer convolve with the receptive field of the input picture in a sliding window fashion, making it easier to learn properties unique to the data. Initially, basic features like lines, edges, and corners are grasped in the early layers, while deeper layers uncover more abstract features[61].

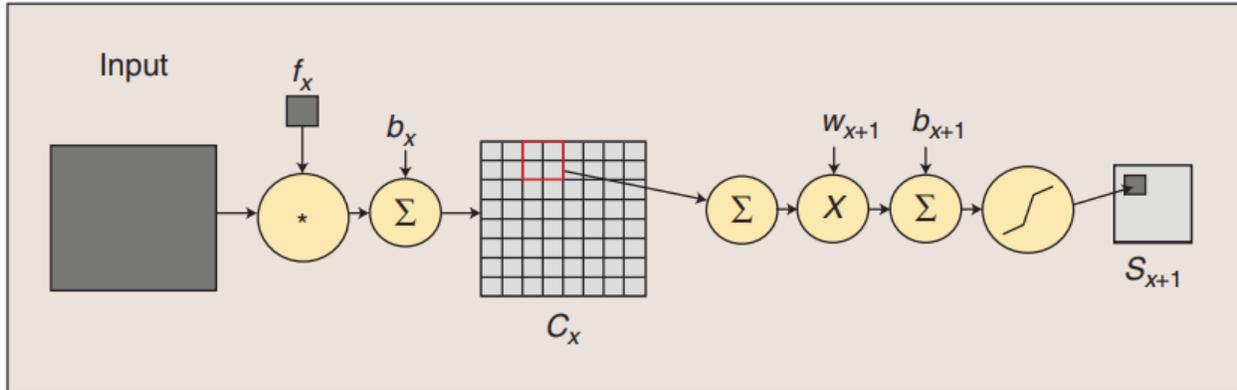


Figure 17: The convolution and subsampling process involves the following steps: The convolution and subsampling process involves the following steps: Initially, a trainable filter f_x is applied to the input, which can be an image in the first stage or a feature map in subsequent stages, resulting in the convolution layer C_x . Then, a trainable bias b_x is added. During subsampling, another trainable bias b_{x+1} , is incorporated, followed by weighting with a scalar w_{x+1} . The neighborhood (four pixels) is summed, and the result is passed through a sigmoid function to produce the feature map S_{x+1} , which is approximately half the size of the original[64].

3.9.3. Pooling Layer

Pooling is a feature seen in the majority of CNN designs. A popular method that primarily aims to minimize the feature map size is pooling. Downsampling of the input feature map is done by the pooling layer, which usually comes after the convolutional layer. Condensing information and transforming the raw data into a more understandable format is its main objective. There are two primary benefits of using pooling layers. First, because the relative locations of the elements that comprise the local pattern might vary greatly, locating the features with comparable local positions gives better dependability. Secondly, it allows for dimensionality reduction in feature representation without introducing additional parameters, thus considerably decreasing both computation time and the overall network's parameter count[65].

There are two primarily used pooling max and average pooling. Max pooling was widely used during the exploration. Max-pooling layers, which take the maximum value in a certain area and pull it into a singular value. Therefore, it speeds up training by increasing the neural network's performance.

3.9.10. Activation Function

A neuron's activation function indicates what it will produce from a certain combination of inputs. It is employed to ascertain if the neuronal output will ignite or not. There are basically two types

of action functions depending on the function they map there result between 0 and 1 and -1 and 1. Non-linear function and linear activation. The result of a linear function, which indicates that the output value is not restricted to a range, is a linear activation function. Rectified Linear Unit (ReLU), a non-linear activation function, will be used in this study. Currently, the most often utilized activation function worldwide is the Rectified Linear Unit, or ReLU. Considering that deep learning and convolutional neural networks both use it. Rectified Linear Unit, or ReLU, activation function sets its output equal to the input x but clips it at 0 from the negative side. If the input x is less than or equal to 0, the output y is set to 0; otherwise, y equals x . Despite its simplicity, ReLU is widely adopted due to its efficiency and speed. While the sigmoid activation function is simpler than ReLU, it's computationally more demanding. ReLU closely resembles a linear activation function, yet it maintains nonlinearity, primarily because of the bend it exhibits after 0. This fundamental characteristic makes ReLU highly effective[58].

