TU WIEN Informatics

Change Detection in Graffiti Images

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Fabian Dachs, B.Sc.
Matrikelnummer 01627961

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dipl.-Ing. Dr.techn. Sebastian Zambanini

Wien, 27. August 2024

Fabian Dachs	Sebastian Zambanini

TU WIEN Informatics

Change Detection in Graffiti Images

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Fabian Dachs, B.Sc.
Registration Number 01627961

to the Faculty of Informatics

at the TU Wien

Advisor: Dipl.-Ing. Dr.techn. Sebastian Zambanini

Vienna, August 27, 2024

Fabian Dachs	Sebastian Zambanini

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Declaration of Authorship

Fabian Dachs, B.Sc.

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.
I further declare that I have used generative AI tools for the purpose of writing assistance, including grammar correction, sentence structuring, and enhancing the overall readability of the text. All scientific content, research findings, and conclusions presented in this thesis are my original work and have been developed without the aid of AI tools. In the appendix "Overview of generative AI tools used" I have listed all generative AI tools that were used in the creation of this work, and described how they were used.

Vienna, August 27, 2024

Fabian Dachs

Kurzfassung

Diese Arbeit beschäftigt sich mit dem Verfahren der Change Detection (dt. Veränderungserkennung) von Graffiti-Wänden, einem weitgehend unerforschten Bereich für die Veränderungserkennung. Dazu wurden drei Schritte durchgeführt:

1. **Aufbereitung der Daten**: Da nur wenige Daten verfügbar sind und diese in starker Korrelation zueinanderstehen, musste ein neuer Datensatz erstellt werden. Dafür wurden digitale Graffiti-Bilder über ein Foto einer Graffitiwand gelegt, um einen synthetischen Datensatz zu generieren.
2. **Trainieren der Modelle**: Um die Performance existierender Change Detection Modelle in Graffiti-Bildern zu erhöhen, wurden die Modelle mit Graffiti-Bildern trainiert. Das Training wurde in verschiedenen Varianten durchgeführt (Finetuning und Training von Grund auf) sowie mit verschiedenen Kombinationen der verfügbaren Daten durchgeführt. Weiters wurde ein einfacheres Modell implementiert, um die Komplexität des Verfahrens bewerten zu können.
3. **Evaluierung**: Schließlich wurden die Modelle anhand der synthetischen Daten und durch einen händisch erzeugten realen Datensatz evaluiert.

Die Evaluierung zeigte, dass die existierenden vortrainierten Modelle nicht für Change Detection in Graffiti-Bildern verwendet werden können, da sie nur einen durchschnittlichen F1-Score von 0,134 erreichten. Alle Modelle konnten durch das Finetuning ihre Performance signifikant verbessern. Im Durchschnitt steigt der F1-Score auf 0,612 für die Modelle, die mit allen verfügbaren Daten trainiert wurden. Bei den meisten Modellen hatten die synthetischen Daten hier einen positiven Effekt, vor allem die Precision konnte bei allen Modellen durch das Verwenden der synthetischen Daten stark verbessert werden. Das Modell mit dem höchsten F1-Score war das eigen-implementierte Modell nur mit realen Daten trainiert, welches einen F1-Score von 0,692 erreicht.

Abstract

The thesis presents an empirical evaluation of change detection in graffiti images, a largely unexplored domain for change detection. To conduct this evaluation three steps were conducted:

1. **Preparation of a dataset**: As the available graffiti data is very scarce and highly correlated a new dataset had to be established. This was achieved by creating a synthetic dataset by adding digital graffiti to images of a graffiti wall.
2. **Training of the models**: To increase the performance of the state-of-the-art change detection models in the domain of graffiti images, the models were trained on graffiti data. The training was performed in different settings (finetuning and training from-scratch) as well as using different combinations of the available data. In this step, a simpler baseline model was implemented as well, to be able to evaluate the complexity of the task.
3. **Evaluation**: Finally, the models were evaluated on the synthetic as well as on a hand-labeled real-world dataset.

The evaluation showed that the original models cannot be used without finetuning for change detection in graffiti images, achieving an average F1-Score of 0.134. All models showed a significant improvement after finetuning, on average the F1-Score increased to 0.612 for the models trained on all available data. For most models, synthetic data had an overall positive effect, especially since the precision could be improved for all models using synthetic data. The best-performing model was the baseline model, only trained on hand-labeled real-world data, achieving an F1-Score of 0.692.

Contents

CHAPTER 1

Introduction

Change detection is the process of identifying changes in images over time [Sin89]. It is performed on image pairs taken at different times to analyze the differences between them [Sin89, MV22]. The difference or change that shall be detected depends on the use case. For instance, Figure 1.1 shows the result of a change detection performed on a sample from the LEVIR-CD satellite image dataset [CS20], where a binary change map indicates the changes between an image pair. In this case, the change detector is used to mark changes in urban development, where e.g. new buildings should be detected as changes but changes in the vegetation (e.g. trees changing to lawn) should be ignored [CS20]. Most of the literature focuses on remote sensing images [LMBM04, HCC+13, JPZ+22]. In theory, the principles of change detection are similar across all domains, regardless of the type of image being analyzed. Other domains for change detection that have been studied include:

- **Medical Images**, for example to detect changes in X-ray or microscopy data [RAAKR05].
- **Industrial Images**, for example to detect defects in manufactured products [CCM+20].
- **Security Images/Camera-trap Images**, for example, to detect changes in surveillance footage for detecting potential threats or wildlife [RAAKR05].
- **Driver Assistance Systems**, for example, to detect changes in the driving environment, such as lane changes, tunnel entry and exit, freeway entry and exit, and overpass ahead [FCF03].

One of the applications of change detection that remains largely unexplored is graffiti images. Change detection in graffiti images is used to document, monitor, and analyze graffiti and street art [WVP23].

Figure 1.1: Example of change detection of ChangeFormer, developed by Gedara et al. [GCBP22], on a sample from the LEVIR-CD [CS20] satellite image dataset. The before-change image is on the left, the after-change image is in the middle and the change map is on the right. The red square should help the viewer see, whether the changes have been detected correctly. (Taken from [GCBP22])

1.1 Problem Statement and Motivation

This thesis is in cooperation with the project INDIGO (INventory and DIsseminate graffiti along the dOnaukanal), which aims to build the basis to systematically document, monitor, and analyze 12.9 km of graffiti along the Donaukanal [WVW+23].

The current method of marking the location of new graffiti works is by manually spotting new artworks either online or onsite [WVP23]. This method works for documenting large graffiti but fails for smaller graffiti like tags or political slogans because they have less presence on social media and are harder to detect onsite. This results in a bias in the graffiti documented. Therefore Wild et al. [WVP23] came to the conclusion that it is impossible to document the Donaukanal's graffiti-scape without using automated change detection.

Change detection on graffiti images holds different challenges as compared to satellite and aerial images [WVP23]. With satellite images, the task is to track changes in land use, vegetation cover, urban development, and other features of the landscape, whereas in the domain of graffiti images, one or more graffiti are added to a scenery filled with graffiti [WVP23, WVMP23].

Figure 1.2 shows exemplarily the diversity graffiti can provide. The desired change maps can describe large new graffiti, new outlines, very small and subtle changes, or no change at all. Even though the main graffiti remained in the second and third example of Figure 1.2, lots of smaller changes to the artwork were made. These changes shall also be detected in detail, while other changes such as changes in the leaves on the ground or the change of the illumination should not be detected as changes [WVP23]. As the image pairs in change detection are not always captured under the same conditions, this can

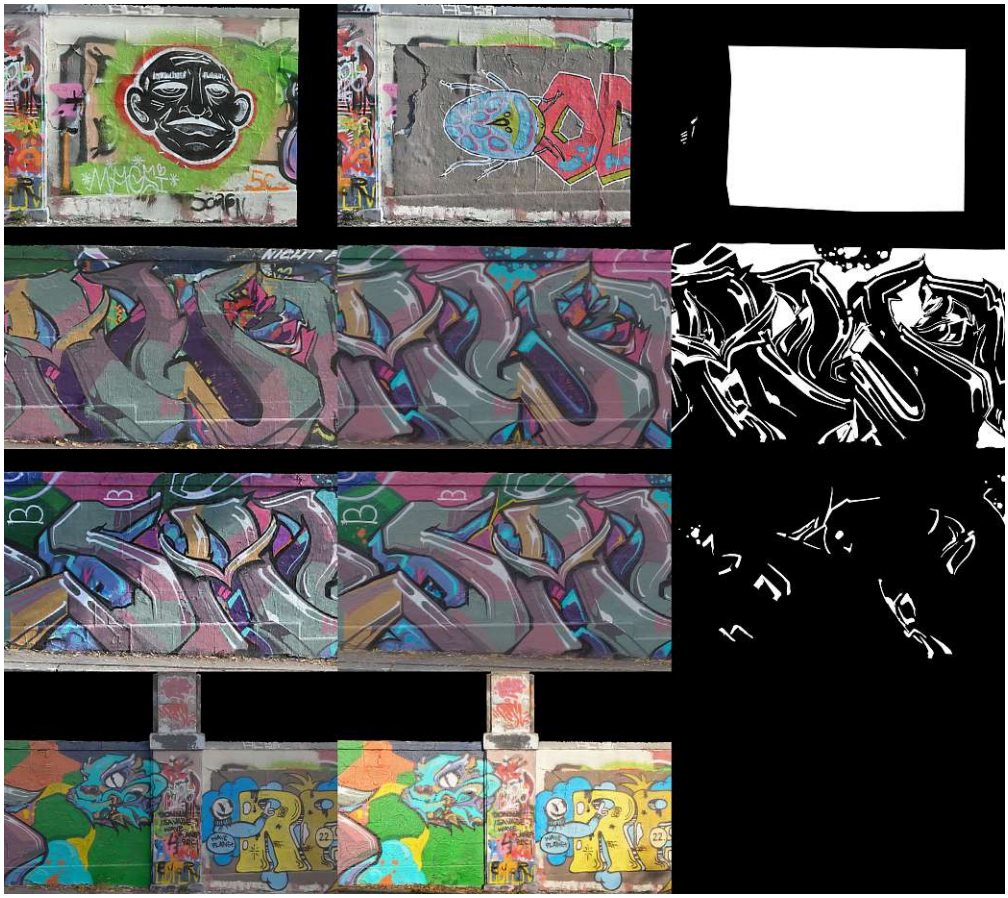


Figure 1.2: Samples of a graffiti-centered change detection dataset developed by Wild et al. [WVMP23]. The before-change images are on the left, the after-change images in the middle, and the change maps are on the right. The dataset replaces all pixels that are not part of the graffiti wall with black pixels.

provide a range of challenges [CHL+23]. For graffiti images, this includes the following challenges [WVW+23]:

1. **Camera and viewpoint**: Using different camera sensors and camera settings can result in significantly different images. This includes differences in the image resolution but also different viewpoints of the subject that can make it difficult to align the image pair correctly. This can provide problems for change detection in aerial or satellite images as well as in graffiti images [CHL+23, WVP23]. An example can be seen on image pair B in Figure 1.3, where the viewpoint slightly changes between images B1 and B2. How much the viewpoint changes depends on the dataset, for the data used by INDIGO the pixels should be mostly co-registered, but may still cause problems for change detection [WVP23].

2. **Heterogeneity in scale**: A change detector should work scale independently. For graffiti images, this includes the detection of new large graffiti as well as smaller artifacts like tags, political slogans, or small changes in a larger graffiti [CHL+23, WVP23]. This presents a different challenge because smaller artifacts do not change the general texture as a large graffiti does. For example in Figure 1.3, changes in image pair A are smaller compared to the large graffiti in image pair B.

3. **Noise and Artifacts**: Natural artifacts can block or change the view on the graffiti temporarily, like for example wind can change the presence of vegetation (branches of trees), wet walls can produce changes in the texture of the walls and reflections can make parts of the wall shiny [CHL+23, WVP23]. None of these artifacts should be detected as new graffiti. For example in image pair B of Figure 1.3, the leaves on the ground should not be classified as new graffiti.

4. **Illumination**: As the image pairs are captured at different times of the day, the shadows and general illumination can change the look of the two images. This is the case for both remote sensing images and graffiti images but appear different for these domains. Remote sensing images mostly have shadows from objects that are present in the image such as hills or buildings which changes the illumination only for small parts of the scenery. For graffiti images, the source of the shadow is not necessarily present in the image, such as a building on the opposite side of the street or a bridge, creating a shadow covering large parts of the image [CHL+23, WVP23]. For example in image pair A of Figure 1.3, half of image A1 is illuminated by the sun while the other half is in the shadow.

It remains unknown how state-of-the-art models can generalise across different domains like graffiti images.

1.2 Aim of Work and Research Questions

This thesis aims to conduct an empirical evaluation of how state-of-the-art change detection models perform on a different domain, namely on graffiti images. The intuition is that models, only trained on remote sensing data perform poorly on a different domain, namely on graffiti images [JPZ+22]. This thesis also compares how state-of-the-art change detection models' performance is enhanced after finetuning the models with real graffiti images and synthetic graffiti images. The models used are described in Section 2.2, and the metrics used for this empirical evaluation are explained in detail in Section 4.1. The research questions of the thesis are as follows:

How do state-of-the-art models perform on graffiti images?
As state-of-the-art models are mostly only trained and evaluated on satellite and aerial images, this thesis studies how the state-of-the-art models can generalize and perform on graffiti images [JPZ+22].

Figure 1.3: Pairs of graffiti images showcasing challenges of change detection. (Pair A from [WVMP23], pair B from Spraycity[1])

How much can the state-of-the-art models benefit from finetuning with graffiti images?
To encounter the problem that state-of-the-art models are predominantly trained on satellite and aerial images, this thesis studies how the state-of-the-art models finetuned on real graffiti images and synthetically generated graffiti images perform on graffiti images [JPZ+22].

