Tunisian Republic
Ministry of Higher Education and Scientific Research
University of Tunis El Manar
National Engineering School of Tunis

Department of Information and Communication Technologies

End of Year Project II

Multimodal Classification Model Exploiting Google Earth Engine Images and OpenStreetMap Footprint

Prepared by:
Achref Kerkeni
Belsem Jemai
Class: 2nd Year TEL

Supervised by:
Mrs. Mariem Zaouali

Academic Year:
2025/2026

Acknowledgements

It is our pleasure to extend to **Mrs. Mariem Zaouali**, Assistant Professor at the Department of Information and Communication Technologies at ENIT, who supervised our final year project, our sincere and heartfelt thanks for her extremely valuable and constructive guidance, support and advice offered throughout the course of our project.

It has been of great importance the immense availability and positive feedback she gave us to help us plan ahead and develop ourselves. It should also be noted how valuable her dedication was to supervise our progress and answer all our questions and guide us whenever any difficulties occurred.

But apart from this technical assistance that we had, there is no way we can overlook the huge level of confidence and motivation that she instilled in all of us in every stage of the process of the project. She was not only a supervisor to us but also a great motivator.

We also extend our sincere thanks to the entire faculty and staff of the **Department of Information and Communication Technologies at ENIT** for providing us with the knowledge, resources, and academic environment necessary to carry out this work. Finally, we wish to express our heartfelt gratitude to all those who, directly or indirectly, contributed to the successful completion of this project.

Abstract

This report presents the work carried out as a second-year engineering end-of-year project at the National Engineering School of Tunis. The project addresses the problem of urban change detection from satellite imagery by proposing a multimodal approach that jointly exploits bi-temporal image pairs and semantic geographic data. The proposed architecture extends an existing image-text fusion model to handle two temporal images through a shared visual encoder and a bi-temporal token sequence, while integrating structured semantic guidance derived from a collaborative geographic database. The system was evaluated on a standard change detection benchmark and produced encouraging results in terms of both detection accuracy and spatial precision. This report describes the project context, the relevant state of the art, the proposed approach, and the experimental results obtained, along with a discussion of current limitations and future perspectives.

Keywords: change detection, remote sensing, multimodal learning, deep learning, satellite imagery, semantic data, urban monitoring.

Résumé

Ce rapport présente le travail réalisé dans le cadre du projet de fin d'année de deuxième année d'ingénierie à l'École Nationale d'Ingénieurs de Tunis. Le projet traite le problème de la détection de changements urbains à partir d'imagerie satellitaire en proposant une approche multimodale qui exploite conjointement des paires d'images bi-temporelles et des données géographiques sémantiques. L'architecture proposée étend un modèle existant de fusion image-texte pour traiter deux images temporelles à travers un encodeur visuel partagé et une séquence de tokens bi-temporelle, tout en intégrant un guidage sémantique structuré issu d'une base de données géographique collaborative. Le système a été évalué sur un benchmark standard de détection de changements et a produit des résultats encourageants en termes de précision de détection et de résolution spatiale. Ce rapport décrit le contexte du projet, l'état de l'art, l'approche proposée et les résultats expérimentaux

obtenus, ainsi qu'une discussion des limites actuelles et des perspectives futures.

Mots-clés : détection de changements, télédétection, apprentissage multimodal, apprentissage profond, imagerie satellitaire, données sémantiques, surveillance urbaine.

Contents

Abstract

List of Figures

List of Tables

List of Acronyms

General Introduction

1 General Context **1**

1.1 Project context 1
1.2 Project objectives 2
1.3 Problem statement 3
1.4 Study and limitations of existing approaches 4
1.5 Proposed solution 6
1.6 Methodology and project management 6
1.7 Conclusion 8

2 Multimodal Change Detection: State of the Art **9**

2.1 Change detection in remote sensing 9
2.2 Deep learning approaches for change detection 10
2.2.1 From convolutional neural networks to Siamese architectures . 10
2.2.2 Transformer-based methods 11
2.3 Multimodal learning in geospatial applications 12
2.3.1 Contrastive Language-Image Pre-training (CLIP) and vision-language models 12
2.3.2 Image-text Encoder Fusion Transformer (IEFT) 13
2.4 Limitations and open challenges 14
2.5 Conclusion 15

3 Proposed Approach: Siamese IEFT with Temporal Fusion and OpenStreetMap (OSM)-Guided Dense Decoding 16
3.1 Data sources used in the project . . . 16
3.1.1 Sentinel-2 imagery from Google Earth Engine . . . 16
3.1.2 LEVIR-CD benchmark dataset . . . 17
3.1.3 OpenStreetMap semantic data . . . 18
3.2 Proposed architecture overview . . . 19
3.3 Shared Siamese visual encoder . . . 20
3.4 Bi-temporal token sequence and TEMP token . . . 21
3.5 Adapted fusion blocks . . . 22
3.6 Local change representation . . . 23
3.7 Prediction heads . . . 23
3.7.1 Global change prediction branch . . . 23
3.7.2 Dense multi-scale decoder . . . 24
3.7.3 Boundary head and deep supervision . . . 24
3.8 OSM semantic guidance integration . . . 24
3.9 Conclusion . . . 26

4 Implementation, Experimental Results, and Analysis 27
4.1 Software environment and project structure . . . 27
4.2 Training configuration and loss design . . . 28
4.3 Export and inference pipeline . . . 29
4.4 Experimental protocol and tested configurations . . . 29
4.5 Experimental results . . . 30
4.5.1 Results on the initial Sentinel-2 phase . . . 30
4.5.2 Results on LEVIR-CD . . . 31
4.5.3 Post-processing with targeted veto . . . 33
4.6 Ablation insights . . . 35
4.7 Discussion and error analysis . . . 35
4.8 Current limitations and future work . . . 36
4.9 Conclusion . . . 36

General Conclusion and Future Perspectives

A Detailed Proposed Architecture

B Implementation Details

C Dataset Examples and Useful Commands

Bibliography

List of Figures

1.1 Original U-Net encoder-decoder architecture with skip connections [1]. 4
1.2 Simplified overview of the proposed multimodal change detection pipeline. 6
1.3 The CRISP-DM process model [5]. 7
1.4 Gantt chart of the project timeline. 7

2.1 Comparison of FC-EF, FC-Siam-diff, and FC-Siam-conc architectures for change detection [7]. 11
2.2 Architecture of the Bitemporal Image Transformer (BIT) [8]. 12
2.3 Overview of the CLIP architecture [13]. 13
2.4 Architecture of the IEFT model 14

3.1 Example of a Sentinel-2 image used during the initial data exploration phase. 17
3.2 Example of a high-resolution LEVIR-CD image used for building change detection. 18
3.3 Overview of the proposed architecture for multimodal bi-temporal urban change detection. 20

4.1 Overview of the project software structure and interactions between main modules. 28
4.2 Example of Sentinel-2 change detection pipeline outputs 31
4.3 Evolution of IoU and Dice/F1 across successive LEVIR-CD versions . 32
4.4 Example prediction on LEVIR-CD 33
4.5 Best-performing patch 33

A.1 Detailed Proposed Architecture . . . . . .

List of Tables

4.1 Summary of key model versions during the Sentinel-2 phase. 30
4.2 Evolution of results on the LEVIR-CD test set across key versions. . . 31
4.3 Effect of targeted veto post-processing on the precision_strict output. 34
4.4 Comparison with representative models reported on LEVIR-CD. . . . 35

B.1 Main project files and their roles. . . . . .

List of Acronyms

BCE	Binary Cross-Entropy
BIT	Bitemporal Image Transformer
BLIP	Bootstrapping Language-Image Pre-training
CLS	Classification Token
CLIP	Contrastive Language-Image Pre-training
CNN	Convolutional Neural Network
CRISP-DM	Cross-Industry Standard Process for Data Mining
CUDA	Compute Unified Device Architecture
DL	Deep Learning
FFN	Feed-Forward Network
FIE	Fusion Intersection Encoder
GEE	Google Earth Engine
GT	Ground Truth
IEFT	Image-text Encoder Fusion Transformer
IoU	Intersection over Union
LEVIR-CD	Learning, Vision and Remote sensing Change Detection dataset
MHSA	Multi-Head Self-Attention
MLP	Multi-Layer Perceptron
NIR	Near-Infrared
OSM	OpenStreetMap
PCA	Principal Component Analysis
RGB	Red, Green, Blue
TEMP	Temporal Token
ViT	Vision Transformer

General Introduction

The rapid pace of global urbanization requires reliable tools to monitor and understand how cities evolve over time. Satellite remote sensing offers the ability to observe large areas repeatedly, providing valuable data for urban planning, environmental assessment, and infrastructure management. However, interpreting satellite images and extracting meaningful information from them remains a complex task, particularly when the goal is to identify changes between two observation dates.

Traditional change detection approaches rely primarily on visual information and often struggle with noise, radiometric variations, and the lack of contextual understanding. These limitations suggest that combining image data with complementary sources of semantic information could lead to more accurate and more interpretable results. Collaborative geographic databases, community-maintained platforms such as OpenStreetMap (OSM) that describe the nature, function, and attributes of urban elements like buildings, roads, and land parcels, represent a promising source of such complementary knowledge.

In this setting, the current work proposes to design a multimodal system for urban change detection by leveraging satellite imagery and structured geographic data. Specifically, the proposed method builds upon an existing state-of-the-art multimodal framework to process pairs of images annotated with textual description in order to generate accurate change maps that are interpretable as well. During this project, several iterations have been performed using different types of data and architectures as progress was being made.

The current report will be structured as follows. In chapter 1, the setting and approach are defined, followed by a literature review in chapter 2. Chapter 3 details the proposed approach, with both the architecture and data. Finally, in chapter 4, the implementation, experimental evaluation, and discussion are presented.

CHAPTER 1

General Context

The present chapter gives a general introduction to the project methodology. It starts with an understanding of how this study fits into the broad area of urban monitoring and geospatial information analysis. The objective of the study is given along with its problem statement. The limitations in the current methodologies used for urban monitoring are discussed briefly, and then the proposed solution is highlighted.

1.1 Project context

Cities are now changing at an unprecedented rate. From new constructions to road expansions, from reduction of green spaces to neighborhood remodeling, it all takes place within a period of just a few years. Recognizing the places and ways of these changes is essential for many reasons. Yet tracking each change individually, particularly when it comes to large-scale regions, is impossible.

Here comes the need for satellite imagery. The satellite takes images of the same area at different points in time, allowing one to see how the area has changed since a particular period. Within the last ten years, the availability of satellite data has greatly improved due to the introduction of open sources that offer free satellite imagery data for the entire world. Such sources enable one to select their area of interest, select different spectral bands, set the dates, cloud cover, and download the final product for use in automated studies on urban development.

In addition to satellite imagery, however, there are other ways to obtain geographic data. Mapping systems that are created and regularly updated by networks of volunteers from around the globe provide detailed accounts of the elements present in specific locations: the buildings, roads, parks, shopping centers, and various other elements of the urban environment. Such databases are constantly being improved and expanded and can be found for many geographic regions, with differing levels of accuracy depending on the region covered. The semantic nature of such geographic data provides a layer of meaning that satellite imagery alone cannot convey.

In fact, the merging of these two kinds of data, satellite imagery analysis and

descriptive data from geographic databases, suggests a promising area of research. It is no longer necessary to depend on only one source of data, but it would be possible to create systems that could recognize urban change visually as well as semantically. This would be especially useful in areas like urban growth, illegal building, disaster damage, and land-use change.

The present project, carried out as a second-year engineering end-of-year project at the National Engineering School of Tunis, sits at the intersection of these two worlds. The idea is to combine satellite imagery with structured geographic data in order to detect urban changes in a way that is both more accurate and easier to interpret. This chapter introduces the broader context of the project before presenting the specific objectives, the problem being addressed, and the methodology that was followed.

1.2 Project objectives

The primary goal of the current project is the development of a multi-modal method for urban change detection which would use both satellite images and structured semantic information about a particular territory. The basic approach is to adapt an existing classifier to work with two images of the same location taken in different time periods, as well as associated textural information, and generate the output in the form of a precise map of changes.