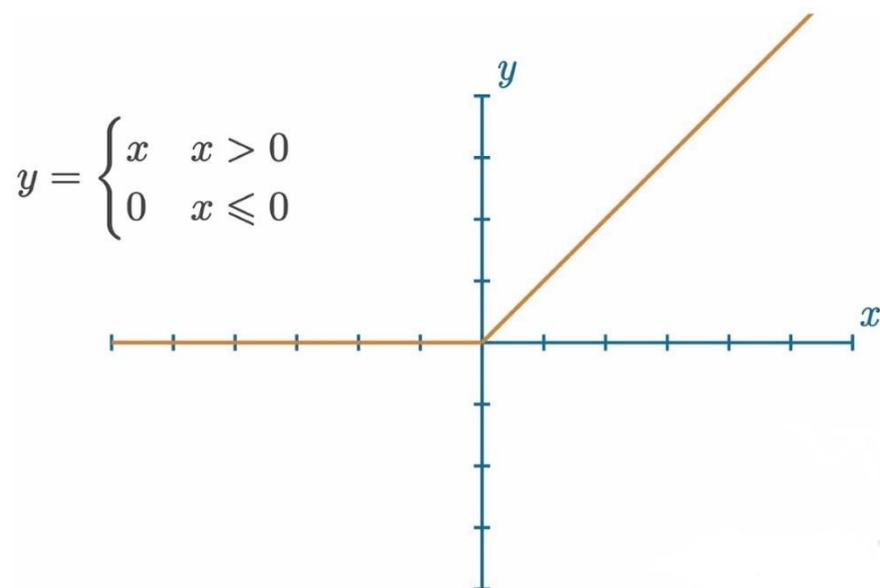


Figure 18: ReLU activation function graph

3.10. Decoder

3.10.10. Deconvolution layer

In order to match the resolution of the original input features, the deconvolution layer serves to recover the quantity of features retrieved by earlier convolutional and pooling layers. This layer offers several advantages: it enables upsampling during the training process itself, and it helps mitigate any loss incurred during the feature resampling procedure[65].

The feature maps are up-sampled and reconstructed using the deconvolutional layers. Deconvolution operates in the opposite direction of conventional convolution and is sometimes known as a convolution that is transposed or convolution with fractionally strided [66]. We use deconvolution as the decoding layer of a convolutional encoder in order to recover the geometry of the original feature map in the encoder stage.

3.11. Loss function

In autoencoder are relay on the reconstructed representation. Essentially Autoencoders are a class of algorithms primarily focused on acquiring a meaningful representation of data through the process of effectively reconstructing a given set of input observations[67]. Hence, there is a need a role that measure the disparity between the input x_i and its corresponding output x'_i .so the concept is to find weights that give the smallest reconstruction error. The most common use in autoencoder is Mean Squared Error (MSE). This Reconstruction Error (RE) metric show us how good or bad CAE able to reconstruct the multitemporal image.

$$RE = MSE = \frac{1}{M} \sum_{i=1}^M |x_i - x'_i|^2$$

Equation 5 : MSE Equation

3.12. Euclidean distance

The distance between two points is known as the Euclidean distance. In other words, the Euclidean distance in Euclidean space is defined as the length of the line segment that separates two locations. The Euclidean distance, often known as the Pythagorean distance, may be computed using Pythagoras' theorem and the coordinate points. The Euclidean distance formula may be used to determine a line segment's distance. Let's say that the two-dimensional coordinate plane has two points, let's say (x_1, y_1) and (x_2, y_2) .

Thus, the Euclidean distance formula is given by:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

Equation 6: Euclidean Distance Equation

In this case, the coordinates of the first point are x_1, y_1 , while the second point is x_2, y_2 . "d" represents the Euclidean distance.

The distance between each multitemporal image's feature vectors at each corresponding location is computed for the purpose of change detection.

3.13. Binary Change Map

To generate the binary change map, we used a popular method called a method that is Otsu threshold. An Otsu threshold algorithm to automatically find the optimal threshold intensity. Finding the threshold intensity that best divides an image into the foreground and background classes is how the Otsu technique operates.

Chapter Four

4. Result

4.1 Overview

In this discussion, we present a comprehensive evaluation of our proposed application of Convolutional Autoencoders (CAEs) for change detection in multitemporal images. CAEs, a specialized subclass of CNNs, are particularly powerful for unsupervised learning tasks. These autoencoders are widely utilized for image reconstruction and feature extraction because of their capability to learn compact and efficient image representations. Our evaluation centers on the binary change maps generated from the features extracted by CAEs. These maps serve as crucial tools for identifying changes between images captured at different times, harnessing the robust representation learning capabilities inherent in CAEs.

4.2. Summary of Dataset Description

The dataset encompasses 6000 meticulously prepared image tiles tailored for the training phase. These tiles represent diverse land features crucial for training the model effectively, including urban settlements, barren land, grasslands, roads, dirt roads, and various others. Due to the unavailability of labeled data, our work is constrained to utilizing freely accessible multi-temporal images sourced from Google Earth.

Data Composition

The data set is distributed as follows:

Training Data:

- Aerial photo: 6000 tiles dated between 2018 and 2021
- Google Earth Data: 4500 tiles dated between 2013 and 2022

4.3.Result

In the results analysis section, we present the outcomes of applying our CAE-based methodology to the dataset. This includes quantitative metrics such as accuracy, precision, recall, and F1-score, which provide a detailed understanding of the model's performance.

