



Leveraging diffusion models for synthetic data generation: Application in mono-temporal change detection from aerial images

Master Thesis

for the attainment of the Master`s degree “Master of Science”,
abbreviated “MSc”

submitted within the University Master Program for Further Education
“Geographical Information Science & Systems – (UNIGIS MSc)”
at the Department of Geoinformatics - Z_GIS,
Faculty of Digital and Analytical Sciences,
University of Salzburg
submitted by

Niclas Julius Homburg

Supervisor:
Dr. Getachew Workineh Gella

Lüdenscheid, March 2026

Content

- Acknowledgements..... IV
- Declaration..... V
- List of Figures..... VI
- List of Tables VI
- Abstract..... VII
- 1 Introduction 1
 - 1.1 Motivation and Problem Statement..... 1
 - 1.2 Objectives of the study 2
 - 1.2.1 General Objective 2
 - 1.2.2 Specific Objectives 2
 - 1.3 Significance of the study 3
 - 1.4 Structure of the thesis..... 3
- 2 Review of the related literature 4
 - 2.1 Change detection in remote sensing and deep learning methods..... 4
 - 2.1.1 Deep learning methods 4
 - 2.1.2 Overview of common change detection datasets..... 5
 - 2.2 Generative models for synthesis 6
 - 2.2.1 Theoretical basics of diffusion models 6
 - 2.2.2 Application areas of generative image synthesis 7
 - 2.3 Role of Generative Data in Change Detection 8
 - 2.3.1 Domain Gap and Semantic Consistency 8
 - 2.3.2 State of research on synthetic training..... 9
 - 2.4 Relevant metrics and evaluation methods in change detection..... 10
- 3 Methodology 11
 - 3.1 Data Acquisition and Processing 11
 - 3.1.1 Data sources 11
 - 3.1.2 Spatial sampling strategy 11
 - 3.1.3 Temporal Consistency and Data Availability..... 11
 - 3.1.4 Building footprint dataset 11
 - 3.2 Semi-synthetic generation pipeline 12

3.2.1	Mask definition	12
3.2.2	Diffusion Inpainting Component	13
3.2.3	Generation Parameters and Prompting.....	14
3.2.4	Structure of the real-world validation data set.....	14
3.2.5	Performance Evaluation	15
3.3	Change detection model.....	15
3.3.1	Model Architecture	15
3.4	Training setup.....	17
3.4.1	Training regime	17
3.4.2	Dataset configuration	17
3.5	Experimental Setup and Evaluation.....	18
3.5.1	Scenarios.....	18
3.5.2	Evaluation metrics	24
3.6	Implementation	24
3.6.1	Technical Infrastructure	24
3.6.2	Software frameworks used	25
3.6.3	Codebase structure, configuration, and run management	25
4	Results	26
4.1	Experiment E1: Diffusion Model comparison	26
4.1.1	Objective and Setup (E1)	26
4.1.2	Training dynamics and convergence (E1)	26
4.1.3	Qualitative generation quality (E1).....	27
4.1.4	Quantitative generalization on ALKIS validation dataset (E1)	32
4.2	Experiment E2: Impact of data augmentation	34
4.2.1	Objective and setup E2	34
4.2.2	Training dynamics E2	34
4.2.3	Qualitative augmentation review E2	34
4.2.4	Quantitative generalization on ALKIS validation dataset (E2)	37
4.3	Experiment E3: Model architecture comparison.....	38
4.3.1	Objective and setup (E3)	38
4.3.1	Training dynamics and convergence.....	39
4.3.2	Quantitative generalization on ALKIS validation dataset (E3)	40

4.4	Experiment E4: Comparison of alternative strategies.....	40
4.4.1	Objective and setup (E4)	40
4.4.2	Training dynamics and convergence.....	40
4.4.3	Qualitative comparison of synthetic-change strategies	42
4.4.4	Quantitative generalization on ALKIS validation dataset.....	43
5	Discussion.....	45
5.1	Base generator effects on diffusion models (Experiment 1).....	45
5.2	Impact of augmentation (Experiment 2).....	45
5.3	Influence of Network Architectures (Experiment 3).....	46
5.4	Comparison to non-diffusion image synthesis strategies (Experiment 4).....	47
5.5	Threshold dynamic and class imbalance.....	47
5.6	Limitations.....	47
5.7	Practical use for ALKIS and governmental surveying.....	48
5.8	Outlook	49
6	Bibliography.....	50

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Getachew Workineh Gella, for his close supervision during my work on the thesis, through always enlightening conversations and the open exchange of ideas. His always important and constructive criticism, as well as his motivational boost were very welcome! I would also like to thank the UniGIS Office Team for their organizational support. Finally, I would like to acknowledge the constant encouragement of my friends and family on the way to my graduation. A special mention goes to my partner, who supported me and showed great patience during the final stages of my thesis.

Declaration

I declare that I have written this work myself and have not used any sources or resources other than those I have cited. All passages that have been taken directly or indirectly from other sources are explicitly marked as such.

AI tools (ChatGPT and DeepL) were used to assist with coding and language revision. All generated results were manually reviewed or modified to ensure technical correctness.

This work has not been previously submitted to any other examination board and has not been published.

Lüdenscheid, 01.03.2026


Niclas Homburg

List of Figures

Figure 1 Directed graphical model of the diffusion process (Ho et al., 2020).....	6
Figure 2 Confusion Table Binary Change evaluation	10
Figure 3 Validation loss across epochs for Experiment 1 generator settings.....	26
Figure 4 Qualitative comparison of synthetic change generation in Experiment 1	31
Figure 5 Prediction mask comparison for Experiment 1	33
Figure 6 Validation loss across epochs for Experiment 2 augmentation levels	34
Figure 7 Prediction mask comparison for Experiment 2	38
Figure 8 Validation loss across epochs for Experiment 3 change architecture	39
Figure 9 Validation loss across epochs for Experiment 4 synthetic change strategies...	41
Figure 10 Prediction mask comparison for different synthetic change strategies.	44
Figure 11 Color histogram comparison synthetic t2 and WMS retrievals	48

List of Tables

Table 1 Overview of experimental scenarios and controlled variables.....	20
Table 2 Configuration and prompting strategies for Experiment 1	21
Table 3 Augmentation intensity parameters for Experiment 2	22
Table 4 Overview of synthetic-change strategies in Experiment 4	23
Table 5 Convergence summary for Experiment 1.....	26
Table 6 Performance metrics by generator on ALKIS validation set in Experiment 1	32
Table 7 Random ‘No Augmentation’ example	34
Table 8 Random ‘Strong Augmentation’ example.....	35
Table 9 Random ‘weak Augmentation’ example	36
Table 10 Performance metrics by augmentation level Experiment 2.....	37
Table 11 Training configurations and hyperparameters for Experiment 3	39
Table 12 Convergence summary for Experiment 3.....	39
Table 13 Performance metrics by backbone architecture Experiment 3	40
Table 14 Convergence summary for Experiment 4	41
Table 15 Comparative analysis of synthetic change processing strategies.....	42
Table 16 Performance metrics by synthesis strategy Experiment 4	43

Abstract

Though Deep learning models showed promising performance in change detection from remotely sensed images, they have critical limitation of demanding extensive amount of properly annotated training data. To overcome this challenge, this study proposes a semi-synthetic change detection data pipeline using the North Rhine-Westphalia cadaster (ALKIS) as a use case. The methodology focused on the systematic exploration of synthetic image synthesis and its application training deep learning based building change detection models where model performance was tested with four experimental setups: two generative diffusion models (SDXL and Flux) with inpainting logics, role of different degrees of augmentation and the influence of network architectures. In addition, the results were fairly compared with existing methods leveraged mono-temporal data for semi-synthetic data generation. The results show it is possible to train change detection models using semi-synthetic datasets. To successfully transfer the model to real-world data, a specific combination proved to be most effective: Structurally coherent image generation (especially through flux models) with variation in image generation must be combined with strong radiometric augmentation. Under these conditions, a DeepLabv3 architecture with a ResNet backbone showed the highest prediction results with an F1 score of 0.68. However, the comparative test also showed that a purely combinatorial, logic-based approach Unpair method is currently still superior to the generative approach (achieved an F-1 score 0.72), as the random pairing of real images produces an almost infinite radiometric variance. The work shows the technical feasibility of the approach. Even if the accuracy achieved is not yet sufficient for fully automatic cadastral updating, the developed technique proves that robust CD models can be trained effectively and without manual labeling. This creates the technological basis for future resource-saving monitoring processes such as pre-filtering systems that flag potentially unregistered buildings for human review.

Keywords

change detection, deep learning, diffusion models, image synthesis

1 Introduction

1.1 Motivation and Problem Statement

Currently there are many earth observation satellites and aerial campaigns that provide a massive amount of remotely sensed data. This provides a greater opportunity to monitor the surface of the Earth with fine grained spatial and temporal resolution. The datasets have broader applications, one of which is building change detection (CD). Change detection is very relevant for various practical real-world applications, from evaluating damages after natural disasters (Zheng, Zhong, et al. 2021) to monitoring urban sprawl in landscape analysis (Jat et al. 2008) and infrastructure development (Al-Ruzouq et al. 2017). Though the availability of massive Earth observation datasets creates good opportunities for monitoring subtle changes in the built environment, information extraction workflows need to be automated.

In the context of modern administrative processes, particularly in Germany, CD plays a key role in the maintenance of geospatial base data (BKG, n.d.), alongside with in-situ surveying. Public authorities, such as land registry offices (*Katasterämter*) and building authorities (*Baubehörden*), rely on up-to-date information to ensure the consistency of the real estate cadaster (ALKIS). Here the goal is distinct from international crisis management: instead of detecting sudden destruction of following catastrophes, the focus lies on identifying continuous structural changes, like new buildings, extensions or demolitions in the built environment using high spatial resolution to update official registers more efficiently.

In this regard deep Learning approaches have revolutionized remote sensing and significantly outperformed older state of the art pixel based methods in recent years (Cheng et al. 2024) but they suffer from a major bottleneck: they require massive amounts of nearly perfect annotated training and validation data (Ball et al. 2017). A model learns to detect changes by processing thousands of examples of accurately labeled "before" and "after" image pairs. Obtaining such bi-temporal annotated dataset is expensive, time-consuming and practically difficult (Ball et al. 2018) given change classes are rare that even with the availability of time and resources it is very hard get sufficient change examples. Most publicly available benchmark datasets, such as xView2 or LEVIR-CD focus heavily on disaster scenarios and specific urbanization patterns that do not necessarily reflect the building structures and vegetation found in Central Europe (H. Chen and Shi 2020a; Gupta et al. 2019). Consequently, a model trained on disaster data may struggle to recognize a typical residential building extension in North Rhine-Westphalia due to domain shift (M. Yang et al. 2019) attributed to changes in background and scene spectral properties as a result of changing geography, season and sensor characteristics. This lack of suitable, region-specific training data remains a persistent challenge. To overcome this shortage of training data without the immense cost of manual

labeling, recent research has increasingly explored the use of Generative Artificial Intelligence. Early approaches like Seo et al. (2023) demonstrated that changes could be synthesized from single temporal images, reducing the need for paired labels.

Parallel to the creation of this thesis, more recently, large models like those proposed models such as Changen2 (Zheng et al. 2025) data generation pipelines like those from (Benidir et al. 2025) proved that diffusion models can create higher-quality synthetic training data for remote sensing. These very recent studies confirm the high relevance of the generative approach. However, they predominantly focus on developing complex, specialized model architectures or testing on global benchmarks. A critical gap remains regarding the accessibility and domain-specific application of these technologies. It remains unclear whether standard open access inpainting models, which are easily accessible to experts working in public planning institutions, without specialized AI departments, are sufficient to meet the performance requirements of the cadastral maintenance (ALKIS) in Germany, particularly regarding the synthesis of contextually consistent ground textures suitable for training robust change detection models.

1.2 Objectives of the study

1.2.1 General Objective

The main objective of this study is to demonstrate that the use of semi-synthetic datasets, generated from mono-temporal aerial imagery using accessible, off-the-shelf diffusion models, leads to robust change detection models capable of generalizing to authentic cadastral data (ALKIS), thereby serving as a viable alternative to manually annotated datasets.

1.2.2 Specific Objectives

To achieve the stated general objective, the study has the following specific objectives:

1. Develop an automated pipeline which utilizes openly available Diffusion models to generate semi-synthetic samples for CD, while maintaining a small domain gap to real Changes through different methods.
2. Train change detection models using different generated synthetic change detection datasets and validate on real observed change detection dataset through zero-shot inference.
3. Compare the approach with other state-of-the art methods that leverage mono-temporal images for change detection.

1.3 Significance of the study

The thesis addresses bottleneck of critical missing training data in deep learning-based CD. By the use of off-the-shelf diffusion models to generate semi-synthetic data, this approach offers a cost- and labor-efficient alternative to manual annotation. Essentially, this research bridges the domain gap for central European settlement structures, which provides public authorities with a scalable solution for flagging overlooked buildings for cadastral maintenance. Furthermore, it should be possible to transfer the workflow to other domains of area objects that are maintained in public datasets.

1.4 Structure of the thesis

The rest of the thesis is structured as follows. Chapter 2 provides the theoretical background and reviews of related work on change detection, diffusion models, and the use of synthetic data in remote sensing. Chapter 3 describes the proposed methodology and technical implementation in detail. This includes the used data sources, the generative data-creation process based on the Flux model, and the setup for training and validating the change detection model. Chapter 4 presents and evaluates the results, while Chapter 5 discusses their scientific as well as practical implications as well as concludes the thesis and outlines potential directions for future work.

2 Review of the related literature

2.1 Change detection in remote sensing and deep learning methods

The CD procedure identifies differences of multi-temporal images of the same region, which are real observations on the surface of the Earth. Since the early stages of digital remote sensing, CD has been defined as a process of comparing bi-temporal image data. The basis is a comparison between bitemporal image data, which are captured with minimal differences in illumination and sensor calibration (SINGH 1989). The process aims to determine spectral or structural differences in the images, which are on one hand caused by real-world changes and on the other hand changes which are caused by external influences, such as light, shadows, atmospheric effects, or sensor incoherences of the hardware (Lu Corresponding author et al. 2004). To achieve this, the captured areal images are geometric and radiometric corrected before they are used to start with the pixel or object-based process (Ban and Yousif 2016).

According to Ban and Yousif (2016) the process to detect changes can be divided in three big steps: 1. Data preparation with corrections, 2. CD from multitemporal Images, 3. Validation of change data through Accuracy Assessment. Depending on the aim of the CD, different techniques are used: pixel based to identify spectral differences, object-based changes for semantic changes or machine learning approaches for more complex changes.

2.1.1 Deep learning methods

Classical methods are commonly lightweight computing tasks, by comparing the pixels with the same coordinates with each other. Due to the work process, methods based on the process are normally sensitive to light exposure or atmospheric changes (Tan et al. 2013), there are several approaches to solve the problem that reduce sensitivity. The simplest approaches are “image differencing”, which uses the difference in pixel values instead of the absolute value, and “image ratioing”, which sets the pixel values in relation to each other (SINGH 1989). Because of the weaknesses of purely pixel-based approaches, object-based image analysis (OBIA) methods were developed (G. Chen et al. 2012). The main idea is to analyze image segments (objects) in their spatial context, rather than treating each pixel independently. This approach is helping in CD with high-resolution aerial images (Blaschke 2010).

From the early 2000s onward, supervised machine-learning classifiers became dominant in remote sensing. Instead of hard per-pixel thresholds, these methods learn decision rules from spectral bands plus engineered features such as indices and texture measures, sometimes complemented by simple shape cues. Support Vector Machine (Mountrakis et al. 2011) and Random Forests (Belgiu and Drăguț 2016) are among the most common choices. In practice, they often hold up better in heterogeneous scenes and with high-resolution or high-dimensional imagery (Maxwell et al. 2018). They are used

for land-cover mapping and, increasingly, in change-detection workflows (Pal and Mather 2005; Mountrakis et al. 2011; Belgiu and Drăguț 2016).

In the field of remote sensing, generative models are increasingly employed for a variety of applications, including super-resolution (Demiray et al. 2021), data augmentation (Mohandoss et al. 2020), domain adaptation (Zhu et al. 2017), and the generation of additional training data (Nguyen et al. 2024; Gella and Lang 2025; Benidir et al. 2025; Zheng et al. 2025). However, it is equally crucial to ensure that these results are structurally consistent. Recent research highlights that while text-to-image models can produce high photorealism, assessing their authenticity is critical for monitoring and verification tasks (Nguyen et al. 2024).

Since the mid-2010s, deep learning has replaced most handcrafted data features. Convolutional neural networks (CNN) (e.g., Siamese CNNs, UNet-Variants) learn multi-scale spatio-spectral patterns directly from the images and typically improve accuracy in complex urban scenes (Caye Daudt et al. 2019). More recently, transformer-based models have been introduced to incorporate long-range context and, in many cases, improve generalization. An example in change detection is ChangeFormer (Bandara et al. 2024), which uses self-attention to model relationships over larger spatial distances compared to typical CNN-based approaches. Review papers summarize this development and remaining problems such as the limited number of labels and domain shift challenges, as well as possible solutions (Cheng et al. 2024).

2.1.2 Overview of common change detection datasets

There are a handful of openly available change detection datasets that have become standardized benchmarks in CD, but they are still only covering a narrow slice of real-world variability. The xBD dataset (often mentioned as xView2) (Gupta et al. 2019) focuses on post-disaster building damage and provides more than 850,000 annotated buildingfootprints in over 22,000 post- and pre-event image pairs with damage categorization of 19 natural disasters.

