How do Transformer models perform in urban change detection with limited satellite datasets, and what strategies can enhance their accuracy for this task?

Jan Bryczkowski[1]

Supervisor(s): Jan van Gemert[1]**, Desislava Petrova-Antonova**[2]

[1]**EEMCS, Delft University of Technology, The Netherlands**
[2]**GATE Institute, Sofia University St. Kliment Ohridski, Bulgaria**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Jan Bryczkowski
Final project course: CSE3000 Research Project
Thesis committee: Jan van Gemert, Desislava Petrova-Antonova, Klaus Hildebrandt

Abstract

As global urbanization reaches an all-time high, effective urban management becomes a crucial factor for efficient development. Enhanced monitoring of these transformations leads to more informed decision-making by policymakers, emphasizing the importance of tracking these changes. One method for monitoring is Change Detection (CD), which involves comparing two satellite images captured at different times to detect changes over a period of time. CD involves numerous difficulties, such as data collection, varying weather conditions, limited availability of datasets, noise, illumination differences, and discrepancies in the equipment used for image capture. Convolutional Neural Networks (CNNs) can address these issues by delivering more effective models with better performance than non-deep learning models. However, the rise of Transformers has led researchers to develop networks based on Transformer architecture, yielding more promising results than CNNs when more data is available. This paper conducts an analysis of two existing Transformer-based models, emphasizing the challenges of handling CD with artificially small datasets. Using smaller datasets reduces the requirements for remote sensing capabilities of satellites, simulating the limitations encountered during data collection and processing. The models under examination are the Bitemporal Image Transformer (BIT) and the Visual change Transformer (VcT).

Index Terms - Change Detection (CD), Remote Sensing, Transformers, Attention Mechanism, Convolution Neural Networks (CNNs)

1 Introduction

In the rapidly evolving landscape of modern cities, the need for effective urban Change Detection (CD) has never been more critical. As urban areas expand and transform, it becomes essential to monitor these changes to ensure sustainable and efficient development. Urban CD plays a vital role in various aspects of city management, from infrastructure planning to environmental conservation. By keeping track of how urban spaces evolve, city planners and policymakers can make informed decisions that address the dynamic needs of a growing population.

From technical perspective, CD can be defined as a task of extracting natural or artificial changes from a specific land area using multiple satellite images from different timeframes [1]. The problem of detecting changes can result in several challenges, including different image resolutions [2], misjudgment caused by illumination variation [1], and image alignment errors [1]. One of the most accurate types of models in recent research efforts are based on Convolutional Neural Networks (CNNs) [1, 2, 3, 4]. Although CNNs are successful at extracting local information and semantics [4], they lack an understanding of the global changes in the images [1]. To tackle this issue, multiple approaches are investigated, where the self-attention mechanism is successfully used to correlate global features [2]. Self-attention allows models to weigh the importance of different parts of the input data, capturing long-range dependencies and global context effectively [5].

Transformer was first introduced in 2017, and its primary goal was to provide solutions for sequence-to-sequence text modeling used in the field of Natural Language Processing (NLP) [5]. Its architecture's effectiveness in handling long-range dependencies and capturing contextual information through self-attention mechanisms, has revolutionized various NLP tasks [5]. As frequently used for text modeling, it has been noticed that splitting text into tokens could be used similarly to split the image into tokens, leading to the creation of a Visual Image Transformer (ViT) [6]. The usage of ViT for the CD tasks increased the performance of recent models, creating a successful way of capturing global features while preserving already established local correlations.

Moreover, acquiring satellite datasets is significantly more challenging compared to obtaining other readily available datasets. Consequently, one of the primary difficulties in developing highly accurate CD models lies in acquiring a sufficient amount, resolution, and quality of data that can be successfully utilized to handle the changes in various urban areas. Given that deep learning models necessitate extensive datasets for training, the large size of these datasets can present a barrier for many researchers. This leads to the central research question: **How do Transformer models perform in urban change detection with limited satellite datasets, and what strategies can enhance their accuracy for this task?**

The primary objective of this paper is to examine the performance of two Transformer-based models, namely the Bitemporal Image Transformer (BIT) [3] and the Visual change Transformer (VcT) [2], in scenarios where data availability is limited. Additionally, it proposes strategies for achieving high quality results without extensive datasets by optimizing the dataset, environment, or runtime, particularly in situations involving limited resources, such as peripheral satellite devices that lack access to cloud computing and must solely rely on their own hardware. Artificially smaller datasets are created from the existing LEVIR-CD dataset [7], using two designed algorithms: Resolution Reduction algorithm, and Image Subset Sampling algorithm. Resolution Reduction algorithm focuses on decreasing images' resolution while preserving the label's condition without leaving any grey artifacts in the label image. Image Subset Sampling algorithm sorts the images based on the number of white pixels, which represent the changes. It then discards every n-th image to maintain the same proportion of changes per image in the smaller dataset as in the original dataset.

This paper is structured as follows: Section 1 introduces the topic and research question. Section 2 provides the related works background and rationale for the study. Section 3 defines the problem and motivation behind the research question. Section 4 details the research methodology. Section 5 presents the results. Section 6 summarizes the findings, implications, and future work. Section 7 outlines the study's limitations. Eventually, Section 8 addresses responsible research and ethical considerations.