Performance measures from the model include:

- Average Precision = 0.7927
- Average Recall = 0.7911
- Average F1 score = 0.7948
- Average Accuracy = 0.8989
- MSE = 0.002

```
# Compute accuracy metrics
precision = precision_score(ground_truth_map_flat, generated_map_flat, average='macro', zero_division=0)
recall = recall_score(ground_truth_map_flat, generated_map_flat, average='macro', zero_division=0)
f1 = f1_score(ground_truth_map_flat, generated_map_flat, average='macro', zero_division=0)
accuracy = accuracy_score(ground_truth_map_flat, generated_map_flat)
```

```
print(f"Average Precision: {average_precision:.4f}")
print(f"Average Recall: {average_recall:.4f}")
print(f"Average F1-score: {average_f1_score:.4f}")
print(f"Average Accuracy: {average_accuracy:.4f}")
```

```
Average Precision: 0.7927
Average Recall: 0.7971
Average F1-score: 0.7948
Average Accuracy: 0.8989
```

Figure 19: The performance of the CAE

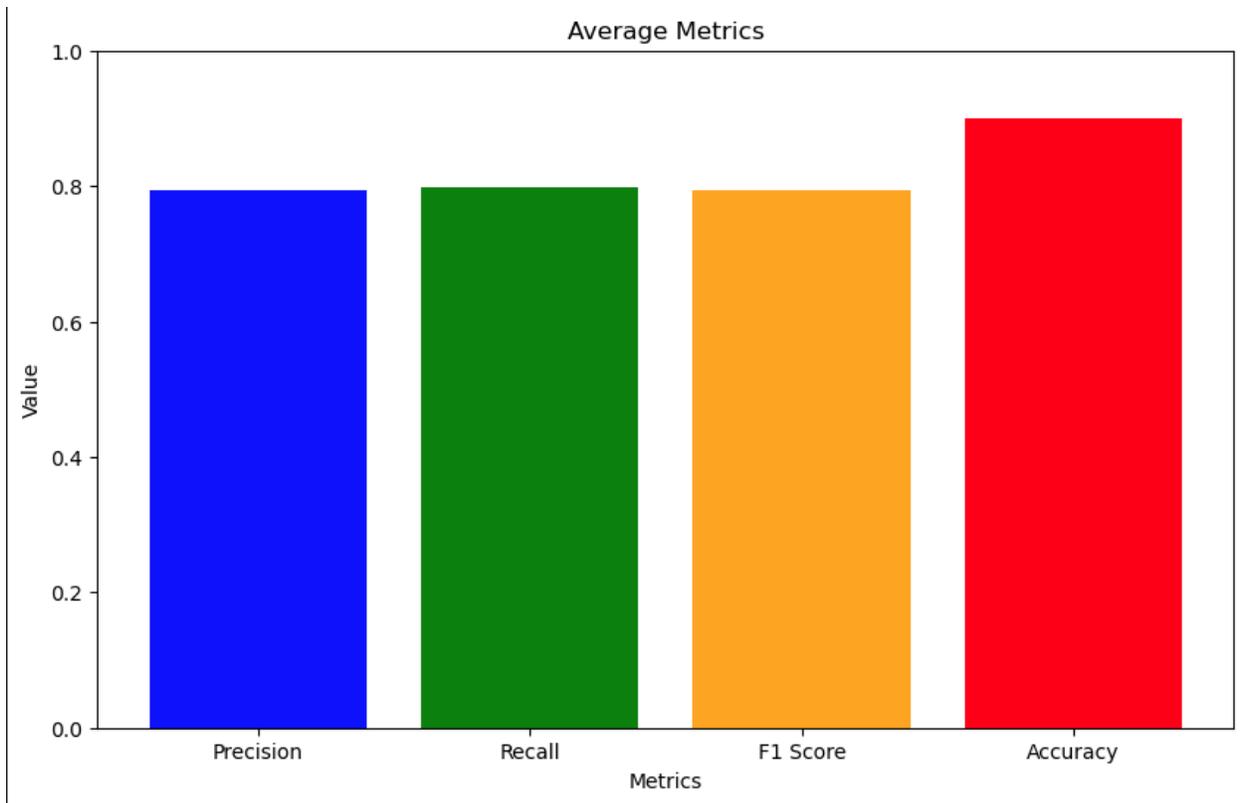


Figure 20: Graph represent of the result

Table 3:Labels for the figures

I: Ground truth label dataset for Google earth	IV: 2018 aerial image
II: Generated binary change map for Google earth dataset	V: 2021 aerial image
II: 2013 Google earth image tile	VI: Generated binary change map for aerial image
III: 2022 Google earth image tile	

4.3.1. Precision Analysis

The ratio of real changes accurately recognized, or genuine positive detections, to the overall number of positive detections, or both true positives and false positives, is known as precision. An accuracy of 0.7927 in the context of change detection indicates that 79.27% of the changes the model detects are real changes. This suggests a low false positive rate, which suggests that changes detected by the model are probably accurate.



I



II



II



III

from the grounded truth data(I) the lable show that all change has occurred our model predicated all of the change.



I



II



II



III



IV



V



VI

4.3.2. Recall Analysis

The ratio of genuine positive detections to the total number of real changes (true positives + false negatives) is called recall, which is sometimes referred to as sensitivity. With a recall of 0.7971, the model is able to identify 79.71% of real changes. This suggests that there aren't many missing changes in the model.



I



II

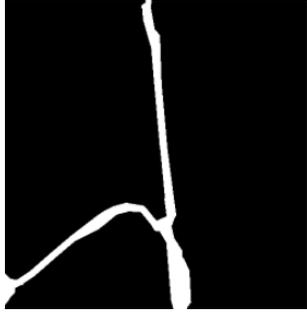


II



III

as we can see from the image the proposed method able to capture most of the change that occurred as we can from the above data it fails to detect the whole trail.