The Onera Satalite Change Detection dataset (OSCD) offers Sentinel-2 multispectral Pre and post image pairs for 24 regions with different levels of urbanization (R. C. Daudt et al. 2018), while the High Resolution Semantic Change Detection (HRSCD) dataset extends this idea to nationwide areal imagery over France with multi-semantic-class change labels (R. Daudt et al. 2018). The dataset is geographically restricted and dominated by European landscape and architecture, which potentially limits the models trained on this dataset to scale under domain shift attributed to shifts in climate, landcover or building structures.

Other popular datasets like LEVIR-CD for building change detection in very-high-resolution Google Earth imagery, are similarly focused on urbanization monitoring (H. Chen and Shi 2020b). Recent reviews therefore emphasize that current open datasets are still limited, which beaks the performance improvement in CD, while the model parameter count growths (Li et al. 2024).

2.2 Generative models for synthesis

In this section, a comprehensive review of generative strategy through diffusion models is provided.

2.2.1 Theoretical basics of diffusion models

Diffusion probabilistic models, also known as Diffusion models (DM), are iterative generative models that aim to generate results through a stochastic process. The generated images are not generated in one step, but in an iterative process consisting of many denoising steps. Data generation can be understood as the “sequential application of denoising autoencoders” (Rombach et al. 2022). Besides this, DMs are offering guiding mechanisms without retraining the model, while they are operating directly in pixel space, which means that optimization is expensive (Rombach et al. 2022).

The diffusion process can be described by two complementary dynamics, both formulated as Markov chains. In the forward process, an original image x_0 is transformed into increasingly noisy versions x_t for $t = 1, \dots, T$, where each state depends only on the previous one $q(x_t|x_{t-1})$. With a suitable noise schedule, the x_0 is progressively added with noise until the final state x_T becomes close to a simple Gaussian distribution of pixels (Ho et al. 2020). The design makes sampling easy at the starting point of generation, because one can begin with random noise. The reverse process aims to reconstruct the denoised version of the image. A model $p_\theta(x_{t-1}|x_t)$ is trained such that, starting from x_T , a realistic sample result is achieved by iteratively applying the denoising process along a Markov chain beginning at $p(x_T) = \mathcal{N}(x_T; 0, I)$ (Ho et al. 2020).

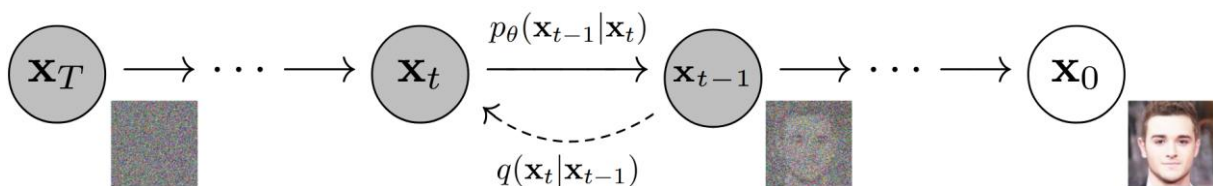


Figure 1 The directed graphical model considered in this work. Source: Ho et al. (2020), Fig. 2.

Core components in a diffusion pipeline often consist of three conceptually separate components:

- I. An image representation component, like an autoencoder that maps the images to a more compact latent space and back (Rombach et al. 2022).
- II. A denoising backbone, which performs the iterative refinement of the space (e.g., U-Net or Transformer-based backbones) (Rombach et al. 2022).
- III. A conditioning pathway, to contribute cross-attention mechanism like text. This condition process is added to the denoiser by an additional y component from $\epsilon_\theta(z_t, t) \rightarrow \epsilon_\theta(z_t, t, y)$ (Rombach et al. 2022).

Due to the high computational cost of high-resolution diffusion in pixel space and the need for many sequential denoising steps, a common strategy is to run the diffusion process not on raw pixels, but rather in a latent space as an efficiency strategy. This is obtained in an autoencoder, which is part of a two-stage system, an encoder E compresses an image into a latent representation $z = E(x)$ and a decoder D reconstructs an image from z (Rombach et al. 2022). Diffusion is then applied in the latent domain, which reduces compute while preserving perceptual quality if the latent space is well trained.

This idea is central for many practical text-to-image and inpainting systems, and it is also compatible with different backbone families: the denoiser operating on latent can be implemented as a CNN (e.g., U-Net) or as a transformer-based network (e.g., Flux.1 (black-forest-labs 2026)). The specific choice affects how local details and global context are represented, but the principle of iterative denoising remains unchanged (Peebles and Xie 2023).

The two most common backbone principles for image-to-image and text-to-image tasks are the U-Net design and Transformer-based architectures. UNets combine a contracting path that captures context with an expanding path that enables precise localization. Skip connections link corresponding resolutions and allow the decoder to recover fine details from high-resolution features computed in the encoder (Ronneberger et al. 2015).

More recent diffusion (and diffusion-related) systems increasingly rely on Transformer backbones instead of purely convolutional U-Net architectures. In these approaches, images are first compressed by an image autoencoder into a latent representation, which can be treated as a sequence of tokens. A Transformer then operates on these latent tokens using self-attention (and, in text-conditioned setups, cross-modal token interaction) to model global dependencies and to predict the denoising or flow-update step in latent space. After the iterative generative process, the resulting latent tokens are decoded back to pixel space by the autoencoder. This design is exemplified by FLUX.1, which performs generation in the latent space of an autoencoder with a Transformer-based architecture and token-level attention mixing (Labs et al. 2025).

2.2.2 Application areas of generative image synthesis

DMs have become established as a versatile framework for image synthesis and image editing. Beyond unconditional image synthesis, they are predominantly used for conditional generation, where additional information guides the output. A prominent example is text-to-image generation in which textual descriptions determine the semantic characteristics of the generated content. Moreover, DMs are particularly effective for image-to-image tasks, because they do not produce an image in a single step, instead they iteratively reconstruct content from noisy representations, so they are trained to reconstruct existing image information (L. Yang et al. 2025; Dhariwal and Nichol 2021).

This thesis is mainly concerned with image editing or inpainting tasks. In the process of inpainting, a specific region of the image is delineated by a mask, indicating the area to be restored (Rombach et al. 2022). The remaining portion of the image is then maintained in its original state. The model is then required to complete the missing part in a plausible manner, whilst remaining consistent with the surrounding context it should match the geometry and the textures of the scene. If such a function is required, a text prompt can be incorporated to instruct the model regarding the content that should be displayed within the masked region. Because of this, inpainting is not only useful for normal image editing, but it can also be used to create controlled, artificial changes in the input scene, for example by removing objects, replacing them, or changing their shape and appearance (Zheng et al. 2025).

In remote sensing, generative models are used more for different tasks, for example super-resolution, data augmentation, domain adaptation, and the generation of additional training data. In Earth observation synthesized images should be visually plausible while maintaining structurally consistent, so that the relevant object properties in the scene are still preserved. This underscores the significance of conditional diffusion and inpainting in this context, particularly when synthetic image pairs are generated for downstream tasks such as segmentation or change detection (Seo et al. 2023).

2.3 Role of Generative Data in Change Detection

In the context of this thesis, 'synthetic' or 'semi-synthetic' data refers to image pairs and labels that do not originate entirely from real bi-temporal observations but rather images partially generated by the generative approach. For instance, a real 'after change' image could be used to generate a plausible 'before change' version, from which the change mask could then be derived. This approach is extensively utilized in the domain of change detection, primarily due to the cost implications of bi-temporal training data, the necessity for accurate annotation, and the rarity of such data in sufficient quantities (Seo et al. 2023).

2.3.1 Domain Gap and Semantic Consistency

The training of change-detection models with synthetic image pairs is somewhat sensitive, because the model relies on visual consistency of unchanged areas not only object recognition. Approaches which ignore this consistency often fail. Seo et al. (2023) illustrate the risks of this limitation in their analyses of “unpaired supervision” where pixels are labeled as unchanged, despite significant visual differences between the two images. These disparities can be attributed to factors such as changes in building colour or texture due to illumination, material properties, or other effects. The occurrence of label appearance mismatches has the potential to disrupt the learning signal, thereby impeding the model's capacity to utilize contextual and structural cues that are typically present in authentic bitemporal pairs (Seo et al. 2023).

Prior work on change detection repeatedly notes that not every difference between two images reflects real-world change. Illumination shifts, shadows, seasonal variation, and atmospheric or sensor effects can all create discrepancies that look like change at first glance (Khelifi and Mignotte 2020). This matters in practice: if a generative pipeline systematically introduces such non-semantic differences, models may latch onto acquisition-style artefacts rather than learning true object change.

2.3.2 State of research on synthetic training

Recent research has explored different generative strategies, ranging from large-scale foundation models to task-specific pipelines. Zheng et al. (2025) introduced "Changen2", a generative foundation model. It simulates change as a stochastic process using a Diffusion Transformer to create large amounts of diverse training data for global applications. Focusing more on specific domains, Gella and Lang (2025) apply a generative approach to identify structural changes in temporary humanitarian settlements. Their work explicitly aimed to overcome the lack of bitemporal data in time-critical emergency responses.

These specific applications are supported by several studies in remote sensing, which suggest that synthetic data can improve model accuracy, in situations where real, labeled training pairs are scarce. Nguyen et al. (2024) discuss how modern text-to-image generators can be adapted to satellite imagery, and they evaluate the generated outputs with a mix of standard metrics and metrics that are more tailored to remote sensing. A key motivation in their work is that synthetic data can help to reduce data scarcity in remote-sensing based machine learning workflows (Nguyen et al. 2024). However, realism by itself is not enough. These works argue that synthetic images become practically useful only when they are controllable and when they stay consistent with the intended semantic content (Nguyen et al. 2024). In other words, it matters that the generated "change" is the change that is desired for a specific task, and not just some visually plausible but semantically unclear modification.

More direct evidence comes from change-detection training setups where synthetic data is created in a task-aligned way. Seo et al. (2023) propose Self-Pair, which constructs synthetic bi-temporal training pairs from single images. In their cross-domain experiments (training on xView2 pre-disaster or SpaceNet2, testing on WHU/LEVIR-CD), Self-Pair consistently outperforms an Unpair baseline across multiple CD architectures. For ChangeStar (Zheng, Ma, et al. 2021), for example, Self-Pair improves F-1 on LEVIR-CD (H. Chen and Shi 2020b) from 68.32 to 81.22 (SpaceNet2 to LEVIR-CD) and on WHU(AmberLi (2021)) from 88.65 to 93.14 (xView2 pre-disaster to WHU). The authors interpret this as evidence that Self-Pair approximates the distribution of real changes better than simpler single-temporal pairing strategies (Seo et al. 2023).

A similar conclusion is reported at larger scale by Benidir et al. (2025), who introduce FSC-180k, a hybrid dataset that mixes real very high-resolution imagery with inpainted

changes and dense labels. They report that pretraining on FSC-180k leads to performance gains compared to training from scratch, and it also outperforms a fully synthetic alternative (SyntheWorld) across different transfer settings. Importantly, the effect is strongest in low-data regimes: with only 1% of target training data, IoU on LEVIR-CD increases from 0.36 (baseline) to 0.55 (+53%) after FSC-180k pretraining, and on S2Looking from 0.10 to 0.15 (+50%) (Benidir et al. 2025).

2.4 Relevant metrics and evaluation methods in change detection.

In the context of binary CD, the evaluation of model performance typically takes place at pixel level, utilizing standard classification metrics derived from the confusion matrix.

Confusion Matrix	Predicted change	Predicted no-change
Ground Truth Change	True Positive (TP)	False Negative (FN)
Ground Truth No-Change	False Positive (FP)	True Negative (TN)

Figure 2 Confusion Table Binary Change evaluation

In addition to overall accuracy, more informative measures such as precision, recall, Intersection over Union (IoU), and the F1-score are commonly reported, as they describe the trade-off between missed changes and falsely recognized changes (Shafique et al. 2022; Li et al. 2024). In instances where class imbalance is a concern, Cohen's Kappa is a commonly utilized metric as a more robust agreement between prediction and ground truth (Leichtle et al. 2017). In practice, for binary mapping, model outputs (i.e., probabilities of the positive class) are often converted to binary maps using a decision threshold which is tuned to maximize a chosen metric, typically F1 or IoU, on a validation dataset.

3 Methodology

3.1 Data Acquisition and Processing

3.1.1 Data sources

This study uses airborne orthophotos from the state of North Rhine-Westphalia, Germany, as the visual basis for both datasets (i) a semi-synthetic change-generation dataset and (ii) a validation dataset. The orthophotos are accessed via standardized OGC web services provided by GeoBasis NRW (Geobasis NRW, n.d.-a). Specifically, the access workflow relies on the “Digitale Orthophotos (DOP) WMS” endpoint, which is listed as a viewing service in the official service overview (Geobasis NRW, n.d.-b).

The DOP products are georeferenced and photogrammetrically corrected orthophotos. According to the dataset documentation, they are provided at a ground sampling distance (GSD) of 10 cm/pixel and are updated in a two-year cycle. GeoBasis NRW reports that processing was moved towards true digital orthophotos since the 2008 flight program (Open.NRW, n.d.). In this work, orthophotos are requested via WMS in PNG format.

3.1.2 Spatial sampling strategy

To generate training and validation samples, a random building is selected as an anchor object. Each sample is a 512×512 pixel tile. Each sample is based on an anchor building selected from the database. The anchor building is randomly moved within the tile to avoid systematic detection of changes in the center. Depending on the density of the buildings, additional buildings can appear in the same patch or not. No explicit exclusion criteria are applied for synthetic-data generation, the sampling is intended to reflect the heterogeneity of NRW’s settlement structures.

3.1.3 Temporal Consistency and Data Availability

The objective of this study is to correlate orthophoto content with the annual building outline time series (see Section 3.1.4). However, not every year is available at every location through the WMS. Therefore, a data availability check is performed during the download process. If a requested year yields a black tile, the request is iteratively shifted to the preceding year until a valid orthophoto is delivered. Samples where the temporal discrepancy between the building outline timestamp and the orthophoto acquisition date exceeds three years are discarded to avoid inconsistencies in the unsupervised dataset.

3.1.4 Building footprint dataset

To represent building geometry and to define “real” building changes, this study uses the official building-outline product Hausumringe (HU) for NRW. The HU are obtained from the ALKIS building and structure objects and provide the geometry of over 10 million building footprints. They represent ground-plan outlines and explicitly do not contain roof geometries or other detailed architectural components. This distinction is relevant

because orthophotos often display roof overhangs and shadows that extend beyond the cadastral ground plan.

The HU datasets are annually available from 2016 onwards and are provided as historized yearly snapshots (Open.NRW, n.d.). For this study, building outlines from 2016 to 2024 are available, while the effective analysis window for change identification is 2019 – 2024, aligned with orthophoto availability in the pipeline. The official HU format documentation lists attributes such as the municipality key (AGS), object identifier (OI), function code, and update date (Geobasis NRW, n.d.-c). In this thesis, only the AGS is required for this workflow as a unique identifier which stays the same over the annual snapshots in most cases; other attributes are excluded.

A building is considered “changed” if its HU geometry differs between two annual snapshots. To distinguish genuine building changes from administrative geometry corrections (e.g., during the “Homogenisierungsprozess” (Frevel 2018)), a PostgreSQL-based “fuzzy matching” filter is implemented via PostGIS. A building is only marked as changed if its center of the geometry lies completely outside the existing footprints of the previous year or if the area variance between identical building identifiers is over 20%. In other words, geometric change of the footprint is used as the operational change criterion (e.g., newly built parts, demolition, or footprint modification with significant modification). Due to long planning and construction cycles in Germany, we found footprint modifications are very rare. Real-world changes predominantly occur as completely new constructions.

Orthophotos (available as DOP10 but utilized as DOP20) are retrieved through GeoBasis NRW’s OGC-compliant WMS services in PNG format, where the annual building outlines (HU) are used as downloaded Shapefile packages (Geobasis NRW, n.d.-b). Both products are published as open data under “Datenlizenz Deutschland – Zero – Version 2.0 (dl-de/zero-2-0)” (GovData, n.d.).

3.2 Semi-synthetic generation pipeline

3.2.1 Mask definition

To ensure exact spatial correspondence between the image data and the semantic change signal, mask creation and orthophoto retrieval are implemented as a coupled workflow, as both are based on the same database retrieval. Since the spatial extent of each patch depends on the random shift applied to the anchor building (see Section 3.1.2), the binary mask is rasterized directly into the specific pixel grid of the retrieved t_1 image. This coupling is essential because the randomized crop cannot be reproduced from the raw data sources later.

Binary inpainting masks are created by rasterizing the selected building footprints into the patch grid. The mask follows the usual inpainting convention: a binary semantic guidance mask where white marks the region that should be replaced, while black marks the region

that should remain unchanged (Hugging Face, n.d.). To account for minor mismatches between cadastral boundaries and visual roof edges, two mask dilation configurations are evaluated (1 m, 2 m). After experimenting with the various mask extensions, we only used the 1-meter mask for image manipulation.

A single consistent masking strategy is applied for the dataset generation: All eligible buildings larger than 40 m² inside a patch are selected for masking. Consequently, the generated image represents a scenario where all building structures within the view have been removed. If masked regions intersect the patch boundary, the mask is clipped by the patch extent.

To prevent masks from extending into adjacent footprints, buffering is applied with a neighborhood constraint. The buffered target footprint is clipped by subtracting the footprints of all non-target buildings before rasterization. This preserves the buffer without encroaching on neighboring buildings. This constraint is particularly critical for structural annexes smaller than the 40 m² selection threshold (e.g., garages attached to a main building), ensuring they are preserved even when the main structure is targeted for deletion.