To achieve the above-stated purpose, it will be necessary to create an adaptive system capable of working with bi-temporal imagery in one unified manner, instead of using two independent pipelines for the processing of each time period. The proposed algorithm should be able to detect important changes based on the comparison of two images while relying on the external semantic information about their contents. This semantic guidance is expected to help the model focus on meaningful changes and reduce false detections caused by noise or irrelevant visual variations.

Another key aim is to generate maps with density changes on a per-pixel basis rather than a binary classification at either an image or a sub-image level. Change detection in urban environments requires a higher level of accuracy with regard to the location of changes that have been identified, ideally with precision up to building or structure level in order to provide a reliable output useful in real-world applications.

A further goal of the research conducted within this project is the application of the proposed algorithm to benchmark data sets often used in change detection studies. This is crucial in ensuring the validity of the outcomes as well as comparison with other techniques described in the relevant literature.

Lastly, it should be pointed out that the scale of the project was somewhat modi-

fied as the research progressed. This is due to the fact that some goals were changed according to the practical difficulties arising during implementation, especially concerning the selection of sources for information and degree of integration of some elements. At times, initial ideas had to be revised to fit the resources at hand.

1.3 Problem statement

Determination of alterations in cities through the use of remote sensing imagery relies on analysis of two images depicting the same area obtained at two distinct moments of time. However, despite its seeming simplicity, this process poses several challenges to the analyst.

First, there is the problem of radiometric differences. Depending on the exact moment of acquisition, the two images depicting the same territory may differ drastically as the result of different time of day, season, weather conditions, and angle of incidence of sunlight. In other words, the images will show the presence of clouds, haze, and shadows that do not reflect any changes but rather cause apparent differences. The use of simplistic comparison algorithms in this case results in numerous false alarms, which makes it difficult to pinpoint true alterations.

There is also the issue of registration errors. Although geometric correction has been performed, the two images are not necessarily completely registered at the pixel level due to such factors as complex relief and buildings that create parallax errors. As a consequence, corresponding elements on different dates shift spatially relative to each other, giving rise to artificial detections on boundaries.

But there is another problem, which is more of semantic nature. While an image can demonstrate how the set of pixels has changed its characteristics, it cannot provide any semantic meaning to the detected change. The appearance of a new building, demolishing of a building, season changes in the vegetation of an area, temporary construction in the territory will result in visually similar changes on the image. It becomes really hard to distinguish actual urban changes from unimportant ones without having additional information about the type of observed area. Such a problem gets even more pronounced in densely populated territories where a lot of different structures exist in a relatively compact area. There are shadows falling from buildings on other buildings, vegetation grows up or is being cut down, and there is a high probability of appearing/disappearing some temporary object.

These considerations give rise to the main concept of this project – the use of additional geographic data alongside with satellite images in order to enhance urban change detection process. Adding semantics to image recognition helps to detect only meaningful changes.

1.4 Study and limitations of existing approaches

A number of approaches have been proposed in the literature to address the problem of change detection from satellite imagery. These methods span several generations of techniques, each bringing improvements but also carrying its own limitations.

The earliest techniques utilized fairly straightforward pixel-based processing, for instance, difference of images, ratio of images, or principal component analysis of images from two different time periods. While these techniques are computationally inexpensive and easy to perform, they are very susceptible to noise and other forms of discrepancies present in the two sets of images. Consequently, these techniques usually generate rather noisy results that necessitate extensive post-processing.

With the rise of deep learning, convolutional neural networks brought a major shift in how change detection is approached. Architectures based on fully convolutional networks, and particularly encoder-decoder designs inspired by the U-Net architecture [1], originally proposed for biomedical image segmentation and later widely adopted in remote sensing tasks, showed that it is possible to learn spatial features directly from the data rather than relying on hand-crafted operators. Figure 1.1 illustrates the original U-Net architecture that served as a foundational reference for these developments. It is important to note that this architecture is used here purely as a scientific reference for non-commercial research purposes.

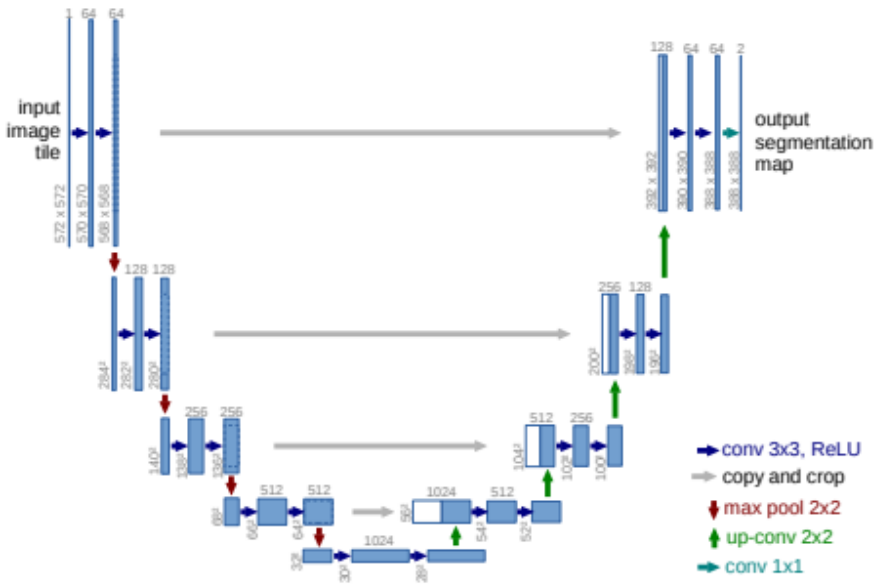


Figure 1.1: Original U-Net encoder-decoder architecture with skip connections [1].

Siamese architectures, which process the two temporal images through shared-weight encoders, further improved temporal consistency by ensuring that the features extracted from both dates live in the same representation space [2]. These

methods significantly improved detection accuracy compared to classical approaches, especially for building-level change detection in urban areas.

More recently, transformer networks were proposed in the domain of change detection. The potential of modeling long-range spatial dependencies by means of self-attention in such networks is superior compared to convolution networks due to their intrinsic limitations related to locality [3]. Recent publications have shown that state-of-the-art results can be obtained using either transformers alone or by combining them with Convolutional Neural Networks (CNNs).

Despite these advances, the vast majority of existing methods operate exclusively on visual information. They take image pairs as input and produce change maps based solely on what can be observed in the pixels. No external knowledge about the content of the scene is used in the process, leaving a significant semantic gap [4]. This means that the model has no way of knowing, for instance, that a particular area corresponds to a residential zone or a park, unless it can somehow infer this from the image content alone.

Some recent works have started exploring the idea of combining images with other modalities. Vision-language models, for example, have shown impressive results in general computer vision tasks by jointly processing images and text. In the remote sensing domain, a few studies have attempted to integrate textual metadata, geographic information system data, or structured annotations into the analysis pipeline. However, these efforts remain sparse and are mostly limited to classification tasks rather than dense spatial prediction problems like change detection.

The integration of structured semantic data, such as the kind of information available in collaborative geographic databases, into a dense bi-temporal change detection framework is still largely unexplored. Most existing multimodal approaches in remote sensing either use text as a simple auxiliary signal or rely on contrastive alignment without deep feature-level fusion. To the best of our knowledge, no prior work fully integrates structured semantic guidance with bi-temporal image processing within a unified architecture specifically tailored for dense urban change mapping.

The above-described gap in the literature is the exact problem that this project attempts to address. In contrast to other methods, which only focus on the visual differences between images to detect any changes, this method tries to improve upon the pipeline of detecting change by adding a layer of semantics to help understand the content of the area observed.

1.5 Proposed solution

To address the limitations identified in the previous section, this project proposes a multimodal approach that combines bi-temporal satellite images with structured semantic data extracted from a collaborative geographic database. The central idea is to extend a state-of-the-art multimodal classification model so that it can process two temporal images together with textual descriptions of the observed area, and produce a dense change map as output.

The architecture leverages a pre-existing image-text fusion structure, which was initially developed to perform multimodal classification on a single image. This baseline structure ensures that the architecture is well-grounded for combining image and textual features at the feature level using deep learning techniques, instead of concatenating outputs from different encoders. Nonetheless, the initial architecture was never meant for multi-temporal or dense predictions; hence, modifications had to be made.

Figure 1.2 gives a schematic representation of the proposed pipeline. The two images are encoded using a common visual encoder, then fused together with the semantic text data, before decoding into a dense change map.

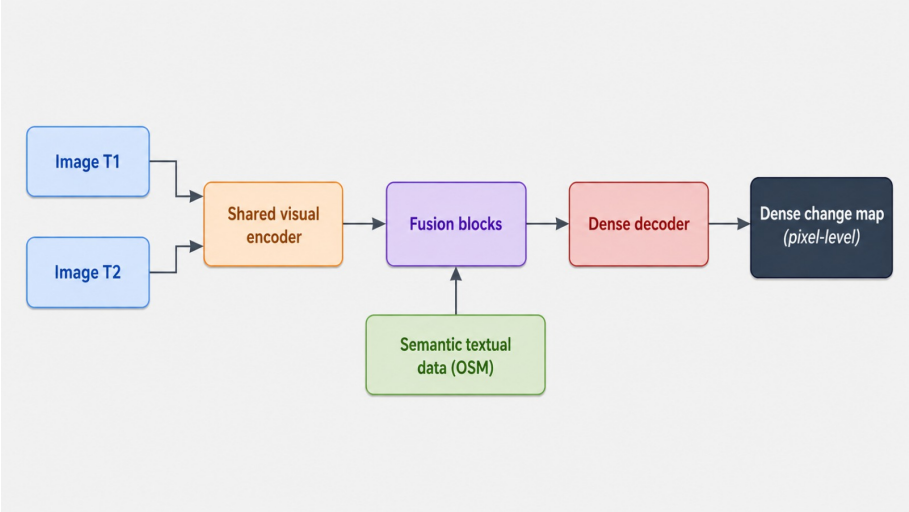


Figure 1.2: Simplified overview of the proposed multimodal change detection pipeline.

1.6 Methodology and project management

The project was carried out according to a set methodology based on the Cross-Industry Standard Process for Data Mining approach [5] that is commonly employed in various data science and machine learning projects. This methodology involves an iterative process including phases such as business understanding, data understanding, data preparation, modeling, and evaluation leading to a final solution.

Figure 1.3 shows the nature of the iterative CRISP-DM process.



Figure 1.3: The CRISP-DM process model [5].

In the first stage, the problem analysis was carried out and the scope of the work was established depending on the results of project statement analysis, literature research, and analysis of problems in multimodal urban change detection. The second stage includes data comprehension and acquisition. Satellite image sources were examined, and also ways of collecting semantic geographic information were investigated. Data processing and preparation were the main tasks of the third stage. The fourth stage included creating and implementing a model architecture, selecting a training strategy, and iterative modeling.

For monitoring the progress of this work, a Gantt chart has been created that divides all stages of the work into key milestones. The Gantt chart is shown in Figure 1.4.

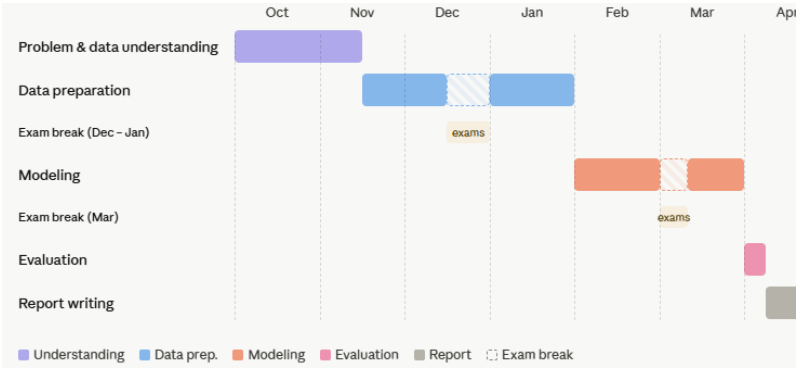


Figure 1.4: Gantt chart of the project timeline.