1.3 Structure of the Thesis

The following structure of the thesis shows how the research questions are answered scientifically:

Chapter 2 provides an overview of change detection and an introduction to the methods and techniques used to tackle this problem. In order to get an understanding of the task and the available state-of-the-art models for change detection, a literature overview of models that achieve comparably good results in different contexts, e.g. satellite or aerial images, is presented.

[1]Source: Spraycity https://spraycity.at/

In Chapter 3 the implementation is presented and explained. This includes the generation of synthetic data. As real labeled data is very sparse it is necessary to work with synthetic data to be able to finetune the available models. Further, a framework to finetune the considered model is described, as well as a simple network, providing a baseline for the evaluation.

In Chapter 4 the models are evaluated on graffiti data, in order to compare the different models (original and trained/finetuned with different settings). This chapter explains the evaluation metrics and presents the evaluation results of all models.

Chapter 5 provides a conclusion of the findings and answers to the research questions.

CHAPTER 2

Related Work

This chapter provides an overview of change detection and an introduction to the methods and techniques used to tackle this problem. Many different approaches for change detection have been developed. (Lu et al. mention 31 approaches in 2004 [LMBM04], Hussain et al. mention 21 techniques [HCC+13] in 2013, and in 2023, Parelius mentions 20 different deep learning approaches for change detection [Par23].) Therefore it is not trivial to select the state-of-the-art models to evaluate, finetune, and train in this thesis. This chapter also serves as a justification for the selection.

2.1 Change Detection

Change detection is a remote sensing task that aims to identify differences in the state of an object or phenomenon by observing it at different times [Sin89]. In 1972 Lillestrand [Lil72] proposed a significant increase in quality and quantity in detecting changes in side-looking radar imagery when the human viewer is only exposed to the changes rather than the whole information. Singh [Sin89] proposed many more applications like land use analysis, monitoring of shifting cultivation, assessment of deforestation, a study of changes in vegetation phenology, seasonal changes in pasture production, damage assessment, crop stress detection, disaster monitoring, snow-melt measurements, day/night analysis of thermal characteristics and other environmental changes.

There are two types of change detection, pixel-based and object-based [JPZ+22]. In pixel-based change detection, change is attempted to be detected on a pixel level [JPZ+22]. Usually, pixel-based change detection models are more sensitive to noise as they compare the spectral or textual values of single pixels without considering the relationship to neighboring pixels. However, modern models, that use a deep learning encoder-decoder architecture that takes spatial information into account, could reduce this sensitivity [JPZ+22]. Object-based change detection analyses the changes on an object level [JPZ+22]. All the pixels of an object, which is a group of local pixel clusters, are assigned to the

same class, which reduces the negative effects of noise of the classification [JPZ+22]. The limitation of this method is the need for a reliable and flexible image segmentation algorithm, which creates groups of local pixel clusters. A possible over-segmentation or under-segmentation can lead to worse change detection results [JPZ+22]. For example, classifying all graffiti on a wall as "graffiti" will not lead to satisfying outcomes.

2.1.1 Change Detection Methods Before Deep Learning

Before deep neural networks were established, various pixel-based and object-based change detection methods were proposed, relying on handcrafted features extracted from complex feature extractors [LMBM04, HCC+13]. In 2004 Lu et al. [LMBM04] analysed 37 different change detection methods, each with different advantages and disadvantages. Similarly, Hussain et al. [HCC+13] in 2013 came to the conclusion, that there is no optimal change detection method for all different kinds of remote sensing data. It was thus unsurprising that a large number of change detection techniques had been developed [HCC+13].

Spectral Mixture Analysis

One of the methods used in land-use and land-cover change detection is Spectral Mixture Analysis (SMA) [SGE17, BMMW94, WPM00]. SMA classifies pixels using the spectral response of multiple wavelengths [SGE17]. As wavelengths are reflected differently depending on the surface, the pixels can be classified using histogram matching between different classes, e.g. water, forest, cleared, or urban [WPM00]. After obtaining the classification maps of an image pair, the change map is computed by comparing the classes pixel by pixel [WPM00].

Support Vector Machine for Change Detection

Another method used in land-use and land-cover change detection is the Support Vector Machine (SVM) [BBM08, HSK+08]. SVM is a supervised algorithm used for classification into two groups. It works by transforming the data in a way that the method can find a hyperplane that separates the two classes [BBM08]. An example is the training data automation-support vector machine (TDA-SVM) by Huang et al. [HSK+08] from 2008. Here the authors tested the method in over 19 study areas selected to cover major forest biomes across the globe and achieved accuracies well above or near 90% for the study areas.

2.1.2 Change Detection Methods Using Deep Learning

The poor expressiveness of the features used in traditional models provided a clear limitation as the data sources got more diverse, spatial resolution increased and image details got richer [JPZ+22]. Similarly to other computer vision tasks, deep learning based methods have achieved success over traditional methods [JPZ+22]: Figure 2.1 shows the increasing trend of using deep learning for change detection in the literature from 2000

to 2022. The values were gathered by using the advanced search on Google Scholar. The values are rounded by Google Scholar but still show a clear trend of increasing papers using deep learning or neural networks[1].

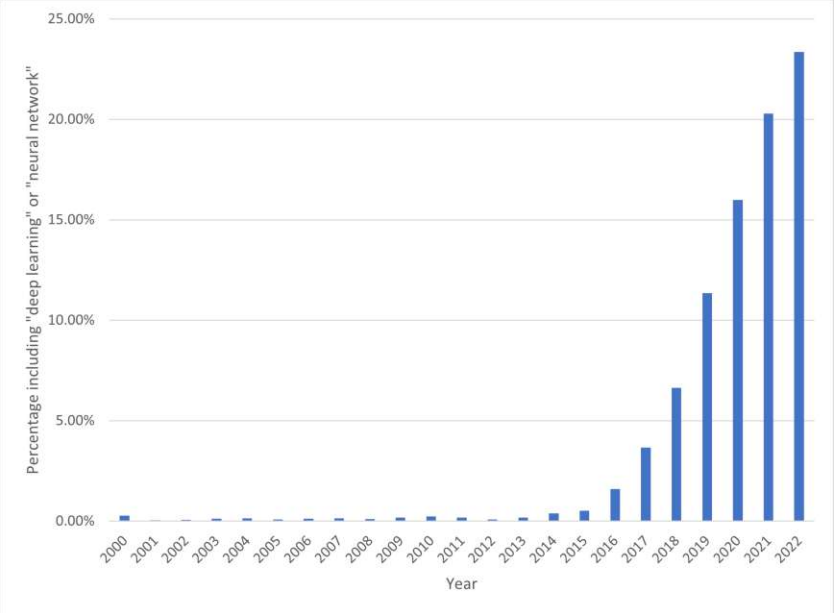


Figure 2.1: Percentage of papers including "change detection" and "neural network" or "deep learning" in the total amount of papers including the term "change detection".

The advantage of neural networks over traditional methods is the ability to craft very complex high-level features from the given data by itself [GBC16]. A deep neural network or multi-layer perceptron consists of many layers each of which maps a set of input values to output values using simple functions [GBC16]. In other words, after each layer, a new representation is computed. During training, the model learns to map the input values to the right output representation. To solve computer vision tasks as well as other tasks more complex neural network architectures have been developed, such as convolutional neural networks, encoder-decoders, and Transformer-Based Networks [JPZ+22].

Convolutional Neural Networks in Change Detection

Convolutional neural networks (CNNs) are neural networks that use a convolution in at least one layer [GBC16]. Using convolutional over fully connected layers gives two advantages, (1) it has far fewer unique parameters because it uses a filter instead of a unique weight for every input value [GBC16], and (2) the layers are much more sparsely connected, where each value of the new layer is only connected to a small number of neighboring values - this way it can focus on its spatial receptive field [JPZ+22].

[1]Source: Google Scholar https://scholar.google.com/

These advantages are very useful for change detection because they allow to derive high-dimensional features from images and improve the accuracy of change detection [JPZ+22]. An example of using CNNs for change detection is the ChangeNet, developed in 2018 by Varghese et al. [VGRB18]. It uses convolutional layers to extract features from the two input images and to compute the change map.

Encoder-Decoder / U-Net in Change Detection

A popular architecture for change detection is an encoder-decoder network. Figure 2.2 shows the basic architecture of the network. It consists of an encoder part, which compresses the input vector into a smaller hidden layer, and the decoder part which reconstructs the hidden layer into an output layer [JPZ+22]. This architecture is popular for sequence-to-sequence models such as natural language processing (NLP) or image-to-image models for change detection or image segmentation, [JPZ+22, BKC17]. The idea is to lose the excess information by compressing the image into a smaller representation. When reconstructing the image to, for example, a change map or segmentation map, only the relevant information is used [BKC17, Par23].

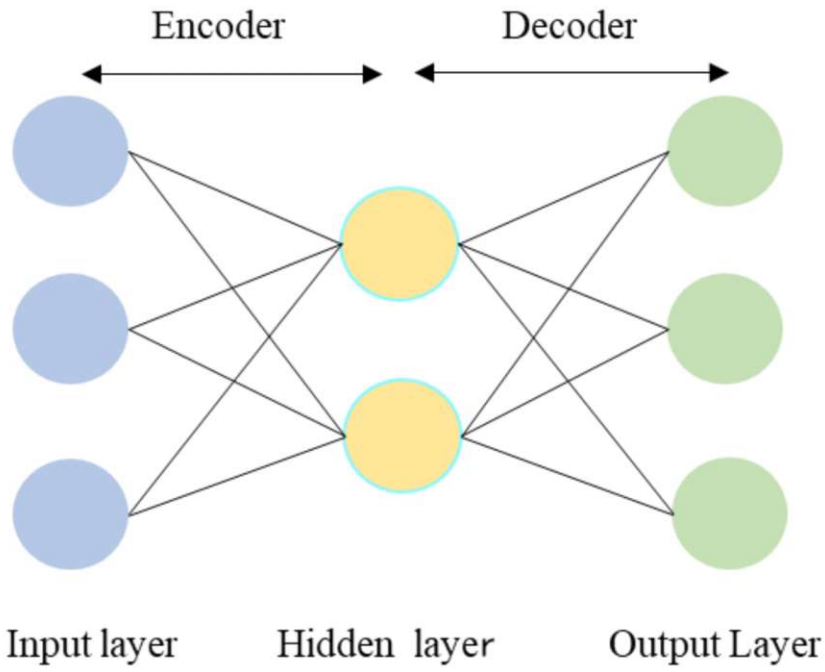


Figure 2.2: Basic encoder-decoder architecture. (Taken from [JPZ+22])

One version of an encoder-decoder architecture is the U-Net from Ronneberger et al. [RFB15], developed in 2015 for biomedical image segmentation. The idea was to add skip connections during the reconstruction of the image, to increase detailed information on the output [RFB15]. The model architecture can be observed in Figure 2.3. To use this

architecture in change detection with two input images, some modifications are necessary [Par23]. The three common methods are, (1) the bi-temporal image (two three-channel images) can be stacked to have one six-channel input image, (2) to use two Siamese decoders, which means that the two decoders share the same parameters, and the output is concatenated during reconstruction, or (3) two Siamese decoders and the absolute value of the difference is concatenated to the reconstruction [Par23]. An example of using a U-Net architecture for change detection is the Fully Convolutional early Fusion (FC-EF) model by Daudt et al. developed in 2018 [DLSB18]. The architecture concatenates the two input images in the first layer and it uses fewer layers, compared to the original U-Net, to combat the lack of available training data.

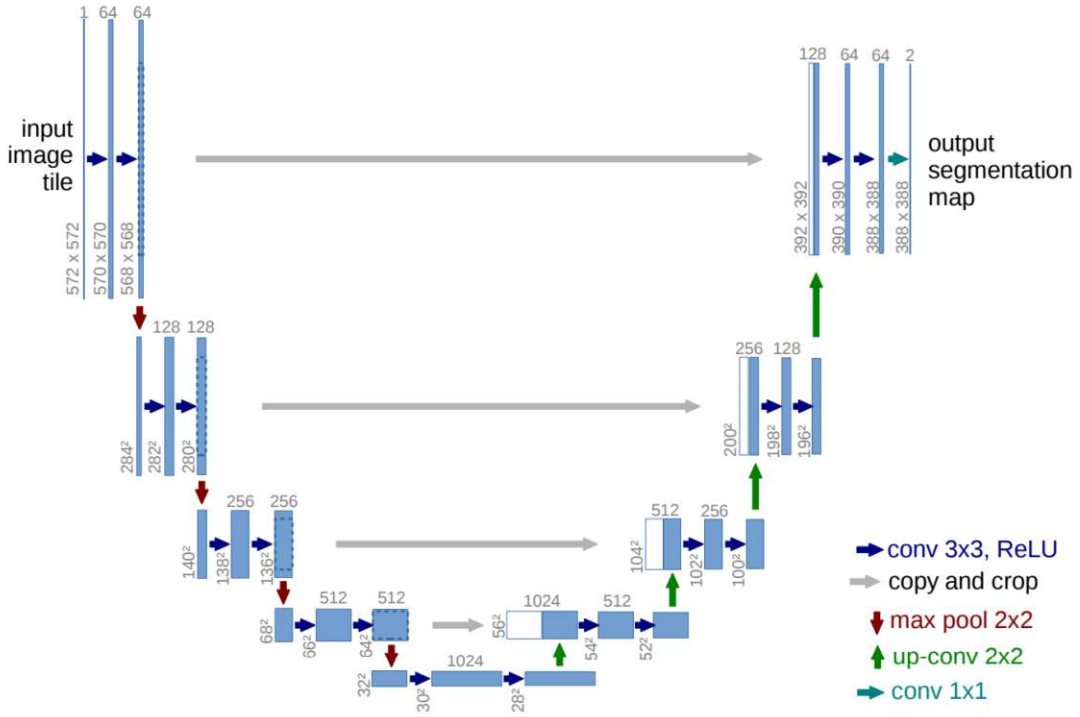


Figure 2.3: The U-Net architecture for biomedical image segmentation. The dark blue arrows indicate a convolutional layer followed by a ReLU activation layer, the grey arrows indicate a copying and cropping of the feature map, the red arrows indicate a max pool operation used to reduce the spatial dimensions, the green arrows indicate an upsampling of the feature map followed by a convolutional layer and the cyan arrow indicates a convolutional layer with a kernel size of 1x1. (Taken from [RFB15])

Transformer-Based Networks in Change Detection

The transformer-based models were developed by Vaswani et al. [VSP+17] in 2017 for NLP tasks. The architecture uses self-attention, which allows the model to focus on

different parts of the input for each element of the sequence, such that the model can process the entire sequence in parallel [VSP+17]. Transformers use an encoder-decoder structure, where the encoder computes a sequence of hidden states which are used by the decoder to generate an output sequence [VSP+17]. After the success of transformer-based models in NLP [VSP+17], transformers for visual computing tasks were developed using the same principles [DBK+20]. In 2020 Dosovitskiy et al. [DBK+20] introduced the Vison Transformer (ViT) for object detection. ViT only uses a transformer encoder which detects objects from the hidden states using an MLP prediction head. As an image is too big to compute self-attention on a pixel basis, ViT splits the image into patches to compute the self-attention [DBK+20]. The models achieve a significantly larger receptive field by using self-attention instead of convolutional layers. This is because the self-attention mechanism can utilize information from the entire image starting from the first self-attention layer [DBK+20]. As change detection is an image-to-image task, different architectures are needed [Par23]. Some transformer-based change detection models use a transformer encoder and a transformer decoder [CQS21, ZWCL22], whereas others use only use transformer encoders and MLP decoders to reconstruct the change map [GCBP22, LZDD22, WZLW21].