3.2.2 Diffusion Inpainting Component

Masked regions are filled using a text-conditioned, diffusion-based inpainting model. In this setup, inpainting is treated as conditional generation: the model synthesizes plausible content inside a user-defined mask, while the unmasked context should remain unchanged. The mask and an optional text prompt together are guiding what is generated and where it appears (Hugging Face, n.d.; Rombach et al. 2022).

During the exploratory phase, two diffusion pipelines were tested: an SDXL inpainting pipeline (Hugging Face 2023) and a FLUX inpainting pipeline using the FLUX.1-Fill-dev checkpoint (black-forest-labs 2026). Both support prompt-guided inpainting, but they differ in model family and their interface details. For SDXL, negative prompts are supported as part guidance conditioning (Diffusers, n.d.). In contrast, the current FLUX inpainting pipeline in Diffusers does not expose a negative prompt argument in the same way, so conditioning is handled slightly differently.

Qualitative comparisons indicated that FLUX.1-Fill-dev produced more consistent results for the specific remote sensing edits required in this study. Therefore, FLUX is selected as the primary generator. Additionally, LoRA-based adaptations are considered to stabilize the style and semantics of the generated content, while the Stable diffusion inpaint pipeline remains as baseline.

3.2.3 Generation Parameters and Prompting

Image synthesis is performed at a fixed spatial resolution of 512×512 pixels to match the resolution of the change map and t_1 . Across all reported runs, the number of denoising steps is kept constant at 30. The strength of prompt conditioning is controlled via the guidance scale parameter, following the common classifier-free guidance idea: higher values enforce closer adherence to the text prompt, but they can also reduce perceptual fidelity. In this study, the guidance scale is typically set to around 45.

Prompts are used for both SDXL and FLUX inpainting. For SDXL, an additional negative prompt can be specified to suppress unwanted attributes. Measures were introduced at various stages to prevent the model from overfitting on repetitive textures (e.g., overly vivid trees). In the final phase, random theme specifications (e.g., agricultural fields, parking lots, construction sites) are used. These are coupled with category-specific Low-Rank Adaptation (LoRA) adapters that have been trained with FluxGym (cocktailpeanut (2024)) using hand-picked examples of typical target landscapes. This ensures that the generative process is aligned with the intended background distribution while maintaining semantic diversity. To ensure reproducibility in all experiments, a constant seed value was set to 0.

The manipulation strategy follows the mask definition introduced in Section 3.2.1. Specifically, all identified building structures within the patch are targeted for removal. Apart from the mask definition itself, no further manipulation-specific constraints are applied at this stage. The pipeline does not use an automatic quality filter for the generated images. Except for the diagnostic naming of files for debugging purposes, which makes it possible to manually verify mask correctness and whether the intended editing region was used.

3.2.4 Structure of the real-world validation data set

The real-world validation data set is generated by first analyzing the HU in the years under investigation for changes in the vector data sets, followed by retrieving the aerial photographs for the respective years via WMS. One image t_2 after the change and one image t_1 before the change. Care is taken to ensure that aerial photographs for the respective years are available. If images are available for both years, the candidates are saved according to the usual scheme in a separate folder with a ground truth mask.

The candidates are then manually reviewed using orthophotos to ensure that the change is visible in the image and that the patch under consideration does not contain any additional building changes that are not covered by the HU difference. Candidates are rejected if the change cannot be visually confirmed as plausible (e.g., due to temporal inconsistency of the images) or if the patch is affected by further changes that cannot be fully captured.

3.2.5 Performance Evaluation

The generalization is evaluated on a separate real-world validation dataset, which is derived from authoritative cadastral-based changes (ALKIS) and aerial imagery. Real change is defined by verified footprint geometry differences, and it is grouped in three practical relevant change types, i.e., the appearance of a building (new construction), the disappearance of a building (demolition), and geometry changes of the footprint (e.g., extension or partial removal). The main difference is the amount of change in each sample, the evaluation dataset has much less changes. The semi-synthetic pipeline generates primary deletion-like changes, while the real dataset additionally contains appearance cases and footprint modifications. The evaluation explicitly tests whether a model trained under controlled deletion scenarios remains transferable to heterogeneous real building changes, and whether it learns structural change cues rather than generator-specific artefacts.

3.3 Change detection model

3.3.1 Model Architecture

As the objective of this study, building CD is formulated as binary, pixelwise segmentation. For this purpose, we modified two commonly used segmentation models, viz, VGG19 with UNet like architecture and DeepLabv3 (L.-C. Chen et al. 2017). They receive two Aerial Images from the same area on two different overflight years as t_1 and t_2 . Both follow a Siamese feature extraction module where bitemporal features are first extracted by the shared encoder. This bitemporal features are ingested into a custom change module which yields a unified change feature representation. These change features are then converted into a dense change prediction via a decoder, where the outcome is mapped to a binary change map, in which every pixel is classified into change or non-change.

The result of calculation is issued in a Logit-Map, a raw prediction without probability interpretation. The Logit-Map is transformed into a Sigmoid-Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

which maps values to the interval $[0, 1]$ (Goodfellow et al. 2016). With a given threshold by the user, a binary change map is processed. In the training process the logits are optimized directly to obtain numerically stable gradients.

The VGG based model is modified into a UNet like encode decoder architecture with ResNet multi-scale change features. The encoder provides multi-scale features both from pre and post mages. These bi-temporal multiscale features were fed into a change module that produces multi-scale change features. These multi-scale change features

were then fed into a decoder where it is fused with bottleneck features through skip-connections followed with upsampling. The custom change module first creates the absolute difference of bitemporal features followed by convolution and batch normalization as follows.

$$F_{diff} = \text{BatchNorm2d}(\text{ReLU}(\text{Conv2D}(|F_{t1} - F_{t2}|)))$$

The DeepLabv3 based configuration utilizes a ResNet backbone (He et al. 2016) (ResNet-50 and ResNet101 are part of experiments), where $t1$ and $t2$ are separately processed through the backbone where feature maps are extracted (L.-C. Chen et al. 2017). This operation is followed by a parameter-free fusion of the changes, which deducts the change from the feature maps. By using this model, the subtraction based difference variant is used, a pixel-by-pixel distance map is first calculated and normalized by the difference between the feature vectors at the deep latent space (Caye Daudt et al. 2018).

$$F_{diff} = |F_{t1} - F_{t2}|$$

These results lead to a weighting of the feature difference, which emphasizes stronger changes more than smaller differences in the value of two points.

Therefore, both model variants deliver a uniform output representation (logits or change probabilities), which means that they can be trained directly on the binary reference masks described in Chapter 3 and compared using an identical evaluation protocol and metrics. The aim of this work is to evaluate the effect of semi-synthetic training data on CD, rather than propose a new CD architecture. Therefore, the effect is tested on two established segmentation techniques of different model families.

DeepLabv3 (ResNet-50/ResNet-101) is chosen, because it is a powerful and widely used semantic segmentation framework. Atrous convolutions allow a larger effective receptive field and still high feature resolution, while Atrous Spatial Pyramid Pooling (ASPP) aggregates information in parallel with different atrous rates, to be robust for object detection over various object sizes. In the original paper, DeepLabv3 is classified as comparable to other state-of-the-art models and showing corresponding benchmark performance (L.-C. Chen et al. 2017).

As an alternative, a U-Net-style encoder–decoder architecture with a VGG-like encoder is used, since this design supports precise localization by combining a contracting path for context with an expanding path and skip connections that preserve fine spatial detail (Ronneberger et al. 2015). The multi-scale feature hierarchy allows the model to represent both local boundary cues and broader contextual patterns relevant for change segmentation (Ronneberger et al. 2015).

3.4 Training setup

3.4.1 Training regime

All CD models are trained on pixelwise binary segmentation paired RGB patches t_1, t_2 of size 512 x 512 pixel. The aerial images are normalized from 8-bit RGB to $[0,1]$ to ensure a consistent input range across generators and keep the image augmentations well defined. Training optimizes binary cross-entropy loss (BCEWithLogitsLoss (PyTorch, n.d.)). Adam (Kingma and Lei 2015) is used as optimizer. The initial learning rate is tuned per model, with a simple learning rate decay schedule during the training. The models are trained up to 50 epochs with early stopping strategy with patience of 10 epochs. While no external validation split was provided, the dataset was partitioned through a routine with a default train ratio of 95%, so 5% of all samples are used for training validation. The validation-set stays identical over the hole training, to prevent fitting to the training sample. The Batch size is set to the largest possible for each architecture to saturate a NVIDIA A100 with 40GB VRAM without running out of memory.

To improve robustness against minor differences between the patches t_1, t_2 , augmentations are applied on the fly during training. This is motivated by the general observation in change detection, that differences between two points in time, which appear in the images, may be caused not only by semantic changes, but also by lighting effects or shadows and seasonal influence (Khelifi and Mignotte 2020). The argumentation pool combines multiple techniques:

1. Geometric transformations which are applied to t_1, t_2 and $mask$, in the form of random horizontal/vertical flips and 90° rotations.
2. Image disturbances applied randomly to each sample within fixed limits in strength, these effects include random RGB noise, smoothed noise, and global brightness/contrast/color shifts.
3. Additionally, random channel permutation (RGB channel swap) is used.
4. The last step is to randomly swap the temporal order of t_1 and t_2 during training, to ensure there is no exclusive sensitivity to missing objects.

3.4.2 Dataset configuration

Training is performed on semi-synthetic image pairs, which are generated from real orthophoto patches and footprint-based inpainting masks. For each generator configuration, a standardized dataset of 5,000 samples is created. In every sample, all eligible buildings within the mask definition (see 3.2.1) are deleted.

The study compares four generator settings on this dataset size:

- I. Stable Diffusion (inpainting)
- II. FLUX (inpainting)

- III. FLUX (inpainting) + generalized LoRA
- IV. FLUX (inpainting) + randomized prompts + prompt-fitted tuned LoRAs

To keep supervision consistent, the same footprint-based masks (final experiments: 1 m buffer) and the same sampling logic for t_1 are used. Only the generated t_2 image differs by generator/prompt/LoRA configuration.

A separate real-world validation dataset is constructed from authoritative cadastral changes (ALKIS) and orthophotos within 2020 – 2024. “Real change” is defined as a verified footprint geometry change (new, removed, or significantly modified building footprints). This dataset is never used for training, and it is reserved for the final evaluation of generalization from semi-synthetic training data to authentic building changes. This dataset contains 300 samples.

3.5 Experimental Setup and Evaluation

3.5.1 Scenarios

The semi-synthetic training datasets represent controlled changes produced via inpainting-based manipulation. To systematically evaluate the impact of specific design decisions systematically, the study is structured into four scenarios. Each experiment is specifically designed to isolate a single variable or a distinct cluster of parameters (see Table 1).

- I. (E1) Generation control scenario
All four generator variants are tested with identical sampling and label logic: Stable-Diffusion-XL-Inpainting, Flux.dev-Inpainting, Flux.dev-Inpainting+LoRA and Flux.dev1-Inpainting with random prompt and matching LoRA (see Table 2). This aims for the isolation of the quality and controllability of the image generation models without augmentation and variation in model architecture.
- II. (E2) Impact of data augmentation
To test the benefit, disadvantages of augmentation on semi-synthetic data, three augmentations levels (off/weak/strong) are tested. This scenario examines if (synthetic) radiometric disturbances have an impact on the transferability to real data. Overall experiment setup visible in Table 3.
- III. (E3) Model comparison
To ensure that the effect is not limited to a single architecture, the generator extension configuration with the best test results is tested on multiple CD backbones (VGG based Encoder-Decoder, DeepLabv3 with ResNET50/ResNET101). Target is to determine whether the results will be reproducible for different model architecture.
- IV. (E4) Comparison to alternative Synthetic Change techniques
In addition to the inpaint-based data generation scenario, a comparison of alternative synthetic change techniques (eg., self-pair/single-pair (Seo et al. 2023),

random pair (Zheng, Ma, et al. 2021), and semantic assignment/training via mask and t_1 (called No-Pair in this experiments)) are tested (see Table 4), which are based on the same aerial training dataset. This setup allows for a quantitative assessment of the performance gains or trade-offs offered by the inpainting method compared to established, simpler synthetic pairing strategies.

Table 1 Overview of experimental scenarios and controlled variables

Experiment	Varied factor(s)	Fixed generator setting	Fixed augmentation	Fixed backbone	Dataset size (train)	Mask buffer	Prompt/LoRA setting	Notes
E1 – Generation control	Generator family + conditioning: SDXL vs FLUX vs FLUX+LoRA vs FLUX+RandPrompt+LoRA	<i>varies</i> (4 generator variants)	fixed no augmentation	fixed Deeplabv3 Resnet 50	5,000 (synthetic train)	Images generated with 1 meter Buffer, CD models trained with 0 m Buffer	SDXL: (prompt + negativeprompt). FLUX: prompt, no negativeprompt. LoRA: prompt + generalized LoRA (scale 0.8); Random prompts + category-LoRA (scale 0.8)	Resolution: 512×512, 30 steps, seed 0 guidance: FLUX 45 SDXL 14
E2 – Augmentation impact	Augmentation level: off / medium / high	fixed: DiffLoRA	<i>varies</i> (off/med/high)	fixed Deeplabv3 Resnet 50	5,000 (synthetic train)	0 m	no prompt	Tests transfer robustness against photometric/geometric disturbances
E3 – Model comparison	CD backbone: VGG-Encoder-Decoder vs DeepLabv3 (R50/R101)	fixed: DiffLoRA	fixed: high	<i>varies</i> (VGG vs DLv3-R50 vs DLv3-R101)	5,000 (synthetic train)	0 m	no prompt	Individual learning rate and batch size in training.
E4 – Alternative synthetic change strategies	Synthetic pairing/label strategy: e.g. Self-Pair / Random Pair / train Model only to recognize buildings	fixed image basis (NRW tiles), generation varies depending on method	augmentation off	fixed Deeplabv3 Resnet 50	5,000 real images	0 m	no prompt	Comparison with ‘simpler’ synthetic CD strategies (same database)
Validation (for all experiments)	–	–	augmentation off	–	300 ALKIS samples	0 m	no prompt	Threshold t via Scan 0–1 (step 0.01), reporting: Precision/Recall/F1/IoU

Table 2 Configuration parameters, prompting strategies, and model checkpoints for the synthetic image generation in Experiment 1 (E1-A – E1-D).

Config	Generator (checkpoint)	Prompting strategy	LoRA(s)	Denoising Steps	Guidance scale	Seed	Prompt (Example in case of E1-D, random example of 10 possible prompts)	Notes
E1-A	SDXL inpainting (diffusers/stable-diffusion-xl-1.0-inpainting-0.1)	Static prompt for seamless texture continuation + negative prompt to suppress buildings/roofs	None	30	14.0	0	prompt = "Seamlessly restore the area where the ground is removed, ensuring the terrain blends perfectly with the surrounding map. Match the style, texture, and colors of the map, preserving all roads, vegetation, and other nearby features. Create a consistent and realistic continuation of the map's visual design." negative_prompt = "buildings, rooftops, urban structures, shelters, domes, walls, man-made structures, visible building remnants, artificial patterns, blurred transitions"	Strength 0.99 Negative prompt available
E1-B	FLUX inpainting (black-forest-labs/FLUX.1-Fill-dev)	Static prompt (no randomized themes)	None	30	45.0	0	prompt = "Replace buildings exclusively with farmland, vegetation, parks, roads, or natural landscapes. No buildings or roofs are allowed."	
E1-C	FLUX inpainting + generalized LoRA (FLUX.1-Fill-dev + LoRA)	Static prompt	Generalized LoRA (style/domain stabilization), scale [0.8]	30	45.0	0	"groun303dnrw20 GROUN303DNRW20 Replace masked areas in the aerial image with only construction site or gardens or parking lots or few trees or roads or natural surfaces or agricultural fields. No buildings, no roofs, no houses! Make sure the vegetation colors look a bit washed out."	LoRA applied to reduce repetitive artifacts / improve domain consistency.
E1-D	FLUX inpainting + randomized prompts + tuned LoRAs (FLUX.1-Fill-dev + LoRAs)	Randomized theme prompts (e.g., gardens/roads/parking lots) + explicit "no buildings/roofs" constraints	Category-specific tuned LoRAs (activated per theme), scale 0.8	30	45.0	0	"Replace masked areas in the aerial image with agricultural fieldsagriculture20dop20. No buildings, no roofs, no houses are visible. "	Designed to increase texture diversity and reduce "tree everywhere" collapse seen with static prompting.

Table 3 Parameterization of augmentation intensities across geometric, photometric, and temporal domains for Experiment 2.