I



II



II



III



IV



V



VI



IV



V



VI

4.3.3. F1-Score Analysis

The F1-score is a statistic that provides a balance between accuracy and recall, calculated as the harmonic mean of the two. It is especially helpful when weighing the trade-off between recall and accuracy. An F1-score of 0.7948 suggests that the model is neither either rigorous nor overly lenient in identifying changes, indicating a reasonable balance between precision and recall.

4.3.4. Accuracy Analysis

The proportion of accurately classified cases—both true positives and true negatives—to all instances is known as accuracy. With an accuracy of 0.8989, the model detects changes and non-changes 89.89% of the time accurately. This high accuracy suggests that the model has good overall performance.



I



II



II



III



I



II



II



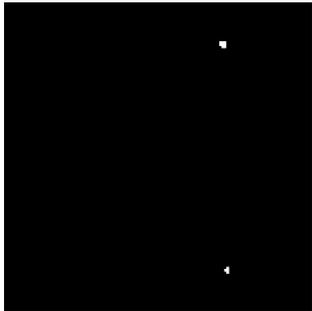
III



IV



V



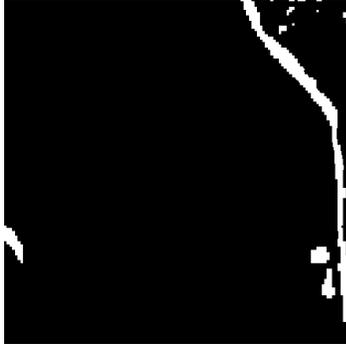
VI



IV



V



VI

4.3.5. MSE Analysis

The small value of MSE, that is, 0.002, evidently gives a good reconstruction of the image from the model. Of course, that is not a proof or guarantee of good change detection since it might reconstruct good but do no damage to the embedded information.

4.4. Discussion on the results

The evaluation metrics for the change detection model using a Deep Learning Convolutional Autoencoder indicate a robust and reliable performance. The model achieved an average precision of 0.7927, an average recall of 0.7971, an average F1-score of 0.7948, and an average accuracy of 0.8989. These results demonstrate the model's effectiveness in identifying changes with a balanced approach, ensuring a good trade-off between precision and recall.

Techniques such as early halting and learning rate reduction were used to further improve the model's effectiveness. Early stopping is a regularization strategy that stops training a model when its performance on a validation set no longer improves, hence preventing overfitting. This prevents the model from overfitting to the training set and guarantees good generalization to new data.

Additionally, a dynamic learning rate reduction technique was utilized to adjust the learning rate dynamically during training. This technique reduces the learning rate when the validation performance metric stops improving.

A noteworthy aspect of the study is the consideration of seasonal changes in the aerial imagery. The images used for this study were taken at different times of the year: April for 2018 and June for 2021. These seasonal differences could introduce variability in the data, potentially confounding the detection of actual land cover changes with seasonal variations. This seasonal variability can affect the model's performance, as it may detect changes that are purely seasonal rather than actual changes in land cover.

For this research, we were unable to obtain data taken at the same time each year, which would have been ideal to prevent such seasonal discrepancies. To mitigate this issue in future studies, it is crucial to use data taken during the same seasons across different years. Aligning the timing of image acquisition ensures that any detected changes are more likely to be genuine land cover changes rather than seasonal effects.

Together, these strategies not only enhance the model's convergence and prevent overfitting but also contribute to the robust performance metrics observed, confirming the model's reliability and effectiveness in change detection tasks despite the challenges posed by seasonal variability in the imagery. When comparing our model to other unsupervised change detection methods, such as CNN-based change detection (CD), several limitations become apparent. CNN-based methods can

extract both 3D and 2D features, whereas our model is limited in this capability. Additionally, CNN and RNN-based architectures can perform multi-class change detection, identifying the types of changes occurring and generating a binary change map with 97% accuracy. This is achieved using labeled data, which was not feasible for implementation in our proposed architecture.

Chapter Five

5. Conclusion and Recommendation

5.2. Conclusion

In conclusion, the change detection model leveraging CAE demonstrates solid performance, achieving an accuracy of 89%. This model's efficacy is particularly beneficial for detecting changes over large areas and extensive datasets, where traditional methods may fall short due to scalability and efficiency limitations. DL models, particularly CAE, are well-suited for processing large datasets and extensive geographical areas. Their ability to automatically extract and learn relevant features from the data makes them highly scalable and efficient.

With a precision and recall of 79%, the model reliably distinguishes between changed and unchanged regions. This level of accuracy is significant for large-scale applications where even small improvements in performance can lead to substantial practical benefits. The model's balanced performance, with high precision and recall, ensures that most detected changes are actual changes, and most actual changes are detected. This balance is critical for minimizing false positives and negatives, which is crucial in large-scale monitoring applications.