2 Background and Rationale of the Study

This section outlines general trends with CD models and explains in more detail the architectural design of BIT and VcT.

2.1 Background

In the field CD, traditional methods such as algebraic algorithms, classification methods, and transformation methods initially achieved notable progress by focusing on detecting changed pixels and classifying them to generate change maps [2]. These methods, including support vector machines (SVM), random forests, decision trees, and Markov random fields (MRF), were once effective but have become insufficient due to their reliance on handcrafted features and threshold settings, leading to limitations in accuracy and generalization [4].

The rapid development of deep learning, particularly deep CNNs, marked a significant improvement in CD tasks. CNN-based models, which extract high-level semantic features from each temporal image, dominated the field and demonstrated superior performance over traditional methods [3, 4]. These models utilize fully convolutional architectures (FCNs) to address dense prediction problems in remote sensing (RS) tasks, including CD [1]. However, purely convolutional approaches face inherent limitations due to their restricted receptive fields (RF), struggling to effectively model long-range spatial and temporal relationships [3]. CD is not purely local because it requires analyzing contextual information and interconnected structures across an image to accurately identify changes. Long-range relationships are essential to capture global consistency and distinguish significant changes from irrelevant variations.

To address these limitations, attention mechanisms, including channel attention, spatial attention, and self-attention, have gained significance [3]. These mechanisms improve feature extraction by focusing on crucial areas and suppressing irrelevant background information, thereby enhancing the detection of subtle changes in scenes [4]. However, challenges remain in detecting small target changes or local area changes.

The most significant advancement in CD has been the rise of Transformers, which employ a non-local self-attention mechanism to model long-range relationships and global dependencies within image pixels [4]. Transformer models, initially proven successful in NLP [4], have been adapted to computer vision tasks, demonstrating outstanding performance. By projecting image blocks into independent sequences for feature extraction, Transformers like ViT have shown the potential to replace traditional convolution as the primary feature extractor [1]. In this paper, an analysis of these models has been conducted to understand their effectiveness and potential for processing smaller datasets, without compromising the eventual results. In the following subsections, two selected Transformer-based models are presented and analyzed.

2.2 Bitemporal Image Transformer (BIT)

The BIT is a widely recognized Transformer-based model employed for CD tasks, often cited and utilized as a baseline for comparison with more advanced models [1, 2, 4]. This model leverages high-level concepts known as visual words or semantic tokens, which are subsequently input into the Transformer encoder to model contexts within a token-based space. The extraction of high-level semantic features from the input image pair is facilitated by a CNN backbone (ResNet) utilizing spatial attention. Following this, the Transformer encoder models the context within the two sets of tokens, which are represented by two image sets taken at different times. The resulting tokens are then input into the Transformer to enhance the original pixel-level features. Ultimately, Feature Difference Images (FDI) are computed from the two redefined maps and fed into a shallow CNN, which is used at the final stage to produce pixel-level change predictions [3]. The CNN is not a main component of the model when learning changes but is employed to generate final stage predictions.

The rationale for selecting the BIT for analysis in the current study is due to its open-source code being readily accessible, as well as its extensive use as a benchmark method for comparison with other models, both Transformer-based and traditional.

2.3 Visual change Transformer (VcT)

The VcT is a more recent model compared to the BIT, incorporating novel approaches in its implementation. VcT extracts feature maps from the given image pairs by using a shared backbone network, a modified version of ResNet18 [8]. ResNet18 plays a role in feature extraction and is not the primary component of the model responsible for processing changes. Subsequently, each pixel of these maps is treated as a graph node within a Graph Neural Network (GNN). The top-k most reliable tokens are then extracted from the map and refined using the k-means clustering algorithm [9]. These reliable tokens are further enhanced through a self/cross-attention scheme and interaction with the original features via an anchor-primary attention learning module. Finally, a prediction head is employed to generate a more accurate change map [2].

The justification for selecting this model in the current study is based on its improved performance compared to the BIT baseline, as reported in the original publication [2]. Furthermore, the model employs a more novel and complex approach, utilizing k-means clustering within a Graph Neural Network (GNN) rather than a simple encoder-decoder structure. This introduces an interesting intersection of fields, making it a compelling choice for analysis.

3 Problem Formulation

In this section, general problems associated with the scarcity of satellite image data and the motivation behind the research question are discussed.

3.1 Satellite Data

Current machine learning models require substantial amounts of data for effective training. For instance, it is relatively straightforward to train a model to differentiate between everyday objects due to the easy availability of numerous photographs. In contrast, satellite imagery, which is essential for applications such as environmental monitoring and urban planning, is more challenging to acquire. The high cost, limited access restricted to certain countries and private companies with satellite capabilities, and general scarcity of such images contribute to this difficulty.

3.2 Inconsistent Data

The process of capturing satellite images is often subject to various environmental and technical factors that can affect the consistency and quality of the data. For example, atmospheric conditions, sensor calibration, and orbital parameters can introduce noise and other artifacts into the images. Consequently, even when satellite data is available, images of the same area taken at different times can vary in representation, adding another layer of complexity to the training process. These challenges underscore the need for specialized techniques and resources to effectively utilize satellite imagery in machine learning applications. Two images depicting the same area but exhibiting significantly different color schemes are presented in Figure 1. Upon investigating the contents within the red and blue rectangles, one can notice substantial color changes, resulting in two distinctly colored images representing the same area.