Augmentation Method	Parameter	No Augmentation	Weak	Strong
Geometric				
Horizontal / Vertical Flip	Probability (p)	0.0	p=0.5 (Fixed)	p=0.5 (Fixed)
Random Rotation	Angle	None	0°,90°,180°,270°	0°,90°,180°,270°
Photometric				
RGB Noise	Strength	0.0	±0.06 (Fixed)	±S, where $S \sim U(0.005, 0.09)$
Smooth RGB Noise	Strength	0.0	±0.2 (Fixed)	±S, where $S \sim U(0.005, 0.09)$
	Smoothness (σ)	-	$\sigma=3.0$ (Fixed)	$\sigma \sim U(0.25, 4.5)$
Brightness	Shift Range	0.0	±0.2 (Fixed)	±B, where $B \sim U(0.05, 0.90)$
Contrast	Scaling Range	0.0	±0.1 (Fixed)	±C, where $C \sim U(0.05, 0.90)$
Color Shift	Channel Shift	0.0	±0.05 (Fixed)	±Cshift, where $C_{\text{shift}} \sim U(0.05, 0.90)$
Spectral				
Channel Swap	Probability (p)	0.0	p=0.3 (Fixed)	$p \sim U(0.015, 0.27)$
Temporal				
Input Flipping (t1 and t2)	Probability (p)	0.0	0.5	0.5

Table 4 E4: Overview of synthetic-change strategies

ID	Method Name	Pairing / supervision logic	Preprocessing required	Label source	Training input	Change evaluation	Augmentation
A	Self-Pair(a) (Seo et al., 2023)	This strategy generates a training pair by extracting four quarters from a single source image, then mix them up and defines the change supervision using the logical XOR difference of their corresponding semantic labels.	No	Self-Pair in training process strategy calculated in the training process	pseudo-Bi-temporal pair $t1, t2$ + change mask	Standard change detection prediction	Off
B	Unpair (Seo et al., 2023)	This method pairs two randomly sampled, spatially unrelated images from the dataset and derives the supervision signal by calculating the logical XOR difference between their semantic segmentation masks.	No	Unpair in training construction strategy calculated in the training process	pseudo-Bi-temporal pair $t1, t2$ + change mask	Standard change detection prediction	Off
C	No-Pair (building segmentation subtraction)	Trains a mono-temporal building segmentation model. Change is obtained by subtracting predicted building masks for $t1$ and $t2$ (post-hoc differencing).	No	Unprocessed building footprints as masks	Single image $t1$ + building mask	Predict buildings separately on $t1$ and $t2$, then compute change by differencing the two predicted building masks	Off
D	Inpainting-based semi-synthetic pipeline (this thesis)	Diffusion inpainting removes buildings in masked regions to create a controlled “before/after” pair; trained as supervised CD on $t1, t2$ with footprint-derived change masks.	Yes	Unprocessed building footprints as masks	pseudo-Bi-temporal pair $t1, t2$ + change mask	Standard change detection prediction	Off
E	Inpainting-based semi-synthetic pipeline (this thesis)	Diffusion inpainting removes buildings in masked regions to create a controlled “before/after” pair; trained as supervised CD on $t1, t2$ with footprint-derived change masks.	Yes	Unprocessed building footprints as masks	pseudo-Bi-temporal pair $t1, t2$ + change mask	Standard change detection prediction	Strong (see 4.2.3)

3.5.2 Evaluation metrics

To evaluate model performance, quantitative metrics at the pixel level are employed, because the models generate continuous probability scores $p(x) \in [0,1]$, a binarization step is necessary to delineate change regions. This requires a threshold τ , where pixels exceeding this value are labeled as changes. Instead of setting τ to a fixed value, as this can negatively impact, the threshold is determined empirically through a search over the range $[0,1]$ in increments of 0.01. The threshold that yields the highest F1-score is selected as optimal. Based on this optimized mask, standard performance indicators are calculated: Precision, Recall, F1-score, and IoU. By tailoring the threshold to the data, there is no subjectivity associated with manual selection, facilitating a fair comparison of model architectures and training configurations.

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

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

$$F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$IoU = \frac{TP}{TP + FP + FN}$$

3.6 Implementation

The technical implementation of this study is based on a hybrid infrastructure in which various workflows were distributed between local servers for database hosting, a workstation for data management, and more powerful cloud-based hardware components for generative diffusion models and CD model training. The following sections describe in detail the hardware specifications, the software used, and the algorithms developed exclusively for this study for data synthesis in CD.

3.6.1 Technical Infrastructure

The task was distributed across three environments depending on the specific resource requirement of the individual tasks. Since Flux.dev.1 alone requires approximately 34 GB to load the checkpoint into the video RAM, a Google Colab Pro runtime environment had to be used. This runtime environment provides 40 GB of VRAM with the NVIDIA A100 configuration. When using 10 LoRAs in the dataset generation phase, out-of-memory errors could still occur, so the machine had to be monitored continuously during synthetic-dataset generation.

In contrast, data processing was performed locally on two different workstations. Managing the HU data set involves processing many millions of vector (shape) features, so CPU time with lower RAM capacity is the most costly factor. For the preparation of the HU data, a database is hosted on a local server that runs day and night anyway (Intel i5-4590T, 16 GB RAM). This setup allowed for low-latency SQL queries independent of internet bandwidth. A separate workstation handled visual validation via QGIS and executed the Python scripts for WMS retrieval.

Data transfer between locally generated data and Google Colab was implemented using Google Drive. While Google Drive allows direct integration into the Colab environment, writing directly to the Google Drive directory proved to be too slow, creating a significant I/O bottleneck. To load training examples quickly, the data was compressed into a single ZIP archive for transfer and then extracted directly onto the local Colab disk. This approach ensured significantly faster data loading times during the training process.

3.6.2 Software frameworks used

The code is written in Python 3.10 and is based on PyTorch. The generative part of the pipeline uses the HuggingFace Diffusers library. On the data engineering side, no standard GIS desktop software was used, as QGIS and ArcGIS struggled with the size of the dataset. Instead, pg8000 was used for direct database interactions and Shapely for geometric operations such as centroid calculations.

To ensure reproducibility, the complete technical framework developed for this work is hosted in a dedicated Git repository. This codebase includes the entire pipeline, from initial SQL-based data management and preprocessing to the generative image synthesis framework and the implementation of all architectures for change detection models. The repository is available at: <https://github.com/iProbi/synthetic-training-data-for-changedetection.git>

3.6.3 Codebase structure, configuration, and run management

The implementation follows a modular pipeline structure based on the methodological steps outlined in Chapter 3.2. Specifically, the code is divided into (i) data processing and HU filtering, (ii) image retrieval (WMS), (iii) mask construction and rasterization, (iv) inpainting-based synthesis, (v) training for change detection, and (vi) evaluation.

To avoid errors in the parameters of the experiments (E1-E4), each trained CD is stored in its own folder, with all applied parameters included in the folder name to prevent confusion. A table with training statistics (validation loss, training loss, learning rate) is stored in this folder. The validation process then saves examples of the ground truth evaluation in the same folder, along with a table containing the results of the metrics. For each model, the last and best models are saved separately from the training. Synchronization with the cloud allows each epoch of training to be restored via version control.

4 Results

4.1 Experiment E1: Diffusion Model comparison

4.1.1 Objective and Setup (E1)

Experiment E1 isolates the influence of the generation diffusion model and conditioning strategy on the training of CD DeepLabv3 with Resnet50 backbone. All models are trained on semi-synthetic datasets generated using the same logic. The generation of synthetic images aims to delete all buildings in the area based on a mask with a buffer of 1 meter (see 3.2.1). Each dataset has a size of 5000 samples with a resolution of 512 x 512 pixels. Only the generator configuration differs: SDXL inpainting, FLUX inpainting, FLUX + generalized LoRA, and FLUX + randomized prompts + fitted LoRAs (see Table 2).

4.1.2 Training dynamics and convergence (E1)

Figure 3 compares the training and validation loss curves for the four generator settings. Convergence is summarized by (i) the epoch with the best validation performance and (ii) the stop epoch under early stopping. This provides contextual information on optimization stability and potential overfitting under each synthetic data variant.

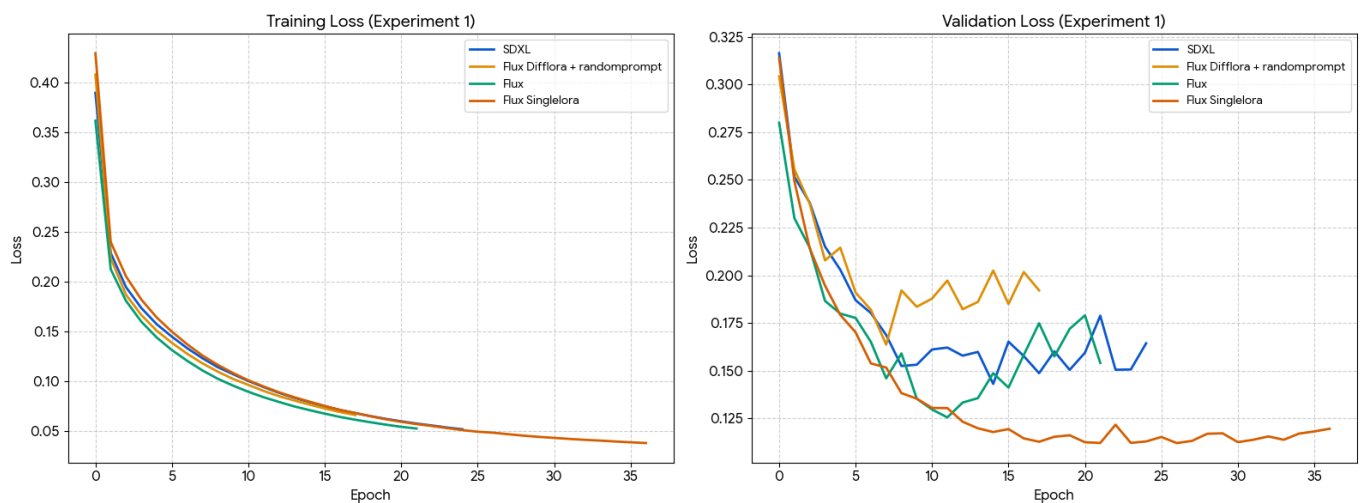


Figure 3 Validation loss (BCEWithLogitsLoss) across epochs for the four generator settings (backbone deeplabv3 + Resnet50, no augmentation, learn rate start 0.00001).

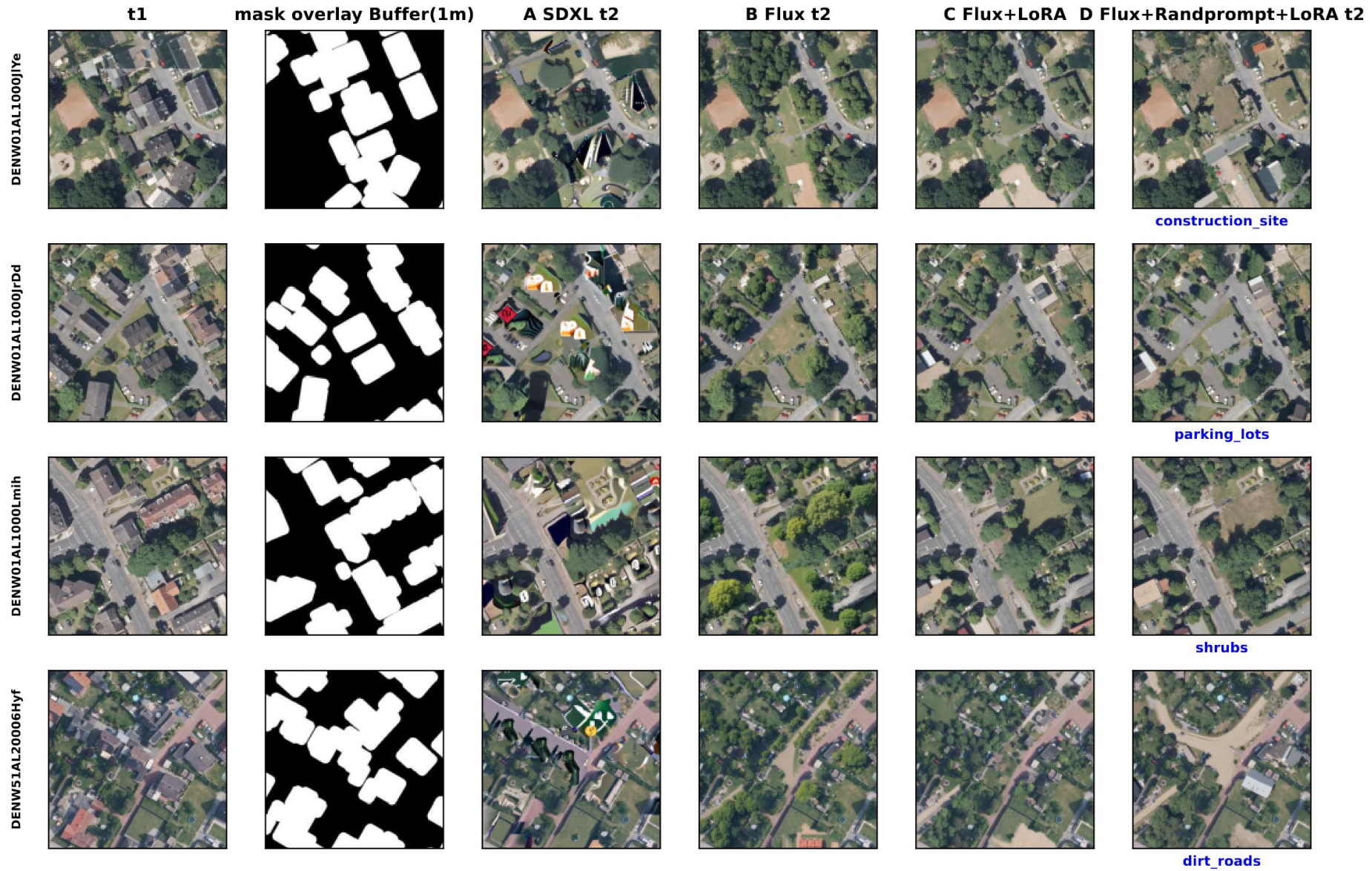
Table 5 E1: convergence summary

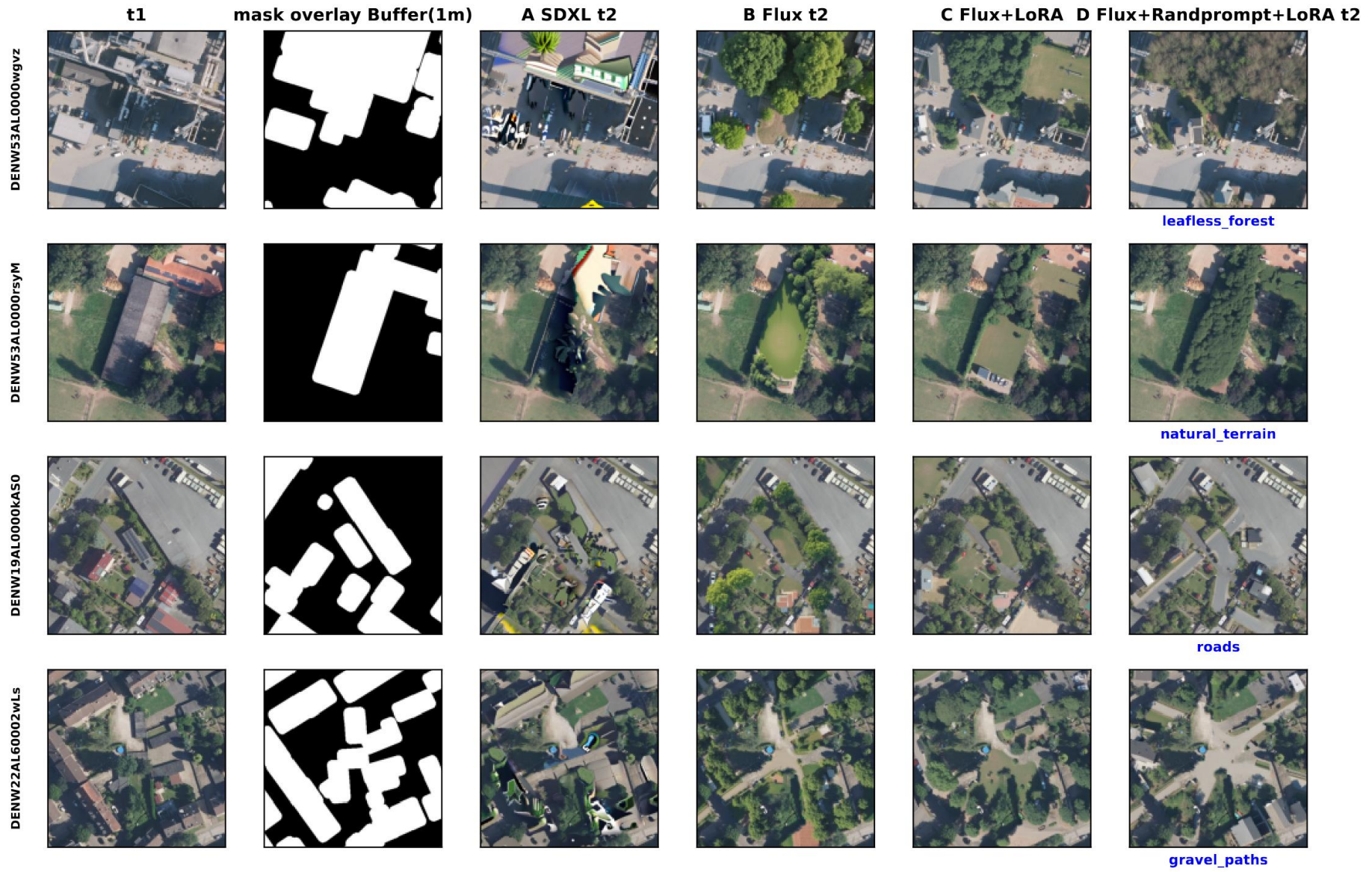
Model	Best validation loss	Best training loss	best epoch
A SDXL	0.1431	0.0518	15
B Flux	0.1255	0.0525	12
C Flux SingleLoRA	0.1121	0.0380	27
D Flux DiffLoRA + randomprompt	0.1636	0.0661	8

Across E1 runs, models A, B, and D reached the best validation performance between epoch 8 and 15. A different learning behavior was observed for C, which showed a longer, more homogeneous learning phase. This can indicate overfitting to the training data.