The practical benefits of this model are vast. For environmental monitoring, the model can efficiently process satellite imagery and other large datasets to detect changes accurately, aiding in deforestation tracking, climate change studies, and wildlife habitat monitoring. In urban planning and development, it can help monitor construction progress, detect illegal land use changes, and assess infrastructure development over vast urban areas. Additionally, the model's capability to detect changes rapidly and accurately is invaluable for disaster management, enabling timely identification of areas affected by natural disasters such as floods, earthquakes, and landslides, thereby facilitating quicker response and recovery efforts.

Future directions for improving the model include enhancing data quality, experimenting with more advanced DL architectures, and applying sophisticated post-processing techniques. Improvements in data quality, including the use of higher-resolution imagery and more diverse datasets, can enhance the model's accuracy and robustness. Advanced architectures and techniques, such as incorporating temporal data or using ensemble methods, could lead to even better performance. Post-processing techniques to refine the model's output can help reduce false positives and negatives, improving the overall reliability of change detection.

The deployment of CAE for change detection presents a powerful approach to managing and analyzing large-scale datasets and geographical areas. With its high accuracy and balanced performance, this model offers significant advantages for applications that require precise and efficient change detection. As technology and data quality continue to improve, the potential for even greater accuracy and applicability of this model will only increase, making it an indispensable tool for modern change detection challenges.

5.3.Recommendation

For future endeavors, it's imperative to address the limitations identified in this study. Collaboration with domain experts is crucial, as their insights can greatly enhance the efficacy of the research. Here are some recommendations and areas for future work:

There is gap that can be addressed for the future works: -

- Incorporate advanced object identification techniques such as You Only Look Once (YOLO) to improve change detection accuracy. YOLO can effectively discern objects like cars and airplanes, preventing them from being mistakenly identified as changes in the scene.
- Integrate models capable of removing shadows from multitemporal images before change detection. Shadows can often obscure true changes, leading to inaccuracies in detection. By pre-processing images to eliminate shadows, the model's accuracy and reliability can be significantly enhanced.
- Develop a user-friendly web application to facilitate the labeling of multitemporal images by other remote sensing experts. This application would streamline the data annotation process, allowing for efficient creation of labeled datasets. Additionally, it would foster collaboration and knowledge-sharing within the remote sensing community.

Reference

- [1] D. A. Mengistu, D. K. Waktola, and M. Woldetsadik, "Detection and analysis of land-use and land-cover changes in the Midwest escarpment of the Ethiopian Rift Valley," *J. Land Use Sci.*, vol. 7, no. 3, pp. 239–260, 2012, doi: 10.1080/1747423X.2011.562556.
- [2] G. Muriuki, L. Seabrook, C. McAlpine, C. Jacobson, B. Price, and G. Baxter, "Land cover change under unplanned human settlements: A study of the Chyulu Hills squatters, Kenya," *Landsc. Urban Plan.*, vol. 99, no. 2, pp. 154–165, 2011, doi: <https://doi.org/10.1016/j.landurbplan.2010.10.002>.
- [3] M. Anteneh, "Analysis of Land Use/Land Cover Change and Its Implication On Natural Resources of the Dedo Watershed, Southwest Ethiopia.," *ScientificWorldJournal.*, vol. 2022, p. 6471291, 2022, doi: 10.1155/2022/6471291.
- [4] B. Chang and G. Zhang, "An unsupervised approach based on geometrical structures to automatic change detection in multitemporal SAR images," *Tien Tzu Hsueh Pao/Acta Electron. Sin.*, vol. 39, no. 9, pp. 2125–2129, 2011.
- [5] Y. Han and J. Wang, "Application of Convolutional Neural Networks in Remote Sensing Image Classification," *Proc. - 2019 2nd Int. Conf. Saf. Prod. Informatiz. IICSPI 2019*, pp. 279–282, 2019, doi: 10.1109/IICSPI48186.2019.9096058.
- [6] K. L. De Jong and A. Sergeevna Bosman, "Unsupervised Change Detection in Satellite Images Using Convolutional Neural Networks," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2019-July, no. July, pp. 1–8, 2019, doi: 10.1109/IJCNN.2019.8851762.
- [7] A. Singh, "Review Article: Digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989, doi: 10.1080/01431168908903939.
- [8] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Computer Vision and Image Understanding Multitask learning for large-scale semantic change detection," pp. 1–12, 2019, [Online]. Available: <https://www.elsevier.com/open-access/userlicense/1.0/>
- [9] Y. Zhan *et al.*, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849,

2017.