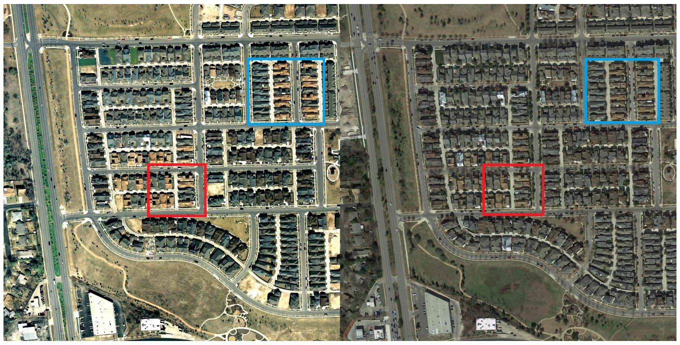


Figure 1: Comparison of the same urban environment at two different timeframes. The left image represents the first timeframe, while the right image represents the second timeframe. The differing colors, highlighted by the red and blue rectangles, illustrate the variations in representation.

3.3 Motivation Behind Research Question

With the increasing use of ViT for image processing tasks, this study aims to investigate the application of Transformers in remote CD. A comprehensive review of existing literature reveals a significant challenge: many models demand substantial data and computing power, resources that are often unavailable in peripheral devices like satellites. In specific cases, these devices need to pre-process images or results within their own infrastructure before transmitting the data back to Earth. This research addresses the gap by exploring how Transformer-based models behave with minimal data input and attempts to find the optimal range of parameters, including image resolution, runtime, and model accuracy. Developing such models is crucial for enhancing research efficiency and enabling rapid CD. This capability is particularly valuable in emergency situations, such as floods or earthquakes [10, 11], where timely and accurate information is essential for effective response.

4 Methodology

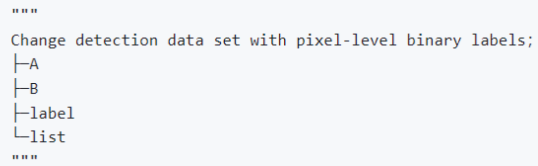
This section presents the data preparation, including two types of data preparation methods, namely Resolution Reduction algorithm and Image Subset Sampling algorithm. Additionally, the experimental setup and details of the runtime environment are also provided.

4.1 Data Pre-processing

Two of the analyzed models, namely BIT and VcT, are open-source models whose code is available on GitHub [12, 13].

A crucial decision for successful implementation of the current study involves selecting an appropriate dataset for the experiments. LEVIR-CD [14] is one of the most widely used datasets in this domain. It is chosen, since it was tested in both of the studies describing the usage of BIT [3] and VcT [2], making it ideal for comparison purposes.

The LEVIR-CD dataset is organized into three groups of images: A, B, and the label, as illustrated in Figure 2. Set A contains the first temporal image of each area, while set B includes an image of the same area taken at a later time. The label set contains the ground truth data indicating changes detected between images in sets A and B. These label images are binary, where white areas represent actual changes and black areas indicate no change. In each of the three directories, namely A, B, and label, there are 446 images for training, 64 images for validation, and 128 images for testing.



```
"""
Change detection data set with pixel-level binary labels;
├─A
├─B
├─label
└─list
"""
```

Figure 2: General structure of the LEVIR-CD dataset folder.

To accurately represent real-life scenarios, two methods have been identified for modifying the LEVIR-CD dataset to reflect the hardware limitations of peripheral satellite devices.

4.1.1 Resolution Reduction

The first method involves reducing the resolution of the images in A, B, and label folders. The algorithm for lowering the resolution is presented in Algorithm 1. When downscaling ground truth label images, the images should remain black and white without any gray artifacts, as white color represents change and black color represents no change. To achieve this, the PIL Python package [15] was used, employing the LANCZOS method [7] for downscaling the image resolution. The LANCZOS method minimizes distortion artifacts known as aliasing when representing a high-resolution image at a lower resolution point, making it suitable for this task.

Algorithm 1 Resolution Reduction Algorithm

```
Define $image\_size$
for $i = 1$ to $number\_of\_images$ do
    Open image $input\_image\_path + i +' .png'$
    Resize image to $(image\_size, image\_size)$ using LANCZOS filter
    Save resized image to $output\_image\_path + i +' .png'$
end for
```

4.1.2 Image Subset Selection

Another method for modifying the original LEVIR-CD dataset is to use fewer images to train the model. This approach introduces its own challenges, such as how to accurately sample a subset of images that represent the dataset without losing information. For instance, some images do not contain any changes, resulting in fully black labels. This creates a need to avoid training the model solely with such images. Therefore, a method has been designed for appropriately sampling the images. First, a procedure is implemented to extract the percentage of white pixels, indicating change, from the image, as presented in Algorithm 2.