The entire project took place between October 2025 and April 2026, and there were breaks during mid-December and March due to exams. The entire process was

organized in a way that allowed the project to be conducted systematically, and the CRISP-DM model fit this work perfectly since it was iterative.

1.7 Conclusion

This chapter offered a broad view of the general environment in which the project is situated, ranging from the higher-level motivations related to urban observation and satellite data processing to the lower-level goals, problem statement, and approach used during the research process. The weaknesses found in the current techniques revealed the importance of leveraging both visual and semantic cues in change detection, and a potential approach was described.

In the following chapter, the current state of the art will be explored more deeply, with particular emphasis on the methods and models currently available for change detection and multimodal learning in geospatial domains.

CHAPTER 2

Multimodal Change Detection: State of the Art

This chapter reviews the main approaches and concepts that are relevant to the present work. It covers change detection methods in remote sensing, from classical techniques to recent deep learning architectures, and examines how multimodal learning has been applied in geospatial contexts. The chapter concludes by identifying the limitations that motivate the approach proposed in this project.

2.1 Change detection in remote sensing

Change detection in remote sensing involves finding the differences that exist in two or more images of a given geographical region that have been captured at different periods in time. This is one of the most widely researched topics within the remote sensing community, being applicable in various areas such as urban development mapping, agricultural mapping, disaster effects analysis, and forestation changes.

The basic premise here is to compare two images that belong to the same region, but which were acquired before and after a given period in order to generate a map indicating regions that underwent significant changes. Unfortunately, it is quite difficult to capture images under exactly the same conditions because there may be differences in sun angles, atmosphere, sensors, and seasons, etc. These pseudo-changes constitute one of the main sources of error in change detection systems [6].

Registration accuracy, spatial resolution, and the lack of semantic understanding also play important roles. Most change detection methods produce binary outputs indicating whether a pixel has changed or not, without providing information about the nature or significance of the change [4]. A newly constructed building and a temporary shadow shift may both be flagged as changes, even though they carry very different implications. This semantic gap motivates the multimodal direction explored in this project.

2.2 Deep learning approaches for change detection

Over the past decade, there have been dramatic changes in how change detection is done using deep learning techniques. Previous approaches tended to rely on manually engineered features, image difference methods, or simple pixel-based approaches. These strategies were effective, but not efficient for handling complicated scenes, noise, shadow effects, seasonality, or minor urban changes.

Deep learning models allow the model to discover relevant visual patterns in the data

2.2.1 From convolutional neural networks to Siamese architectures

Convolutional neural networks (CNNs) have been amongst the earliest Deep learning (DL) models used for change detection. CNNs allow automatic feature extraction of spatial information hierarchically, where features learned in early layers include low-level information such as edge and texture information, and high-level features are extracted in later layers. The ability to learn features specific to the task directly from training data instead of using hand-engineered features was a significant leap over classical pixel-wise methods.

The earliest DL models adopted the aforementioned architecture and passed the concatenated pair of images of the two time instances as inputs into a fully convolutional encoder-decoder model inspired by the U-Net model [1]. This FC-EF approach is straightforward and works well; however, it does not consider the temporal relation between the two date images since it has to learn to distinguish between the contribution of each date to the fused data.

The Siamese architecture solves this problem. It includes two encoder branches that are shared across the two branches and perform operations on the two temporal images separately. Because both branches use the same learned parameters, the features extracted from T1 and T2 are guaranteed to live in the same representation space, making them directly comparable. This weight-sharing mechanism provides a strong temporal consistency prior that is particularly well suited for change detection.

Building on this Siamese principle, Daudt et al. [7] proposed two influential variants. FC-Siam-diff processes the two images through shared-weight encoder branches and computes the element-wise difference between the corresponding feature maps at each encoder level. These difference features are then fed into a decoder that reconstructs the spatial resolution and produces the final change map. By operating on feature differences rather than raw pixel differences, FC-Siam-diff captures more abstract and semantically meaningful changes. FC-Siam-conc follows a simi-

lar dual-encoder design but concatenates the feature maps from the two branches instead of computing their difference. This concatenation preserves the full information from both dates and lets the decoder learn to combine them in the most useful way. Figure 2.1 illustrates the three architectures side by side.

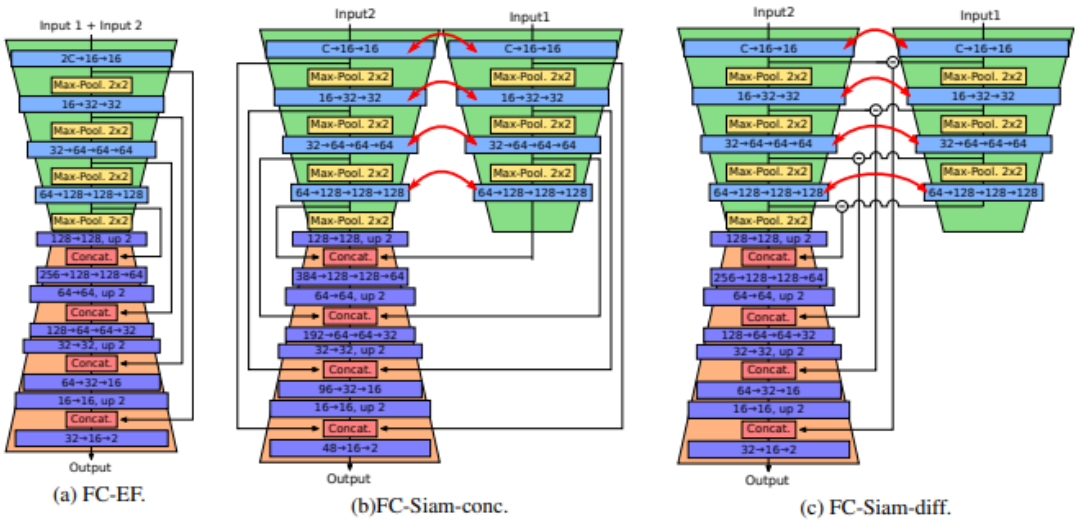


Figure 2.1: Comparison of FC-EF, FC-Siam-diff, and FC-Siam-conc architectures for change detection [7].

2.2.2 Transformer-based methods

More recently, transformer architectures have started to attract attention in change detection. The key advantage of this approach lies in the ability of the model to consider correlations between areas that are far from each other within an image due to the use of the self-attention technique. While convolutional layers tend to capture mostly local dependencies, transformers allow modeling connections between distant objects within the image.

An example of this research trend would be Bitemporal Image Transformer (BIT)[8]. The BIT model has been devised to merge some of the benefits of CNNs and transformers. First, the two images captured on two different dates are used to extract their features using a CNN-based backbone network. Next, relations between the pixels within each image and relations between the two views of the images taken at different times are learned using the transformer-based encoder network. However, instead of using all the pixel values, BIT makes use of semantic tokens.

ChangeFormer [9] is another representative transformer-based approach for change detection. Unlike BIT, it relies on a fully transformer-based architecture. It uses a hierarchical vision transformer encoder to extract multi-scale features from the input images. This makes it suitable for dense change detection, where both fine details and broader spatial relationships are important. Results on several benchmark

datasets have shown that transformer-based models can be a strong alternative to traditional CNN-based approaches for detecting changes in remote sensing images.

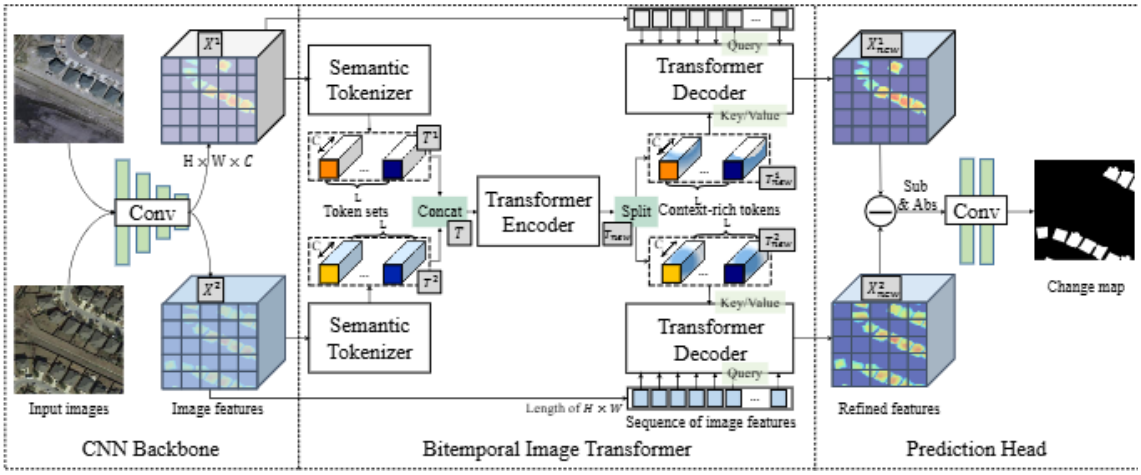


Figure 2.2: Architecture of the Bitemporal Image Transformer (BIT) [8].

Other hybrid approaches have combined convolutional and transformer components, using CNNs for efficient local feature extraction and transformers for global context aggregation, often achieving a good balance between computational efficiency and detection accuracy [10].

2.3 Multimodal learning in geospatial applications

While the methods discussed in the previous section focus exclusively on visual information, a growing body of work has explored the idea of combining images with other types of data to improve understanding of geospatial scenes [11, 12]. This section examines how multimodal learning has been applied in remote sensing contexts, focusing on two models directly relevant to the present project.

2.3.1 Contrastive Language-Image Pre-training (CLIP) and vision-language models

Vision-language models have attracted considerable attention in the general computer vision community in recent years [13]. These models are trained to jointly understand images and natural language, enabling tasks such as zero-shot image classification, image retrieval from text queries, and visual question answering. Their success is largely based on the idea of learning a shared embedding space where semantically related images and texts are mapped close to each other.

Contrastive Language-Image Pre-training (CLIP), introduced by Radford et al. [13], consists of two separate encoders, an image encoder (Vision Transformer or ResNet) and a Transformer-based text encoder, trained with a contrastive objective

on large numbers of image-text pairs. This produces embeddings capturing rich semantic relationships between visual and textual content.

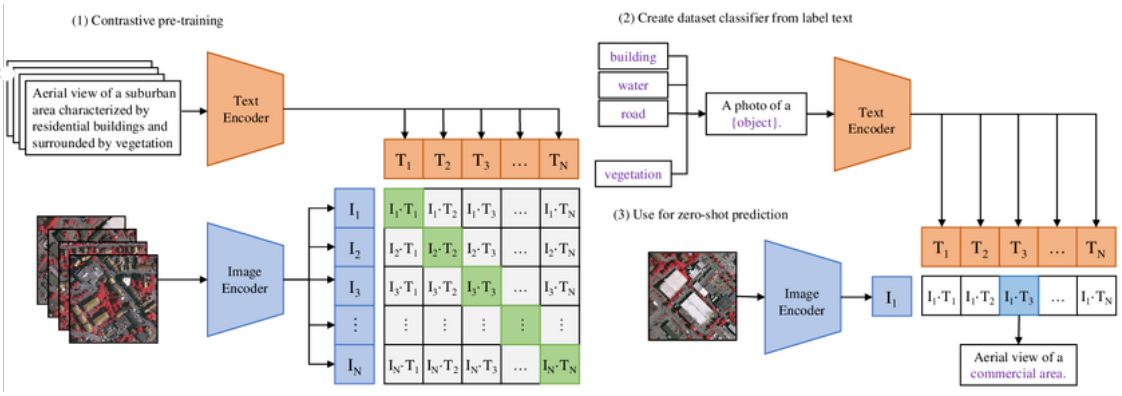


Figure 2.3: Overview of the CLIP architecture [13].

As far as scene classification, image retrieval, and land-use detection are concerned, some previous works on remote sensing have focused on leveraging CLIP to perform those operations [11]. Nevertheless, CLIP generates one global embedding vector for each image, and this means that no spatial information is included in that embedding regarding the positions of features, which makes the method unsuitable for being utilized as a backbone for any kind of pixel-based task. From the perspective of the present project, CLIP can be utilized as a semantic ranking/filtering mechanism.