- [10] W. A. Malila, "Change Vector Analysis: an Approach for Detecting Forest Changes With Landsat.," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, 1980, pp. 326–336. [Online]. Available: <https://api.semanticscholar.org/CorpusID:35554562>
- [11] J. R. Shepard, "A concept of change detection," in *Photogrammetric Engineering*, 1964, vol. 30, no. 4, pp. 648–651. [Online]. Available: <https://api.semanticscholar.org/CorpusID:209442693>
- [12] D. Lu, P. Mausel, E. Brondízio, and E. Moran, "Change Detection Techniques," *Int. J. Remote Sens.*, vol. 25, 2004.
- [13] N. Audebert, B. Le Saux, and S. Lefèvre, "Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 140, pp. 20–32, 2018, doi: 10.1016/j.isprsjprs.2017.11.011.
- [14] C. Benedek and T. Szirányi, "Change detection in optical aerial images by a multilayer conditional mixed Markov model," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3416–3430, 2009, doi: 10.1109/TGRS.2009.2022633.
- [15] N. H. E. A. M. T. D. T. M. T. M. D. A. Dagnenet Sultan Atsushi Tsunekawa and K. Ebabu, "Analyzing the runoff response to soil and water conservation measures in a tropical humid Ethiopian highland," *Phys. Geogr.*, vol. 38, no. 5, pp. 423–447, 2017, doi: 10.1080/02723646.2017.1302869.
- [16] J. M. Maitima *et al.*, "The linkages between land use change , land degradation and biodiversity across East Africa," *African J. Environ. Sci. Technol.*, vol. 3, no. 10, pp. 310–325, 2009, doi: 10.5897/AJEST08.173.
- [17] N. Ramankutty and J. A. Foley, "Estimating historical changes in global land cover: Croplands from 1700 to 1992," *Global Biogeochem. Cycles*, vol. 13, no. 4, pp. 997–1027, 1999, doi: 10.1029/1999GB900046.
- [18] J. Slater and R. Brown, "Changing landscapes: Monitoring Environmentally Sensitive

- Areas using satellite imagery,” *Int. J. Remote Sens.*, vol. 21, no. 13–14, pp. 2753–2767, 2000, doi: 10.1080/01431160050110278.
- [19] T. A. Stone and P. Lefebvre, “Using multi-temporal satellite data to evaluate selective logging in para, brazil,” *Int. J. Remote Sens.*, vol. 19, no. 13, pp. 2517–2526, 1998, doi: 10.1080/014311698214604.
- [20] J. E. Ball, D. T. Anderson, and C. S. Chan, “Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community,” *J. Appl. Remote Sens.*, vol. 11, no. 04, p. 1, 2017, doi: 10.1117/1.jrs.11.042609.
- [21] R. Das Kangabam, M. Selvaraj, and M. Govindaraju, “Spatio-temporal analysis of floating islands and their behavioral changes in Loktak Lake with respect to biodiversity using remote sensing and GIS techniques,” *Environ. Monit. Assess.*, vol. 190, no. 3, 2018, doi: 10.1007/s10661-018-6485-x.
- [22] D. Lu, P. Mausel, E. Brondízio, and E. Moran, “Change detection techniques,” *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, 2004, doi: 10.1080/0143116031000139863.
- [23] J.-P. Thirion and G. Calmon, “Deformation analysis to detect and quantify active lesions in three-dimensional medical image sequences,” *IEEE Trans. Med. Imaging*, vol. 18, no. 5, pp. 429–441, 1999, doi: 10.1109/42.774170.
- [24] L. Lemieux, U. C. Wiesmann, N. F. Moran, D. R. Fish, and S. D. Shorvon, “The detection and significance of subtle changes in mixed-signal brain lesions by serial MRI scan matching and spatial normalization,” *Med. Image Anal.*, vol. 2, no. 3, pp. 227–242, Sep. 1998, doi: 10.1016/s1361-8415(98)80021-2.
- [25] D. Rey, G. Subsol, H. Delingette, and N. Ayache, “Automatic detection and segmentation of evolving processes in 3D medical images: Application to multiple sclerosis,” *Med. Image Anal.*, vol. 6, no. 2, pp. 163–179, 2002, doi: [https://doi.org/10.1016/S1361-8415\(02\)00056-7](https://doi.org/10.1016/S1361-8415(02)00056-7).
- [26] M. Bosc, F. Heitz, J.-P. Armspach, I. Namer, D. Gounot, and L. Rumbach, “Automatic change detection in multimodal serial MRI: application to multiple sclerosis lesion

- evolution,” *Neuroimage*, vol. 20, no. 2, pp. 643–656, 2003, doi: [https://doi.org/10.1016/S1053-8119\(03\)00406-3](https://doi.org/10.1016/S1053-8119(03)00406-3).
- [27] L. E. N., N. E. N., K. D. T., and N. George, “Technique to Measure 3D Work-of-Fracture of Concrete in Compression,” *J. Eng. Mech.*, vol. 125, no. 6, pp. 599–605, Jun. 1999, doi: 10.1061/(ASCE)0733-9399(1999)125:6(599).
- [28] G. Nagy, T. Zhang, W. R. Franklin, E. Landis, E. Nagy, and D. T. Keane, “Volume and Surface Area Distributions of Cracks in Concrete,” in *Visual Form 2001*, 2001, pp. 759–768.
- [29] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, “Automatic change detection of driving environments in a vision-based driver assistance system,” *IEEE Trans. Neural Networks*, vol. 14, no. 3, pp. 646–657, 2003, doi: 10.1109/TNN.2003.811353.
- [30] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: real-time tracking of the human body,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, 1997, doi: 10.1109/34.598236.
- [31] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, “Deep Learning-Based Change Detection in Remote Sensing Images: A Review,” *Remote Sens.*, vol. 14, no. 4, pp. 1–40, 2022, doi: 10.3390/rs14040871.
- [32] S. Liu, L. Bruzzone, F. Bovolo, and P. Du, “Unsupervised hierarchical spectral analysis for change detection in hyperspectral images,” *Work. Hyperspectral Image Signal Process. Evol. Remote Sens.*, pp. 1–4, 2012, doi: 10.1109/WHISPERS.2012.6874245.
- [33] L. Khelifi and M. Mignotte, “Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis,” *IEEE Access*, vol. 8, no. Cd, pp. 126385–126400, 2020, doi: 10.1109/ACCESS.2020.3008036.
- [34] D. B. Morris and P. H. A. Martin Kaye, “Remote sensing of the environment,” *Endeavour*, vol. 32, no. 117, pp. 117–121, 1973.
- [35] D. R. Thompson *et al.*, “Retrieval of Atmospheric Parameters and Surface Reflectance from Visible and Shortwave Infrared Imaging Spectroscopy Data,” *Surv. Geophys.*, vol.