2.2 Overview of the State-Of-The-Art Models

This section presents the state-of-the-art models that are finetuned and evaluated on graffiti images in this thesis. To get a diverse overview of the available models, three models with different structures and techniques are chosen:

2.2.1 ChangeNet-v2 (2019)

ChangeNet-v2, developed by Prabhakar et al. [PRB+19] is based on the convolutional architecture of the ChangeNet, developed in 2018 by Varghese et al. [VGRB18]. As can be observed in Figure 2.4, the network consists of three main components. First, there is a Siamese network, which means that the parts of the network for each input image share the same weights. This Siamese network is used to extract the features with six convolutional layers, including two max-poolings in later layers. In the second component, for each of the six feature map pairs a correlation map is computed, followed by three convolutional layers. For some of the feature maps transposed convolutions are used as the feature maps have different resolutions. In the final part, the six feature maps are concatenated and after a final convolutional block, the change map is created [PRB+19].

ChangeNet-v2 was chosen for this thesis as it uses a simpler architecture, mainly relying on convolutional layers. By comparing this architecture to more complex transformer architectures an evaluation of the necessity of such complexity in the models is possible.

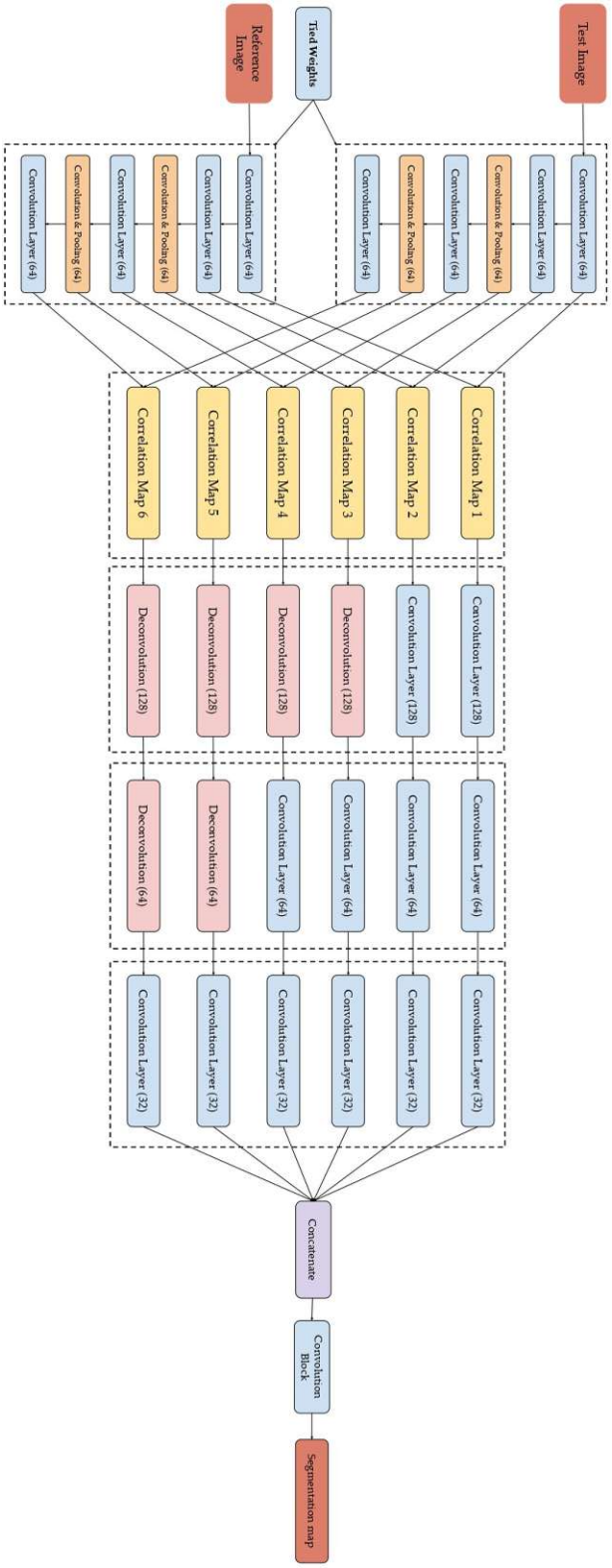


Figure 2.4: ChangeNet-v2 model for change detection. (Taken from [PRB+19])

2.2.2 Bitemporal Image Transformer (2021)

The Bitemporal Image Transformer (BIT) developed by Chen et al. [CQS21] is a transformer architecture with a convolutional backbone. The illustration of the network can be observed in Figure 2.5. It consists of three parts. First a CNN as a backbone to extract two feature maps from the two input images. In the second part, a semantic tokenizer is used to generate compact semantic tokens for each feature map. These sets of tokens are concatenated and fed into a transformer encoder-decoder network, which outputs two upsampled feature maps, that incorporates the high-level semantic information that is useful to detect the changes. In the last part, the two feature maps are pixel-wisely subtracted and a convolutional prediction head generates the binary change map.

BIT was chosen for this thesis as it uses a transformer encoder-decoder network that has shown success in various domains [CQS21]. By comparing it to different transformer architectures an evaluation of which transformer architecture is suitable for graffiti change detection is possible.

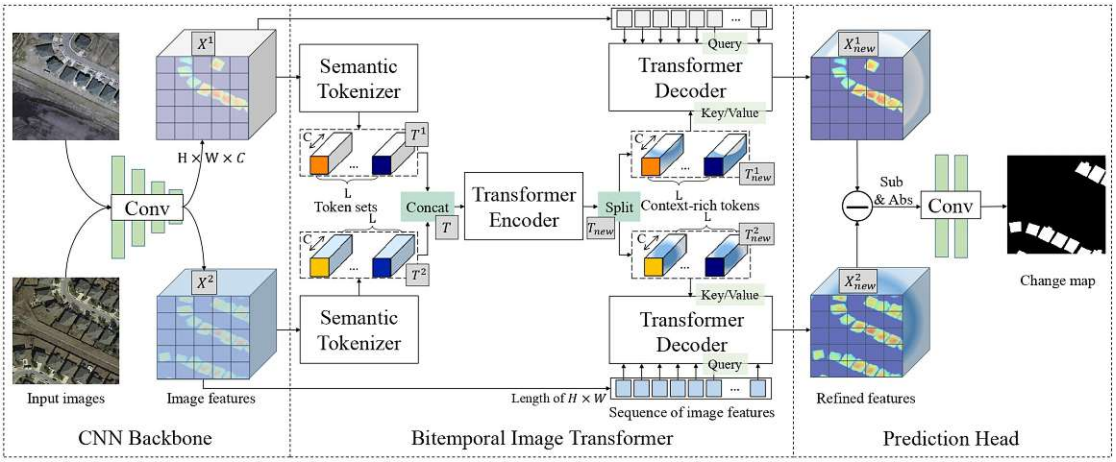


Figure 2.5: Illustration of the BIT-based model for change detection. (Taken from [CQS21])

2.2.3 ChangeFormer (2022)

The third state-of-the-art model that is finetuned and evaluated is ChangeFormer from Gedara et al. [GCBP22]. As can be seen in Figure 2.6, to extract features of the two input images ChangeFormer has a Siamese hierarchical transformer encoder. The hierarchical transformer encoder consists of four transformer blocks. After each block, a difference-module concatenates the feature pair from the two input images and feeds the concatenated feature map into a convolutional layer. During training, this convolutional layer learns the optimal distance metric at each scale. This is different from the BIT model,

which calculates the absolute difference from the extracted features [CQS21]. These four difference feature maps are then fed into a lightweight Multi-Layer Perceptron (MLP) decoder, where the feature maps are upsampled to the same dimensions, concatenated, and the change map is calculated [GCBP22].

ChangeFormer was chosen for this thesis as it uses a hierarchical transformer encoder and a difference-module, that learns the optimal distance metric. By comparing it to different transformer architectures an evaluation of which transformer architecture is suitable for graffiti change detection is possible.

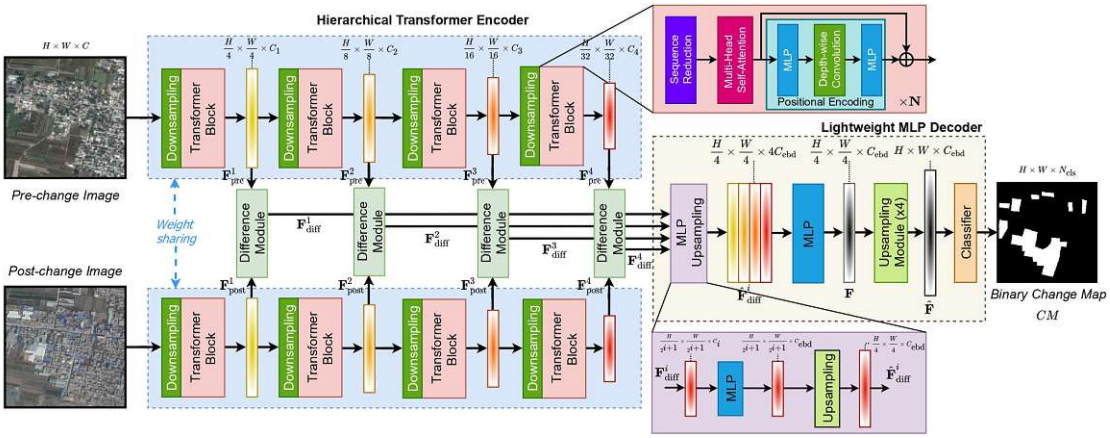


Figure 2.6: ChangeFormer network for CD. (Taken from [GCBP22])

2.3 Summary

This chapter discusses the aged and modern methods and techniques of change detection. The chapter highlights that traditional change detection methods are becoming less popular due to their limitations. In contrast, approaches using deep learning have shown promising results. The rationale behind the selection of specific state-of-the-art models is given. ChangeNet-v2 is largely based on CNNs, while ChangeFormer and BIT are transformer architectures using different encoder-decoder structures. By using a mix of approaches, a comparison of different approaches in the domain of graffiti images is possible.

CHAPTER 3

Implementation

To conduct the experiments, it is necessary to implement the generation of the synthetic dataset to train the models. Further, the Simple U-Net, a simpler neural network that functions as a baseline is implemented as well as the scripts for finetuning the models. These implementations are explained in detail in this chapter.

3.1 Synthetic Dataset Generation

As change detection emerged from remote sensing [Sin89], which mainly focuses on the goals of resources monitoring, urban planning, or disaster assessment, most of the available datasets focus on aerial and satellite images [JPZ+22]. Jiang et al. [JPZ+22] present in a survey from 2022 a summary of popular high-resolution remote sensing datasets for change detection tasks, nine datasets for aerial and six datasets for satellite images [JPZ+22].

In the course of project INDIGO Wild et al. [WVW+23, WVMP23] developed a graffiti-centred change detection dataset. The images were collected at an approximately 50-meter-long strip along Vienna's Donaukanal on eleven days between October and December 2022. It takes 17 images to cover the 50 meters of the Donaukanal. Using different cameras and different settings resulted in 29 distinct image sets of the scene. By using all combinations of the 29 distinct image sets, this amounts to 406 unique image pairs per location. So for 17 locations, this amounts to 6,902 unique image pairs with associated change maps [WVMP23].

As this dataset originates from only 17 different locations, the data is highly correlated and the 6,902 unique image pairs are not sufficient to train or finetune the chosen state-of-the-art models. It is very time-consuming to label change maps by hand, hence labeling more image pairs is also not a viable option. Therefore a synthetic dataset was created by adding digital graffiti to an image of a graffiti wall. Using this method the

change map can be easily obtained during the creation of the graffiti image. Figure 3.1 shows examples of the synthetic dataset.

Figure 3.1: Example of the synthetic dataset, the before-change images are on the left, the after-change images in the middle and the change maps are on the right.