2.3.2 Image-text Encoder Fusion Transformer (IEFT)

While CLIP aligns images and text through contrastive learning at the output level, the Image-text Encoder Fusion Transformer (IEFT) addresses this limitation by introducing deep feature-level fusion at intermediate representation stages [14]. The IEFT pipeline consists of a visual encoder that processes the input image into patch-level feature tokens, a text encoder that produces text tokens, and a Fusion Intersection Encoder (FIE) that applies cross-attention and self-attention mechanisms to enable deep interaction between the visual and textual features. The output of the FIE is passed to classification heads that produce both local patch-level predictions and a global image-level classification.

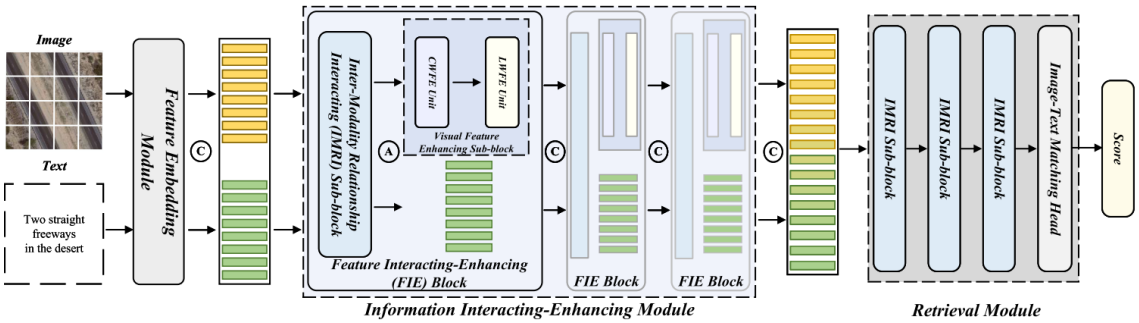


Figure 2.4: Architecture of the IEFT model

The key advantage of IEFT is the depth of the multimodal interaction: the FIE blocks allow the model to learn complex correspondences between specific visual regions and specific textual concepts, enabling spatially grounded multimodal reasoning. However, the standard IEFT architecture was designed for single-image multimodal classification, does not natively support bi-temporal inputs, and lacks mechanisms for dense pixel-level prediction.

2.4 Limitations and open challenges

The methods reviewed in this chapter show the important progress achieved in both change detection and multimodal learning for geospatial applications. Deep learning models such as FC-EF, FC-Siam-conc, and FC-Siam-diff introduced fully convolutional and Siamese architectures to compare bi-temporal images in an end-to-end way [7]. Later methods, such as SNUNet-CD, improved spatial detail extraction using dense skip connections and nested U-Net structures [15]. More recently, transformer-based models such as BIT and ChangeFormer used self-attention to better capture long-range spatial and temporal relationships between the two dates [8, 9].

Nevertheless, all these techniques continue to depend primarily on the visual data. They are capable of identifying the changes in the pixels; however, they lack the incorporation of the semantic knowledge that is external to the scene being observed. As a result, the actual urban change may be misinterpreted with the change in the visuals such as shadows or lighting conditions.

At the same time, multimodal models such as CLIP have shown the value of aligning images and text in a shared representation space [13]. In remote sensing, models such as RemoteCLIP and IEFT extend this idea to satellite imagery and image-text retrieval tasks [16, 14]. Nevertheless, these models are not directly designed for dense bi-temporal change detection. CLIP and RemoteCLIP mainly produce global image-text representations, while IEFT focuses on image-text retrieval rather than comparing two temporal images or generating pixel-level change

maps.

Therefore, an important gap remains between visual change detection and multimodal semantic understanding. Existing change detection models are strong in bi-temporal comparison and dense prediction, but they usually lack semantic guidance. In contrast, multimodal models provide semantic alignment, but they do not naturally produce dense change masks. This motivates the direction explored in this project, which combines bi-temporal image processing, semantic geographic information, multimodal fusion, and pixel-level change map generation within a single framework.

2.5 Conclusion

This chapter reviewed the main approaches relevant to this project, covering classical and deep learning methods for change detection, as well as multimodal learning strategies based on CLIP and IEFT. The analysis highlighted that while each of these directions has produced valuable contributions, the combined challenge of bi-temporal change detection with dense prediction and structured semantic guidance represents an active open research problem. The next chapter presents the proposed approach that aims to contribute to addressing this challenge.

CHAPTER 3

Proposed Approach: Siamese IEFT with Temporal Fusion and OpenStreetMap (OSM)-Guided Dense Decoding

This chapter presents the proposed approach for multimodal bi-temporal urban change detection. The data sources used in the project are first introduced, then the architecture is described starting from its overall pipeline and each component is detailed individually.

3.1 Data sources used in the project

Before presenting the internal architecture, it is important to introduce the data sources used in this work. The project relies on three complementary sources: Sentinel-2 imagery exported from Google Earth Engine, the LEVIR-CD benchmark dataset, and semantic information extracted from OpenStreetMap. Sentinel-2 was used during the first exploratory phase to validate the general idea of the pipeline, LEVIR-CD was used for the final experimental evaluation, and OpenStreetMap provided the external semantic context used to guide the multimodal component.

3.1.1 Sentinel-2 imagery from Google Earth Engine

The initial phase of the project used Sentinel-2 satellite imagery exported through Google Earth Engine. Sentinel-2 provides multispectral images that are useful for observing urban areas over time. For each selected region, two images were acquired at two different dates, corresponding to T1 and T2. Each image contained four spectral bands: B2 (Blue), B3 (Green), B4 (Red), and B8 (Near-Infrared). The two temporal images were then stacked to form an 8-channel input tensor, where the first four channels correspond to T1 and the last four channels correspond to T2.



Figure 3.1: Example of a Sentinel-2 image used during the initial data exploration phase.

This first dataset allowed the general bi-temporal processing pipeline to be tested in a realistic remote sensing context. However, Sentinel-2 imagery has a spatial resolution of around 10m for the visible and near-infrared bands used in this work, which is not sufficient for precise building-level change detection. For this reason, the project later moved toward a benchmark dataset with higher spatial resolution and pixel-level annotations.

3.1.2 LEVIR-CD benchmark dataset

For the final evaluation, the project used the LEVIR-CD benchmark dataset [10]. LEVIR-CD is a widely used dataset for building change detection. It contains pairs of high-resolution RGB remote sensing images acquired at two different dates. Each image pair is associated with a binary ground truth mask that identifies building changes at the pixel level.



Figure 3.2: Example of a high-resolution LEVIR-CD image used for building change detection.

LEVIR-CD provides a more rigorous experimental setting than the initial Sentinel-2 data because it offers standardized train, validation, and test splits, as well as manually annotated pixel-level change masks. In this project, the dataset was organized according to its standard structure, with folders `A/` for T1 images, `B/` for T2 images, and `label/` for the binary ground truth masks. The images were processed as RGB pairs, which corresponds to six visual channels when the two dates are considered together.

3.1.3 OpenStreetMap semantic data

OpenStreetMap (OSM) was used as an external source of semantic geographic information. For each image patch, OSM elements such as buildings, roads, vegetation, water areas, residential zones, and industrial zones were extracted using geographic coordinates associated with the patch. These elements were then transformed into textual and structured representations.

The textual representation provides an overview of the patch contents by using linguistic descriptions, whereas the structured representation highlights the semantics of the patch by using numerical features. The numerical features may be those that refer to buildings, roads, trees, water, and other types of land use. This data serves as semantic guidance for discriminating between significant and insignificant urban modifications within the new architecture.

For the LEVIR-CD phase, the OSM data was organized in JSON format. Each

patch may include fields such as `main_text`, `osm_texts`, `has_osm_text`, `osm_struct`, and `osm_reliability`. The field `main_text` contains a compact textual summary of the area. The field `osm_texts` contains candidate semantic descriptions. The field `osm_struct` provides a structured numerical vector, and `osm_reliability` indicates the estimated reliability of the available OSM information.

3.2 Proposed architecture overview

The proposed architecture is built upon the IEFT model, which was originally designed for single-image multimodal classification by fusing visual and textual features through its Fusion Intersection Encoder (FIE) blocks. In this project, the standard IEFT pipeline is adapted to handle bi-temporal satellite image pairs and to produce dense pixel-level change maps instead of classification labels. CLIP is integrated into the framework as a semantic ranking tool for filtering OSM textual descriptors, but it does not serve as the main processing backbone.

How does the entire process work? Two satellite images for one and the same location that were captured on two distinct occasions (T1 and T2) are provided as input to the algorithm together with the semantic information available in OSM, resulting in an output of a dense binary change map and a global change score.

Firstly, two separate images are encoded individually by the Siamese Visual Encoder using Vision Transformers, generating two sequences of patch-wise feature tokens referred to as F_{t1} and F_{t2}. These features are arranged as a unified bi-temporal sequence in [CLS | F_{t1} | TEMP | F_{t2}] order. This sequence is further processed using fusion blocks, followed by calculation of local change features used as input in three parallel prediction branches: the Global Change Prediction branch, the Dense Multi-scale Decoder, and the Boundary Head. Structured semantic information from OpenStreetMap is introduced at several layers.

The overall architecture is illustrated in Figure 3.3. A complete and detailed version of the architecture diagram, showing all internal modules, data flows, tensor dimensions, and the four processing blocks, is provided in Appendix A.

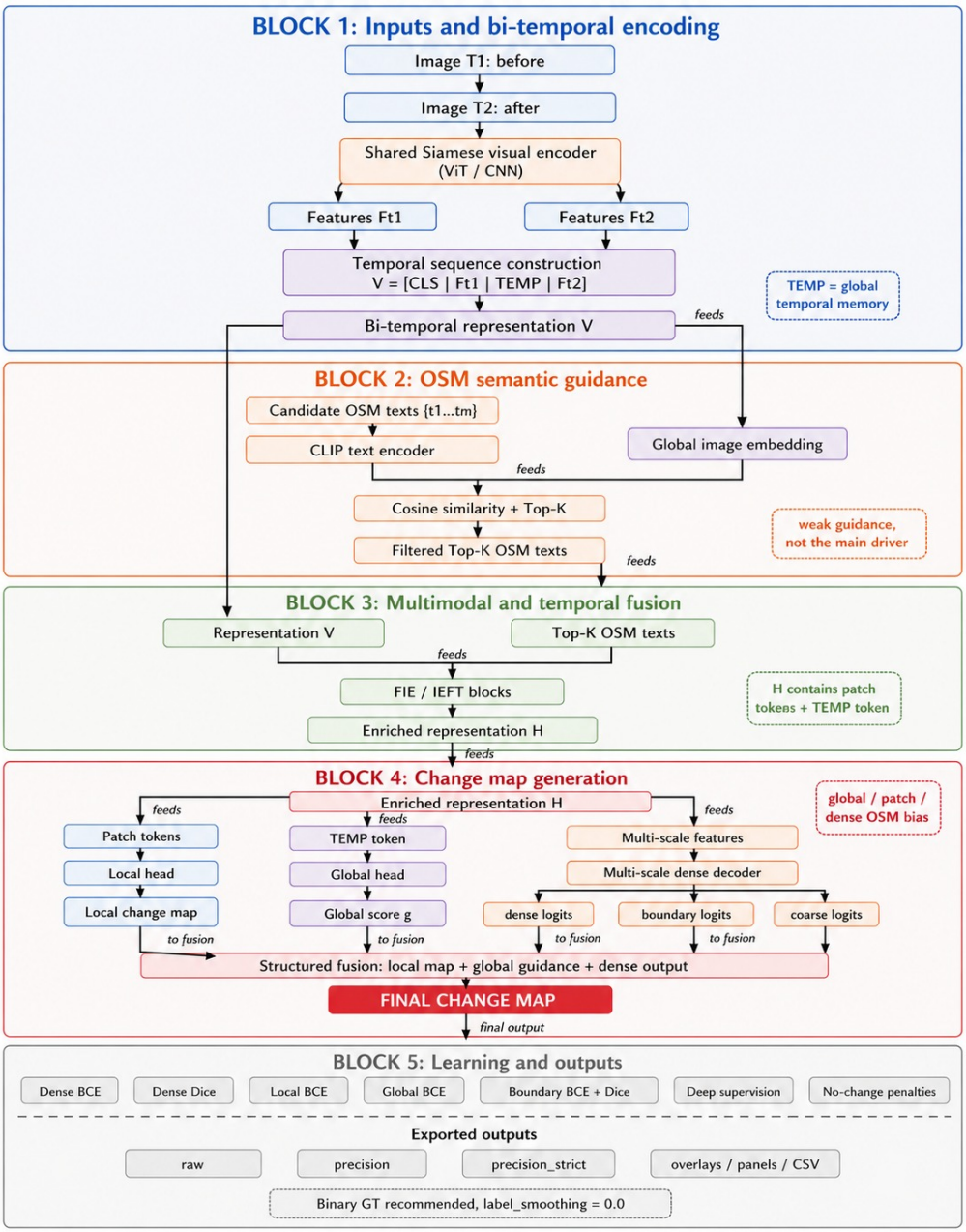


Figure 3.3: Overview of the proposed architecture for multimodal bi-temporal urban change detection.