- 40, no. 3, pp. 333–360, 2019, doi: 10.1007/s10712-018-9488-9.
- [36] X. Ma, A. Huete, N. N. Tran, J. Bi, S. Gao, and Y. Zeng, “Sun-Angle Effects on Remote-Sensing Phenology Observed and Modelled Using Himawari-8,” *Remote Sens.*, vol. 12, no. 8, 2020, doi: 10.3390/rs12081339.
- [37] S. N. Araya, A. Fryjoff-Hung, A. Anderson, J. H. Viers, and T. A. Ghezzehei, “Advances in soil moisture retrieval from multispectral remote sensing using unoccupied aircraft systems and machine learning techniques,” *Hydrol. Earth Syst. Sci.*, vol. 25, no. 5, pp. 2739–2758, 2021, doi: 10.5194/hess-25-2739-2021.
- [38] R. D. Macleod and R. G. Congalton, “A quantitative comparison of change-detection algorithms for monitoring eelgrass from remotely sensed data,” *Photogramm. Eng. Remote Sensing*, vol. 64, no. 3, pp. 207–216, 1998.
- [39] E. F. Lambin and D. Ehrlich, “Land-cover changes in Sub-Saharan Africa (1982-1991): Application of a change index based on remotely sensed surface temperature and vegetation indices at a continental scale,” *Remote Sens. Environ.*, vol. 61, no. 2, pp. 181–200, 1997, doi: 10.1016/S0034-4257(97)00001-1.
- [40] X. Dai, “The effects of image misregistration on the accuracy of remotely sensed change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 5 PART 1, pp. 1566–1577, 1998, doi: 10.1109/36.718860.
- [41] S. Boles and D. L. Verbyla, “Comparison of three AVHRR-based fire detection algorithms for interior Alaska,” *Remote Sens. Environ.*, vol. 72, pp. 1–16, 2000, [Online]. Available: <https://api.semanticscholar.org/CorpusID:462767>
- [42] L. Zhu, J. Suomalainen, J. Liu, J. Hyypä, H. Kaartinen, and H. Haggren, “A Review: Remote Sensing Sensors,” *Multi-purposeful Appl. Geospatial Data*, 2018, doi: 10.5772/intechopen.71049.
- [43] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1717–1724, 2014, doi: 10.1109/CVPR.2014.222.

- [44] J. Naranjo *et al.*, “No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title,” *J. Algorith.*, vol. 12, no. 1, pp. 579–587, 2016, [Online]. Available: <http://jurtek.akprind.ac.id/bib/rancang-bangun-website-penyediaan-layanan-weblog>
- [45] Y. Liu, L. Liu, and Y. Yan, “Network Topology Change Detection Based on Statistical Process Control,” *ACM Int. Conf. Proceeding Ser.*, pp. 145–151, 2020, doi: 10.1145/3409501.3409532.
- [46] X. Li *et al.*, “Big Data in Earth system science and progress towards a digital twin,” *Nat. Rev. Earth Environ.*, vol. 4, 2023, doi: 10.1038/s43017-023-00409-w.
- [47] *Deep Learning for the Earth Sciences*. 2021. doi: 10.1002/9781119646181.
- [49] L. Mou and X. X. Zhu, “A recurrent convolutional neural network for land cover change detection in multispectral images,” *Int. Geosci. Remote Sens. Symp.*, vol. 2018-July, pp. 4363–4366, 2018, doi: 10.1109/IGARSS.2018.8517375.
- [50] W. Wiratama, J. Lee, and D. Sim, “Change Detection on Multi-Spectral Images Based on Feature-level U-Net,” *IEEE Access*, vol. 8, pp. 12279–12289, 2020, doi: 10.1109/ACCESS.2020.2964798.
- [51] S. T. Seydi, M. Hasanlou, and M. Amani, “A new end-to-end multi-dimensional CNN framework for land cover/land use change detection in multi-source remote sensing datasets,” *Remote Sens.*, vol. 12, no. 12, 2020, doi: 10.3390/rs12122010.
- [52] G. J. Awcock and R. Thomas, “Image Preprocessing,” in *Applied Image Processing*, London: Macmillan Education UK, 1995, pp. 90–125. doi: 10.1007/978-1-349-13049-8_4.
- [53] J. L. Morgan, S. E. Gergel, and N. C. Coops, “Aerial photography: A rapidly evolving tool for ecological management,” *Bioscience*, vol. 60, no. 1, pp. 47–59, 2010, doi: 10.1525/bio.2010.60.1.9.
- [54] V. Kristollari and V. Karathanassi, “Change Detection in VHR Imagery With Severe Co-Registration Errors Using Deep Learning: A Comparative Study,” *IEEE Access*, vol. 10,

pp. 33723–33741, 2022, doi: 10.1109/ACCESS.2022.3161978.