It is not guaranteed that the automatic synthetic image generation algorithm places the synthetic graffiti in a plausible location in the background image. This should not be a problem though, as most of the background images predominantly consist of a graffiti wall. In some cases, it can happen though that the location of the placed graffiti is implausible or impossible. The algorithm then learns to detect graffiti in scenes that will not occur in the real world and may have a harder time detecting real graffiti or be distracted by other changes more easily. Figure 3.2 shows examples of impossible or implausible graffiti placements, in image A half of the graffiti is placed on the floor, in image B the graffiti stretches over the fence and also the grass, in image C the graffiti

stretches into the river and image D the graffiti is placed above the wall in front of the substructure of a bridge. Due to the large number of images generated, unfortunately, this cannot be avoided but it is expected that it will not have a big impact on the final results.

Figure 3.2: Examples of impossible or implausible graffiti placements of the synthetic image generation.

3.1.1 Source of Images

The graffiti images are scraped from different websites offering free images with a transparent background, these websites include FAVPNG[1], KindPNG[2], NicePNG[3], pngfind[4], StickPNG[5] and a graffiti generator from Graffiti Empire[6]. Using these websites 787 graffiti, street art, or similar images are gathered.

For the background images, 1,267 different backgrounds are scraped from Spraycity[7], a website that collects and publishes images of graffiti walls from Vienna's Donaukanal.

3.1.2 Augmentations

To enhance the variety of the dataset some augmentations are applied. Random Gaussian noise is added, the global brightness of the images is changed randomly to simulate changes in the weather, and the image is moved randomly (up to 10 pixels) and rotated randomly (up to 2 degrees) to simulate changes in the perspective. The augmentations do not drastically change the image visually, but now the pixels are no longer co-registered, i.e. in the same location. Figure 3.3 shows an example of the augmentations - on the left is the original image and on the right is the augmented image.

Figure 3.3: Example of the augmentations, the left image shows the original image and the right image shows the augmented image.

[1]Source: FAVPNG https://favpng.com/
[2]Source: KindPNG https://www.kindpng.com/
[3]Source: NicePNG https://www.nicepng.com/
[4]Source: pngfind https://www.pngfind.com/
[5]Source: StickPNG https://www.stickpng.com/
[6]Source: Graffiti Empire https://www.graffiti-empire.com/
[7]Source: Spraycity https://spraycity.at/

3.1.3 Hybrid Dataset

Additionally to the pure synthetic dataset, a hybrid dataset is developed as well. Here a real image pair recorded at two different times from a graffiti wall recorded by project INDIGO [WVW+23] is used as a background. The intuition of this dataset is that it is closer to a real image pair and the trained models can subsequently better recognize which changes shall not be detected, such as changes in vegetation, watermarks on the wall, and illumination. Figure 3.4 shows examples of the hybrid dataset.

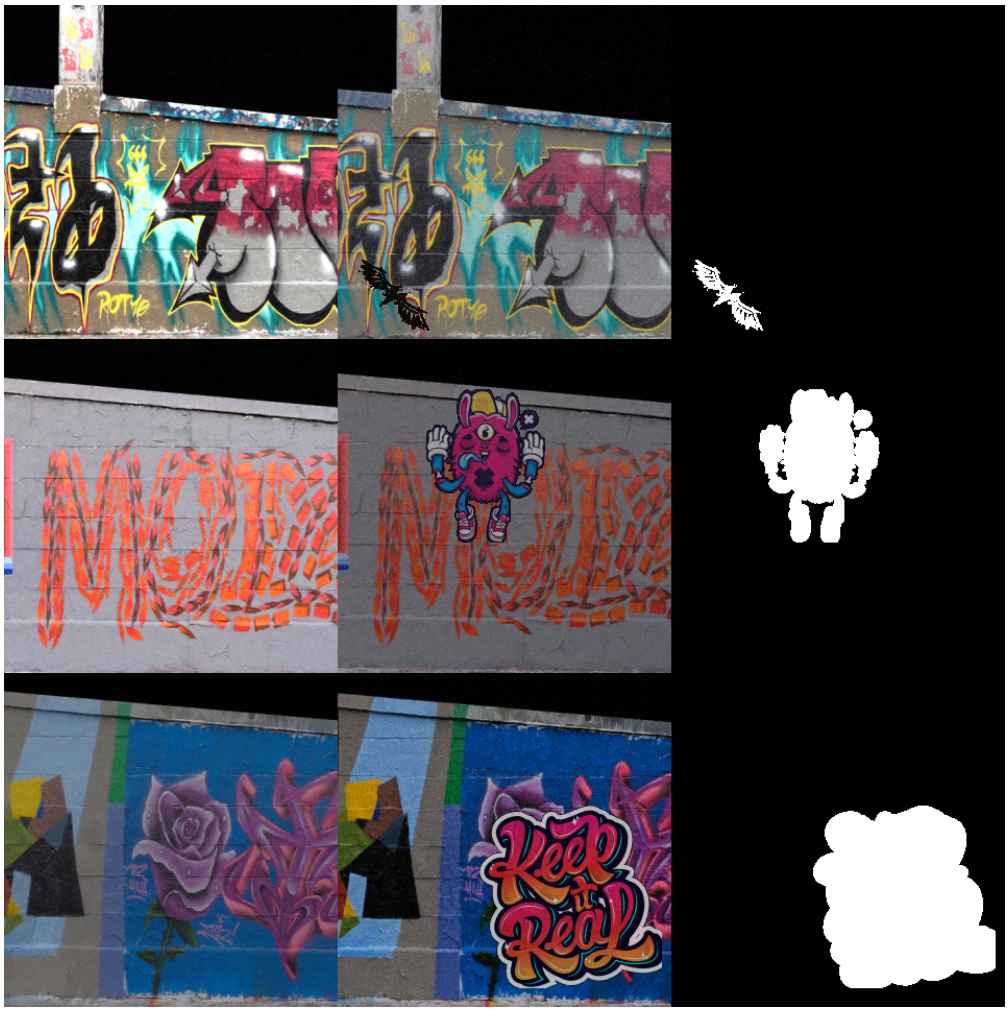


Figure 3.4: Example of the hybrid dataset, the before-change images are on the left, the after-change images in the middle and the change maps are on the right.

3.1.4 Train/Validation/Test Split

The scraped graffiti and backgrounds are used across the datasets and multiple times within the single sets, and the locations where the images for the hybrid dataset are similar to the locations of the dataset labeled by Wild et al. [WVMP23]. Therefore the train/validation/test split cannot be performed carelessly. Otherwise, one graffiti or location could occur in the train set of one dataset while being part of the test set in a different dataset. In order to take care of this, the graffiti and locations are split before generating the synthetic data, in this way no graffiti used in one of the training sets occurs in a validation set or a test set or vice versa.

In the end, 6,000 synthetic image pairs were generated. 5,000 were purely synthetic, with a split of 4,700 for training, 200 for validation, and 100 for testing. Of the 1,000 hybrid synthetic image pairs, 700 are used for training, 200 for validation, and 100 for testing.

3.2 Simple U-Net for Change Detection

As discussed in Section 1.1, change detection in graffiti images provides a different task than change detection on remote sensing images, and all change detection models are only evaluated on satellite or aerial images [SZZ+20, JPZ+22]. It is unknown how difficult change detection in graffiti images is. For this reason, this thesis also evaluates a simpler model to be able to compare the results of the state-of-the-art models with a simpler model to quantify the benefits of using a more complex model. The simpler model follows the U-Net architecture from Ronneberger et al. [RFB15]. As can be seen in Figure 3.5, the two three-channel input images are concatenated into one six-channel image. The first part of the network encodes the bi-temporal input image in four stages of convolutions and pooling, the second part decodes the hidden state using residuals from the encoding stages.

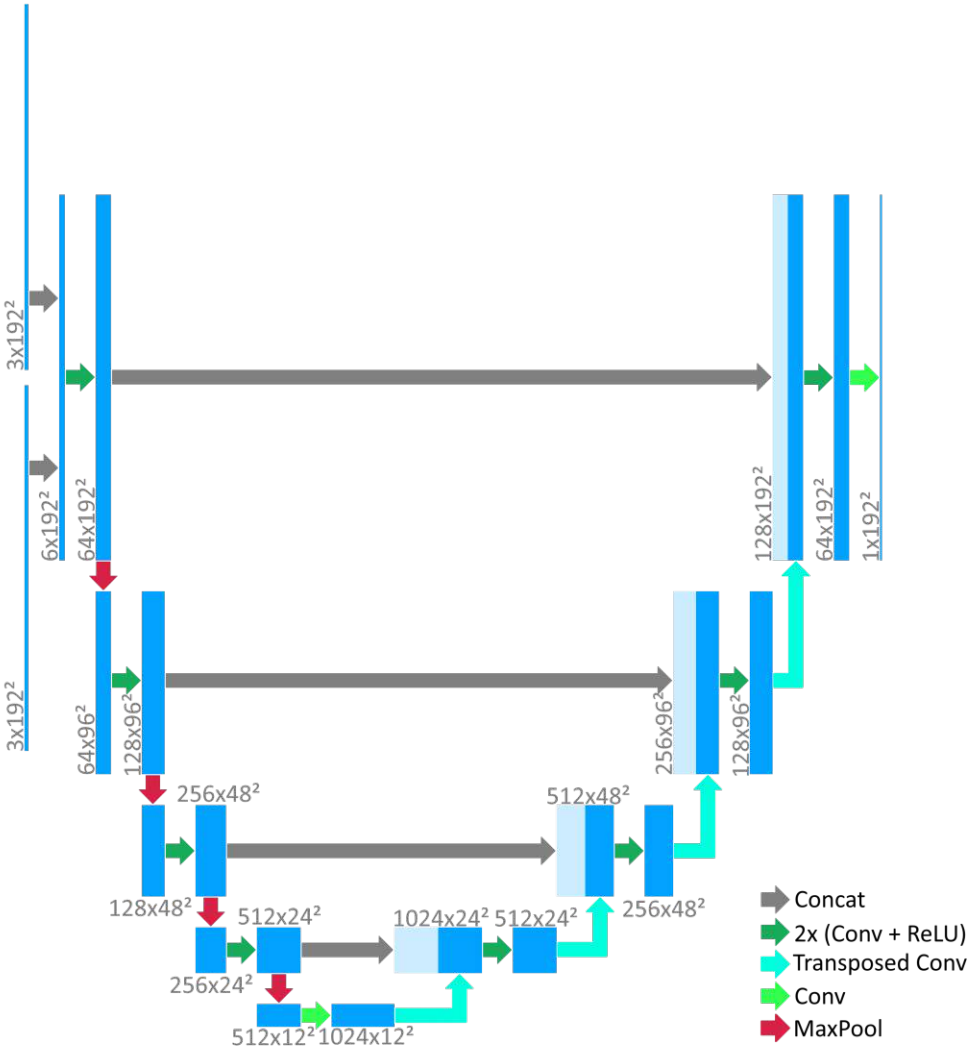


Figure 3.5: Architecture of the Simple U-Net for Change Detection, following the U-Net architecture from Ronneberger et al. [RFB15]. The grey arrows indicate a copying and concatenation of the layer. The green arrows indicate a a block of two convolutional layers, each with a ReLU activation. The cyan arrows indicate a transposed convolution for upsampling. The light green arrows indicate a single convolutional layer. The red arrows indicate a max pool operation used to reduce the spatial dimensions.

3.3 Finetuning the of State-Of-The-Art Models

Finetuning is a technique used to improve the performance of pre-trained models on a specific task [TSG+16]. In this case, change detection models trained on satellite or aerial data are finetuned using graffiti data. Further, this chapter discusses how the finetuning of the models is implemented in Python using the PyTorch library.

Preparing the Data

The first step for finetuning the models is to prepare the data. Three different datasets are available for training, the synthetic dataset, the hybrid dataset and the hand-labeled "real" dataset from Wild et al. [WVMP23]. For the purpose of evaluating the benefit the models can extract from each dataset, the models are trained on different combinations of sets: every model is trained on (1) all three data sets, (2) the synthetic dataset and the hybrid dataset, (3) only on the synthetic dataset, (4) only on the real dataset. For training 4,700 synthetic, 700 hybrid, and 2,527 real image pairs were used; for the validation, 200 synthetic, 200 hybrid, and 1,805 real image pairs were used; and for testing 100 synthetic, 100 hybrid, and 1,805 real image pairs were used. The resolution of the input images for all models is 192 by 192.

Loading the Pre-trained Model

In the next step, the pre-trained model is loaded. The state-of-the-art models used, ChangeNet-v2 from Prabhakar et al. [PRB+19], BIT from Chen et al. [CQS21] and ChangeFormer from Gedara et al. [GCBP22], provide a pre-trained model and the code for the PyTorch model on their respective online repositories[8,9,10]. The models are loaded and initialized with the trained parameters. In addition to using the pre-trained parameters, the models are also trained from-scratch, which means no pre-trained parameters are loaded but the training starts with randomly initialized parameters. This makes it possible to evaluate the benefit of finetuning as well.

Training the Model

The final step is to train the models on the finetuning dataset. The models are trained on an NVIDIA GeForce RTX 3050 Ti GPU. Given the limited GPU resources, the models are trained on batches of size 1. All models use the Adam optimizer with a different learning rate for each model, the Simple U-Net is trained with a learning rate of 3e-3, BIT and ChangeFormer with 3e-5, and ChangeNet-v2 with 1e-6. ChangeNet-v2, BIT, and Simple U-Net use cross-entropy as a loss function, and ChangeFormer uses a variation of the Intersection-over-Union loss (IoU) function, the *mmIoULoss*. The *mmIoULoss* calculates the IoU between the predicted and target masks and returns the negative of

[8]Source: ChangeNet-v2 https://github.com/suvaansh/ChangeNet-v2
[9]Source: BIT https://github.com/justchenhao/BIT_CD
[10]Source: ChangeFormer https://github.com/wgcban/ChangeFormer

the minimum IoU and the mean IoU. The models are trained for 30 epochs, to prevent overfitting after each epoch the loss on the validation set is calculated for the current model, and the model with the lowest loss on the validation set during training is selected for evaluation.