3.3 Shared Siamese visual encoder

The visual encoder is based on the Vision Transformer (ViT) architecture [17]. ViT operates by first dividing each input image of size $H \times W \times C$ into a regular grid of non-overlapping patches of size $p \times p$, producing $N = (H/p) \times (W/p)$ patches in total. Each patch is then flattened into a one-dimensional vector and linearly projected into a d-dimensional embedding through a learnable projection matrix $\mathbf{E} \in \mathbb{R}^{(p^2 \cdot C) \times d}$. To preserve information about the spatial arrangement of patches, a set of learnable positional embeddings $\mathbf{E}_{pos} \in \mathbb{R}^{N \times d}$ is added element-wise to the

patch embeddings before they enter the transformer layers.

The resulting sequence of N patch tokens is then processed through a stack of L transformer encoder layers. Each layer consists of a Multi-Head Self-Attention (MHSA) sublayer followed by a feed-forward network (FFN) sublayer, with layer normalization applied before each sublayer and residual connections applied after. In the self-attention mechanism, each token computes attention weights over all other tokens in the sequence, allowing the model to capture both local and global spatial relationships within the image.

The suggested model structure involves the same ViT encoder with identical parameters being used for processing images of both T1 and T2. In this regard, the so-called Siamese architecture will allow us to use comparable sets of features since the same parameters transform the images into the same feature representation. Specifically, given a $256 \times 256 \times 3$ image with patches of size $p = 16$, the resulting number of patch tokens would be $N = 16 \times 16 = 256$, each of which will have d dimensions.

The ViT encoder is initialized from pretrained weights (ViLT `vilt_200k_mlm_itm.ckpt`) to leverage visual representations learned from large-scale image-text data. During training, the encoder parameters are updated with a reduced learning rate (1×10^{-5}) compared to the decoder (1×10^{-4}), in order to fine-tune the pretrained representations without destroying them.

3.4 Bi-temporal token sequence and TEMP token

Once both images have been encoded, the patch tokens from T1 and T2 are combined into a unified bi-temporal sequence, as shown in Equation 3.1:

$$V = [\text{CLS} \mid F_{t1} \mid \text{TEMP} \mid F_{t2}] \tag{3.1}$$

where $F_{t1} = [f_{t1}^1, f_{t1}^2, \ldots, f_{t1}^N]$ and $F_{t2} = [f_{t2}^1, f_{t2}^2, \ldots, f_{t2}^N]$ are the patch token sequences from the two dates, each containing N tokens of dimension d.

The CLS token is a learned vector of size d, which is added at the beginning of the sequence. The purpose of the CLS token is to summarize the representation of all tokens on both temporal dates using the self-attention function within the fusion blocks. The output of the CLS token acts as a summary of the bitemporal inputs for the global change prediction task.

The TEMP token is another learnable vector of dimension d, inserted between F_{t1} and F_{t2}. It is used to mark an explicit temporal boundary that enables the self-attention layers to tell apart tokens of the first and the second dates from each other. If there were no such a separator, the attention process would not have had any positional clues to separate the temporal context of the two categories of tokens,

given that they use the same positional embeddings.

The total length of the sequence V is $1 + N + 1 + N = 2N + 2$ tokens, which for $N = 256$ gives a sequence of 514 tokens.

3.5 Adapted fusion blocks

The fusion blocks are adapted from the FIE in the original IEFT architecture [14]. Each fusion block implements the standard transformer encoder layer structure, consisting of two sublayers applied sequentially.

The first sublayer is MHSA. Given the input sequence V, the attention mechanism computes queries, keys, and values through learned linear projections, as defined in Equation 3.2:

$$Q = VW_Q, \quad K = VW_K, \quad V_{\text{val}} = VW_V \tag{3.2}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_h}$ and $d_h = d/h$ for h attention heads. The attention output for each head is computed according to Equation 3.3:

$$\text{Attention}(Q, K, V_{\text{val}}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_h}}\right) V_{\text{val}} \tag{3.3}$$

The outputs of all heads are concatenated and linearly projected back to dimension d.

The second sublayer is a FFN, consisting of two linear transformations with a GELU activation in between, as expressed in Equation 3.4:

$$\text{FFN}(x) = W_2 \,\text{GELU}(W_1 x + b_1) + b_2 \tag{3.4}$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff} \times d}$, with d_{ff} typically set to $4d$.

Both sublayers are preceded by layer normalization and followed by residual connections, as shown in Equations 3.5 and 3.6:

$$x' = x + \text{MHSA}(\text{LN}(x)) \tag{3.5}$$

$$x'' = x' + \text{FFN}(\text{LN}(x')) \tag{3.6}$$

The self-attention in these fusion blocks operates over the entire bi-temporal sequence V, meaning that each token can attend to all other tokens regardless of their temporal origin. This enables the model to learn cross-temporal correspondences: a token from T1 can attend to the corresponding spatial position in T2 to detect changes, while also attending to neighboring positions to capture contextual information. Multiple fusion blocks are stacked for progressive refinement of the

temporal and spatial representations.

3.6 Local change representation

After the fusion blocks have processed the bi-temporal sequence V, the enriched token representations are split back into their temporal components $\hat{F}_{t1}$ and $\hat{F}_{t2}$, each containing N tokens. At each spatial position i, three complementary change features are computed to capture different aspects of the temporal difference.

The absolute difference $|\hat{f}^i_{t2} - \hat{f}^i_{t1}|$ captures the magnitude of change at each feature dimension, highlighting positions where the representation has shifted significantly between the two dates, regardless of the direction of the change.

The element-wise product $\hat{f}^i_{t1} \odot \hat{f}^i_{t2}$ captures feature co-activation patterns. When both temporal representations are active in the same dimensions, the product is large; when one is active and the other is not (indicating a change), the product tends to be smaller. This provides a complementary signal to the absolute difference.

The cosine similarity $\cos(\hat{f}^i_{t1}, \hat{f}^i_{t2}) = \frac{\hat{f}^i_{t1} \cdot \hat{f}^i_{t2}}{\|\hat{f}^i_{t1}\| \, \|\hat{f}^i_{t2}\|}$ measures the angular similarity between the two temporal representations, providing a normalized measure of change that is invariant to the magnitude of the feature vectors.

These three features are concatenated with the original tokens from both dates to form a rich local change descriptor at each spatial position, as defined in Equation 3.7:

$$c^i = [\hat{f}^i_{t1} \mid \hat{f}^i_{t2} \mid |\hat{f}^i_{t2} - \hat{f}^i_{t1}| \mid \hat{f}^i_{t1} \odot \hat{f}^i_{t2} \mid \cos(\hat{f}^i_{t1}, \hat{f}^i_{t2})] \tag{3.7}$$

This multi-faceted representation allows the subsequent prediction heads to leverage different types of temporal signals depending on the nature of the change being detected.

3.7 Prediction heads

3.7.1 Global change prediction branch

The CLS token, after passing through all fusion blocks, contains a global summary of the bi-temporal content. It is fed into a Multi-Layer Perceptron (MLP) consisting of two linear layers with a GELU activation and dropout in between, producing a single scalar logit. A sigmoid activation converts this logit into a global change probability $p_{global} \in [0, 1]$.

In the context of the inference process, the role of p_{global} can be thought of as a no-change veto procedure: whenever the global change score becomes lower than a preset value, the entire prediction map becomes nullified (becomes zero), thereby preventing any false positives due to the prediction of change where there has not

been one. This is especially crucial when working with rural areas or patches that have had no changes at all.

3.7.2 Dense multi-scale decoder

The decoder follows a progressive upsampling strategy inspired by UNet-style architectures. It receives feature maps extracted at multiple depths from the ViT encoder through skip connections, specifically from transformer blocks 4, 8, and 12 (the final block). Each skip connection feature map is first linearly projected to the decoder hidden dimension ($d_{dec} = 128$), then progressively upsampled through a series of convolutional and bilinear interpolation stages.

At each upscaling step, the decoder integrates the upsampled features with the skip-connection features from the corresponding layer of the encoder to allow the reconstruction of small spatial details that would have been neglected while going through the encoder layers. In each integration step, first the concatenated features undergo convolutions with batch normalization and ReLU activation functions. The decoder's final output will be an $H \times W$ sized probability map for change at each pixel, created using 1×1 convolutions along with a sigmoid activation function.

3.7.3 Boundary head and deep supervision

The Boundary Head consists of a lightweight convolution layer which receives as input the output of the decoder and generates the boundary map, identifying the edges of areas that have experienced changes. This component is trained using the ground truth boundary masks obtained through morphological gradient processing (dilation minus erosion) applied to the ground truth change masks. The Boundary Loss helps create sharper and more accurate change boundaries, preventing the blurring effect generated by dense decoders.

The use of deep supervision involves introducing auxiliary losses at different levels of decoder architecture. Unlike the supervision of the model's output only, the ground truth images are resized according to the resolutions of different levels of the decoder and losses are computed at these levels. This helps the earlier layers of the decoder also to receive some gradient information from the losses.

3.8 OSM semantic guidance integration

The use of OSM semantic data, which were presented as another source of data to be used in the project earlier, is one of the unique elements of the approach suggested. Even though the basis of the architecture is based on being image first, which means

that the process of analysis performed by the image pipeline is independent of any text at all, OSM data can provide the model with additional context.

For each image patch or tile in the dataset, several OSM-derived fields are extracted and organized into a structured representation. The field `main_text` contains a primary textual description of the area covered by the patch, summarizing the dominant land use or the most prominent features. The field `osm_texts` contains a list of individual textual descriptors corresponding to specific OSM elements within the patch, such as building types, road categories, or vegetation areas. The binary flag `has_osm_text` indicates whether any OSM data is available for the given patch, allowing the model to handle areas with sparse or missing coverage. The field `osm_struct` provides a structured feature vector encoding quantitative properties extracted from OSM, such as the number of buildings, total road length, or proportion of green area. Finally, `osm_reliability` provides a confidence score reflecting the estimated quality and completeness of the available OSM data.

Such semantic features get integrated within the architecture on various levels. The global level involves the integration of information provided by OSM, which helps the overall change score get influenced by the contextual prior of what type of area is being looked at. On the patch level, the effect of the semantic features is to influence the representations of local change in a way that can help the network discriminate actual structural changes from visual artifacts. At the dense prediction level, OSM guidance affects the output of the decoder.

A particular part played by CLIP in this process of integration is outlined below. Prior to adding the candidate OSM texts to the neural network, CLIP performs filtering of those texts based on their semantic relation to the visual aspect of the patch. An image encoder provided by CLIP generates embeddings for the patch and for all candidate OSM texts by using the CLIP text encoder. Then, cosine similarity is calculated between embeddings of the image and of each text. Finally, only Top-K texts are left based on their cosine similarity scores with the image. This way, the input to the model gets filtered for only semantically relevant information. The CLIP-based ranking module is partially implemented at the moment, and improvements are currently in progress.