APPENDIX

1.Code for CAE

```
def autoencoder():  
    #Define the patch size  
    original_img_size = (target_size, target_size, 3)  
  
    #encoder  
    input_img = Input(shape=original_img_size)  
    x = Conv2D(64, (3, 3), padding='same')(input_img)  
    x = Activation('relu')(x)  
    x = MaxPooling2D((2, 2))(x)  
    x = Conv2D(32, (3, 3), padding='same')(x)  
    x = Activation('relu')(x)  
    x = MaxPooling2D((2, 2))(x)  
    x = Conv2D(16, (3, 3), padding='same')(x)  
    encoded = Activation('relu')(x)  
  
    #decoder  
    x = Conv2DTranspose(32, (4,4), strides=(2,2), padding='same')(encoded)  
    x = Activation('relu')(x)  
    x = Conv2DTranspose(64, (4,4), strides=(2,2), padding='same')(x)  
    x = Activation('relu')(x)  
    x = Conv2D(4, (3, 3), padding='same')(x)  
    decoded = Activation('sigmoid')(x)  
  
    # Compile the model  
    model.compile(optimizer='adam', loss='mean_squared_error')  
  
    return model
```

2.Feature extractor

```
def feat(x_train):  
    feat1 = model1.predict(x_train)  
    feat2 = model2.predict(x_train)  
    feat3 = model3.predict(x_train)  
    feat4 = model4.predict(x_train)  
  
    x1 = tf.image.resize(feat1,[128,128])  
    x2 = tf.image.resize(feat2,[128,128])  
    x3 = tf.image.resize(feat3,[128,128])  
    x4 = tf.image.resize(feat4,[128,128])  
  
    F = tf.concat([x2,x1,x4,x3],3)  
    return F
```

3.Code for loading the data and calculation of Euclidean distance

```
for i in range(0,number_files,1):

    #Print index of the file
    print(f" i is {i}")

    #Access the first date folder
    os.chdir(dir1_path)
    #Read the first date image
    im1 = gdal.Open(dir1_sorted_list[i])
    im1=np.array(im1.ReadAsArray())
    # print(dir1_sorted_list[i])
    #Change the order of the axes to width,height,channels
    im1=im1.transpose(1,2,0)
    # resized_patch1 = cv2.resize(im1, (224, 224))
    #Convert values to [0,1]
    im1=im1/255
    #Expand the dimensions so that the model can read it (batch_size=1)
    im1=np.expand_dims(im1,0)
    os.chdir(dir2_path)
    im2 = gdal.Open(dir2_sorted_list[i])
    im2=np.array(im2.ReadAsArray())
    im2=im2.transpose(1,2,0)

    im2= im2/255
    im2=np.expand_dims(im2,0)

    im1_shape=im1.shape[2]

    if (im1_shape== target_size):
        #Store the file name and define name of the change map
        filename=os.path.basename(dir1_sorted_list[i])
        filename2=filename.split('.')[0] + "change.tiff"
        F1=feat(im1) #Features from image patch 1
        #Calculate the square value of the feature maps
        F1=tf.square(F1)
        F2=feat(im2) #Features from image patch 2
        F2=tf.square(F2)
        #Subtract the feature maps from the 2 dates
        d=tf.subtract(F1,F2)
        d=tf.square(d)
        #Create the change map by summing values from the feature maps
        d=tf.reduce_sum(d,axis=3)

        #Resize the change map
        dis=d.numpy()
        dis=dis.transpose(1,2,0)
        dis = tf.image.resize(dis,[224,224], method="nearest")
        dis=d.numpy()
        dis=np.resize(dis,[224,224])
        #Save the change map
        os.chdir(dir3_path)
        tiff.imwrite(filename2,dis)
```

4. Otsu thresholding

```
#Iterate over the files and create the binary change maps
for i in range (number_files):

    #Print the index of the file
    print(" The file index is %s" %i)

    #Access the directory of the change maps (float)
    os.chdir(dir1_path)
    #Read image
    im1 = gdal.Open(dir1_list[i])
    im1=np.array(im1.ReadAsArray())

    #Store the file name and define the name of the binary change map
    filename=os.path.basename(dir1_list[i])
    filename2=filename.split('.')[0] + "_otsu.png"

    #Access the directory of the binary change maps
    os.chdir(dir2_path)

    #Calculate the Otsu threshold
    if im1.min() == 0 and im1.max()==0:
        val=0
    else:
        val = filters.threshold_otsu(im1[:,:])

    #Create array to store the binary values
    A=np.uint8(np.zeros((224,224)))
    for i1 in range(224):
        for i2 in range(224):
            if im1[i1,i2]>val:
                A[i1,i2]=255
            else:
                A[i1,i2]=0

    cv2.imwrite(filename2, A)
```