Algorithm 2 Percentage Extraction of White Pixels

```
Require: $image\_name$, $path$
Open image $path + image\_name$
Get pixel data
$white\_pixels \leftarrow$ Count of white pixels
$total\_pixels \leftarrow$ Total number of pixels
$white\_percentage \leftarrow (white\_pixels/total\_pixels) \times 100$
return $white\_percentage$
```

Next, the complete algorithm for sampling the appropriate subset is defined in Algorithm 3.

Algorithm 3 Image Subset Sampling Algorithm

```
Require: $path$, $req\_percent$, $num\_keep$
$res \leftarrow empty\_array$
for $i = 1$ to $number\_of\_images$ do
    $img\_name \leftarrow' train\_' + i +' .png'$
    $result \leftarrow$ Algorithm 2($img\_name$)
    Append $result$ to $res$
end for
$sorted\_res \leftarrow$ Sort $res$ by $white\_percent$
$step \leftarrow$ len($sorted\_res$)$/num\_keep$
$reduced\_res \leftarrow empty\_array$
for $i = 0$ to $num\_keep - 1$ do
    Append $sorted\_res$[int($i \times step$)] to $reduced\_res$
end for
$img\_names \leftarrow$ Extract $name$ from $reduced\_res$
return $img\_names$
```

The sampling algorithm ensures that all images in the training dataset are sorted based on the percentage of white pixels in the label image, placing all-black images at the beginning of the collection. Images with more white pixels, indicating more change, are positioned towards the end of the collection. Depending on the chosen number of images, every n-th result is discarded, ensuring a uniform reduction in dataset size while maintaining the same distribution of white pixels representing change. An illustration of discarding 25% of the images is depicted in Figure 3, wherein the data denotes sorted images according to the percentage of white pixels. The black columns signify retained images, while the red columns denote discarded images.

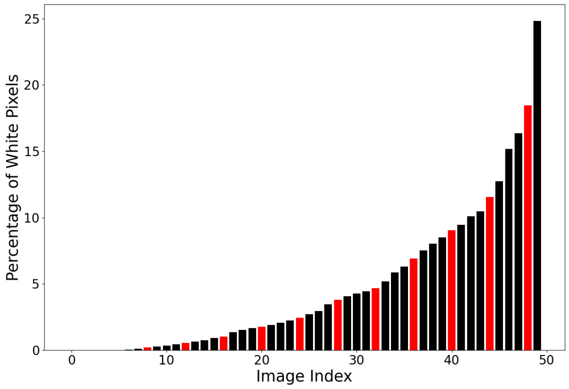


Figure 3: Discarding 25% of the images based on white pixel percentage. Black columns represent retained images, while red columns represent discarded images. The dataset is therefore reduced by 25%.

4.2 Computation

Each experiment involves utilizing the modified LEVIR-CD dataset on both the BIT and VcT frameworks. The procedure for each experiment entails training the model on the specified modified dataset, followed by validation of the model after each epoch, and subsequently testing the newly trained model on the testing partition of the dataset. During each training session, evaluation metrics are recorded to facilitate comparative analysis over time, as detailed in Section 5.

Each training cycle is conducted over 200 epochs. This duration is chosen based on preliminary experiments which indicated that 200 epochs are sufficient for evaluating all given scenarios. Moreover, this epoch count is widely adopted in research literature as it allows the model to stabilize within a specific region of the accuracy.

The baseline reference training experiment involved training the model on the entire LEVIR_CD dataset, denoted as LEVIR_100. Subsequent training runs are compared against this baseline to evaluate performance. The training experiments are categorized into two main groups: decreasing the resolution, as detailed in Section 4.1.1 **Resolution Reduction**, and decreasing the number of images, as detailed in Section 4.1.2 **Image Subset Selection**. For each category, eight experiments are conducted, as visible in Table 1.

To refer to one of the experiments, one can state, for example, "VcT, Category 1, By size," which indicates a 20% reduction by size from the original dataset input into the VcT

BIT		
Category	**By size***	**By resolution****
Category 1	20%	205x205
Category 2	30%	307x307
Category 3	40%	410x410
Category 4	50%	512x512
Category 5	60%	614x614
Category 6	70%	717x717
Category 7	80%	819x819
Category 8	90%	922x922
Category 9	100%, 1024x1024 (baseline)	
VcT		
Category	**By size***	**By resolution****
Category 1	20%	205x205
Category 2	30%	307x307
Category 3	40%	410x410
Category 4	50%	512x512
Category 5	60%	614x614
Category 6	70%	717x717
Category 7	80%	819x819
Category 8	90%	922x922
Category 9	100%, 1024x1024 (baseline)	

Table 1: Generated Datasets, * percentage of the original dataset - using Image Subset Sampling algorithm, ** pixels - using Resolution Reduction algorithm.

model. The process results in a total of 32 experiments, plus two baseline experiments, culminating in 34 experiments in total. Each experiment consisted of training, validation, and testing phases.

All computations are conducted using Google Colab's Pro+ GPU option, leveraging the NVIDIA A100 GPU, which provides a maximum GPU RAM of 80GB [16].

5 Results

This section elaborates on the limitations of the VcT, highlighting the changes made to the experimental methodology. Subsequently, the evaluation metrics are outlined, and the results are presented.