4.1.3 Qualitative generation quality (E1)

Prior to evaluating the transferability to real-world data, the generated t_2 images were visually inspected to evaluate (i) contextual consistency within the masked region, (ii) seam artefacts along the mask boundaries, and (iii) the diversity and plausibility of non-building textures.





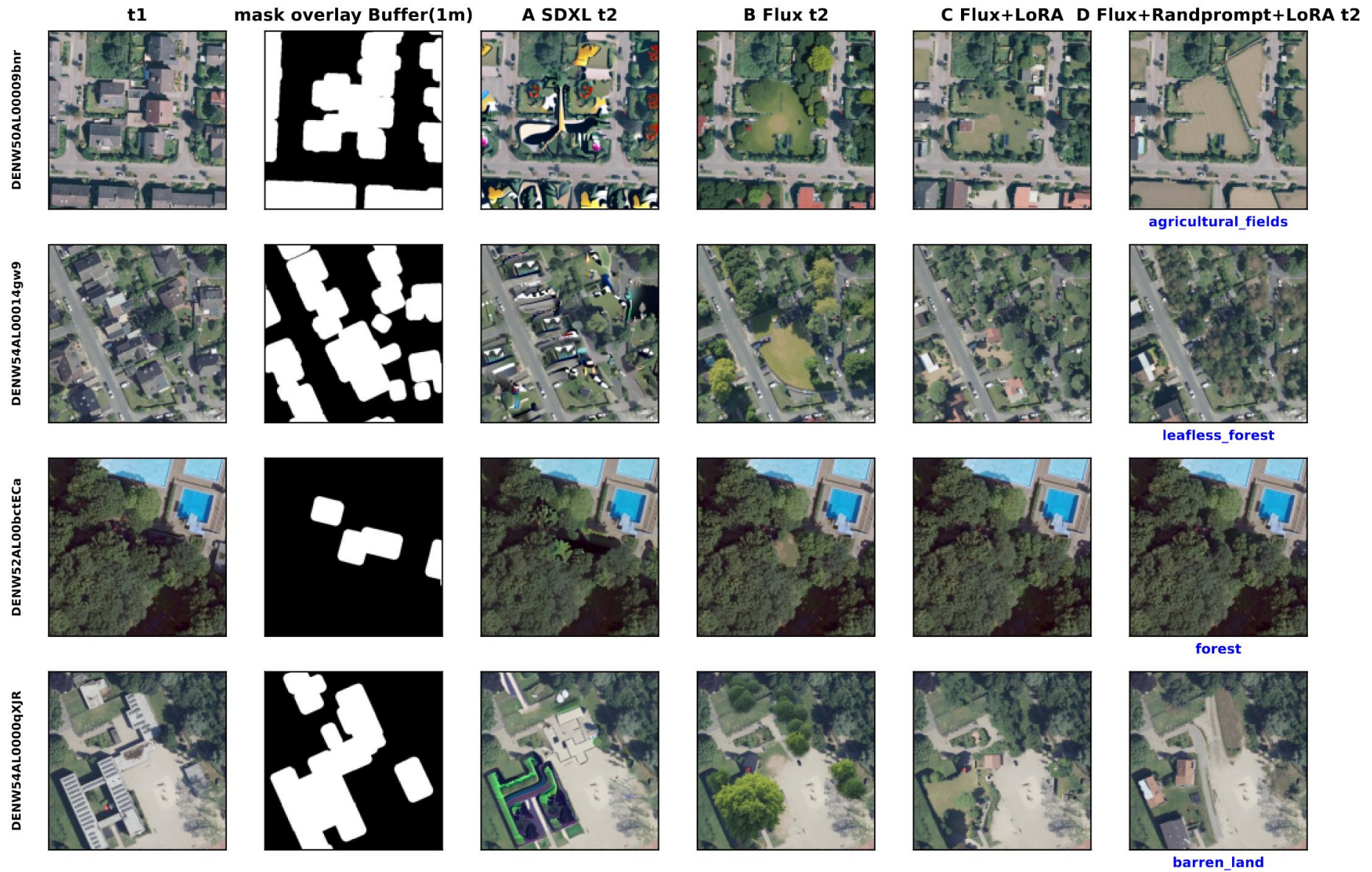




Figure 4 Qualitative comparison of synthetic change generation in Experiment 1. The vertical labels on the left indicate the ALKIS ID as a spatial reference. Original T1; target area mask; SDXL (E1-A) with illogical structures; Flux (E1-B) with extremely vivid textures; Flux + LoRA (E1-C) with realistic but monotonous results; Flux + random prompts/LoRAs (E1-D) with both realism and diversity. The labels under column indicate the specific target class (e.g., “gardens,” “agriculture fields”).

Qualitatively, SDXL (A) tends to produce artifacts (e.g., exaggerated unrealistic textures), particularly visible in ALKIS scene “DENW50AL00009bnr” (see Figure 4: agriculture fields from DiffLoRA). Flux (B) produces sharper, more coherent textures with much fewer boundary seams over all image samples. Adding the generally trained LoRA to the Flux inpaint model (C), reduces the vivid colors of vegetation in all situations and blends in better with the context of the aerial image. Generator type D increases the diversity of the samples by generating specific objects into the given patterns while mostly respecting the non-building constraint. In summary, Flux inpaint with randomized prompts and randomized LoRAs (D) offer the largest variance in the generations, while the generations produced by the model with SingleLoRA (C) appear the most realistic.

4.1.4 Quantitative generalization on ALKIS validation dataset (E1)

Using semi-synthetic change detection datasets generated by different variants of the generator family and scenarios, change detection models were trained to evaluate the impact of synthetic data quality. The results are indicated in Table 6.

Table 6 Performance metrics by generator on ALKIS validation set in Experiment 1

Model	Threshold	IoU	F1-Score	Precision	Recall	Accuracy
D Flux DiffLoRA + randomprompt	0.95	0.4186	0.5585	0.6299	0.6307	0.9484
A SDXL	0.89	0.3640	0.5018	0.5459	0.6552	0.9350
B Flux	0.87	0.3391	0.4801	0.5266	0.6054	0.9351
C Flux SingleLoRA	0.35	0.3200	0.4556	0.4677	0.6196	0.9271

Table 6 reports on the zero-shot inference performance of each generator setting in training of the DeepLabv3 with ResNet50 Backbone in CD on real-world ALKIS HU dataset. The results show a clear ranking (see Table 6) across the four settings. Configuration (D) achieves the strongest overall performance, with F1 = 0.559 and IoU = 0.419 at an optimal threshold of $t = 0.95$. SDXL (A) follows with F1 = 0.502 and IoU = 0.364, showing the highest recall among all variants (0.655) but lower precision (0.546) compared to (D), which indicates more false positives while maintaining similar sensitivity. The other Flux configurations (A, C) showed limited performance, especially the general LoRA setup (C) produces much lower optimal threshold $t = 0.35$, which indicates a much more confident decision between changed and not changed pixels. Overall, setup D provides the best tradeoff between precision, recall on ALKIS.

In order to correlate the quantitative values with actual behavior, Figure 5 compares the predicted change maps with the actual values.

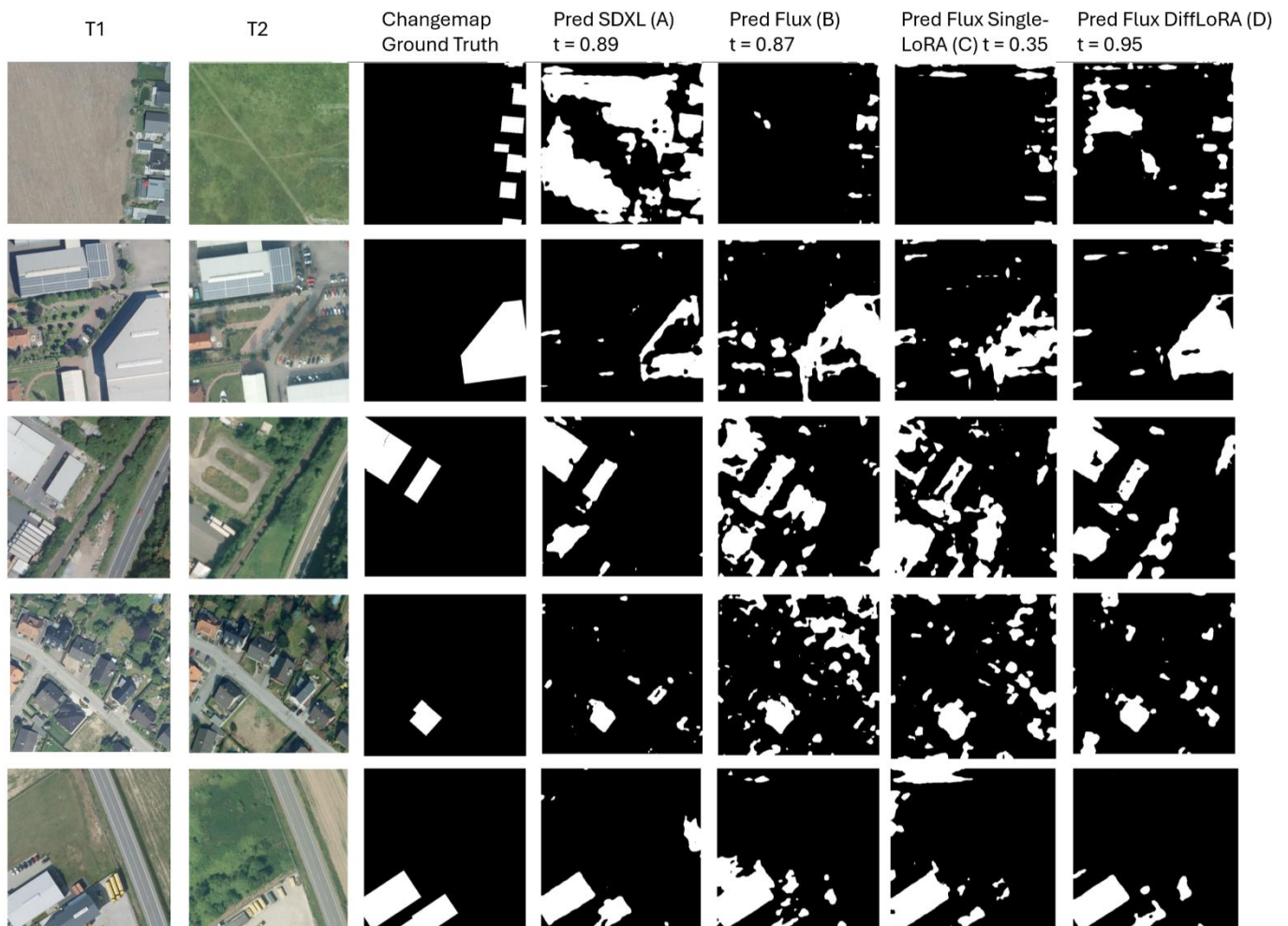


Figure 5 Qualitative comparison of the prediction masks generated by the four model configurations (A to D) using the real ALKIS validation dataset. The layout is structured to show the multi-temporal input sequence consisting of T1 and T2, followed by the reference ground truth change map.

A qualitative analysis of the prediction masks shows different performance characteristics in the different configurations. Model A shows inconsistent behavior, producing good results in some cases but failing completely in others due to extremely high false positive (FP) rates. Model B consistently produces fragmented predictions for all scenes. Although it identifies most of the actual changes, the masks lack structural density compared to Model A. This fragmentation is even more prominent in model C, which has an even higher false negative rate (FN) within the changed areas. Model D delivers the most robust results, characterized by the lowest FN rate and the highest geometric accuracy in detecting change boundaries. Although Model D still generates false positive results, it is more stable than Model A, as it avoids extreme outliers and displays better structural consistency. Overall, however, none of the models provide reliable results in their predictions.

4.2 Experiment E2: Impact of data augmentation

4.2.1 Objective and setup E2

Experiment E2 isolates the influence of augmentation strength on the transferability of the change detection model to real data. Building on the results of the previous phase, the generator configuration is set to the most advanced variant: Flux Inpainting in combination with random prompts and fitted LoRAs. Only the intensity of radiometric and geometric data augmentation differs between test runs and is divided into three levels: no augmentation, weak augmentation, and high augmentation.

4.2.2 Training dynamics E2

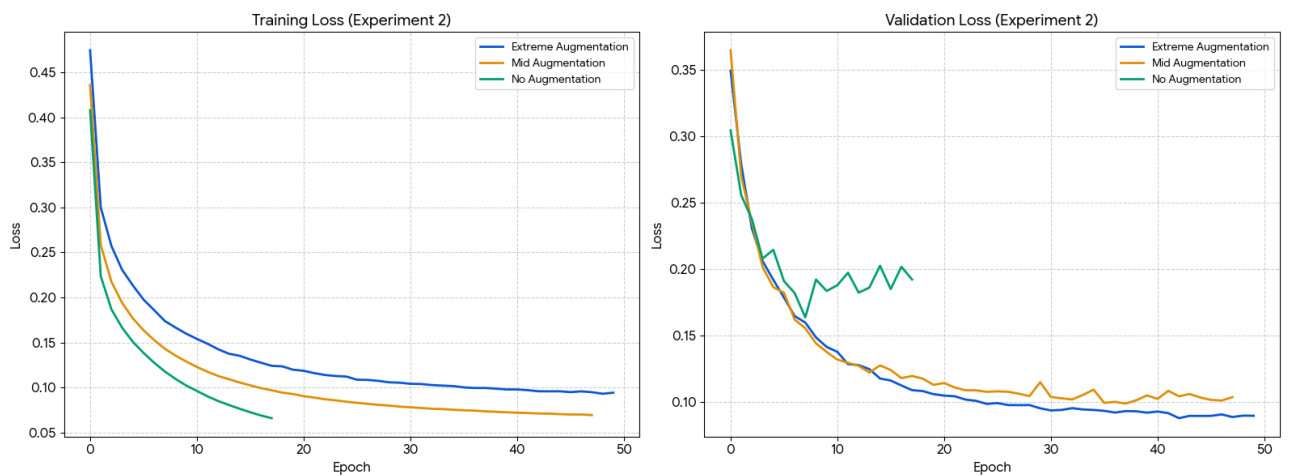


Figure 6 E2: Training and Validation loss vs epoch

Augmentation changes convergence and stability: The ‘strong’ setting leads to lower variability in validation loss and smaller generalization gaps. In this case, the higher the augmentation of the training data, the more robust and less overfitted the learned model seems to be.

4.2.3 Qualitative augmentation review E2

Table 7 Random ‘No Augmentation’ example




T1	T2	Change Label
		

Table 8 Random 'Strong Augmentation' example






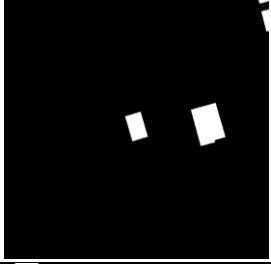


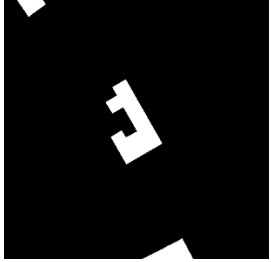





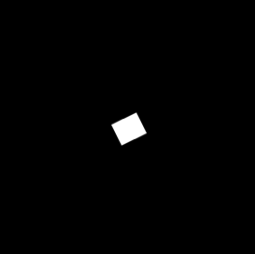


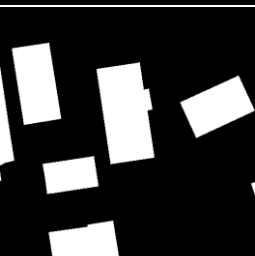
T1	T2	Change Label
		
		
		

Table 9 Random 'weak Augmentation' example

T1	T2	Change Label
		
		
		

Qualitatively, the No Augmentation baseline (see Table 7) maintains clear, noise free textures with consistent textures around the changes between t_1 and t_2 . The weak Augmentation (see Table 9) introduces moderate photometric shifts, while synthetic noise and slight color variations are visible, the weak augmentation level delivers a near real color shift between seasons for the human eye. In contrast, strong Augmentation (see Table 8) creates severe radiometric distortion. High magnitude brightness and contrast changes can lead to near black- or white-out scenarios. While strong color changes almost destroy texture coherence, most structural boundaries are still recognizable.

4.2.4 Quantitative generalization on ALKIS validation dataset (E2)

Using the best generator configuration determined in Experiment 1, this section examines how different levels of data augmentation affect model performance on real ALKIS data. The quantitative results are shown in Table 10, and predictions are shown in Figure 7.

Table 10 E2: ALKIS results

Configuration	Threshold	IoU	F1-Score	Precision	Recall	Accuracy
Strong Augmentation	0.86	0.5436	0.6815	0.7118	0.7016	0.9695
Weak Augmentation	0.88	0.4581	0.6011	0.6591	0.6561	0.9518
No Augmentation	0.95	0.4186	0.5585	0.6299	0.6307	0.9484

Strong augmentation significantly improves performance. The IoU increases from approximately 0.419 (no augmentation) to 0.544. This confirms that the model learns to generalize much more robustly to the test data through the massive image changes in training.