It should be highlighted that at the moment, the model is functioning predominantly in the image-first regime. Although the OSM guidance serves as supplementary background information, the prediction of dense changes is mostly facilitated by the image processing stream. The incorporation of the semantic data in its fullest extent, such that the elements of OSM would take part in the attention mechanism in the fusion modules, still awaits future research.

3.9 Conclusion

This chapter presented the data sources and the proposed architecture for multimodal bi-temporal urban change detection. The chapter introduced Sentinel-2 imagery, the LEVIR-CD benchmark, and OpenStreetMap semantic data, then detailed each component of the architecture: the shared Siamese encoder, the bi-temporal token sequence with TEMP token, the adapted fusion blocks, the local change representation, the prediction heads, the dense decoder with boundary supervision, and the OSM semantic guidance. The next chapter presents the implementation details, the evolution of the experimental results, and a critical analysis of the outcomes.

CHAPTER 4

Implementation, Experimental Results, and Analysis

This chapter describes the practical implementation of the proposed system, the experimental protocol, and the results obtained on both the Sentinel-2 and LEVIR-CD datasets. It also includes a discussion of the main observations, an analysis of the errors and limitations encountered, and directions for future work.

4.1 Software environment and project structure

The whole project was created and executed within a Conda environment running on a Windows desktop PC. Deep Learning is performed with the help of PyTorch library with PyTorch Lightning for managing the training process. Vision Transformer encoder is provided by the `timm` library while the part responsible for text encoding uses the `transformers` library developed by Hugging Face. Mixed-precision training is accomplished with PyTorch Automatic Mixed Precision feature. Acceleration is provided via GPU utilization with the use of CUDA. Training visualization is done via TensorBoard. Sacred is used to configure the experiments, which can be fully reproduced due to sacred-serialization library for hyperparameters versioning. All training operations and exports are invoked via `run.py` script.

The codebase is organized around several key modules. The main model logic resides in `vilt_module.py`, which implements the adapted IEFT architecture including the Siamese encoder, the bi-temporal token sequence, the fusion blocks, and the prediction heads. The multi-scale dense decoder and boundary head are implemented in `multiscale_change_head.py`. The loss functions, including the composite change detection losses and regularization terms, are defined in `objectives.py`. Dataset handling for Sentinel-2 and LEVIR-CD is implemented in separate dataset and datamodule files. The export and evaluation pipeline is handled by `export_change_maps.py` and `compare_overlay_quality.py`.

Figure 4.1 summarizes the main software bricks of the project and their relationships.

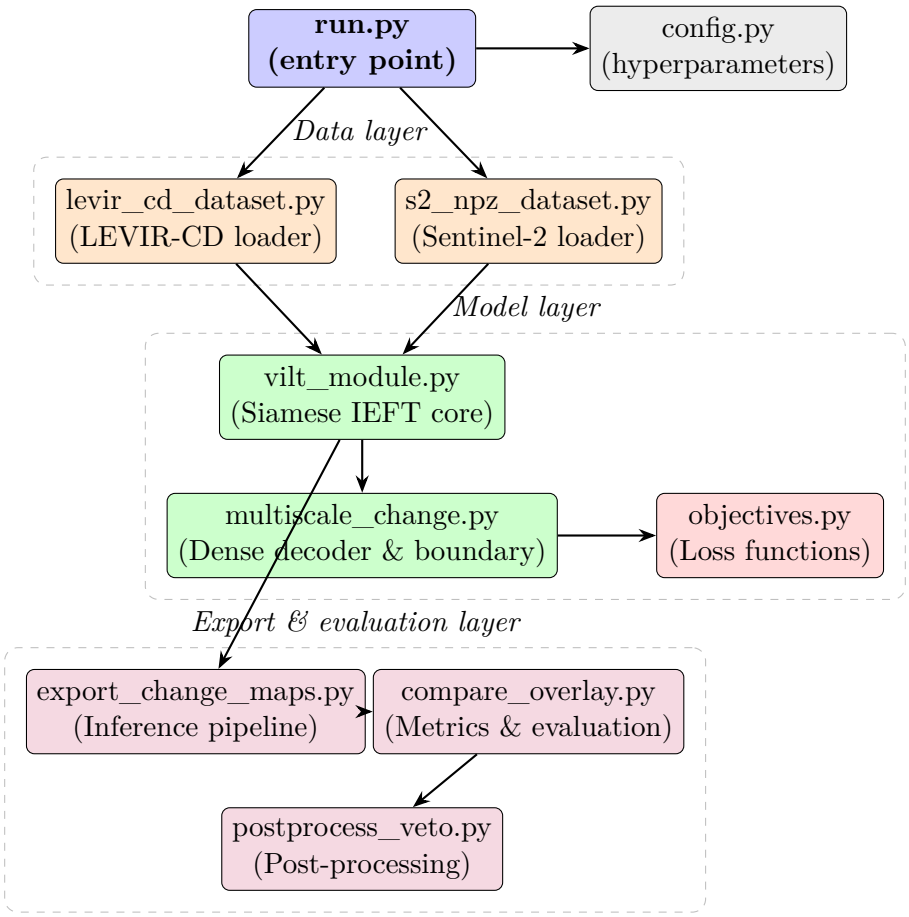


Figure 4.1: Overview of the project software structure and interactions between main modules.

4.2 Training configuration and loss design

Training is done using the AdamW optimization technique with an initial learning rate of 1×10^{-4} for the decoder and heads and a much smaller learning rate of 1×10^{-5} for the ViT encoder that was pre-trained. This helps prevent catastrophic forgetting during fine-tuning. A batch size of 1 is chosen because of limited GPU memory, while mixed precision FP16 is used to save memory and improve speed. Training checkpoints are periodically saved, allowing training to resume from any saved checkpoint.

The loss function is a weighted combination of several complementary terms. The dense Binary Cross-Entropy (BCE) loss and the dense Dice loss are applied to the pixel-level change map output. The local BCE loss supervises the patch-level change predictions. The global BCE loss supervises the CLS-based global change score. The boundary BCE and boundary Dice losses are applied to the boundary head output. Deep supervision losses are added at intermediate decoder stages.

In addition to these standard losses, several regularization mechanisms are employed. A no-change penalty discourages the model from predicting changes in

patches that are labeled as entirely unchanged in the ground truth. A no-change max penalty and a no-change top-k penalty further suppress high-confidence false predictions in no-change regions. A color-only penalty discourages the model from relying solely on color differences between T1 and T2, which may reflect radiometric variations rather than actual structural changes.

The label smoothing value for the LEVIR-CD is set to 0.0 so that the ground truths are kept binary. The binary GT constraint is enabled using a threshold value of 0.5.

4.3 Export and inference pipeline

The export pipeline converts the model predictions into usable change maps. For LEVIR-CD, images are processed using a sliding window with tile size 256×256 and stride 128, and the overlapping predictions are averaged. Three output modes are produced for each test image.

The **raw** mode is a binary classification using a static threshold of 0.50 for the predicted probabilities while maintaining all connected regions above the minimum area value. The raw mode gives the best recall value but results in higher false-positive values.

In the **precision** mode, other rules apply, namely precision gate, which is a measure of mean predicted probability and building support score, discarding any component which might be a road or noise, and no-change veto, which excludes all predictions if the overall change score is below a certain threshold.

The **precision_strict** method has the most stringent filtering criteria, where the threshold for components scores is tighter, the criterion for not changing is stricter, and there are more spurious objects removed. This method is aimed at optimizing the precision-recall trade-off and serves as the main output reference.

The export pipeline also generates overlay visualizations, side-by-side panel comparisons with ground truth, NPZ files containing the raw probability maps, and CSV files with per-patch metrics.

4.4 Experimental protocol and tested configurations

The experimental evaluation follows a standard protocol. Models are trained on the training split, hyperparameters are selected based on validation performance, and final metrics are reported on the held-out test split. For LEVIR-CD, the test set contains 2048 tiles of 256×256 pixels.

The main evaluation metrics are: Intersection over Union (IoU), Dice coefficient (equivalent to F1-score), Precision, Recall, and overall Accuracy. These metrics are

computed per patch and then averaged. In addition to global averages, a changed-only analysis is performed by computing metrics only on patches where the ground truth contains at least some changed pixels, which provides a fairer assessment of detection quality by excluding trivially correct no-change predictions.

4.5 Experimental results

4.5.1 Results on the initial Sentinel-2 phase

The Sentinel-2 phase served as the development and validation platform where the conceptual pipeline was built and iteratively refined. Working with multi-spectral imagery (4 bands: B2, B3, B4, B8) at 128×128 resolution, the project explored numerous model versions that progressively improved the quality of change detection outputs.

The earliest versions (v10 and below) produced very coarse change maps at the resolution of the ViT token grid (12×12). The breakthrough came from transitioning to a coarse-to-fine prediction strategy, where the initial coarse map served as a spatial prior refined using dense spectral difference maps (diff_rgb, diff_ndvi, diff_nir, diff_edge). This refinement process, combined with a boundary head, fundamentally changed the nature of the output from smooth blobs to spatially precise masks.

Table 4.1 summarizes the progression of key versions during the Sentinel-2 phase, reporting both the main contribution of each version and the corresponding quantitative results on the 225-patch test set. Note that the evaluation during this phase relied on a pseudo-reference constructed automatically from spectral difference maps rather than human-annotated ground truth; the absolute metric values should therefore be interpreted with appropriate caution.

Table 4.1: Summary of key model versions during the Sentinel-2 phase.

Version	**IoU**	**Dice**	**Precision**	**Recall**	**Key contribution**
v10 (baseline)	< 0.10	< 0.15	—	—	Coarse token-grid map
v14 (stable)	0.201	0.314	0.350	0.397	Stable training reference
v16	—	0.528*	—	—	Dense decoder improvement
v17 (best S2)	0.488	0.646	0.684	0.615	Refined map + adaptive export

*changed-only Dice

The evaluation during this phase relied on a pseudo-reference constructed automatically from spectral difference maps rather than human-annotated ground truth. While this allowed consistent comparison between versions, it also meant that the absolute metric values should be interpreted with appropriate caution.

Figure 4.2 shows an example of the Sentinel-2 prediction pipeline outputs, illustrating the various intermediate representations (raw prediction probability, normalized probability, fixed and adaptive masks, pseudo-reference, and final overlay) that were used to analyze and refine the change detection results.

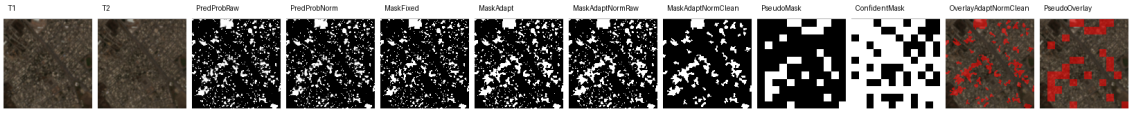


Figure 4.2: Example of Sentinel-2 change detection pipeline outputs

Despite the encouraging progression during this phase, several fundamental limitations motivated the transition to LEVIR-CD: the 10m/pixel spatial resolution of Sentinel-2 was insufficient for individual building detection, the 12×12 token grid produced outputs too coarse for precise footprints, and the absence of standardized pixel-level annotations prevented meaningful comparison with methods from the literature.

4.5.2 Results on LEVIR-CD

The transition to LEVIR-CD allowed the system to be evaluated against standardized pixel-level ground truth at high spatial resolution (0.5 m/pixel). The model underwent a substantial evolution on this benchmark, progressing from fragile early adaptations to a robust dense prediction system. Table 4.2 presents the main milestones across the most significant versions.

Table 4.2: Evolution of results on the LEVIR-CD test set across key versions.

Version	**IoU**	**Dice/F1**	**Precision**	**Recall**	**Key insight**
v12 dense256	0.1209	0.1991	0.2004	0.2997	First dense branch
v14 stable	0.2010	0.3143	0.3500	0.3971	First stable base
v14 postfix	0.2326	0.3465	0.2634	0.6562	More recall, wider masks
v15 strict (ch-only)	0.3934	0.5290	–	–	Pixel/object logic
Best strict	0.4073	0.5217	0.7335	0.4951	Best early compromise
v20d raw	0.4888	0.5965	0.5501	0.7897	Strong recall
v20d precision	0.5109	0.6213	0.6143	0.7129	Balanced output
v20d prec. strict	**0.5487**	**0.6617**	**0.6677**	0.7140	Best model output

Figure 4.3 shows the evolution of the two main metrics (IoU and Dice/F1) across the successive versions, illustrating the progressive improvement from early dense attempts to the final precision_strict output.