5.1 VcT Limitations

All BIT-related experiments, as presented in Table 1, were conducted as planned. However, certain difficulties arose with the VcT model. Specifically, when running the baseline experiment on the full LEVIR-CD dataset for 200 epochs using VcT, the model's estimated runtime was projected to be approximately 16-24 hours. This observation confirms the argument in this paper that experiments with smaller datasets are worth investigating to achieve faster computation and reduced hardware usage. Additionally, it was observed that around the 30th epoch, the results began to converge and subsequently oscillated around a horizontal line, indicating minimal further improvement. Figure 4 illustrates that when executing the baseline VcT LEVIR-CD, Category 9, 100%, experiment, the results converge at approximately 0.86 represented by the orange line, and oscillate around that value.

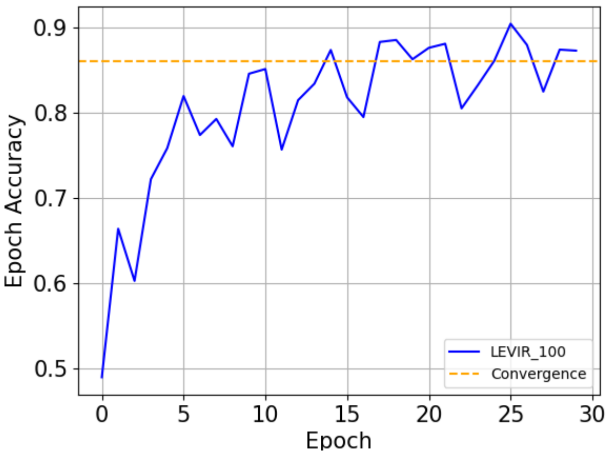


Figure 4: After 30 epochs, the results do not exhibit significant improvement.

To conduct the experiments within a reasonable timeframe, it was decided to run the VcT experiments up to the 30th epoch.

Additionally, when running full-resolution experiments, the model encountered excessive GPU RAM usage. When attempting to run the 512x512 images on the model, GPU RAM usage was still too high. To address this issue, the method proposed by Wang et al. [2] was employed: the 1024x1024 images were split into sixteen 256x256 images for both the A and B timeframe datasets, as well as the label dataset. Although the total amount of information (pixels) in the system remained the same, the processing method changed significantly. Therefore, it has been decided to abandon the approach of decreasing the images' resolution for testing the VcT model. Instead, the model has only been tested by limiting the dataset size, as in Algorithm 3. Consequently, as shown in Table 2, the final number of experiments was reduced from 34 (32 + 2 baselines) to 26 (24 + 2 baselines).

5.2 Evaluation Metrics

When evaluating the performance of an algorithm within this framework, three fundamental aspects must be taken into account: execution time, hardware capabilities, and model's performance.

Execution time is straightforward to measure, defined as the interval between the initiation and completion of the algorithm's commands. However, depending on hardware usage, time may not be a reliable measurement of model's performance. Therefore, it is used as an indicator of a trend rather than as an exact quantitative assessment.

This study identifies **maximum GPU RAM** as a pivotal factor limiting computational capacity. GPU RAM is easily accessible through Google Colab's interface, making it easy to measure. The rationale for choosing this metric is that the availability of more GPU RAM often correlates with the

BIT		
Category	**By size**	**By resolution**
Category 1	20%	205x205
Category 2	30%	307x307
Category 3	40%	410x410
Category 4	50%	512x512
Category 5	60%	614x614
Category 6	70%	717x717
Category 7	80%	819x819
Category 8	90%	922x922
Category 9	100%, 1024x1024 (baseline)	

VcT	
Category	**By size**
Category 1	20%
Category 2	30%
Category 3	40%
Category 4	50%
Category 5	60%
Category 6	70%
Category 7	80%
Category 8	90%
Category 9	100% (baseline)

Table 2: Generated datasets used for the experiments.

presence of more complex hardware components, which can significantly impact overall system costs and performance.

The final evaluation metric is **model performance**. As both BIT and VcT are already equipped with calculating the chosen metrics throughout the training after each epoch and for the testing dataset, it was concluded that these metrics are appropriate for evaluating model performance. These metrics include the F1 score (Equation 1), Precision (Equation 2), Recall (Equation 3), Intersection over Union (IoU) (Equation 4) of the change category, and Overall Accuracy (OA) (Equation 5):

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{4}$$

$$\text{OA} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively.

5.3 Experimental Results

The results of all of the experiments are present in the tables, including BIT by size in Table 3, BIT by resolution in Table 4, and VcT by size in Table 5. All of the most optimal results for each evaluation metric have been highlighted in bold.

5.4 Result Analysis

In this subsection, an analysis of the provided results is conducted along with general observations.

5.4.1 Time Analysis

Time analysis can be summarized by noticing that time increases almost every time when the dataset is increased either by size or by resolution. The time comparison for BIT, presented in Figure 5, and for VcT, shown in Figure 6, increase almost linearly.

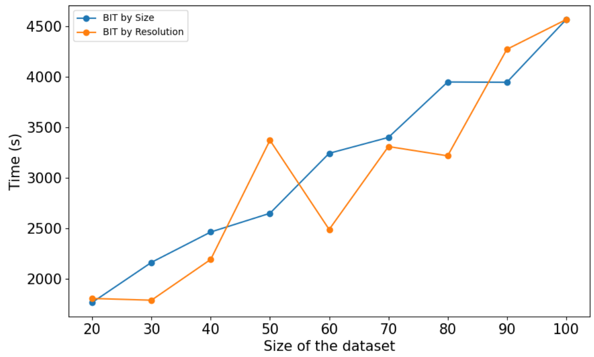


Figure 5: Time comparison for both BIT datasets - by size and by resolution. Blue represents BIT by size, while orange represents BIT by resolution.