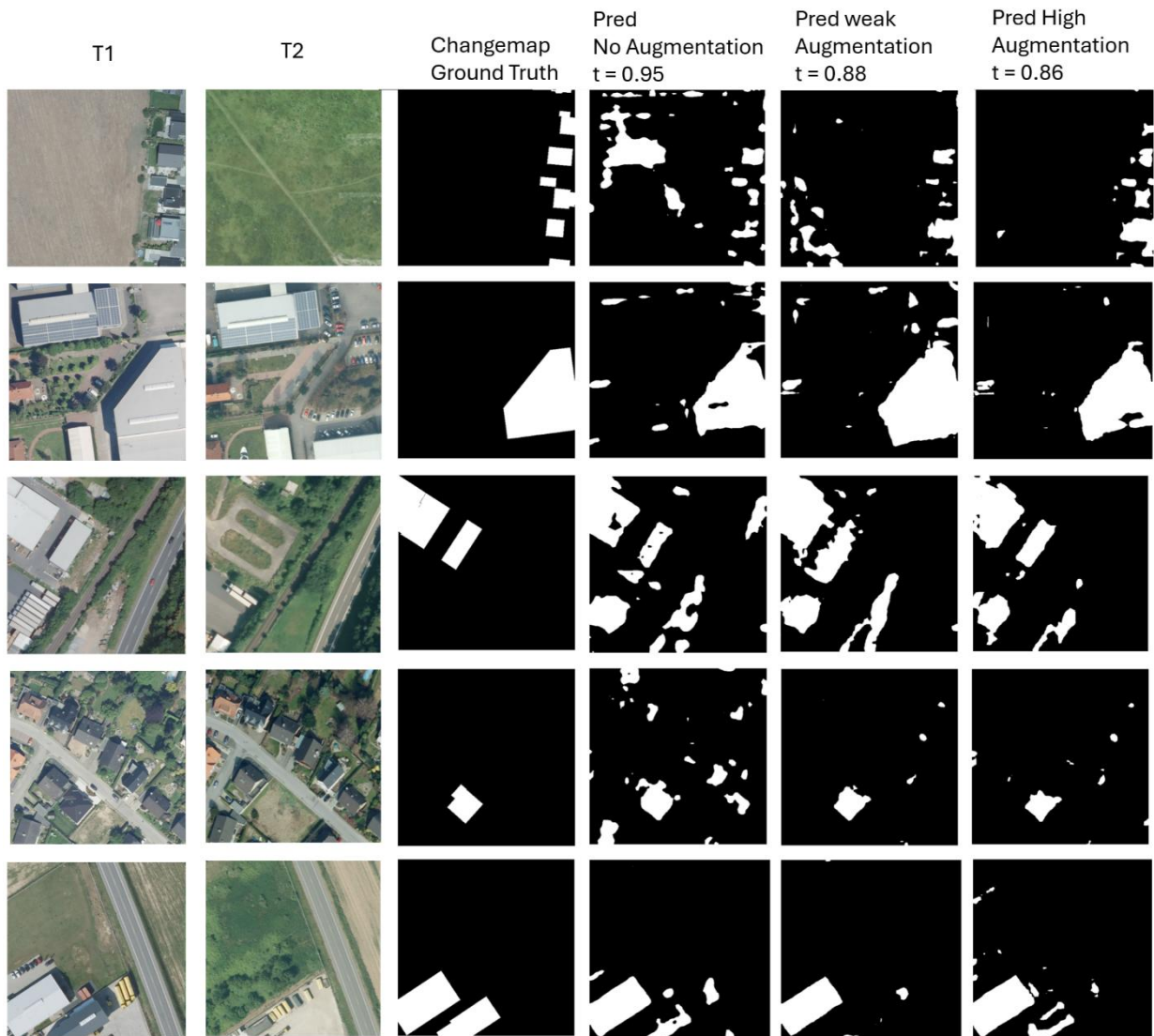


Figure 7 Qualitative comparison of the prediction masks generated with the different augmentation levels using the real ALKIS validation dataset. The layout is structured to show the multi-temporal input sequence consisting of T1 and T2, followed by the reference ground truth change map and the predictions by the augmentation levels no, weak, and strong augmentation from left to right.

By comparing augmentation levels in predictions, it becomes apparent that the fragmentation of the prediction decreases steadily with increasing augmentation on the training data, and fewer False Positive (FP) predictions appear.

4.3 Experiment E3: Model architecture comparison

4.3.1 Objective and setup (E3)

Experiment E3 evaluates whether the observed data effects persist across model families. The generator and augmentation are fixed to Flux Inpainting + random prompt + fitted LoRA and strong augmentation. The CD backbone is varied: VGG-based encoder-decoder, Deeplabv3-ResNet50 and Deeplabv3-ResNet101. The learning rate and batch size must be adjusted during the experiment to optimize each change model to a similar level (see Table 11).

Table 11 E3 configurations (Backbone, Batch size, Learn rate)

Backbone	Batch size	Learn rate
VGG-based-encoder-decoder	28	0.0001
Deeplabv3-ResNet50	15	0.00001
Deeplabv3-ResNet101	9	0.00002

4.3.1 Training dynamics and convergence

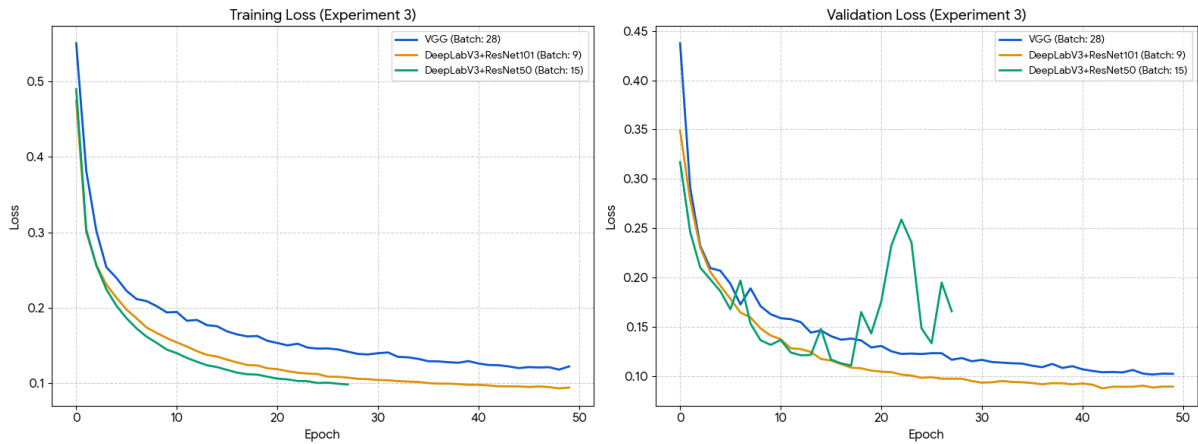


Figure 8 E3: Training and Validation loss vs epoch

Table 12 E3: convergence summary

Backbone	Best Validation Loss	best training Loss	Best epoch
VGG-based-encoder-decoder	0.1213	0.1019	43
DeepLabv3 + ResNet50	0.1111	0.0983	18
Deeplabv3 + ResNet101	0.0896	0.0943	47

Figure 8 summarizes the training and validation loss curves for the three backbones under experiment settings (Table 11). The VGG based and Deeplabv3 with ResNet101 Backbone show smooth and stable convergence over the full training until the stop epoch (50), reaching their best validation loss rate late at epoch 43 and 47 (Table 12). In contrast, the ResNet50 backbone reaches the best validation loss at epoch 18, followed by a clear deterioration in validation performance. Overall, E3 indicates that backbone choice affects optimization stability and the risk of overfitting on semi-synthetic training data, which provides important context for the result report in ALKIS generalization in the next section.

4.3.2 Quantitative generalization on ALKIS validation dataset (E3)

Based on the best-performing configuration from the previous experiments, Experiment 3 evaluates the influence of the backbone architecture. By comparing different backbones, this phase determines the optimal structural balance between model size and generalization on real data. The performance metrics are listed in Table 13.

Table 13 E3: ALKIS results by backbone

Backbone	Threshold	IoU	F1-Score	Precision	Recall	Accuracy
VGG-based-encoder-decoder	0.97	0.2518	0.3745	0.3600	0.4827	0.9226
DeepLabV3+ResNet50	0.86	0.5436	0.6815	0.7118	0.7016	0.9695
DeepLabV3+ResNet101	0.77	0.4055	0.5404	0.5811	0.6946	0.9303

DeepLabV3 with a ResNet-50 backbone achieves the best overall performance, reaching F1 score of 0.6815 and IoU of 0.5436 at an optimal threshold of $\tau = 0.86$. This configuration also shows the highest precision (0.7118) while maintaining a balanced recall (0.7016), indicating a favorable trade-off between false positives and missed changes. DeepLabV3-ResNet101 performs noticeably weaker, with F1 = 0.5404 and IoU = 0.4055 at $\tau = 0.77$. Although its recall remains relatively high (0.6946), lower precision (0.5811) suggests an increased false-positive tendency compared to ResNet-50. The VGG-based encoder-decoder yields the lowest scores, with F1 = 0.3745 and IoU = 0.2518 at a very high threshold ($\tau = 0.97$), and it also shows the lowest recall (0.4827), indicating that many true change pixels are missed under this setting.

4.4 Experiment E4: Comparison of alternative strategies

4.4.1 Objective and setup (E4)

Experiment E4 compares the inpainting-based pipeline against alternative synthetic change generation methods based on mono-temporal imagery, including Self-Pair variant from Seo et al. (2023) and Unpair (Seo et al. 2023). In addition, the “No-Pair” option uses a model that has been trained exclusively for mono-temporal building detection. To evaluate change detection, this model generates separate building masks for T1 and T2. These prediction masks are then compared using mask subtraction to generate the final change predictions. Backbone is fixed to DeepLabv3 + ResNet50.

4.4.2 Training dynamics and convergence

Figure 9 shows the training and validation loss curves for the five different preprocessing and pairing strategies in Experiment 4. The learning process is characterized by varying degrees of stability and convergence speeds, which depend on the method used to

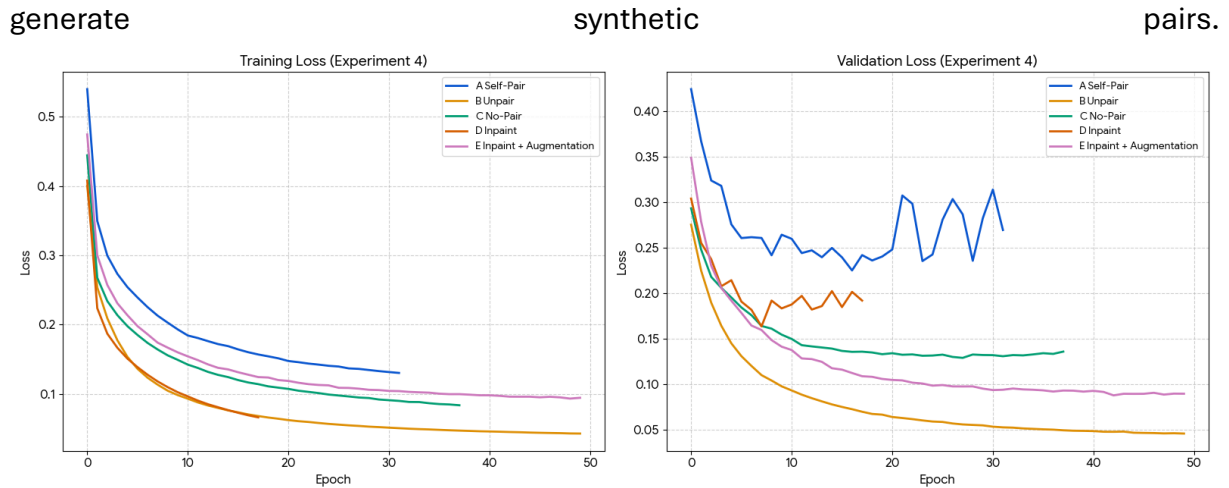


Figure 9 E4: Training and Validation loss vs epoch

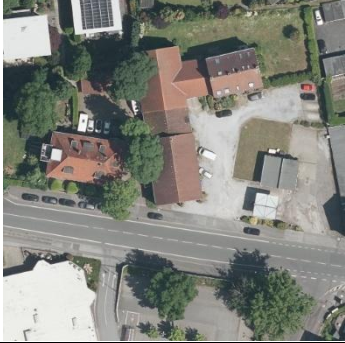


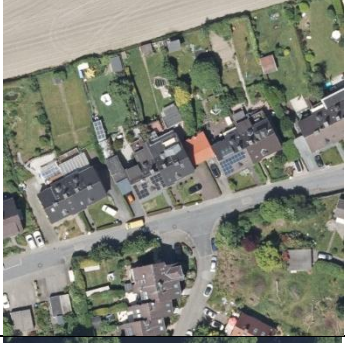
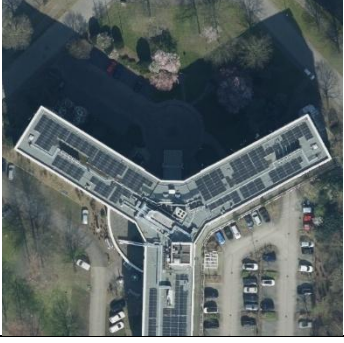

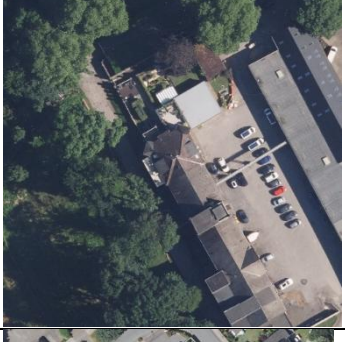
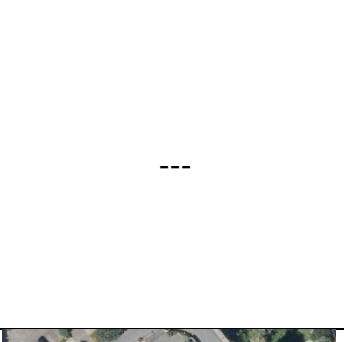
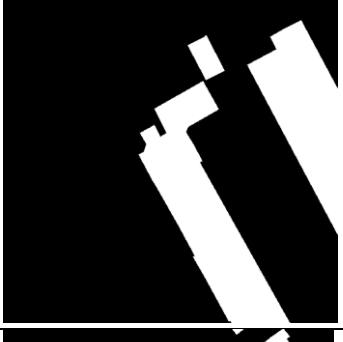
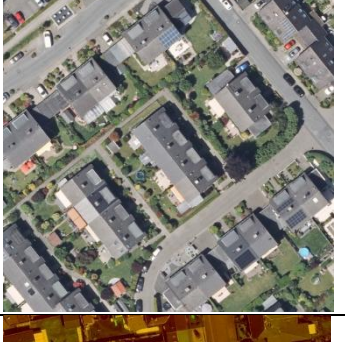
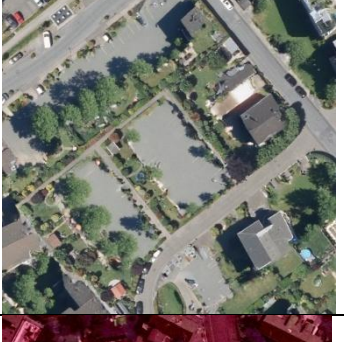
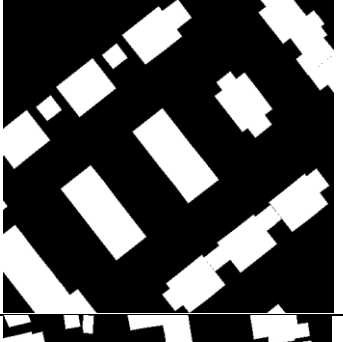


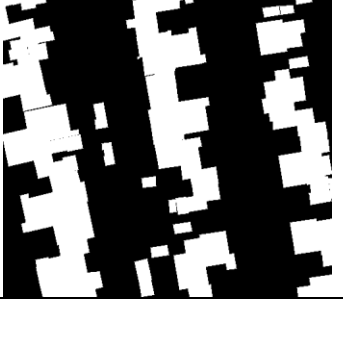
Table 14 E4: convergence summary

Model	Best validation loss	Best training loss	Best epoch
A Self-Pair	0.2252	0.1302	17
B Unpair	0.0459	0.0428	50
C No-Pair	0.1291	0.0836	28
D Inpaint	0.1636	0.0661	8
E Inpaint + Augmentation	0.0879	0.0932	43

In this experiment, Model B (Unpair) and Model E (Inpaint + Augmentation) demonstrated the highest optimization stability, both utilizing the full 50 epochs (Figure 9). While Model B achieved the overall lowest validation loss (Table 14). In contrast, Model D (Inpaint) reached its peak validation performance very early (epoch 8), followed by a rapid onset of overfitting, as indicated by the diverging validation curve. Model A (Self-Pair) showed the most volatile behavior with the highest overall loss, suggesting that learning the change features from identical images with only geometric transformations presents the most challenging optimization task. The performance of Model C (No-Pair) is about average in all respects.

4.4.3 Qualitative comparison of synthetic-change strategies

Table 15 Comparative analysis of qualitative synthetic change processing strategies

ID	Synthetic Strategy	T1	T2	Change Label
A	Self-Pair			
B	Unpair			
C	No-pair			
D	Inpaint			
E	Inpaint + strong Augmentation			

From a qualitative perspective, the strategies differ significantly (see Table 15). Self-Pair (A) offers realistic radiometric consistency due to identical recording conditions but does not produce plausible structural changes. Unpair (B) represents similar but stronger: it offers highly realistic radiometric diversity by pairing unrelated real images, but here too, logical structural transitions are lacking. No-Pair (C) provides the most accurate and realistic mask, but suffers from a lack of intra-pair augmentation, which limits the overall training diversity.