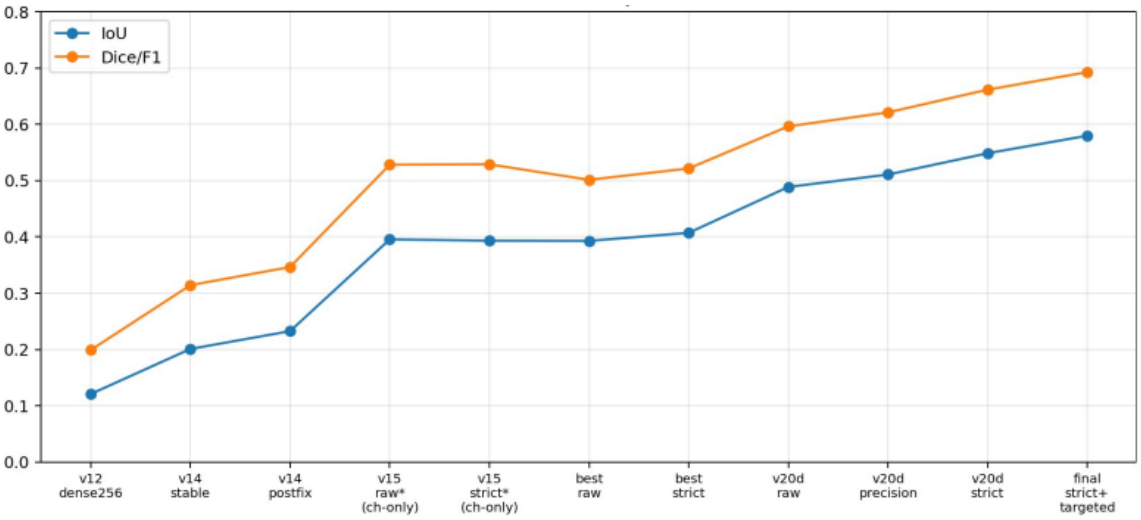


Figure 4.3: Evolution of IoU and Dice/F1 across successive LEVIR-CD versions

The best model output corresponds to the precision_strict export of checkpoint v_20d_full_118k_v2, achieving an IoU of 0.5487 and a Dice/F1 of 0.6617. This result represents the culmination of several architectural and training improvements: the dense multi-scale decoder with skip connections at ViT blocks 4, 8, and 12, the enriched local change representation combining absolute difference, element-wise product, and cosine similarity, the boundary head with deep supervision, and the structured export pipeline with no-change veto logic.

Figure 4.4 illustrates a representative example from the test set (test_120), displayed as a multi-panel strip. From left to right, the panels show: (1) the T1 image, depicting the area before change, predominantly open agricultural land with sparse development; (2) the T2 image, showing extensive new residential construction that has appeared between the two dates; (3) the model confidence map, a grayscale probability output where brighter pixels indicate higher predicted change likelihood; (4) the temporal attention map, visualizing which regions the model attends to most strongly when comparing T1 and T2; (5) the raw predicted mask after fixed thresholding; (6) the precision_strict predicted mask after filtering and veto post-processing; (7) the ground truth binary change mask; and (8–9) the overlay visualizations on T1 and T2, where predicted change regions are highlighted in red. This example demonstrates the ability of the system to detect large-scale building changes with reasonable spatial accuracy while filtering out noise through the precision_strict pipeline. The comparison between the raw and precision_strict masks shows how post-processing removes small spurious detections while preserving the main change regions.

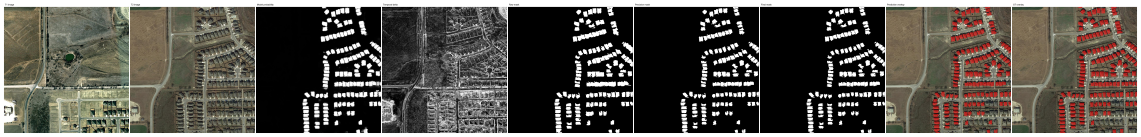


Figure 4.4: Example prediction on LEVIR-CD

Among the 2048 test patches, the best-performing patch was test_89, which achieved near-perfect detection metrics: Dice = 0.9543, IoU = 0.9126, Precision = 0.9601, and Recall = 0.9486. Figure 4.5 shows the prediction for this patch in the same multi-panel format. The T1 image shows an open field with surrounding vegetation and a road, while the T2 image reveals the construction of a large institutional building complex (with clearly visible structures including what appears to be a school or sports facility with an adjacent running track). The model confidence map shows strong, well-localized activation precisely over the new buildings, with minimal noise in the surrounding unchanged areas. The raw and precision_strict masks are nearly identical in this case, indicating that the change signal is strong and unambiguous. The ground truth mask confirms that the predicted change region closely matches the annotated building footprints. The overlay panels show the detected changes highlighted in red on both temporal images, illustrating that the model successfully captured both the main building mass and the smaller adjacent structures. This patch illustrates the upper bound of the system's capability when the change pattern is well-defined and the visual contrast between T1 and T2 is clear.

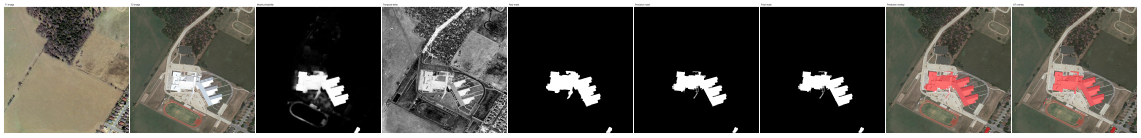


Figure 4.5: Best-performing patch

4.5.3 Post-processing with targeted veto

Although the precision_strict export option itself already guarantees an excellent result, there still exists an opportunity for enhancement, which involves a particular post-processing procedure that is performed only on the results of model prediction. It should be noted that such post-processing does not affect model weights at all and is purely based on the processed output data.

The targeted veto scheme helps to solve one type of residual errors, namely those consisting of weak false positives in regions with weak, fragmented, and spatially uncoherent predictions of the model. Such regions often represent rural territories, vegetation edges, or other patches where the difference in T1 and T2 is due to seasonal or lighting variations instead of real structure change.

The veto decision for each patch is based on several signals already available in the export pipeline: the predicted change ratio (proportion of pixels flagged as changed), the mean and peak prediction probability, the global change score from the CLS token, the building support mean (a measure of how consistent the prediction is with building-like structures in T2), the number of connected components, the maximum component area, and the dominant component ratio. Based on these signals, three veto rules are applied: a weak small false positive rule that suppresses patches with very low prediction ratio and low mean probability, a fragmented false positive rule that targets patches with many small disconnected components and low overall confidence, and a weak global false positive rule that catches patches with low global change scores. To prevent the veto from suppressing legitimate detections, three protection conditions are evaluated: patches with a strong probability peak, patches with at least one large connected component, and patches with strong semantic support from the OSM structural features.

Table 4.3 presents the results before and after the targeted veto post-processing.

Table 4.3: Effect of targeted veto post-processing on the precision_strict output.

Configuration	**IoU**	**Dice/F1**	**Precision**	**Recall**	**Accuracy**
prec. strict (model)	0.5487	0.6617	0.6677	0.7140	0.9779
prec. strict + targeted veto	**0.5799**	**0.6929**	**0.6989**	**0.7452**	**0.9780**

The targeted veto improved IoU from 54.87% to 57.99% and Dice/F1 from 66.17% to 69.29%. This corresponds to an absolute gain of +3.12% for both metrics. It also improved the balance between precision and recall compared with the previous best export configuration. The simultaneous improvement in both precision and recall is explained by the nature of the correction: the veto suppresses entire patches that were previously counted as false positives, which removes those patches from the false positive count (increasing precision) while the true positive patches remain unchanged. The overall accuracy remained essentially stable at 0.9780, confirming that the veto targeted only problematic patches without degrading the rest of the predictions.

This constitutes the highest score that has been achieved by the model. The targeted veto procedure is designed as a standalone post-processing script (`postprocess_targeted_veto.py`) that parses the NPZ files generated by the pipeline, enforces the veto rules on a per-patch basis, and writes out the final masks and overlays.

As a means of providing more information, Table 4.4 compares the above results to other state-of-the-art models available from other sources in literature on LEVIR-CD data. Note that the comparison is indicative in nature, since some variations in

terms of training methods are expected.

Table 4.4: Comparison with representative models reported on LEVIR-CD.

Model	Type	Precision	Recall	Dice/F1	IoU
FC-Siam-Diff [7]	Siamese CNN	0.8953	0.8331	0.8631	0.7592
SNUNet-CD [15]	Siamese Nested U-Net	0.8918	0.8717	0.8816	0.7883
BIT [8]	CNN + Transformer	0.8924	0.8937	0.8931	0.8068
ChangeFormer [9]	Siamese Transformer	0.9205	0.8880	0.9040	0.8248
Proposed model (precision_strict)	Siamese IEFT dense	0.6677	0.7140	0.6617	0.5487
Proposed model + targeted veto	Siamese IEFT dense + post-processing	0.6989	0.7452	0.6929	0.5799

4.6 Ablation insights

Ablation studies were performed by disabling parts of the architecture and measuring the effects on the performance of the system. Here are some of the key observations.

The absence of the boundary head and deep supervision resulted in poor sharpness of the boundaries and reduction of Dice by 2–3%. The no-change veto approach was instrumental in suppressing the false positives in unchanged regions since its absence caused a considerable drop in precision without impacting the recall value. The OSM-based structural information (`osm_struct`) yielded an insignificant but consistent gain in performance mainly by suppressing the false positives in rural and uncertain areas. Finally, the TEMP token was essential for understanding global changes, and its absence impacted the global change score accuracy negatively.

4.7 Discussion and error analysis

The results obtained on LEVIR-CD are encouraging but also reveal several patterns of errors that deserve discussion.

The type of error that occurs frequently in the rural area is the false positive case. There are cases where roads, vegetation borders, agriculture land, and textures are classified as change because of the significant difference observed from T1 and T2 data because of the seasonal effect and change in illumination conditions. In such a case, the use of no-change veto and precision_strict filters comes in handy but does not completely solve the problem.

In some rare cases, there are patches with building changes which may not be easy to deal with. If small buildings are observed side by side in one location, there are chances that the model will either lump them together as a single entity or miss some of the buildings.

The precision-recall trade-off is an inherent challenge. The raw output captures most true changes but includes substantial noise. The precision_strict mode produces cleaner maps but may suppress legitimate detections, particularly small or

low-confidence ones. Finding the optimal operating point depends on the specific application requirements.

4.8 Current limitations and future work

Several limitations should be acknowledged. The OSM semantic guidance currently operates as a structured side signal rather than as a deeply integrated modality. A tighter integration, where OSM features participate directly in the attention computations within the fusion blocks, could potentially improve the semantic grounding of the predictions. The CLIP-based text ranking component has been partially integrated and is being progressively refined; completing this integration with full Top-K semantic filtering of OSM texts is a priority for future development.

The sign version of the temporal difference $(f_{t2} - f_{t1})$ was also investigated as a replacement for the absolute one but did not show superior performance in the current setup. This research path can be explored further in some other way of normalization or integration.

A number of directions require future research. The most important among them is completing the progressive CLIP integration for the Top-K OSM text filtering. Another one is to improve the modeling of dense building clusters by using instance-aware prediction heads. Also, it would be beneficial to find a more universal way to learn no-changes than penalizing. Finally, using pre-trained ViTs can bring some improvement.