3.4 Summary

This chapter provides details of the implementation of the conducted empirical evaluation. It provides the reasons for the necessity of synthetic graffiti generation and the details of how it was generated. The Simple U-Net is introduced as a baseline model for graffiti change detection and its architecture is shown in detail. Lastly, details on the finetuning of the models are given, including the necessary hyperparameters to reproduce the results. With this chapter, reproducibility should be ensured across the whole evaluation.

CHAPTER 4

Evaluation

This chapter presents the evaluation of the state-of-the-art change detection models on graffiti images. The chapter explains the metrics used to measure the performance of the models and the results of the finetuning and training experiments. The chapter provides quantitative evidence to answer the proposed research questions, as well as qualitative results to show visually how the models perform.

4.1 Metrics

The metrics are chosen as they are the most common metrics used in the literature [GCBP22, JPZ+22, MV22]. The following equations use true-positive (TP) as the number of pixels that are changed in the ground-truth and a change was detected, true-negative (TN) as the number of pixels that remain unchanged and are also detected as such, false-negative (FN) is the number of pixels that are changed in the ground-truth but were not detected, and false-positive (FP) is the number of pixels that did not change in the ground truth but were wrongly detected as changed [JPZ+22].

Precision measures the fraction of pixels that are correctly classified as 'changed' among all detected pixels:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

Recall measures the fraction of pixels that were detected as changed to all the changed pixels:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

The ***F1-Score*** is the harmonic mean between precision and recall:

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

***Overall Accuracy* (OA)** is the fraction of the correctly predicted pixels to all pixels:

$$\text{OA} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.4}$$

The ***Intersection over Union (IoU)*** or Jaccard index [Jac12], measures how well the predicted changes and the ground-truth changes overlap. It is the fraction of the correctly predicted as changed pixels (TP) to the number of pixels that are either predicted changed (TP and FP) or should be predicted as changed (TP and FN):

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{4.5}$$

4.2 Results

The models are evaluated on the test set of three different datasets, the synthetic dataset created in the course of this thesis, labeled as "Synth", the hybrid dataset created in the course of this thesis, labeled as "Hybrid" and the hand-labeled "real" dataset created by Wild et al. [WVMP23] labeled as "Real". The values are rounded to three decimals and the best scores are highlighted in bold. The true-positives, true-negatives, false-positives, and false-negatives are counted for the entire test set and then the metrics are calculated.

4.2.1 Models Individual Results

The following tables show the results of the four models. The column "Training Data" shows the data the model is finetuned on, this can be "-" for the original model, "Synth" if the model is finetuned only on the synthetic dataset, "Synth+Hybrid" if the model is finetuned on the synthetic dataset and the hybrid dataset, "Real" if the models are finetuned only on real data, or "Synth+Hybrid+Real" if the real dataset created by Wild et al. [WVMP23] is used as well for finetuning. Table 4.1 shows the results of the ChangeNet-v2, Table 4.2 shows the results of the BIT, Table 4.3 shows the results of the ChangeFormer and Table 4.4 shows the results of the Simple U-Net.

Test Data	Training Data	OA	Precision	Recall	F1-Score	IoU
Synth	-	0.928	0.942	0.287	0.440	0.282
	Synth	**0.985**	0.989	**0.857**	**0.918**	**0.849**
	Synth+Hybrid	**0.985**	**0.991**	0.855	**0.918**	0.848
	Real	0.927	0.956	0.271	0.422	0.267
	Synth+Hybrid+Real	0.979	0.945	0.837	0.888	0.798
Hybrid	-	0.926	0.934	0.298	0.452	0.292
	Synth	**0.982**	0.980	0.836	0.902	0.822
	Synth+Hybrid	**0.982**	**0.983**	**0.842**	**0.907**	**0.829**
	Real	0935	0.931	0.391	0.551	0.380
	Synth+Hybrid+Real	0.975	0.919	0.829	0.871	0.772
Real	-	0.944	0.953	0.115	0.206	0.115
	Synth	0.964	0.992	0.434	0.604	0.433
	Synth+Hybrid	0.964	**0.993**	0.435	0.605	0.433
	Real	0.960	0.935	0.395	0.555	0.384
	Synth+Hybrid+Real	**0.966**	0.964	**0.481**	**0.642**	**0.473**

Table 4.1: Results of ChangeNet-v2 trained on different datasets.

Test Data	Training Data	OA	Precision	Recall	F1-Score	IoU
Synth	-	0.908	**0.981**	0.071	0.132	0.071
	Synth	**0.980**	0.967	0.827	**0.892**	**0.805**
	Synth+Hybrid	0.979	0.976	0.810	0.885	0.794
	Real	0.895	0.472	0.544	0.505	0.338
	Synth+Hybrid+Real	0.979	0.930	**0.846**	0.886	0.796
Hybrid	-	0.903	0.852	0.059	0.110	0.058
	Synth	**0.968**	0.941	0.733	**0.825**	**0.701**
	Synth+Hybrid	0.967	**0.959**	0.706	0.813	0.685
	Real	0.866	0.376	0.475	0.420	0.266
	Synth+Hybrid+Real	0.965	0.857	**0.792**	0.823	0.700
Real	-	0.936	0.437	0.018	0.034	0.017
	Synth	0.950	0.833	0.269	0.407	0.255
	Synth+Hybrid	0.951	**0.857**	0.266	0.406	0.255
	Real	0.909	0.325	**0.406**	0.361	0.221
	Synth+Hybrid+Real	**0.953**	0.749	0.383	**0.507**	**0.340**

Table 4.2: Results of BIT trained on different datasets.

Test Data	Training Data	OA	Precision	Recall	F1-Score	IoU
Synth	-	0.875	0.345	0.295	0.318	0.189
	Synth	**0.986**	0.988	**0.872**	**0.927**	**0.863**
	Synth+Hybrid	**0.986**	**0.991**	0.867	0.925	0.860
	Real	0.931	0.679	0.564	0.616	0.446
	Synth+Hybrid+Real	**0.986**	0.989	0.870	0.926	0.862
Hybrid	-	0.827	0.259	0.373	0.305	0.180
	Synth	0.974	0.896	0.845	0.870	0.770
	Synth+Hybrid	**0.982**	**0.967**	0.853	0.906	0.829
	Real	0.885	0.455	0.653	0.537	0.367
	Synth+Hybrid+Real	**0.982**	0.941	**0.879**	**0.909**	**0.833**
Real	-	0.837	0.121	0.250	0.163	0.089
	Synth	0.958	0.750	0.498	0.599	0.427
	Synth+Hybrid	0.964	**0.936**	0.466	0.622	0.451
	Real	0.922	0.396	0.426	0.411	0.258
	Synth+Hybrid+Real	**0.968**	0.922	**0.546**	**0.686**	**0.522**

Table 4.3: Results of ChangeFormer trained on different datasets.

Test Data	Training Data	OA	Precision	Recall	F1-Score	IoU
Synth	Synth	**0.991**	**0.989**	**0.923**	**0.955**	**0.913**
	Synth+Hybrid	**0.991**	0.986	0.920	0.952	0.908
	Real	0.856	0.354	0.551	0.431	0.274
	Synth+Hybrid+Real	0.988	0.958	0.922	0.939	0.885
Hybrid	Synth	0.980	0.918	0.886	0.902	0.821
	Synth+Hybrid	**0.989**	**0.987**	**0.901**	**0.942**	**0.891**
	Real	0.823	0.305	0.538	0.389	0.242
	Synth+Hybrid+Real	0.987	0.970	0.897	0.932	0.872
Real	Synth	0.952	0.764	0.346	0.476	0.312
	Synth+Hybrid	0.960	**0.961**	0.383	0.548	0.377
	Real	0.962	0.715	**0.671**	**0.692**	**0.529**
	Synth+Hybrid+Real	**0.964**	0.945	0.462	0.620	0.450

Table 4.4: Results of Simple U-Net trained on different datasets.

It can be observed that for ChangeNet-v2 and Simple U-Net, the model that is trained on synthetic data tends to perform better on the synthetic test data, the model trained on synthetic and hybrid data tends to perform better on the hybrid test data and the model trained on all datasets tends to perform better on the test data of the real dataset from Wild et al. [WVMP23]. This effect is less clear for ChangeFormer and BIT, the two transformer architectures. Especially ChangeFormer tends to perform best when trained on all three sets, although the margin can be very slim.

As visualized in Figure 4.1, the original models, only trained for different domains, can never outperform their finetuned counterparts in terms of F1-Score in the real dataset. The original models of ChangeNet-v2 and BIT yield a comparatively good precision, but the is recall much lower than the finetuned models. For ChangeFormer and BIT also the precision performs far worse than at the finetuned models.

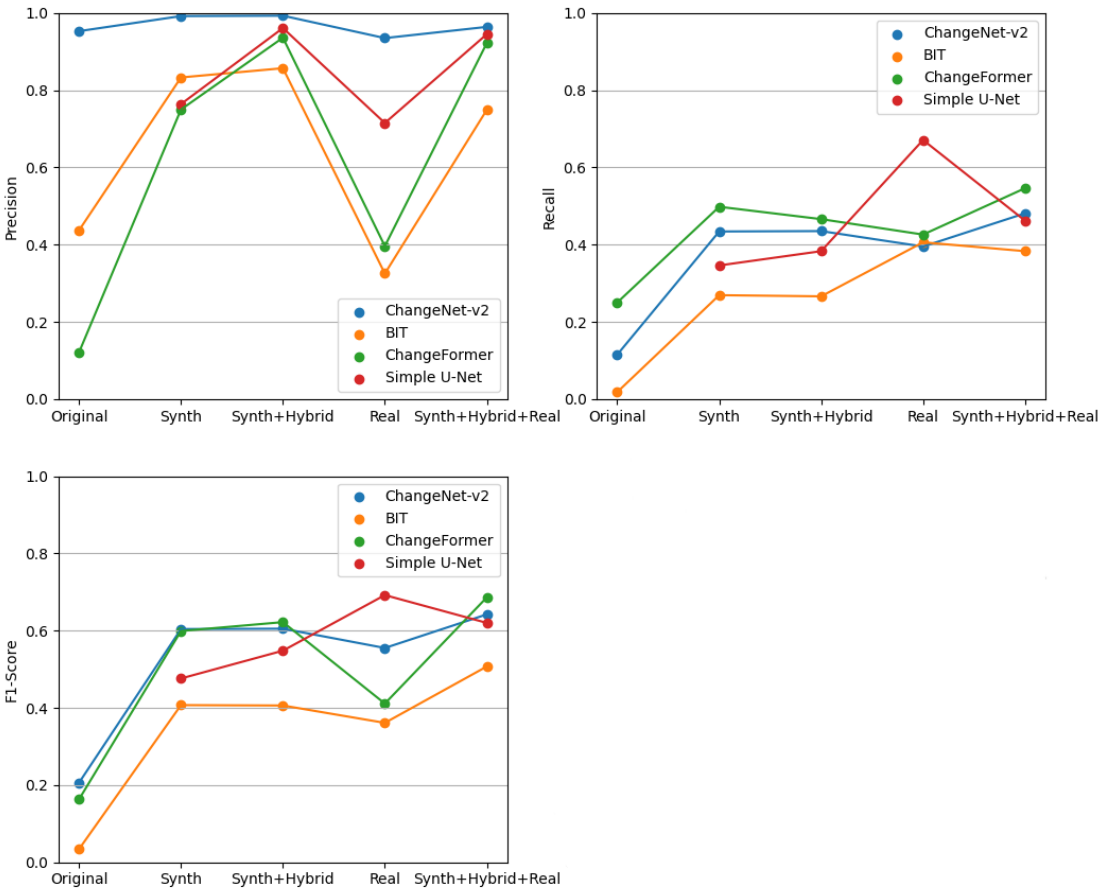


Figure 4.1: Comparison of precision, recall, and F1-Score of models finetuned on the different dataset configurations and evaluated on the real dataset.

Using synthetic data increases the precision for all models. Comparing the models trained

on real data and the models trained on synthetic and hybrid synthetic data, the models on average could increase the precision by 0.344 evaluated on the real data. This effect is strongest for the transformer models, BIT and ChangeFormer, which increase their precision by 0.532 and 0.540. In contrast, the recall decreases slightly, by 0.087. Most effected here is the Simple U-Net, where the recall decreases by 0.288, when trained on synthetic and hybrid data instead of only real data.

All models except for the Simple U-Net can achieve their best F1-score when trained with all available data, synthetic, hybrid and real. The Simple U-Net achieves the highest F1-Score when trained only with real data and thereby achieves the highest F1-score among all other models as well. The main reason for the good F1-Score of the Simple U-Net is comparatively the good recall. The model reaches a worse precision than all other models trained on the full training data, however, the model can achieve the best recall among all models, as can be observed in Figure 4.2.