The Inpaint strategy (D) manages to produce logical structural changes, such as the coherent addition or removal of buildings, but remains susceptible to radiometric shifts in unmodified areas due to a lack of diversity, as with the No-Pair method (C). This gap is closed in model E (inpaint + augmentation). Although the strong augmentations in model E can occasionally reach a degree of visual unreality, they compensate for the lack of natural bi-temporal variance with a lack of realism.

4.4.4 Quantitative generalization on ALKIS validation dataset

Using the best configurations established in the previous experiments, change detection models were trained to evaluate the impact of different image synthesis strategies. This final experiment compares structural inpainting techniques with combinatorial pairing methods to identify the most effective data generation logic. The results are indicated in Table 16.

Table 16 E4: ALKIS results by strategy

Technique	Threshold	IoU	F1-Score	Precision	Recall	Accuracy
A Self-Pair	0.67	0.4706	0.6010	0.6388	0.6820	0.9559
B Unpair	0.99	0.5998	0.7276	0.7855	0.7135	0.9739
C No-Pair	0.65	0.3748	0.5099	0.4292	0.7515	0.9342
D Inpaint	0.95	0.4186	0.5585	0.6299	0.6307	0.9484
E Inpaint + Augmentation	0.86	0.5436	0.6815	0.7118	0.7016	0.9695

The quantitative test (Table 16) shows the highest results for the B Unpair strategy in terms of IoU (0.60) and F1-Score (0.73). This suggests that the high spectral diversity and hard edges of the building masks from pairing completely different images ensure that the CD model learns most reliably to ignore all other natural influences. Model E (Inpaint + Augmentation) follows as the second most efficient strategy with its more logical and plausible structures but strong augmentation, which leads also to a high robustness. Model C (No-Pair) has the least complex strategy as baseline shows the lowest scores, likely due to its inability to teach the model radiometric shifts without augmentation and limited training data. The Self-pair-strategy (A) offers similar precision (see Table 16) level as D Inpaint, while scoring much higher in F1-Score and IoU, which indicates a much

higher level of confidence on the part of A, which is also reflected in the threshold t (0.67 A vs 0.95 D).

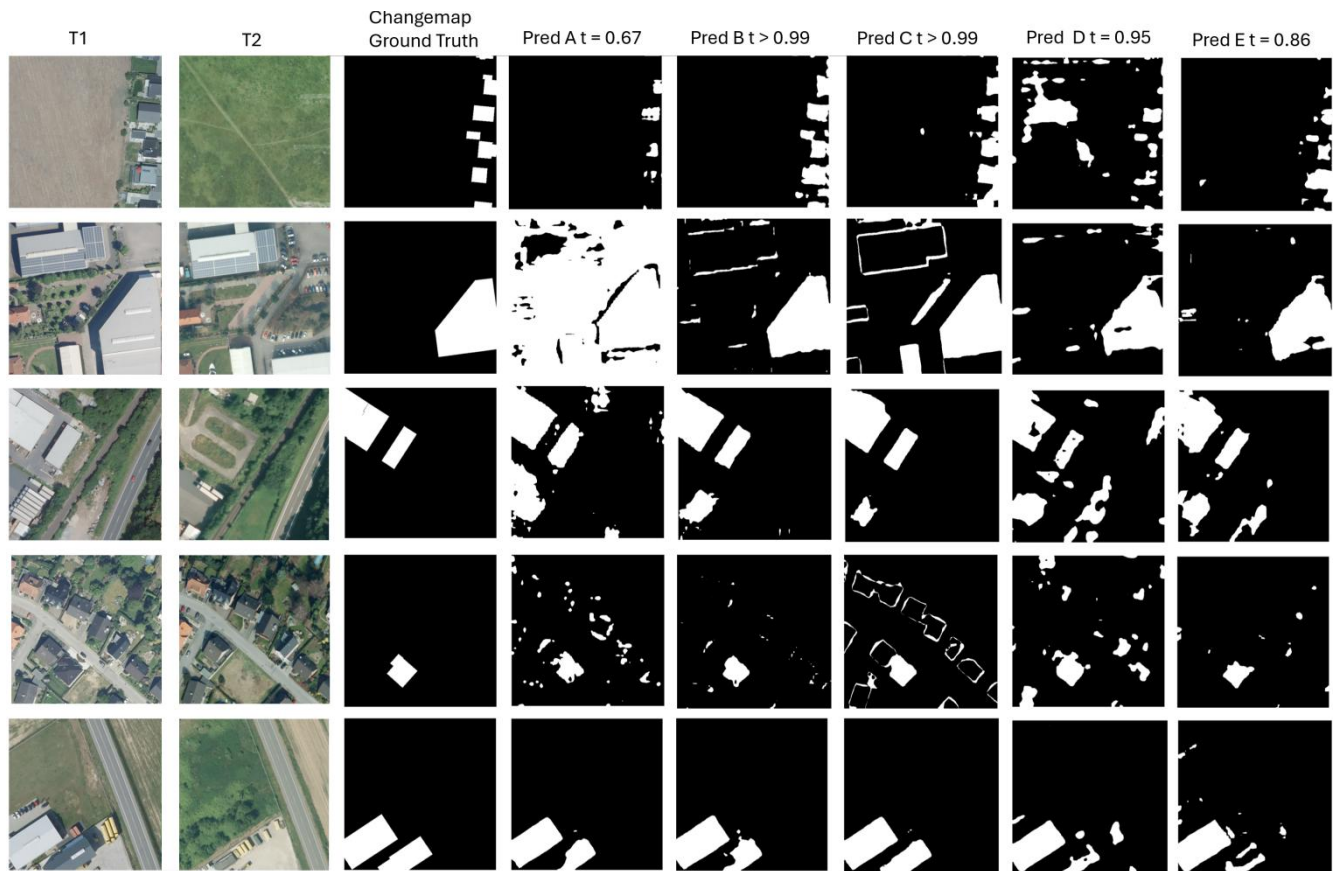


Figure 10 Qualitative comparison of the prediction masks generated by models trained with different systematically changed training data. All models are validated on the real-world ALKIS validation dataset. The layout is structured to show the multi-temporal input sequence consisting of T1 and T2, followed by the reference ground truth change map and the predictions by the synthetic-change strategies ordered from A to E (see Table 4).

A qualitative comparison of the predictions made by the training strategies reveals characteristic patterns in the change predictions. The model trained with strategy A (Self-Pair) shows a mostly inconsistent estimate, varying from conservative predictions to an almost completely false positive (FP) failure. Model B, (Unpair) provides the most complete change masks, as expected from Table 16. However, this strategy also leads to some false positive (FP) results, while all actual changes (TP) are almost completely captured. In contrast, model C (No-Pair) shows the best accuracy for large building areas and provides the most solid filling, but fragmented false positive (FP) areas arise due to regular surrounds around unchanged buildings. Strategy D (Inpaint) produces the least sharp change boundaries of all variants, with noticeable soft edges and several small round artifacts. Finally, Model E (Inpaint with strong augmentation) shows hardly any shadows of unchanged buildings in the examples and reacts similarly sensitively to small buildings as strategy E. However, small gaps sometimes appear within buildings (FN).

5 Discussion

The main target of this thesis was to evaluate synthetic image data generation for the training of deep learning-based CD models in the context of the North Rhine-Westphalia cadaster (ALKS). The results of the experiments E1 - E4 show a complex synergy between all four tested variables and logics, which are covered by the experiments. In this section, the results will be discussed.

5.1 Base generator effects on diffusion models (Experiment 1)

Experiment 1 (E1) evaluated the suitability of two Diffusion Models (SDXL and Flux) with various LoRA and prompt configurations (SingleLoRA, DiffLoRA) for image synthesis. By generating in native 512 x 512 pixel, in all configurations, the Flux solution showed better visual performance over SDXL. This is attributed to better coherence in image quality and generation of more realistic synthetic changes. This test showed that not only is the coherence of the generated images important for training CD models but also the variation has wider importance. While the SingleLoRA variant generated the most realistic images for the human eye, the CD model trained with this dataset achieved limited performance when validated with the real-world change dataset.

The changes that were most unrealistic for humans resulted from the SDXL workflow, but these were the second best in the quantitative test (section 4.1.4). Only by varying the prompts and LoRAs, the learning process was improved while maintaining a high level of coherence. Overall, it can therefore be confirmed that semantic meaning and variation are equally important to create “safe” augmentation (Shorten and Khoshgoftaar 2019), where the physical coherence of the change is preserved while providing enough variation to the learning model.

However, all these methodologies are missing a critical point. All these images are nearly untouched in comparison to the initial image. This increases the risk that a model will not initially learn where changes have taken place but will recognize where no changes have taken place. This will relate to a common problem, shortcut learning (Geirhos et al. 2020) and overfitting, where the model learns static training data instead of the underlying change mechanism (Shorten and Khoshgoftaar 2019).

5.2 Impact of augmentation (Experiment 2)

While experiment 1 (E1) established a baseline, Experiment 2 (E2) investigated the influence of data augmentation. To counteract the risk of training in unchanged areas (see section 5.1), a comparison was made here to determine the extent to which different image augmentations are quantitatively beneficial to CD training. The randomness in augmentation of the training data counteracted any systematization of augmentation in the training. It turned out that transferability to real data worked best with completely unrealistic levels of radiometric abstraction. Samples in CD training that were partially

barely recognizable in the “strong” augmentation setting to the human eye led to the best results.

This aligns with the concept of domain randomization (Tobin et al. 2017). While E1 showed that physical and structural coherence is necessary during generation, E2 demonstrates that radiometric coherence must be viewed separately. Training with highly to non-realistic textures in case of radiometric augmentation forces the model to ignore color domain shifts and focus on structural differences (Tobin et al. 2017). A CD model trained with a combination of the SingleLoRA dataset (worst in E1) in with “strong” augmentation again led to superior results compared to the DiffLoRA dataset (best in E1) (F1 score 0.697 vs 0.682 respectively). This shows that too much variance can also lead to worse results in terms of transferability to the real-world.

5.3 Influence of Network Architectures (Experiment 3)

Experiment 3 focused on the influence on how change feature extraction is embedded in model architecture. The direct comparison showed that ResNet-based architectures (ResNet50/101) in DeepLabv3 performed significantly more robustly than VGG backbones under strong augmentation. This result is consistent with the analyses of Zhang et al. (2020). The authors show that VGG has an extremely high fitting ability due to its small 3×3 convolution kernels and performs particularly well in the extraction of large, homogeneous areas. However, precisely because of this adaptability, this architecture tends to overfit to the superficial artefacts of diffusion models when used with artificial data. The identity shortcut connections introduced by ResNet, on the other hand, allow for a deeper, more abstract semantic representation without falling down to the vanishing gradient problem (Zhang et al. 2020).

More importantly, Zhang et al. (2020) also noted that ResNet architectures have a superior ability to extract small, complex objects. For the present ALKIS scenario, this ability was crucial for mastering the transfer to the highly noisy real data and for reliably detecting even small building outlines regardless of their texture.

Interestingly, when comparing ResNet50 and ResNet101, the backbone with fewer parameters performs better than the other. ResNet101 probably tends to overfit, which can be seen in the low validation and training loss in the training dynamics (see section 4.4.2). This is counterintuitive but can be partly explained by the smaller dataset size. As already mentioned, highly complex models needing massive amounts of data to avoid overfitting (Shorten and Khoshgoftaar 2019). A much larger dataset would probably benefit from the more complex backbone.

5.4 Comparison to non-diffusion image synthesis strategies (Experiment 4)

Experiment 4 compared the generation of logical structural changes (Inpaint) with pure radiometric diversity (Unpair). Although Model B (Unpair) delivered the best result quantitatively with an IoU of 0.60, this may be possible due to a massive combinatorial advantage. The random pairing of 5,000 real image sections during training resulted in unique radiometric combinations. This nearly infinite variance in sensor noise and lighting differences acts as a perfect domain randomization mechanism (Tobin et al. 2017). As Shorten and Khoshgoftaar (2019) point out, deep learning models are relying on big data to avoid overfitting. Model B (Unpair) successfully simulates this amount of data.

The Inpaint strategies (models D and E) (see Table 14) on the other hand, were limited to 5,000 fixed pairs due to the high inference costs of the diffusion models. If it were computationally possible to generate Inpaint images “on the fly” during training, this method could potentially outperform Unpair, as it combines coherent structural logic with necessary radiometric diversity.

The inpainting methods are still in their early stages, and augmentation could also be extended to a diffusion method, allowing coherent logic in changes to be combined with realistic augmentation if necessary. In addition, there is still potential in terms of computing power and costs, and the associated dataset size, which future work must prioritize to fully unlock the potential of generating semi-synthetic datasets.

5.5 Threshold dynamic and class imbalance

One methodological finding is the strong variation in optimal thresholds, which fell from nearly 1.0 for Unpair (see 4.5.3) to 0.35 (see 4.2.3) for SingleLoRA. This is a symptom of extreme class imbalance in change detection and overfitting to training data. As discussed in section 5.1, the SingleLoRA model likely overfitted to the static background, leading to low confidence when predicting real-world changes. Consequently, the best possible threshold is very low (0.35) to capture real changes, but this increased noise in predictions. A higher weighting of the ‘change’ class in the loss function empirically forced the network to make more aggressive predictions, resulting in lower thresholds. This suggests that future work should prioritize dynamic metrics such as focal loss (Lin et al. 2018) in order to stabilize model confidence.

5.6 Limitations

There are two main limitations in the Inpainting techniques at the moment, which are disadvantages to the Self-pair and Unpair logics from experiment 4 and a main disadvantage over real-world data.

The first limitation is the scalability as mentioned in section 5.4. Generative diffusion models suffer from highly demanding computational costs, a critical challenge that limits the datasets size in comparison to the real-time dataset creation like Self-pair and Unpair

logics, without the use of deep learning techniques. The second limitation is the lack of implementation of natural variations that were not explicitly modeled in the synthetic pipeline. Leafy trees that partially cover buildings in summer or drop their leaves. While the Unpair strategy could randomly introduce some of these variations, the inpaint models lack a specific logic for simulating seasonal coverings.

The third weakness compared to purely logic-based strategies (A and B experiment 4) becomes apparent when looking at the color histogram of the retrieved WMS samples and generated samples. While the generated images have a rather smooth color histogram (see Figure 11) due to suppression of small details (mainly texture), the direct WMS retrievals are rather jagged. This leads to a natural domain gap, which could maybe be bridged in future research by adding such a compression effect to the augmentation pipeline.

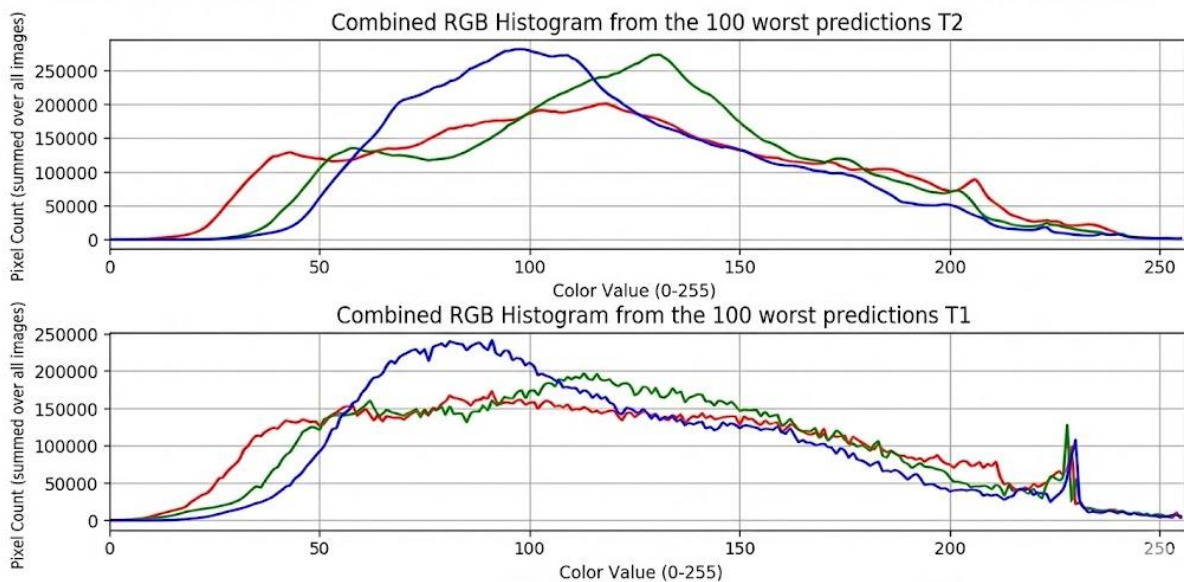


Figure 11 Comparison Color histogram 100 worst predictions generated by diffusion model (synthesized T2, top) and WMS retrievals (T1, bottom)

5.7 Practical use for ALKIS and governmental surveying

The results of this thesis provide a clear picture of how deep-learning-based CD can be integrated into governmental workflows in North Rhine-Westphalia. While IoU of 0.54 and F1-Score of 0.68 are insufficient for a fully automated mapping workflow to update parts of the ALKIS database, the system is viable for pre-filtering systems. Given the predictions are not precise (fragmentation, over or under generalized geometries, committed or omitted changed buildings) connotes that a human operator is still required to verify and refine the geometry of changes while vectorizing them.