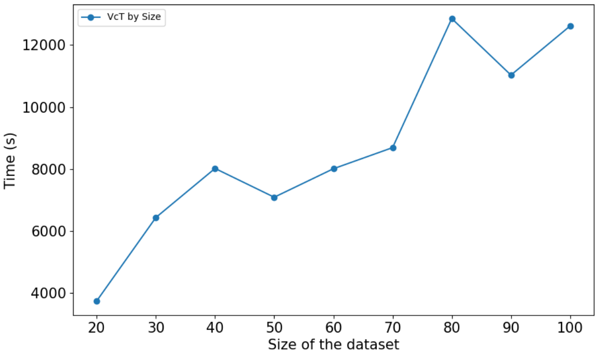


Figure 6: Time comparison for VcT dataset - by size.

5.4.2 Evaluation of Learning Capabilities

Datasets limited in size pose a critical question in modeling scenarios: is the model genuinely learning from the data? In this specific scenario, learning is defined as achieving an Overall Accuracy, as in Algorithm 5, above 0.5. A value of

Table 3: BIT by size.

	20%	30%	40%	50%	60%	70%	80%	90%	100%
F1	0.57924	0.58979	0.66704	0.68964	0.68639	0.71341	0.71391	**0.72250**	0.71283
Precision	0.64990	0.68181	0.68203	0.67659	0.71516	0.73047	0.73128	0.73922	**0.75886**
Recall	0.52243	0.51965	0.65269	0.70321	0.65985	0.69714	0.69734	**0.70653**	0.67206
IoU	0.40769	0.41823	0.50042	0.52630	0.52252	0.55450	0.55510	**0.56556**	0.55379
OA	0.7795	0.7853	0.8248	0.8363	0.8351	0.8488	0.8495	**0.8540**	0.8538
Time	**1770s**	2166s	2466s	2651s	3244s	3401s	3949s	3946s	4566s
GPU	**3.1GB**	3.7GB	3.9GB	4.2GB	4.1GB	4.3GB	4.3GB	4.3GB	4.4GB

Table 4: BIT by resolution.

	20%	30%	40%	50%	60%	70%	80%	90%	100%
F1	0.00000	0.14889	0.40432	0.53815	0.51532	0.63640	0.55026	0.66577	**0.71283**
Precision	0.00000	0.71200	0.59265	0.64801	0.63025	0.64595	0.66225	0.68296	**0.75886**
Recall	0.00000	0.08314	0.30682	0.46014	0.43585	0.62713	0.47067	0.64943	**0.67206**
IoU	0.00000	0.08044	0.25339	0.36813	0.34709	0.46671	0.37956	0.49899	**0.55379**
OA	0.4948	0.5806	0.7102	0.7765	0.7679	0.8113	0.7923	0.8259	**0.8538**
Time	1809s	**1792s**	2195s	3373s	2491s	3311s	3219s	4272s	4566s
GPU	**4.3GB**	**4.3GB**	**4.3GB**	**4.3GB**	**4.3GB**	**4.3GB**	**4.3GB**	**4.3GB**	4.4GB

Table 5: VcT by size.

	20%	30%	40%	50%	60%	70%	80%	90%	100%
F1	0.68772	0.68755	0.69116	0.76096	0.76512	0.79770	0.79817	**0.83601**	0.75456
Precision	0.84713	0.89200	0.84253	0.90504	0.91035	0.93147	0.89238	0.91727	**0.94213**
Recall	0.57881	0.55935	0.58590	0.65645	0.65984	0.69753	0.72196	**0.76798**	0.62928
IoU	0.52407	0.52387	0.52807	0.61415	0.61959	0.66348	0.66413	**0.71823**	0.60586
OA	0.8320	0.8444	0.8269	0.8769	0.8789	0.8971	0.8955	**0.9131**	0.9044
Time	**3748s**	6439s	8022s	7094s	8012s	8693s	12855s	11032s	12616s
GPU	6.3GB	**5.9GB**	6.3GB	6.5GB	6.4GB	6.7GB	7.0GB	9.5GB	12.0GB

0.5 signifies a random guess between black and white pixels, indicating change or no change. As shown, for VcT, this is not an issue, as all trials resulted in significantly higher values than 0.5. For BIT by size, there has also not been an issue. However, for BIT by resolution, the model does not appear to learn anything, with its final OA being 0.4948 when the dataset is in Category 1. To test the hypothesis whether more time resolves the issue, additional epochs were added to observe the model's reaction. The training with 400 epochs, instead of 200, for BIT Category 1 by resolution, is illustrated in Figure 7, showing epoch accuracy over time. It demonstrates that with sufficient time, the model does learn, but the accuracy still oscillates around 0.55, indicating only moderate improvement and not accurate results.