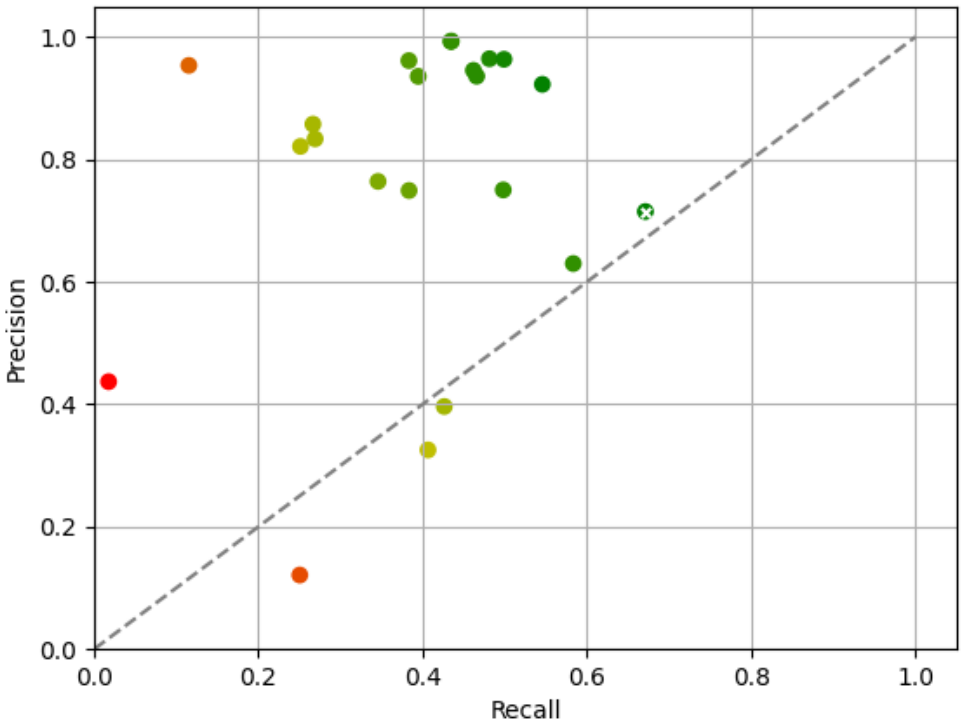


Figure 4.2: Comparison of all models' precision and recall evaluated on the real test set. The color of the markers indicates the F1-Score, from red for the lowest F1-Score to green for the highest. The Simple U-Net, the model with the highest F1-Score is marked with a white cross.

To investigate the behavior of the precision and the recall further, Figure 4.3 shows the precision-recall curve for the two models yielding the highest F1-Scores. The Simple U-Net, trained on real data only, and the ChangeFormer, trained on all data. Figure 4.3 shows the precision-recall curve of the two models. It can be observed that the ChangeFormer tends to have a higher precision, and the Simple U-Net performs better in terms of recall while being slightly more balanced.

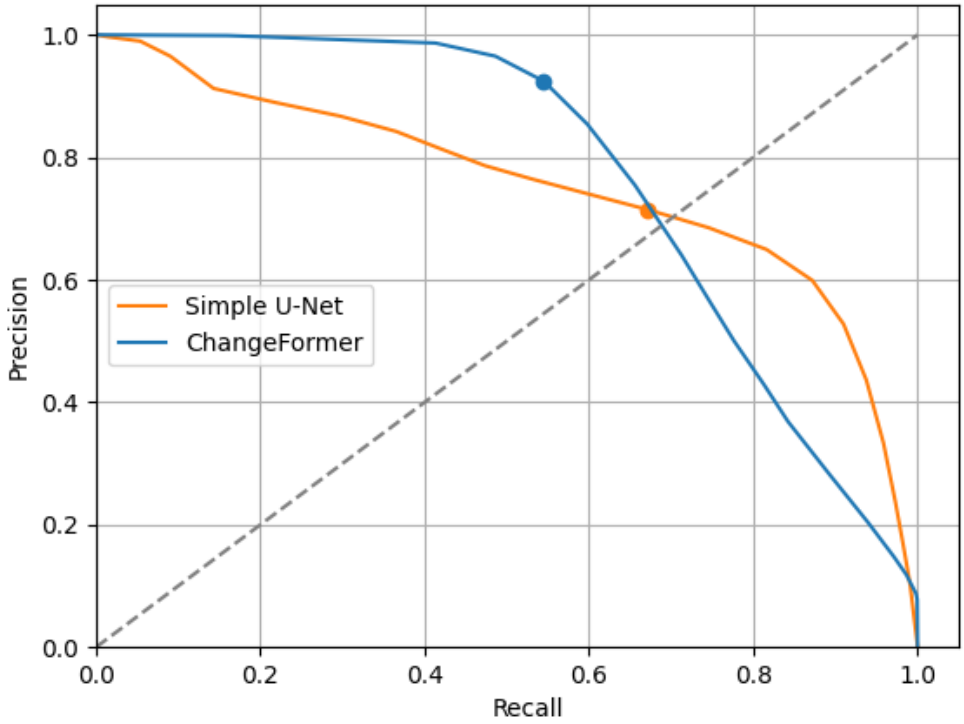


Figure 4.3: Precision-recall curve of the Simple U-Net, trained on real data only and the ChangeFormer trained on all data. The results of the models without changing the threshold are marked with a dot.

4.2.2 Benefit of Model Finetuning

The following Table 4.5 shows the effect of finetuning over training the models from-scratch, i.e. the benefit of using parameters trained on a different domain, over using randomly initialized parameters of models. The models are evaluated on the real test set from Wild et al. [WVMP23]. The finetuned models are finetuned on all three datasets (synthetic, hybrid, and real). It can be observed that finetuning tends to outperform training the model from scratch, although the improvement in this case often is very

small. For some metrics, the model trained from scratch outperforms the finetuned one. Figure 4.4 shows a comparison of the finetuned models versus the models trained from-scratch visually.

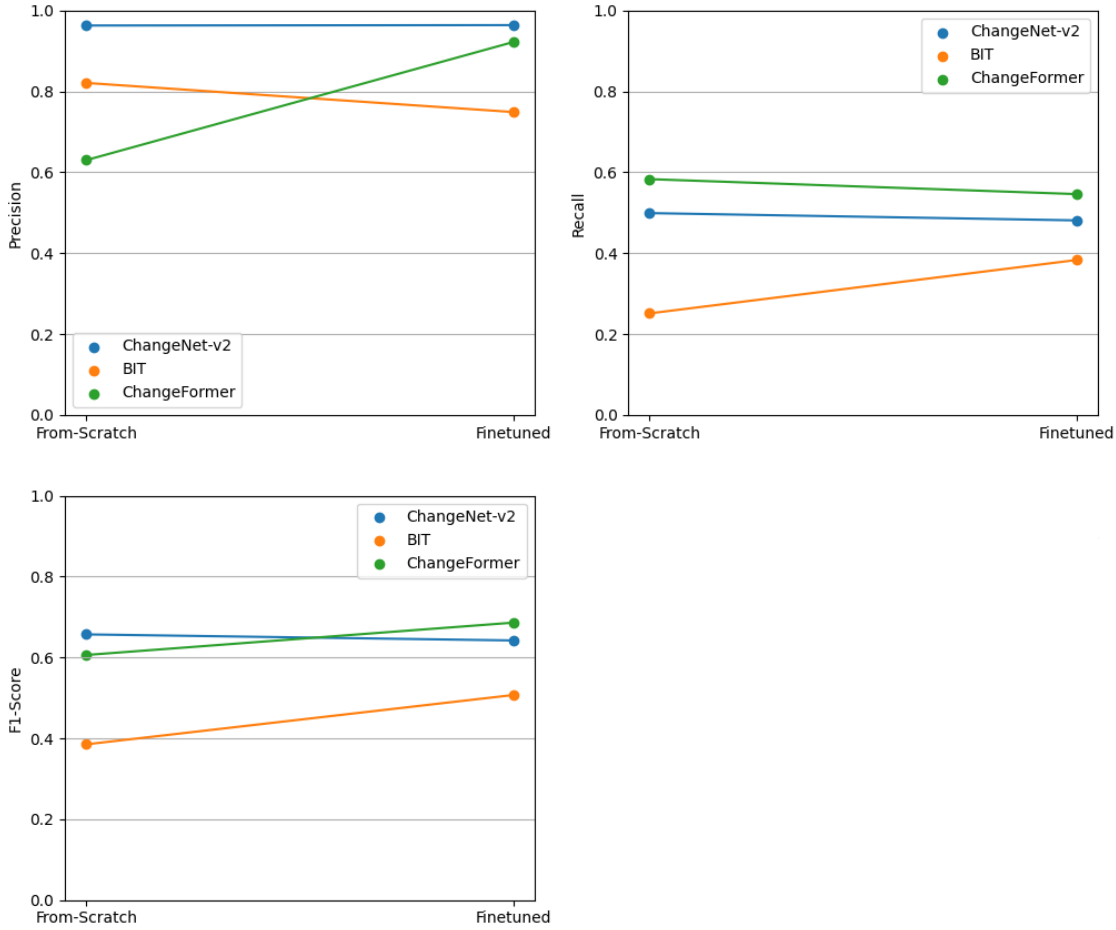


Figure 4.4: Comparison of precision, recall, and F1-Score of finetuning models over training from-scratch.

Model	Type	OA	Precision	Recall	F1-Score	IoU
ChangeNet-v2	From-Scratch	**0.967**	0.963	**0.499**	**0.657**	**0.490**
	Finetuned	0.966	**0.964**	0.481	0.642	0.473
BIT	From-Scratch	0.950	**0.821**	0.251	0.385	0.238
	Finetuned	**0.953**	0.749	**0.383**	**0.507**	**0.340**
ChangeFormer	From-Scratch	0.951	0.630	**0.583**	0.606	0.434
	Finetuned	**0.968**	**0.922**	0.546	**0.686**	**0.522**

Table 4.5: Results of finetuned models compared to models trained from-scratch.

4.2.3 Qualitative Results

This subsection provides qualitative results. Figure 4.5 shows the change maps of the ChangeNet-v2 (CN), BIT, ChangeFormer (CF) and the Simple U-Net (SUN) with the different configurations, "Original" for the original model that has not been finetuned or trained with graffiti data, "Synth" for the models only finetuned on synthetic data, "Synth+Hybrid" for the models finetuned on synthetic and hybrid data and "Synth+Hybrid+Real" for the models finetuned on real data as well. Sample "(a)" is from the synthetic dataset, "(b)" is from the hybrid dataset, and samples "(c)", "(d)" "(e)" and (f) are from the real dataset.

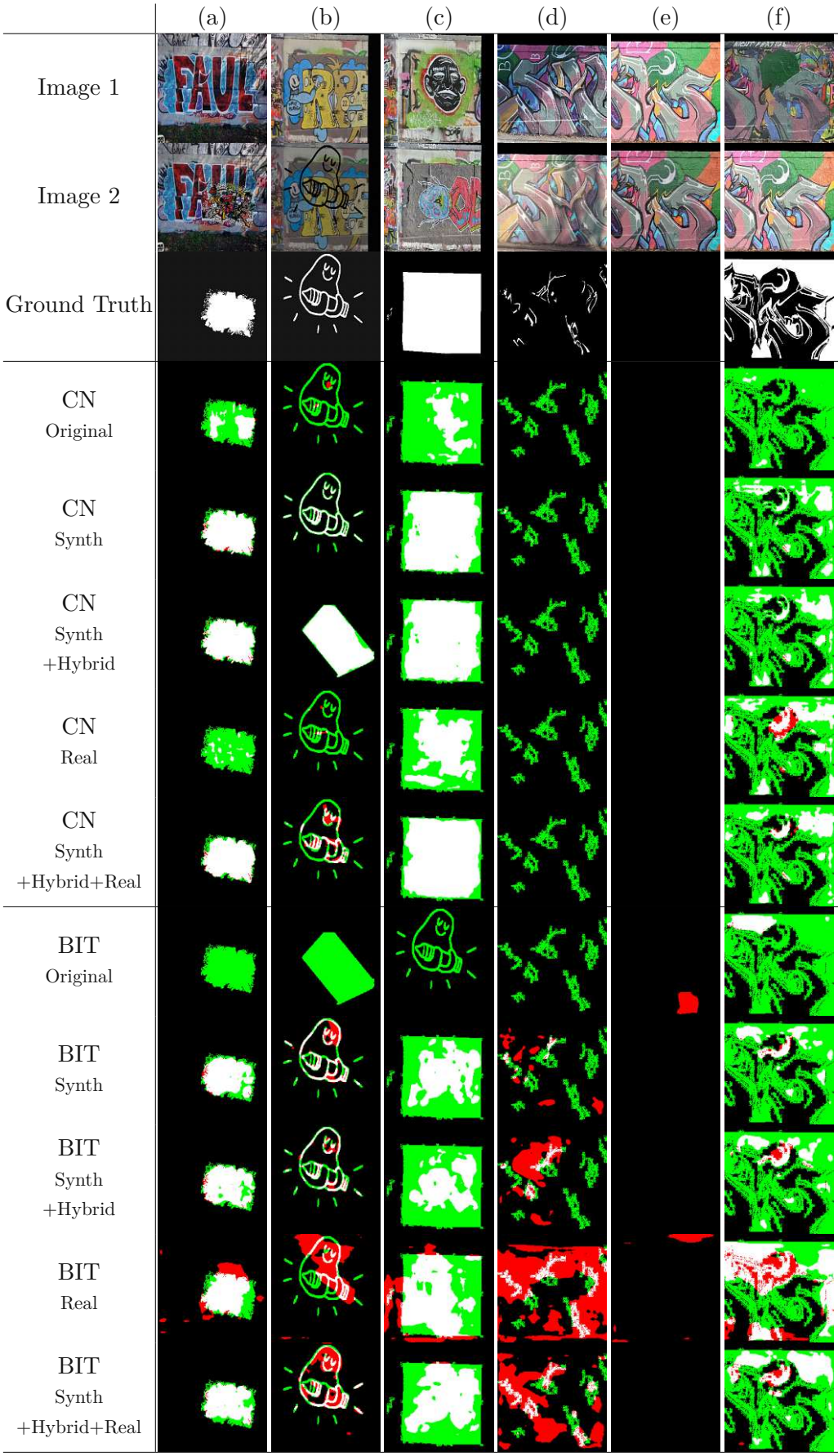
The qualitative results allow a visual analysis of the models' performance. It can be seen that the original models fail to detect useful changes. While ChangeFormer and ChangeNet-v2 detect some changes, BIT barely seems to detect any changes. This matches with the results from Table 4.2 where the original model of BIT only reaches a recall of 0.018.