They also have wider potential for the detection of unofficial constructions on the ground in nature conservation areas. In this case, the system also only needs to flag potential building footprints which have not been registered. For such pre-filtering based workflow,

the threshold could be tuned to a high recall score, because human reviewers can easily reject false positive alarms, but the overall would benefit from a more sensitive flagging system. This could lead to a lower human effort workflow in terms of controlling e.g., nature conservation areas or building construction on private ground without proper legal permits. The manual label bottleneck can be bypassed by using the Unpair or Inpainting strategies allowing the training of robust models without spending substantial man-hours on manual annotations or search for real-world building change objects on bi-temporal remotely sensed imagery.

5.8 Outlook

Future research should address the limitation of the current Inpaint strategy, the application of highly efficient distilled generative diffusion models with contextual prompting, such as Flux.1 Kontext (Labs et al. 2025) should be investigated. The use of this technique could make it possible to generate images on the fly, as in the Unpair strategy, while maintaining the beneficial structural logic in changes, as demonstrated in E1. The generation of semi-synthetic training images in real time also makes it possible to expand the pool of training samples in a similar way to the Unpair strategy. This expanded dataset would allow models with more parameters, such as ResNet101 as backbone, to be trained more effectively, which could result in improved prediction accuracy.

Furthermore, the domain gap caused by the compression of Web Map Service (WMS) data must be addressed as well. While the augmentation methods in E2 try to mimic real changes between real aerial images taken at different times, future approaches could split this task: classical augmentation could focus specifically on simulating technical compression artifacts, while environmental variations (e.g., seasons or lighting) should be mimicked through the contextual input in the image diffusion process.

6 Bibliography

- Al-Ruzouq, Rami, Khaled Hamad, Abdallah Shanableh, and Mohamad Khalil. 2017. 'Infrastructure Growth Assessment of Urban Areas Based on Multi-Temporal Satellite Images and Linear Features'. *Annals of GIS* 23 (3): 183–201. <https://doi.org/10.1080/19475683.2017.1325935>.
- AmberLi. (2021). *AmberHen/WHU-OPT-SAR-Dataset*. June 3, released January 21. <https://github.com/AmberHen/WHU-OPT-SAR-dataset>.
- Ball, John E., Derek T. Anderson, and Chee Seng Chan Sr. 2017. 'Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools, and Challenges for the Community'. *Journal of Applied Remote Sensing* 11 (4): 042609. <https://doi.org/10.1117/1.JRS.11.042609>.
- Ball, John E., Derek T. Anderson, and Pan Wei. 2018. 'State-of-the-Art and Gaps for Deep Learning on Limited Training Data in Remote Sensing'. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July, 4119–22. <https://doi.org/10.1109/IGARSS.2018.8518681>.
- Ban, Yifang, and Osama Yousif. 2016. 'Change Detection Techniques: A Review'. In *Multitemporal Remote Sensing: Methods and Applications*, edited by Yifang Ban. Springer International Publishing. https://doi.org/10.1007/978-3-319-47037-5_2.
- Bandara, Wele Gedara Chaminda, Nithin Gopalakrishnan Nair, and Vishal M. Patel. 2024. 'DDPM-CD: Denoising Diffusion Probabilistic Models as Feature Extractors for Change Detection'. arXiv:2206.11892. Preprint, arXiv, January 12. <https://doi.org/10.48550/arXiv.2206.11892>.
- Belgiu, Mariana, and Lucian Drăguț. 2016. 'Random Forest in Remote Sensing: A Review of Applications and Future Directions'. *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (April): 24–31. <https://doi.org/10.1016/j.isprsjprs.2016.01.011>.
- Benidir, Yanis, Nicolas Gonthier, and Clement Mallet. 2025. 'The Change You Want To Detect: Semantic Change Detection In Earth Observation With Hybrid Data Generation'. arXiv:2503.15683. Preprint, arXiv, April 4. <https://doi.org/10.48550/arXiv.2503.15683>.
- BKG. n.d. 'BKG - Projekte - Cop4All-DE'. Accessed 2 February 2026. https://www.bkg.bund.de/DE/Forschung/Projekte/Cop4All-DE/Cop4All-DE_cont.html.
- black-forest-labs. 2026. 'FLUX.1-Fill-Dev · Hugging Face'. January 2. <https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev>.
- Blaschke, T. 2010. 'Object Based Image Analysis for Remote Sensing'. *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (1): 2–16. <https://doi.org/10.1016/j.isprsjprs.2009.06.004>.

- Caye Daudt, Rodrigo, Bertr Le Saux, and Alexandre Boulch. 2018. 'Fully Convolutional Siamese Networks for Change Detection'. *2018 25th IEEE International Conference on Image Processing (ICIP)*, October, 4063–67. <https://doi.org/10.1109/ICIP.2018.8451652>.
- Caye Daudt, Rodrigo, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. 2019. 'Multitask Learning for Large-Scale Semantic Change Detection'. *Computer Vision and Image Understanding* 187 (October): 102783. <https://doi.org/10.1016/j.cviu.2019.07.003>.
- Chen, Gang, Geoffrey J. Hay, Luis M. T. Carvalho, and Michael A. Wulder. 2012. 'Object-Based Change Detection'. *International Journal of Remote Sensing* 33 (14): 4434–57. <https://doi.org/10.1080/01431161.2011.648285>.
- Chen, Hao, and Zhenwei Shi. 2020a. 'A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection'. *Remote Sensing* 12 (10). <https://doi.org/10.3390/rs12101662>.
- Chen, Hao, and Zhenwei Shi. 2020b. 'A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection'. *Remote Sensing* 12 (10): 1662. <https://doi.org/10.3390/rs12101662>.
- Chen, Liang-Chieh, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. 'Rethinking Atrous Convolution for Semantic Image Segmentation'. arXiv:1706.05587. Preprint, arXiv, December 5. <https://doi.org/10.48550/arXiv.1706.05587>.
- Cheng, Guangliang, Yunmeng Huang, Xiangtai Li, et al. 2024. 'Change Detection Methods for Remote Sensing in the Last Decade: A Comprehensive Review'. *Remote Sensing* 16 (13): 2355. <https://doi.org/10.3390/rs16132355>.
- cocktailpeanut. (2024). *Cocktailpeanut/Fluxgym*. Python. September 5, released January 29. <https://github.com/cocktailpeanut/fluxgym>.
- Daudt, Rodrigo Caye, Bertr Le Saux, Alexandre Boulch, and Yann Gousseau. 2018. 'Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks'. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July, 2115–18. <https://doi.org/10.1109/IGARSS.2018.8518015>.
- Daudt, Rodrigo, Bertrand Saux, Alexandre Boulch, and Yann Gousseau. 2018. *High Resolution Semantic Change Detection*. <https://doi.org/10.48550/arXiv.1810.08452>.
- Demiray, Bekir Z., Muhammed Sit, and Ibrahim Demir. 2021. 'D-SRGAN: DEM Super-Resolution with Generative Adversarial Networks'. *SN Computer Science* 2 (1): 48. <https://doi.org/10.1007/s42979-020-00442-2>.

- Dhariwal, Prafulla, and Alex Nichol. 2021. 'Diffusion Models Beat GANs on Image Synthesis'. arXiv:2105.05233. Preprint, arXiv, June 1. <https://doi.org/10.48550/arXiv.2105.05233>.
- Diffusers. n.d. 'Stable Diffusion XL'. Accessed 23 January 2026. https://huggingface.co/docs/diffusers/main/en/api/pipelines/stable_diffusion/stable_diffusion_xl.
- Frevel, Martin. 2018. *Karten- und Koordinatenanpassung durch Homogenisierung im Zuge von Liegenschaftsvermessungen*. June 1.
- Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, et al. 2020. 'Shortcut Learning in Deep Neural Networks'. *Nature Machine Intelligence* 2 (11): 665–73. <https://doi.org/10.1038/s42256-020-00257-z>.
- Gella, Getachew W., and Stefan Lang. 2025. 'Generative Approach for Building Change Detection in Temporary Settlement Areas for Humanitarian Emergency Response'. *IGARSS 2025 - 2025 IEEE International Geoscience and Remote Sensing Symposium*, August, 1536–40. <https://doi.org/10.1109/IGARSS55030.2025.11242694>.
- Geobasis NRW. n.d.-a. 'Geobasis NRW'. Accessed 7 February 2026. <https://www.bezreg-koeln.nrw.de/geobasis-nrw>.
- Geobasis NRW. n.d.-b. 'Geodatendienste'. Accessed 23 January 2026. <https://www.bezreg-koeln.nrw.de/geobasis-nrw/webdienste/geodatendienste>.
- Geobasis NRW. n.d.-c. 'Hausumringe'. Accessed 23 January 2026. <https://www.bezreg-koeln.nrw.de/geobasis-nrw/produkte-und-dienste/liegenschaftskataster/aktuelles-liegenschaftskataster/hausumringe>.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- GovData. n.d. 'DL-DE->Zero-2.0'. Accessed 23 January 2026. <https://www.govdata.de/dl-de/zero-2-0>.
- Gupta, Ritwik, Richard Hosfelt, Sandra Sajeev, et al. 2019. 'xBD: A Dataset for Assessing Building Damage from Satellite Imagery'. arXiv:1911.09296. Preprint, arXiv, November 21. <https://doi.org/10.48550/arXiv.1911.09296>.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. 'Deep Residual Learning for Image Recognition'. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 770–78. <https://doi.org/10.1109/CVPR.2016.90>.
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. 'Denoising Diffusion Probabilistic Models'. arXiv:2006.11239. Preprint, arXiv, December 16. <https://doi.org/10.48550/arXiv.2006.11239>.
- Hugging Face. 2023. 'Diffusers/Stable-Diffusion-XL-1.0-Inpainting-0.1'. September 7. <https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1>.

- Hugging Face. n.d. 'Inpainting (Diffusers Documentation)'. Accessed 23 January 2026. <https://huggingface.co/docs/diffusers/en/using-diffusers/inpaint>.
- Jat, Mahesh Kumar, P. K. Garg, and Deepak Khare. 2008. 'Monitoring and Modelling of Urban Sprawl Using Remote Sensing and GIS Techniques'. *International Journal of Applied Earth Observation and Geoinformation* 10 (1): 26–43. <https://doi.org/10.1016/j.jag.2007.04.002>.
- Khelifi, Lazhar, and Max Mignotte. 2020. 'Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis'. *IEEE Access* 8: 126385–400. <https://doi.org/10.1109/ACCESS.2020.3008036>.
- Kingma, Diederik P., and Jimmy Lei. 2015. *Adam: A Method for Stochastic Optimization*.
- Labs, Black Forest, Stephen Batifol, Andreas Blattmann, et al. 2025. 'FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space'. arXiv:2506.15742. Preprint, arXiv, June 24. <https://doi.org/10.48550/arXiv.2506.15742>.
- Leichtle, Tobias, Christian Geiß, Tobia Lakes, and Hannes Taubenböck. 2017. 'Class Imbalance in Unsupervised Change Detection – A Diagnostic Analysis from Urban Remote Sensing'. *International Journal of Applied Earth Observation and Geoinformation* 60 (August): 83–98. <https://doi.org/10.1016/j.jag.2017.04.002>.
- Li, Kaiyu, Xiangyong Cao, and Deyu Meng. 2024. 'A New Learning Paradigm for Foundation Model-Based Remote-Sensing Change Detection'. *IEEE Transactions on Geoscience and Remote Sensing* 62: 1–12. <https://doi.org/10.1109/TGRS.2024.3365825>.
- Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. 'Focal Loss for Dense Object Detection'. arXiv:1708.02002. Preprint, arXiv, February 7. <https://doi.org/10.48550/arXiv.1708.02002>.
- Lu Corresponding author, D., P. Mausel, E. Brondízio, and E. Moran. 2004. 'Change Detection Techniques'. *International Journal of Remote Sensing* 25 (12): 2365–401. <https://doi.org/10.1080/0143116031000139863>.
- Maxwell, Aaron E., Timothy A. Warner, and Fang Fang. 2018. 'Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review'. *International Journal of Remote Sensing* 39 (9): 2784–817. <https://doi.org/10.1080/01431161.2018.1433343>.
- Mohandoss, Tharun, Aditya Kulkarni, Daniel Northrup, Ernest Mwebaze, and Hamed Alemohammad. 2020. 'Generating Synthetic Multispectral Satellite Imagery from Sentinel-2'. arXiv:2012.03108. Preprint, arXiv, December 5. <https://doi.org/10.48550/arXiv.2012.03108>.
- Mountrakis, Giorgos, Jungho Im, and Caesar Ogole. 2011. 'Support Vector Machines in Remote Sensing: A Review'. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (3): 247–59. <https://doi.org/10.1016/j.isprsjprs.2010.11.001>.

- Nguyen, Tuong Vy, Alexander Glaser, and Felix Biessmann. 2024. 'Generating Synthetic Satellite Imagery With Deep-Learning Text-to-Image Models -- Technical Challenges and Implications for Monitoring and Verification'. arXiv:2404.07754. Preprint, arXiv, April 11. <http://arxiv.org/abs/2404.07754>.
- Open.NRW. n.d. 'Datensatz Details | Open.NRW'. Accessed 23 January 2026. <https://open.nrw/dataset/56fb584b-10cf-4009-a405-0bef06bb3e00>.
- Pal, M., and P. M. Mather. 2005. 'Support Vector Machines for Classification in Remote Sensing'. *International Journal of Remote Sensing* 26 (5): 1007–11. <https://doi.org/10.1080/01431160512331314083>.
- Peebles, William, and Saining Xie. 2023. 'Scalable Diffusion Models with Transformers'. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, October 1, 4172–82. <https://doi.org/10.1109/ICCV51070.2023.00387>.
- PyTorch. n.d. 'BCEWithLogitsLoss — PyTorch 2.10 Documentation'. Accessed 24 January 2026. <https://docs.pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>.
- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. 'High-Resolution Image Synthesis with Latent Diffusion Models'. arXiv:2112.10752. Preprint, arXiv, April 13. <https://doi.org/10.48550/arXiv.2112.10752>.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28.
- Seo, Minseok, Hakjin Lee, Yongjin Jeon, and Junghoon Seo. 2023. 'Self-Pair: Synthesizing Changes from Single Source for Object Change Detection in Remote Sensing Imagery'. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January, 6363–72. <https://doi.org/10.1109/WACV56688.2023.00631>.
- Shafique, Ayesha, Guo Cao, Zia Khan, Muhammad Asad, and Muhammad Aslam. 2022. 'Deep Learning-Based Change Detection in Remote Sensing Images: A Review'. *Remote Sensing* 14 (4): 871. <https://doi.org/10.3390/rs14040871>.
- Shorten, Connor, and Taghi M. Khoshgoftaar. 2019. 'A Survey on Image Data Augmentation for Deep Learning'. *Journal of Big Data* 6 (1): 60. <https://doi.org/10.1186/s40537-019-0197-0>.
- SINGH, ASHBINDU. 1989. 'Review Article Digital Change Detection Techniques Using Remotely-Sensed Data'. *International Journal of Remote Sensing* 10 (6): 989–1003. <https://doi.org/10.1080/01431168908903939>.

- Tan, Bin, Jeffrey G. Masek, Robert Wolfe, et al. 2013. 'Improved Forest Change Detection with Terrain Illumination Corrected Landsat Images'. *Remote Sensing of Environment* 136 (September): 469–83. <https://doi.org/10.1016/j.rse.2013.05.013>.
- Tobin, Josh, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. 'Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real-world'. arXiv:1703.06907. Preprint, arXiv, March 20. <https://doi.org/10.48550/arXiv.1703.06907>.
- Yang, Ling, Zhilong Zhang, Yang Song, et al. 2025. 'Diffusion Models: A Comprehensive Survey of Methods and Applications'. arXiv:2209.00796. Preprint, arXiv, September 27. <https://doi.org/10.48550/arXiv.2209.00796>.
- Yang, Meijuan, Licheng Jiao, Fang Liu, Biao Hou, and Shuyuan Yang. 2019. 'Transferred Deep Learning-Based Change Detection in Remote Sensing Images'. *IEEE Transactions on Geoscience and Remote Sensing* 57 (9): 6960–73. <https://doi.org/10.1109/TGRS.2019.2909781>.
- Zhang, Rongyu, Lixuan Du, Qi Xiao, and Jiaming Liu. 2020. 'Comparison of Backbones for Semantic Segmentation Network'. *Journal of Physics: Conference Series* 1544 (1): 012196. <https://doi.org/10.1088/1742-6596/1544/1/012196>.
- Zheng, Zhuo, Stefano Ermon, Dongjun Kim, Liangpei Zhang, and Yanfei Zhong. 2025. 'Changen2: Multi-Temporal Remote Sensing Generative Change Foundation Model'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47 (2): 725–41. <https://doi.org/10.1109/TPAMI.2024.3475824>.
- Zheng, Zhuo, Ailong Ma, Liangpei Zhang, and Yanfei Zhong. 2021. 'Change Is Everywhere: Single-Temporal Supervised Object Change Detection in Remote Sensing Imagery'. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, October, 15173–82. <https://doi.org/10.1109/ICCV48922.2021.01491>.
- Zheng, Zhuo, Yanfei Zhong, Junjue Wang, Ailong Ma, and Liangpei Zhang. 2021. 'Building Damage Assessment for Rapid Disaster Response with a Deep Object-Based Semantic Change Detection Framework: From Natural Disasters to Man-Made Disasters'. *Remote Sensing of Environment* 265 (November): 112636. <https://doi.org/10.1016/j.rse.2021.112636>.
- Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. 'Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks'. *2017 IEEE International Conference on Computer Vision (ICCV)*, October, 2242–51. <https://doi.org/10.1109/ICCV.2017.244>.