5.4.3 General Observations

One of the most notable observations is that the more complicated VcT model learns significantly faster than the baseline model, BIT. VcT demonstrates promising results even after the first epoch, whereas BIT takes considerably longer to start learning effectively. This difference is particularly pronounced when the dataset is small, such as BIT, Category 1 by size or resolution. A compelling comparison to illustrate this phenomenon involves comparing BIT Category 9, which serves as the baseline with theoretically the largest dataset, with VcT Category 1, which represents the theoretically weakest dataset, and BIT Category 8 for further context. For BIT Category 9 and BIT Category 8, the initial spike in learning occurs at the 10th and 14th epochs in purple and blue, respectively. In contrast, VcT Category 1 shows accuracy higher than 0.5 starting from the first epoch in green, increasing to over 0.6 before the 5th epoch. This comparison is illustrated in Figure 8.

6 Conclusions and Future Work

This section presents the conclusions derived from the experiments and elaborates on how these findings can be utilized by other researchers in the future work section.

6.1 Conclusions

General trends observed within the experiment align with the research hypothesis: the more time or the larger the dataset, the better the results. One significant observation is that even with much smaller datasets, such as VcT with 20% of the data, the model can achieve promising results, with accuracy exceeding 83%, as shown in Table 5. Additionally, it has been observed that the main phase of training occurs within the first 30 epochs for all models, after which the accuracy stabilizes and oscillates around a certain value.

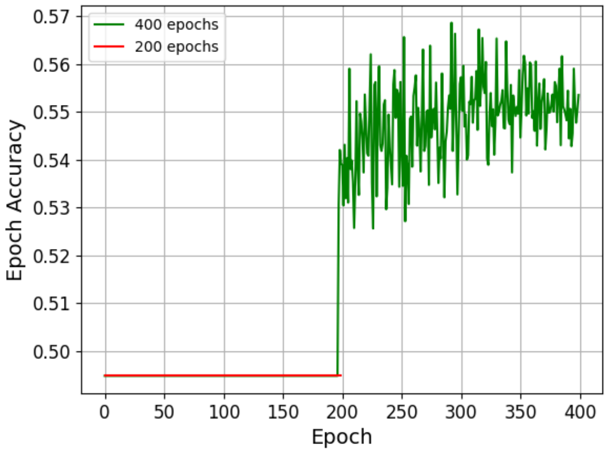


Figure 7: Epoch accuracy over extended time, BIT, Category 1, by resolution.

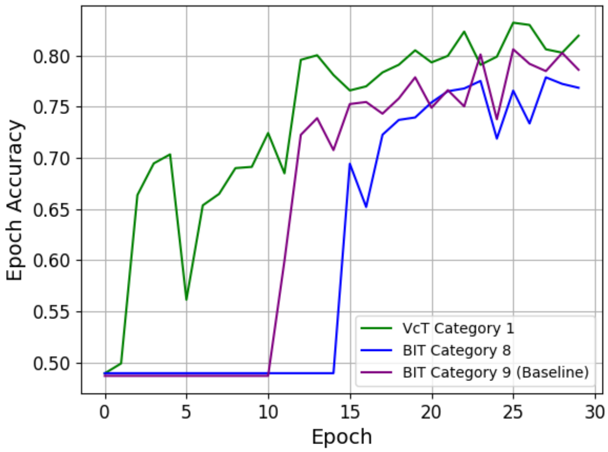


Figure 8: Epoch accuracy over time for the first 30 epochs. Purple - BIT, Category 9, 100%, Blue - BIT, Category 8, 90%, Green - VcT, Category 1, 20%.

The primary conclusion from the obtained results is that existing models for CD, which utilize Transformer technology, can indeed be trained with limited datasets. The more complex and recent VcT model performed well even with significant data reduction, such as in Category 1. Furthermore, reducing the dataset by the number of images appears to have a smaller negative effect on the final OA, as demonstrated by the OA for 20% in both BIT by size, presented in Table 3, and BIT by resolution, presented in Table 4.

6.2 Future Work

Identifying how smaller datasets used for training models can still achieve effective results is significant in the CD research field. This is particularly important in collaborative artificial intelligence, where both humans and machine learning models cooperate to obtain optimal outcomes. For instance, a smaller dataset might be employed to identify "interesting" or noteworthy areas in an image with approximately 80% accuracy, a task that can be accomplished relatively quickly. Subsequently, a human can verify these identified areas by inspecting the suggested regions. This approach allows for cropping or resizing the image, which can then be input into more advanced models requiring longer training periods and more resources. This method exemplifies one of the many practical applications of the findings discussed in this paper.

Furthermore, the utilization of smaller datasets is critical for the successful deployment of models and algorithms in production environments. Although numerous research studies produce highly accurate results, the significant costs associated with their deployment frequently restrict these findings to theoretical experimentation. This separation underscores the need for developing cost-effective methodologies that can leverage limited data without compromising the robustness and accuracy of the models. Addressing these constraints makes transitioning from theoretical research to practical applications feasible, enhancing the real-world impact of CD Transformer-based model innovations.

7 Limitations

Due to the limited timeframe, only one experiment per category was conducted. However, given more time, multiple experiments should be conducted in future research to average the results and achieve greater accuracy and less error. Additionally, this work does not aim for the highest accuracy or the best-performing model, which is another limitation of the research presented in this paper.