It can be observed that ChangeNet-v2 and BIT seem to increase the amount of FN predictions (visualized in red) when the model is trained on the real data in addition to the synthetic and hybrid data for image pair (b). Image pair (b) features a line art creature where only the lines should be detected as changed as the body of the creature is transparent and should therefore not be detected as changed. As the real dataset does not include such artwork, the model focuses less on detecting such art forms when less training data includes it. ChangeFormer and the Simple U-Net seem not to have a problem with this type of graffiti.

The changes in image pair (d) only include changes made to an existing graffiti, which are lines and contours that are added. All models seem to have a problem with image pair (d): most models do not detect any changes, and some, for example, BIT trained on synthetic, hybrid, and real data detect larger parts of the graffiti as changed.

Image pair (e) does not include any changes, most of the models seem to detect this correctly, except the original models of BIT and ChangeFormer, both detecting false-positive artifacts.

As shown in Table 4.4 and Figure 4.1, the Simple U-Net trained only on the real data is able to achieve the best F1-Score and recall among all models, while having a comparatively low precision. To investigate this further, Table 4.6 shows additional examples of the two models with the two best F1-Scores, ChangeFormer trained on all available data and the Simple U-Net trained only on the real data. Similar to what the qualitative results suggest, the visual comparison shows that the Simple U-Net classifies much larger regions as changes compared to the other model. In image pairs (c), (d), (e) and (h) it can be observed that much larger regions are classified as changes than the actual graffiti that was added. This causes a decrease in the precision but an increase in the recall.



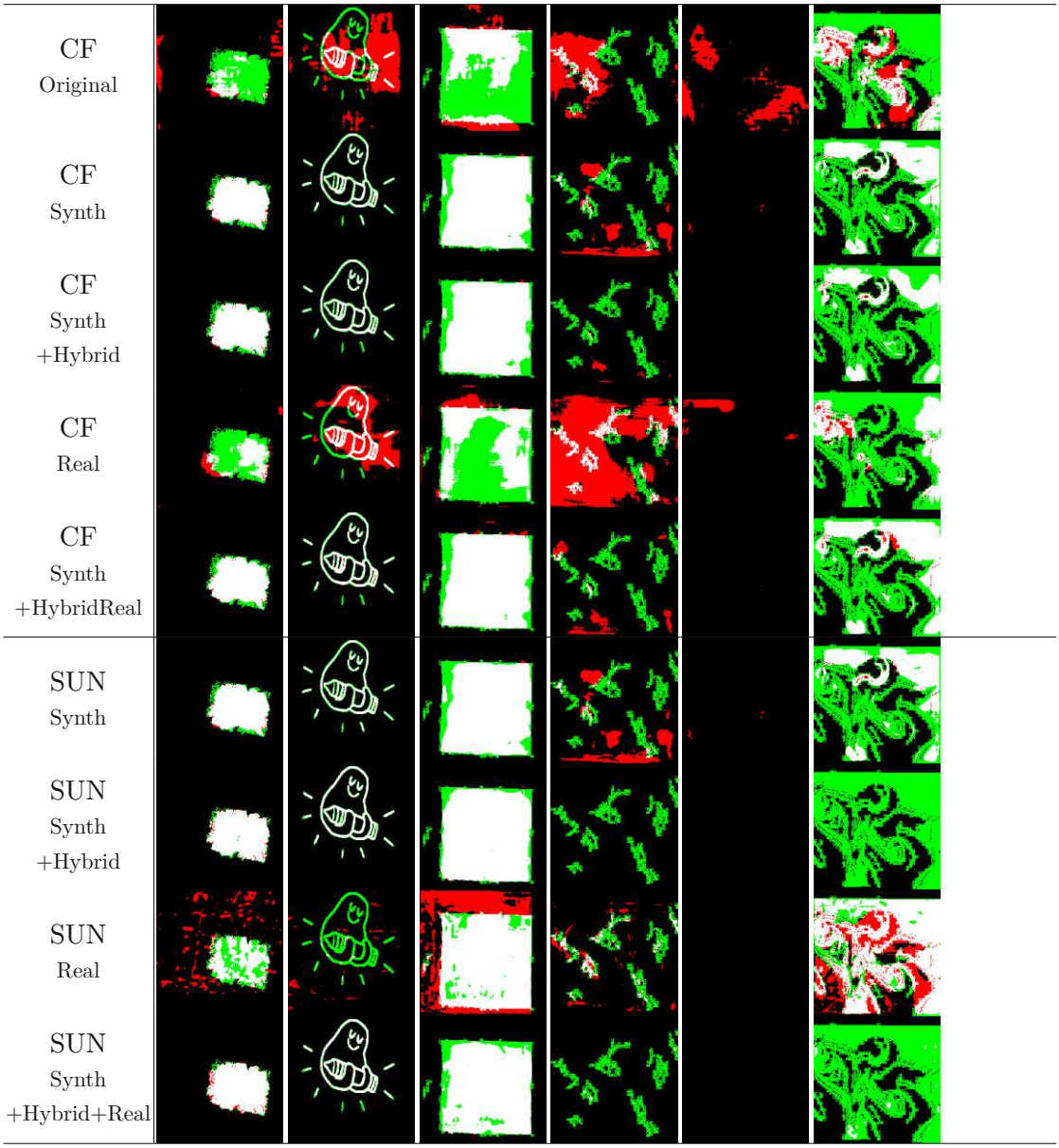


Figure 4.5: Qualitative results of the original and finetuned models of the ChangeNet-v2 (CN), BIT, ChangeFormer (CF) and the Simple U-Net (SUN) with the different configurations, "Original" for the original model, "Synth" for the models only finetuned on synthetic data, "Synth+Hybrid" for the models finetuned on synthetic and hybrid data and "Synth+Hybrid+Real" for the models finetuned on real data as well. Sample "(a)" is from the synthetic dataset, "(b)" is from the hybrid dataset, and samples "(c)", "(d)" "(e)" and (f) are from the real dataset. White pixels indicate true positives, black pixels indicate true negatives, red pixels indicate false positives, and green pixels indicate false negatives.

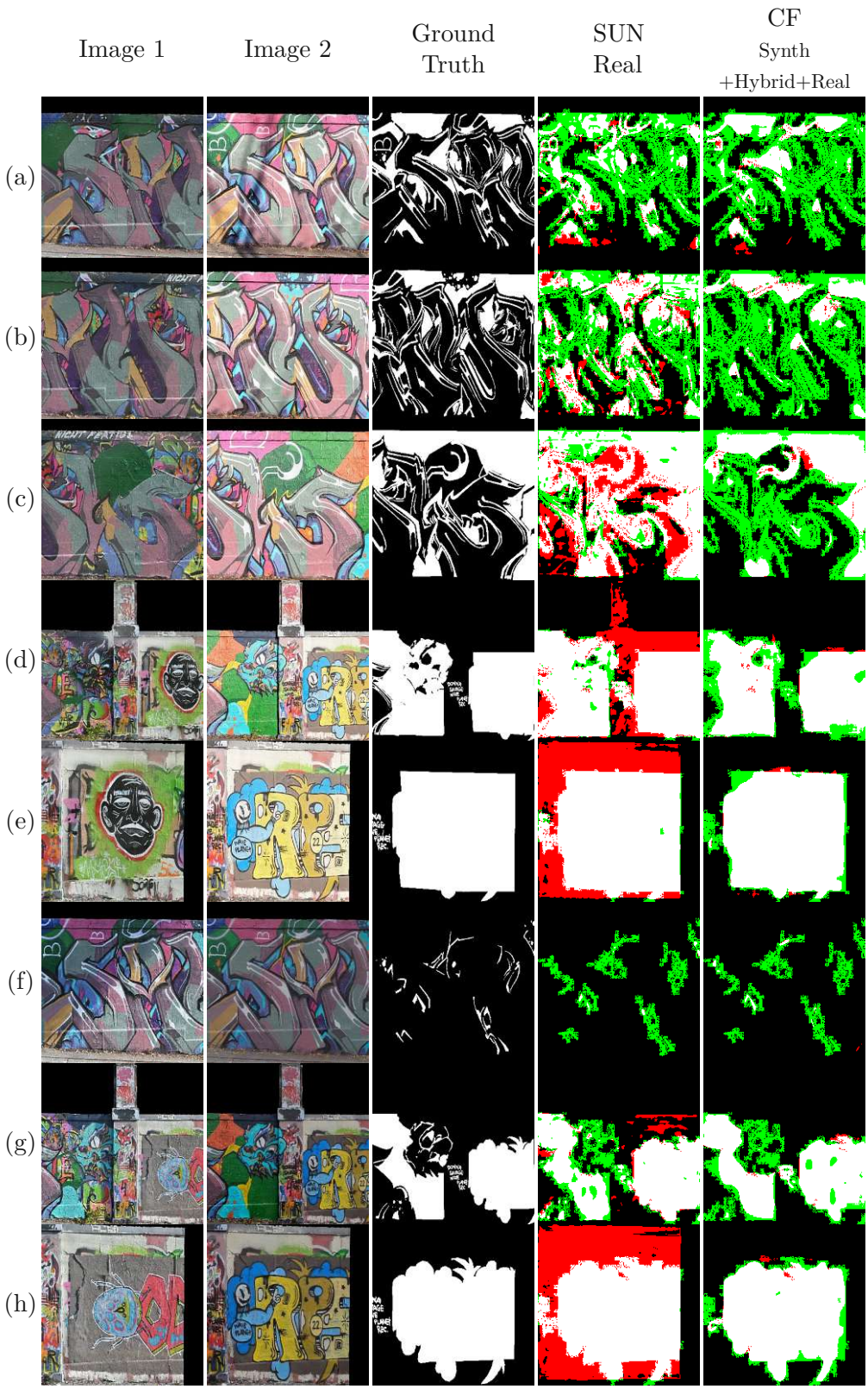


Figure 4.6: Qualitative results of the Simple U-Net trained only on the real data and ChangeFormer trained on all available data. The input image pairs are from the real dataset.

4.3 Summary

Chapter 4 presents the evaluation metrics and the results of the finetuned models on three different datasets: synthetic, hybrid, and real graffiti images.

The results show that the models perform better when finetuned on graffiti data than using the original pre-trained parameters. The performance generally increases the more data is used to finetune the models except for the Simple U-Net, which achieves the highest F1-Score among all models, while only being trained on real data. To investigate this further the Simple U-Net is compared to the model achieving the second highest F1-Score, the more advanced ChangeFormer. The comparison shows that the Simple U-Net can outperform the more advanced model because of its high recall while having the lowest recall of all models trained on all available data.

Further, the comparison of the finetuned models, which are pre-trained on different domains, versus the models trained from-scratch, with randomly initialized parameters does not show a clear trend with the finetuned models slightly performing better than the models trained from-scratch.

CHAPTER 5

Conclusion

This thesis aims to empirically evaluate the state-of-the-art change detection models and finetuned state-of-the-art change detection models on the domain of graffiti images. This is achieved by generating synthetic data, as the available data is very sparse. Then finetuning state-of-the-art change detection models on the new domain and evaluating the models trained and finetuned with different configurations.

5.1 Research Questions Revisited

Based on the results, presented in Section 4.2, the research questions can be answered.

How do state-of-the-art models perform on graffiti images?
ChangeFormer, which achieves F1-Scores between 0.867 and 0.904 on remote sensing datasets, only achieves an F1-Score of 0.163 on the real graffiti data [GCBP22]. Other models like ChangeNet-v2, which achieves F1-Scores between 0.688 and 0.938 on remote sensing datasets, achieves an F1-Score of 0.206 on the real graffiti data [PRB+19]. BIT, achieving between 0.693 and 0.893 on remote sensing datasets achieves an F1-Score of 0.034 on the real graffiti data [CQS21].

The performance of all models is much worse than their performance on remote sensing tasks. This is not surprising, as the models were only designed and trained to function in a different domain and graffiti images present different challenges. The state-of-the-art change detection models cannot be used as they are for the domain of graffiti images but need to be altered, either by training with dedicated data or if possible, even in their architecture as well.

How much can the state-of-the-art models benefit from finetuning with graffiti images?
All models show a clear increase in performance after finetuning, in terms of Overall Accuracy, F1-Score, and the Intersection over Union. For the precision, this trend is

not so clear as the precision for the original models of ChangeNet-v2 and BIT are very competitive to the finetuned results. The original model of BIT even achieves the highest precision for the real test set. In terms of F1-Score, the trend is very clear, ChangeNet-v2 increases the F1-Score on the real test set from the original 0.206 to 0.604 finetuned with synthetic data, and to 0.605 finetuned with synthetic and hybrid data and to 0.642 finetuned with real data as well. The average increase of the F1-Score of the model evaluated on the same type of data it was trained on is 0.456. BIT increases the F1-Score on the real test set from the original 0.034 to 0.407 finetuned with synthetic data, and to 0.406 finetuned with synthetic and hybrid data, and to 0.507 finetuned with real data as well. The average increase of the F1-Score of the model evaluated on the same type of data it was trained on is 0.645. For ChangeFormer the F1-Score on the real test set from the original 0.163 to 0.599 finetuned with synthetic data, and to 0.622 finetuned with synthetic and hybrid data, and to 0.686 finetuned with real data as well. The average increase of the F1-Score of the model evaluated on the same type of data it was trained on is 0.578.

Using synthetic data for finetuning the models generally seems beneficial for the models, ChangeNet-v2, BIT, and ChangeFormer perform better when only trained on synthetic and hybrid data as opposed to only trained on real data. The best performance can be achieved when all three datasets, synthetic, hybrid, and real, are used. The problem with only using the real dataset is the harsh drop in the precision of the models, while the recall can mostly be preserved or even increased. BIT achieves its highest recall when only trained with real data. This is different for the Simple U-Net, which achieves the highest overall F1-Score of all models when only trained with real data. This may be due to the simpler architecture that does not require the amounts of training data that the state-of-the-art models need.