4.9 Conclusion

This chapter presented the implementation, training, export pipeline, and experimental evaluation of the proposed system. The results on LEVIR-CD demonstrate a substantial improvement over the baseline, with the best model output (precision_strict) achieving a Dice/F1 of 0.6617 and an IoU of 0.5487, further improved to a Dice/F1 of 0.6929 and an IoU of 0.5799 through targeted post-processing. The analysis of errors and limitations highlighted the main challenges that remain, particularly regarding false positives in non-urban areas and the ongoing refinement of the CLIP-based OSM filtering. These observations define clear directions for future work aimed at further improving the precision, robustness, and semantic grounding of the system.

General Conclusion and Future Perspectives

This report presented the work carried out as part of a second-year engineering end-of-year project at the National Engineering School of Tunis, addressing the problem of multimodal urban change detection from satellite imagery.

The project proposed an approach that extends the Image-text Encoder Fusion Transformer (IEFT) architecture for bi-temporal change detection. The key architectural contributions include a shared Siamese visual encoder that processes both temporal images in the same representation space, a bi-temporal token sequence with a learnable TEMP token for temporal separation, adapted fusion blocks for cross-temporal feature integration, a local change representation based on absolute difference, element-wise product, and cosine similarity, a dense multi-scale decoder with boundary head and deep supervision, and structured OpenStreetMap semantic guidance injected at multiple levels of the pipeline.

From being able to work with the system using the first phase which consisted of imagery sourced from Sentinel-2 imagery via Google Earth Engine, the assessment process proceeded to being done with a better understanding, since now we were able to use ground truths from the LEVIR-CD dataset at a higher spatial resolution while maintaining the multi-modal nature of the system.

From the experiments performed for LEVIR-CD, there was a considerable increase in performance from the baseline to the best performance achieved from the model in precision_strict mode, increasing from a Dice score of 0.31 to a Dice score of 0.66, and further increased to a Dice score of 0.69 after applying post-processing. This is indicative of the capability of the dense decoder, the use of composite loss, no-change penalties, the pipeline for exportation, and the veto process to create useful and spatially precise change detection maps.

Some limitations do exist. False detections in rural regions, inaccurate localization of tightly clustered buildings, and continued progressive incorporation of CLIP-based OSM text ranking are some of the major limitations. These observations point out the clear path ahead.

The main perspectives for continuation include: completing the progressive integration of CLIP for full Top-K semantic filtering of OSM texts, developing deeper OSM integration where semantic features participate directly in the attention mech-

anism, improving the detection of dense or small building changes through instance-aware prediction, exploring remote sensing-specific ViT backbones, and evaluating the approach on additional datasets to assess cross-domain generalization. Despite these remaining challenges, the project demonstrated that combining bi-temporal visual processing with structured semantic guidance is a viable and promising direction for urban change detection.

APPENDIX A

Detailed Proposed Architecture

This appendix presents the complete proposed architecture in detail, showing all internal modules, data flows, tensor dimensions, and processing stages.

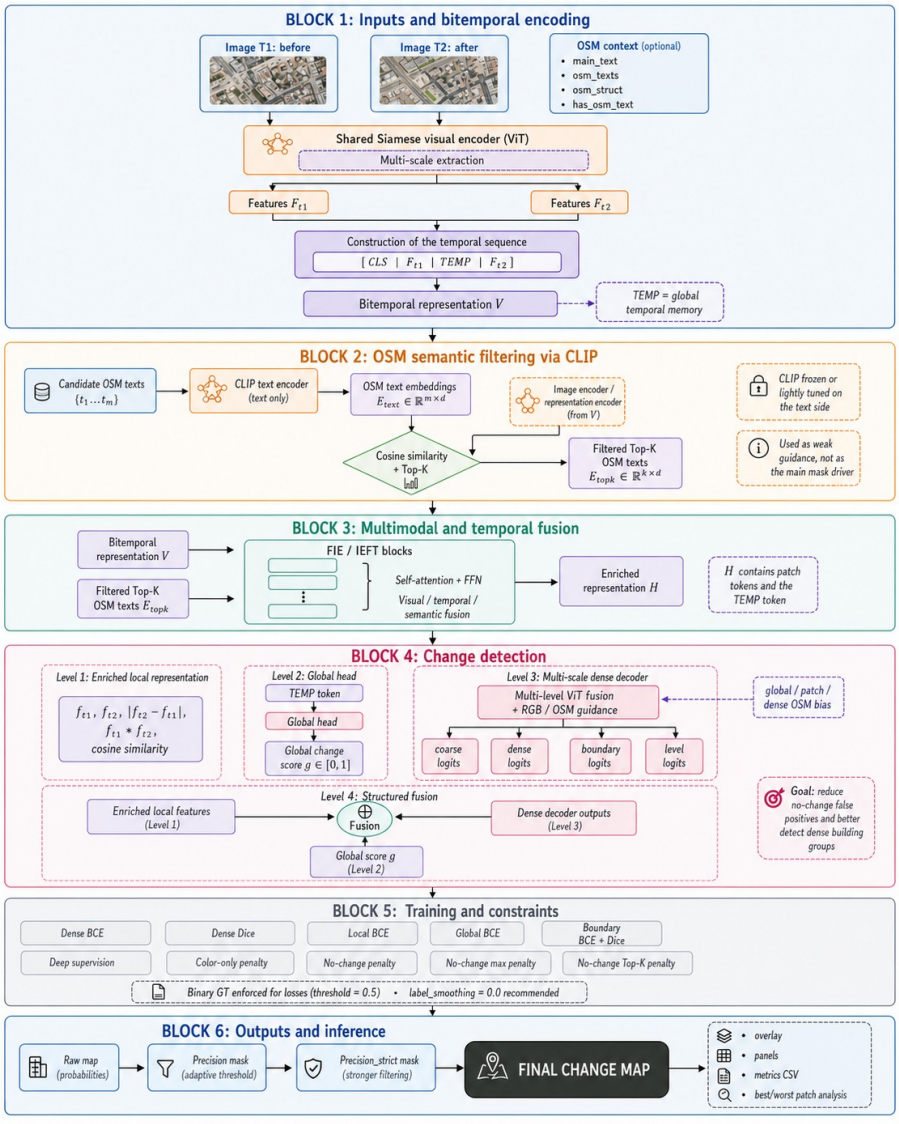


Figure A.1: Detailed Proposed Architecture

APPENDIX B

Implementation Details

The project codebase is organized as follows:

Table B.1: Main project files and their roles.

File	Role
`run.py`	Main entry point for training
`IEFT/config.py`	Hyperparameter configuration
`IEFT/modules/vilt_module.py`	Core model (Siamese encoder, fusion, heads)
`IEFT/modules/objectives.py`	Loss functions
`IEFT/modules/multiscale_change.py`	Dense decoder and boundary head
`IEFT/datasets/levir_cd_dataset.py`	LEVIR-CD data loading
`IEFT/datasets/s2_npz_dataset.py`	Sentinel-2 NPZ data loading
`export_change_maps.py`	Inference and export pipeline
`compare_overlay_quality.py`	Per-patch metric computation
`postprocess_targeted_veto.py`	Targeted veto post-processing

The software environment consists of Python 3.8 (Conda), PyTorch 1.13+ with CUDA, PyTorch Lightning, `timm` for Vision Transformer models, `transformers` from Hugging Face, and TensorBoard for monitoring. All experiments were run on a single GPU under Windows.

APPENDIX C

Dataset Examples and Useful Commands

LEVIR-CD dataset structure:

```
data_levir_cd/raw/LEVIR CD/
    train/
        A/      (T1 images)
        B/      (T2 images)
        label/  (binary change masks)
    val/
        A/ B/ label/
    test/
        A/ B/ label/
```

Sentinel-2 NPZ dataset structure:

```
data_npz/
    train/  (4 NPZ files, 1673 patches)
    val/    (1 NPZ file, ~60 patches)
    test/   (225 patches)
    meta/dataset.json
```

Each Sentinel-2 patch contains 8 channels: T1_B2, T1_B3, T1_B4, T1_B8, T2_B2, T2_B3, T2_B4, T2_B8, at 128×128 resolution.

Example training command (LEVIR-CD):

```
python run.py with \
  task_levir_cd_v20_cliprank_vitb16_levir_dense_completion \
  exp_name="v_20d_full_118k_v2" \
  data_root="data_levir_cd/raw/LEVIR CD" \
  load_path="weights/vilt_200k_mlm_itm.ckpt" \
  levir_use_osm=True \
  levir_osm_texts_json="data_osm/osm_texts_v21.json" \
```

```
  change_use_osm_struct=True \
  levir_label_smoothing=0.0 \
  change_force_binary_gt_for_losses=True \
  batch_size=1 per_gpu_batchsize=1 \
  max_steps=118000 max_epoch=80 \
  precision=16 num_workers=0 \
  ms_change_decoder_dim=128 \
  learning_rate=1e-4 encoder_learning_rate=1e-5
```

Example export command (LEVIR-CD):

```
python export_change_maps.py \
  --ckpt_path "result/.../last.ckpt" \
  --data_root "data_levir_cd/raw/LEVIR CD" \
  --split test \
  --output_dir "outputs/change_maps_final" \
  --model_input_size 256 --tile_size 256 \
  --tile_stride 128 --fixed_threshold 0.50 \
  --nochange_global_thr 0.21 \
  --nochange_prob_mean_thr 0.028 \
  --nochange_peak_thr 0.24 \
  --min_region_size 28 \
  --component_score_thr 0.31 \
  --precision_gate_prob 0.20 \
  --precision_gate_build 0.38
```

Example evaluation command:

```
python compare_overlay_quality.py \
  --input_dir "outputs/change_maps_final" \
  --output_csv "outputs/compare_versions/results.csv" \
  --border_band 8 --top_k 10 --rank_metric dice
```

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[2] Yuan Liu et al. "Building Change Detection for Remote Sensing Images Using a Dual-Task Constrained Deep Siamese Convolutional Network Model". In: *IEEE Geoscience and Remote Sensing Letters* 18.5 (2021), pp. 811–815.

[3] Hao Chen and Zhenwei Shi. "A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection". In: *Remote Sensing* 12.10 (2020), p. 1662.

[4] Lorenzo Bruzzone and Francesca Bovolo. "A Novel Framework for the Design of Change-Detection Systems for Very-High-Resolution Remote Sensing Images". In: *Proceedings of the IEEE* 101.3 (2013), pp. 609–630.

[5] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a Standard Process Model for Data Mining". In: *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*. 2000, pp. 29–39.

[6] Qiqi Zhu et al. "Recent Progress and Challenges in Foundation Models for Remote Sensing Change Detection". In: *IEEE Geoscience and Remote Sensing Magazine* 12.1 (2024), pp. 49–72.

[7] Rodrigo Caye Daudt, Bertrand Le Saux, and Alexandre Boulch. "Fully Convolutional Siamese Networks for Change Detection". In: *IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 4063–4067.

[8] Hao Chen, Zhenwei Qi, and Zhenwei Shi. "Remote Sensing Image Change Detection With Transformers". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–14.

[9] Wele Gedara Chaminda Bandara and Vishal M. Patel. "A Transformer-Based Siamese Network for Change Detection". In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2022, pp. 207–210.

[10] Hao Chen and Zhenwei Shi. “A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection”. In: *Remote Sensing* 12.10 (2020), p. 1662.

[11] Zilun Hu et al. “Exploring CLIP for Assessing the Look and Feel of Streetscapes and Remote Sensing Images”. In: *arXiv preprint arXiv:2207.02087* (2022).

[12] Xuran Li, He Yin, and Zhiyuan Li. “Multi-Source Remote Sensing Data Fusion for Urban Area Change Detection”. In: *International Journal of Remote Sensing* 41.22 (2020), pp. 8654–8679.

[13] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8748–8763.

[14] Xu Tang, Yifan Zhang, et al. “Interacting-Enhancing Feature Transformer for Cross-Modal Remote Sensing Image and Text Retrieval”. In: *Remote Sensing* (2023).

[15] Sheng Fang et al. “SNUNet-CD: A Densely Connected Siamese Network for Change Detection of VHR Images”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2021), pp. 1–5.

[16] Fan Liu et al. “RemoteCLIP: A Vision Language Foundation Model for Remote Sensing”. In: *arXiv preprint arXiv:2306.11029* (2023).

[17] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations (ICLR)*. 2021.