8 Responsible Research

Although fast CD algorithms may lead to significant improvements in many areas, including agricultural activities and urban planning, they may also pose potential dangers if misused. One of the significant risks lies in the military application of these algorithms, where they could be used to gain an advantage over adversaries. The recognition of potential targets based on recent changes in satellite imagery may lead to unnecessary attacks on civilians. Furthermore, if these tools fall into the hands of terrorists, they could also pose a danger to civilian lives. Another critical aspect of using these algorithms is the consideration of privacy issues. As the datasets used for CD are predominantly collected by private corporations and countries, there needs to be a mechanism to ensure privacy. This could be achieved by blurring private areas or using techniques such as mmWave [17] to detect image changes without directly revealing the image itself. In conclusion, while CD algorithms have the potential to bring about substantial advancements, it is crucial to address their ethical and privacy implications to prevent misuse and protect people's privacy.

References

[1] Weichao Sun Huan Yang Bin Wang Yunhe Teng, Shuo Liu and Jintong Jia. *A VHR Bi-Temporal Remote-Sensing Image Change Detection Network Based on Swin Transformer*, 2023.

[2] Xixi Wang Ziyan Zhang Lan Chen Xiao Wang Bo Jiang, Zitian Wang and Bin Luo. *VcT: Visual change Transformer for Remote Sensing Image Change Detection*, 2023.

[3] Zipeng Qi Hao Chen and Zhenwei Shi. *Remote Sensing Image Change Detection with Transformers*, 2021.

[4] Xiangtai Li Shuchang Lyu Zhaoyang Xu Qi Zhao Guangliang Cheng, Yunmeng Huang and Shiming Xiang. *Change Detection Methods for Remote Sensing in the Last Decade: A Comprehensive Review*, 2023.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention is all you need*, 2017.

[6] Alexander Kolesnikov Dirk Weissenborn Xiaohua Zhai Thomas Unterthiner Mostafa Dehghani Matthias Minderer Georg Heigold Sylvain Gelly Jakob Uszkoreit Alexey Dosovitskiy, Lucas Beyer and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2020.

[7] Ken Turkowski and Steve Gabriel. *Filters for Common Resampling Tasks*, 1990.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*, 2016.

[9] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. *The Global k-Means Clustering Algorithm*, 2003.

[10] MG Shanmuga Priya Suresh Shalini Mandyam, Sukeerthi and Kavitha Srinivasan. *Natural Disaster Analysis using Satellite Imagery and Social-Media Data for Emergency Response Situations*, 2023.

[11] Y. Kim and M.-J. Lee. *Rapid Change Detection of Flood Affected Area after Collapse of the Laos Xe-Pian Xe-Namnoy Dam using Sentinel-1 GRD Data*, 2020.

[12] justchenhao. *GitHub Repository*. https://github.com/justchenhao/BIT_CD.

[13] Event-AHU. *GitHub Repository*. https://github.com/Event-AHU/VcT_Remote_Sensing_Change_Detection.

[14] Hao Chen and Zhenwei Shi. *A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection*, 2020. https://www.mdpi.com/2072-4292/12/10/1662.

[15] Pillow Contributors. *Pillow 2.7.0 Release Notes*, 2020. https://pillow.readthedocs.io/en/stable/releasenotes/2.7.0.html#blur-radius.

[16] Google Research. *Colaboratory FAQ*, 2024. Available at: https://research.google.com/colaboratory/faq.html#:~:text=Colab%20is%20a%20hosted%20Jupyter,%2C%20data%20science%2C%20and%20education.

[17] Jia Zhang, Rui Xi, Yuan He, Yimiao Sun, Xiuzhen Guo, Weiguo Wang, Xin Na, Yunhao Liu, Zhenguo Shi, and Tao Gu. *A Survey of mmWave-Based Human Sensing: Technology, Platforms and Applications*, 2023.

A ChatGPT Usage

This appendix provides the usage of ChatGPT for the Research Project. ChatGPT was primarily used for inspiration, latex formatting, and code bugs debugging.

- **Prompt:** why are cnns good for image processing
- **Prompt:** Can you change: [1] A VHR Bi-Temporal Remote-Sensing Image Change Detection Network Based on Swin Transformer Yunhe Teng 1 , Shuo Liu 2,*, Weichao Sun 2 , Huan Yang 1 , Bin Wang 1 and Jintong Jia 1 in this format: @Manualexample, title = example referance, author = Firstname Lastname, year = 1900,
- **Prompt:** is Intel(R) Iris(R) Xe Graphics a GPU?
- **Prompt:** how to do this: Verify your CUDA installation. Make sure CUDA is installed correctly and the PATH variables are set up properly.
- **Prompt:** whats the difference between google colab t4 gpu, a100 gpu, l4 gpu? what are the specs and which one is the fastest one
- **Prompt:** where to add reference so that I can refer to this table, how to add it into the latex code
- **Prompt:** how to do new page after references, and remeber it is two columns format in latex
- **Prompt:** when I have a research paper and I have pseudocode for my own designed algorithms, should I add it in the appendix or the main text?
- **Prompt:** are there any research papers that look on how visual image transformers for change detection act in the environment with limited data, such as small datasets or not high resolution datasets