5.2 Further Findings

Table 4.5 shows the results comparing the finetuned models and the models trained from-scratch. It shows that the finetuned models perform better in most metrics than the models trained from-scratch, but this effect is smaller than expected and for some metrics, the model trained from-scratch performs better than the finetuned one. The benefit of finetuning, which means using pre-trained models seems to be stronger for the precision, and less for the recall. An explanation for this might be the comparatively good performance of the original models on the precision, which can be observed in Figure 4.1.

In order to establish a baseline and evaluate the complexity of the task, a simpler neural network was implemented. The Simple U-Net performs best in terms of F1-Score in all test sets, for the synthetic and hybrid test sets the model trained on all data outperforms other models. For the real test set the Simple U-Net trained only on real data outperforms the other models.

The main reason for the good F1-Score of the Simple U-Net is comparatively the good

recall. Especially in the model trained only on real data, this model reaches a worse precision than all other models trained on the full training data. However, the model can achieve the best recall among all models. As can be observed in Figure 4.2 most models achieve a much better precision than recall.

Figure 4.3 shows a comparison of the precision-recall curves of the Simple U-Net and the next best model, the ChangeFormer. It can be observed that by changing the threshold the two models show a different behavior, where the Simple U-Net tends to have a better recall and the ChangeFormer a better precision. By adjusting the loss function the ChangeFormer might be able to sacrifice some of their precision to increase the recall and subsequently increase the F1-Score. However, which metric shall be optimized depends on the use case of the change detector. The qualitative results show that models performing better in precision might be more useful for the task of documenting and monitoring new graffiti in an automated way. Models with a higher recall often detect wrong changes, which might make it harder for a heuristic to crop the outlines of the new graffiti, whereas missing some pixels in a larger changed graffiti might not have severe effects on a heuristic. For instance, comparing the models using the $F_{0.5}$-Score could be beneficial for this use case. The $F_{0.5}$-Score is calculated as follows:

$$F_\beta\text{-Score} = \frac{(1+\beta^2)\cdot \text{Precision}\cdot \text{Recall}}{\beta^2\cdot \text{Precision} + \text{Recall}} \tag{5.1}$$

where $\beta = 0.5$. Using this metric the Simple U-Net, trained on real data only, achieves an $F_{0.5}$-Score of 0.705, and the ChangeFormer, trained on all data, an $F_{0.5}$-Score of 0.811. Figure 5.1 shows the comparison of the two models, as well as a curve along which the $F_{0.5}$-Score remains constant as 0.811.

Rewarding a model's precision more than its recall could increase the usability of the models for the task of documenting and monitoring new graffiti in an automated way.

Nonetheless, a much simpler model was able to outperform the state-of-the-art models in most metrics. Another reason for this might be due to the architecture of the models, that are designed for a different task. For example, BIT uses a feature extractor for each image of the bi-temporal image. The absolute difference between the two feature maps is fed into a prediction head. For remote sensing images, such as the task of detecting urban development, the feature extractor is used to extract the presence of urban structures [CQS21]. When the two feature maps are subtracted, only the new urban structures remain in the feature map [CQS21]. For graffiti images, this method might be less useful because usually the graffiti wall is already filled with graffiti, and old graffiti is replaced with new ones. Therefore the feature extractor cannot just detect graffiti in both input images and subtract them from each other. ChangeFormer uses a convolutional difference-module instead of a strict subtraction but still performs worse than the Simple U-Net. This indicates that a dedicated architecture is needed.

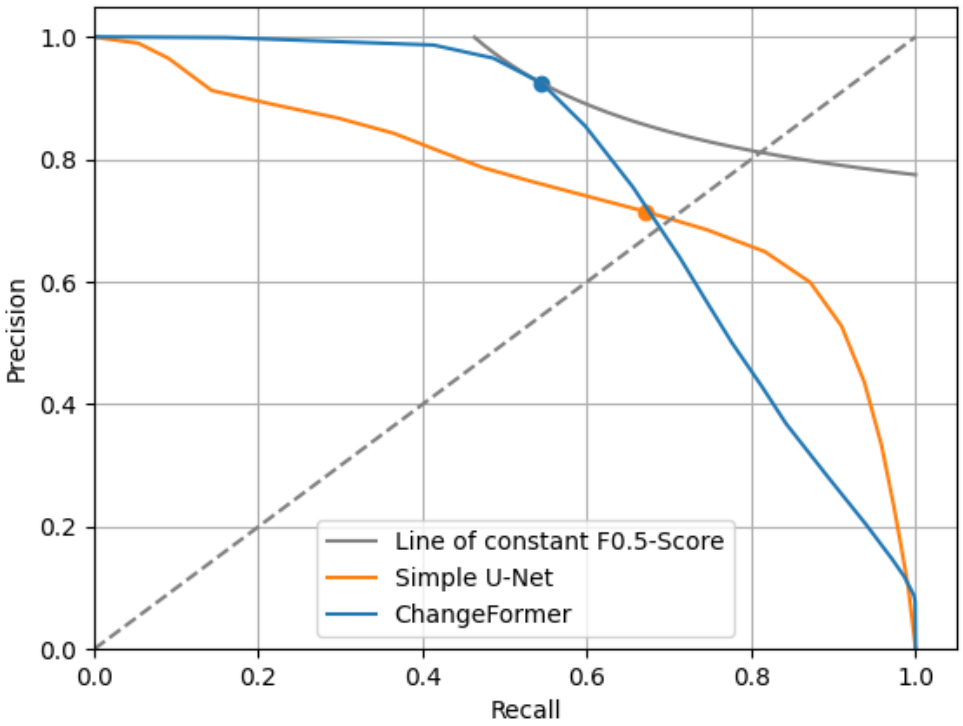


Figure 5.1: Precision-recall curve of the Simple U-Net, trained on real data only and the ChangeFormer trained on all data. The results of the models without changing the threshold are marked with a dot. The grey line indicates a line along which the $F_{0.5}$-Score remains 0.811.

5.3 Limitations and Future Work

The models show a positive effect in all metrics when being finetuned, but the available data is very limited. The dataset created by Wild et al. [WVMP23] contains 6,902 image pairs, but these image pairs originate from only 17 different locations. The same 17 locations are used as the backgrounds of the hybrid set. In contrast, the synthetic dataset provides 1,267 different backgrounds and 787 different graffiti. Although the synthetic dataset can increase the performance significantly, as can be seen in the tables from Section 4.2, more real data is needed. Only 17 different locations with slight variations of changes in lightning and the present graffiti is not enough to generalize to more scenarios.

In the scope of this thesis the models were finetuned in their entirety, without researching the effect of freezing layers [GBC16]. By freezing layers of the models, the trainable parameters can be significantly reduced to combat the shortage of training data. Although

the frozen layers need to be selected carefully, most change detection models use a feature extraction stage as well as the prediction head and it may be necessary that both stages have to adapt their behavior to increase the performance on the new domain.

As mentioned in Section 5.2 future graffiti change detection models should be designed dedicated to the task at hand. Extracting features and creating a change map from a form of difference can be useful for remote sensing images but for graffiti images, this may be different. This can be seen with the Simple U-Net, which outperforms more advanced and complex transformer architectures. In future work, a dedicated change detection model designed for graffiti images could be established.

5.4 Contribution

The contribution of this thesis is to evaluate how state-of-the-art change detection models perform on a different domain, namely graffiti images, and how much they can benefit from finetuning with synthetic and real graffiti data. The thesis provides insights into the challenges and opportunities of applying change detection to graffiti images, which is a largely unexplored application. The thesis also demonstrates the usefulness of synthetic data generation for change detection tasks, especially when real labeled data is scarce.

List of Figures

List of Tables

Bibliography

[BBM08] Francesca Bovolo, Lorenzo Bruzzone, and Mattia Marconcini. A novel approach to unsupervised change detection based on a semisupervised svm and a similarity measure. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2070–2082, 2008.

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[BMMW94] Eduardo S Brondizio, Emilio F Moran, Paul Mausel, and You Wu. Land use change in the amazon estuary: Patterns of caboclo settlement and landscape management. *Human Ecology*, 22:249–278, 1994.

[CCM+20] Tamás Czimmermann, Gastone Ciuti, Mario Milazzo, Marcello Chiurazzi, Stefano Roccella, Calogero Maria Oddo, and Paolo Dario. Visual-based defect detection and classification approaches for industrial applications—a survey. *Sensors*, 20(5):1459, 2020.

[CHL+23] Guangliang Cheng, Yunmeng Huang, Xiangtai Li, Shuchang Lyu, Zhaoyang Xu, Qi Zhao, and Shiming Xiang. Change detection methods for remote sensing in the last decade: A comprehensive review. *arXiv preprint arXiv:2305.05813*, 2023.

[CQS21] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

[CS20] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sensing*, 12(10):1662, 2020.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DLSB18] Rodrigo Caye Daudt, Bertr Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 4063–4067. IEEE, 2018.

[FCF03] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. Automatic change detection of driving environments in a vision-based driver assistance system. *IEEE Transactions on Neural Networks*, 14(3):646–657, 2003.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GCBP22] Wele Gedara Chaminda Bandara and Vishal M Patel. A transformer-based siamese network for change detection. *arXiv e-prints*, pages arXiv–2201, 2022.

[HCC+13] Masroor Hussain, Dongmei Chen, Angela Cheng, Hui Wei, and David Stanley. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:91–106, 2013.

[HSK+08] Chengquan Huang, Kuan Song, Sunghee Kim, John RG Townshend, Paul Davis, Jeffrey G Masek, and Samuel N Goward. Use of a dark object concept and support vector machines to automate forest cover change analysis. *Remote Sensing of Environment*, 112(3):970–985, 2008.

[Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[JPZ+22] Huiwei Jiang, Min Peng, Yuanjun Zhong, Haofeng Xie, Zemin Hao, Jingming Lin, Xiaoli Ma, and Xiangyun Hu. A survey on deep learning-based change detection from high-resolution remote sensing images. *Remote Sensing*, 14(7):1552, 2022.

[Lil72] Robert L Lillestrand. Techniques for change detection. *IEEE Transactions on Computers*, 100(7):654–659, 1972.

[LMBM04] D. Lu, P. Mausel, E. Brondízio, and E. Moran. Change detection techniques. *International Journal of Remote Sensing*, 25(12):2365–2401, 2004.

[LZDD22] Qingyang Li, Ruofei Zhong, Xin Du, and Yu Du. Transunetcd: A hybrid transformer network for change detection in optical remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

[MV22] Murari Mandal and Santosh Kumar Vipparthi. An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6101–6122, 2022.

[Par23] Eleonora Jonasova Parelius. A review of deep-learning methods for change detection in multispectral remote sensing images. *Remote Sensing*, 15(8):2092, 2023.

[PRB+19] K. Ram Prabhakar, Akshaya Ramasamy, Suvaansh Bhambri, Jayavardhana Gubbi, R. Venkatesh Babu, and Balamuralidhar Purushothaman. Changenet-v2: Semantic change detection with convolutional neural networks. 2019. Unpublished manuscript available on GitHub: `https://github.com/suvaansh/ChangeNet-v2`.

[RAAKR05] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[SGE17] Abdelrahim AM Salih, El-Tyeb Ganawa, and Anwer Alsadat Elmahl. Spectral mixture analysis (sma) and change vector analysis (cva) methods for monitoring and mapping land degradation/desertification in arid and semi-arid areas (sudan), using landsat imagery. *The Egyptian Journal of Remote Sensing and Space Science*, 20:S21–S29, 2017.

[Sin89] Ashbindu Singh. Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6):989–1003, 1989.

[SZZ+20] Wenzhong Shi, Min Zhang, Rui Zhang, Shanxiong Chen, and Zhao Zhan. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sensing*, 12(10):1688, 2020.

[TSG+16] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

[VGRB18] Ashley Varghese, Jayavardhana Gubbi, Akshaya Ramaswamy, and P Balamuralidhar. Changenet: A deep learning architecture for visual change

detection. In *Computer Vision – ECCV 2018 Workshops*, pages 129–145, 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, volume 30, pages 5998—6008, 2017.

[WPM00] Douglas Ward, Stuart R Phinn, and Alan T Murray. Monitoring growth in rapidly urbanizing areas using remotely sensed data. *The Professional Geographer*, 52(3):371–386, 2000.

[WVMP23] Benjamin Wild, Geert Verhoeven, Rafał Muszyński, and Norbert Pfeifer. Indigo change detection reference dataset. *TU Wien Research Data*, 10.48436/ayj4e-v4864, 2023.

[WVP23] Benjamin Wild, Geert Verhoeven, and Norbert Pfeifer. Tracking the urban chameleon–towards a hybrid change detection of graffiti. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 285–292, 2023.

[WVW+23] Benjamin Wild, Geert Verhoeven, Martin Wieser, Camillo Ressl, Johannes Otepka-Schremmer, and Norbert Pfeifer. Graffiti-dokumentation: Projekt indigo. In *22. Internationale Geodätische Woche Obergurgl 2023*, pages 322–325. Wichmann, 2023.

[WZLW21] Zhixue Wang, Yu Zhang, Lin Luo, and Nan Wang. Transcd: scene change detection via transformer-based architecture. *Optics Express*, 29(25):41409–41427, 2021.

[ZWCL22] Cui Zhang, Liejun Wang, Shuli Cheng, and Yongming Li. Swinsunet: Pure transformer network for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

Appendix

Overview of generative AI tools used

AI Tool	Description of usage
Microsoft Copilot	Microsoft Copilot is an AI assistant powered by a large language model (LLM). This tool was used for inspiration on how to phrase sentences to enhance the overall readability of the text. Source: `https://copilot.microsoft.com/`
Grammarly	Grammarly is a typing assistant for grammar and spelling mistakes and also analyses written text in real-time to suggest improvements in the writing style. This tool was used to correct grammar and spelling mistakes and sentence structuring. Source: `https://app.grammarly.com/`