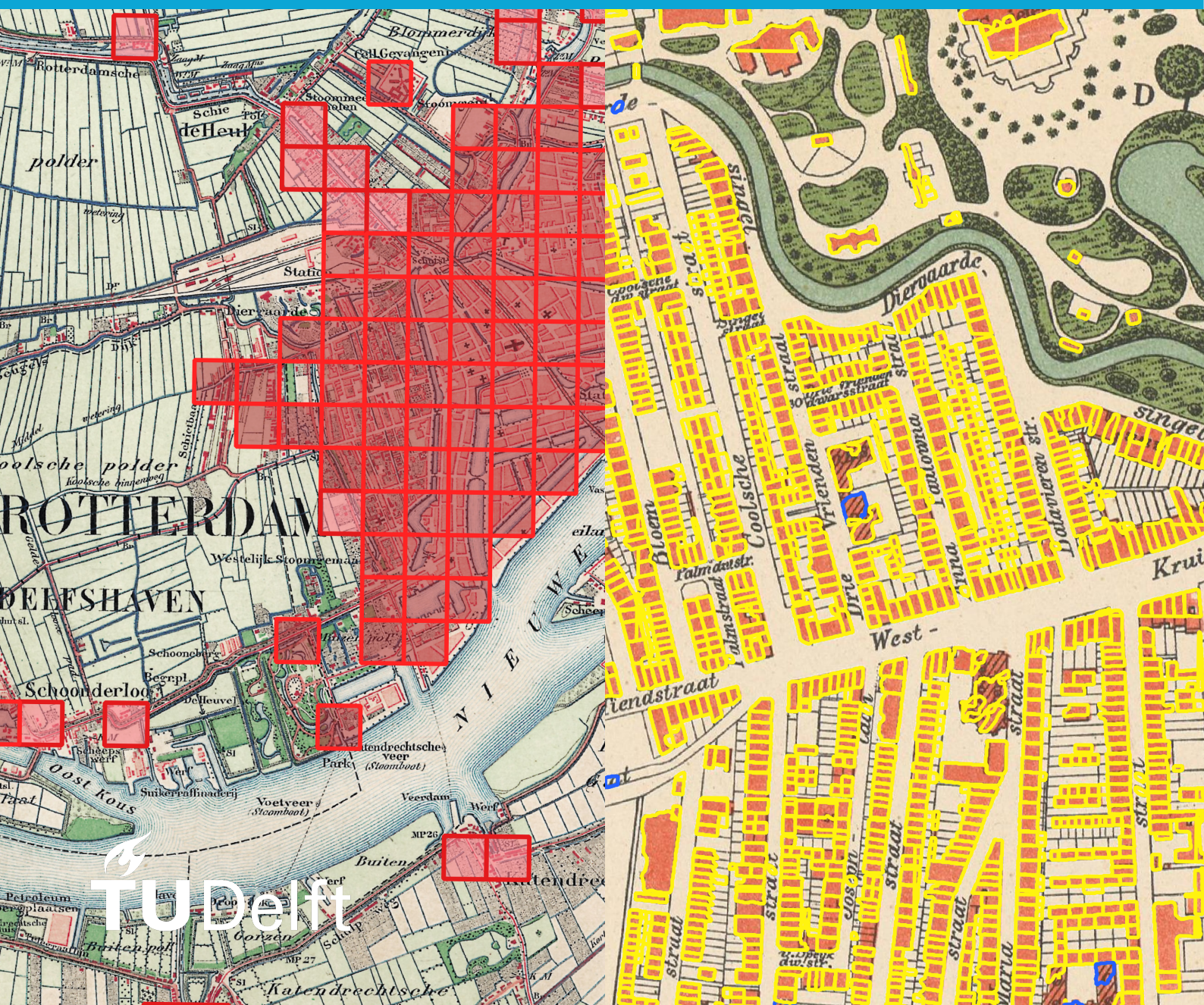


Patch-Level and Object-Level Deep Learning for Urban Extent Extraction from Multi-Temporal Historical Maps

Andika Hadi Hutama
2026



MSc thesis in Geomatics

Patch-Level and Object-Level Deep Learning for Urban Extent Extraction from Multi-Temporal Historical Maps

Andika Hadi Hutama

June 2026

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Andika Hadi Utama: *Patch-Level and Object-Level Deep Learning for Urban Extent Extraction from Multi-Temporal Historical Maps* (2026)

© This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

The work in this thesis was carried out in the:



Geo-Database Management Centre
Delft University of Technology

Supervisors: Dr.ir. B.M. (Martijn) Meijers
Prof.dr. A.R. (Ana) Pereira Roders

Co-reader: Dr. Giorgio Agugiaro

Abstract

Historical maps are a rich source of information about past urban landscapes, but their contents remain largely inaccessible for computational analysis. This thesis investigates and compares two deep learning approaches for automated urban extent extraction from Dutch historical topographic maps, and applies the results to urban change analysis.

Method 1 applies patch-level image classification using the MapReader framework on the Bonnebladen map series (1:25,000) covering the Province of South Holland. Map sheets are divided into 200-meter square patches and classified into three classes using a fine-tuned ResNet-101 model. A weak classifier trained on 1,768 manually annotated patches achieved a macro-averaged F1-score of 0.900. Inference was applied to 74 sheets across two temporal editions (T1: 1869–1916, T2: 1910–1943), producing georeferenced classified patch datasets at provincial scale. Post-classification comparison between T1 and T2 detected approximately 65 km² of total urban growth in South Holland, with new building blocks as the dominant growth pattern. The results are validated against the HGN-1900 land use dataset, showing broad spatial agreement in urban areas.

Method 2 applies object-level instance segmentation using Mask R-CNN on Rotterdam 1897 municipality maps (1:5,000). To avoid manual annotation, BGT vector data is style-transferred to approximate the historical cartographic style and translated using CycleGAN to generate a synthetic training dataset with automatically aligned annotation masks. The epoch 6 CycleGAN checkpoint is selected based on FID evaluation and visual inspection of building boundary preservation. The Mask R-CNN model trained on this synthetic data achieved a validation mAP50 of 0.8784. After post-processing and shape regularisation, 12,299 building polygon predictions are produced from sheet 3. External evaluation against BGT buildings confirmed to predate 1897 yielded a recall of 41.4%, reflecting both model detection limitations and the substantial post-war urban renewal of Rotterdam following the 1940 bombing.

The two methods differ substantially in their capabilities and requirements. Method 1 scales to provincial level with limited annotation effort and supports statistical-level change analysis, but is sensitive to cartographic style shifts across temporal editions and produces coarse 200-meter raster output. Method 2 produces building-level vector polygons enabling urban persistence analysis, but requires a complex multi-stage pipeline, is constrained by georeferencing quality, and depends on synthetic training data that underrepresents dense historic urban configurations. Both methods are fully operational on a consumer-grade GPU laptop, confirming their accessibility without HPC infrastructure. Together, they demonstrate that IIIF-based map access infrastructure and modern deep learning techniques can be combined to support reproducible urban change analysis from historical map collections at multiple scales of spatial granularity.

Keywords: historical maps, deep learning, urban extent extraction, patch classification, Mask R-CNN, instance segmentation, CycleGAN, Bonnebladen, Netherlands, urban change analysis

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my first supervisor, Martijn Meijers, for his invaluable guidance throughout this thesis. His patient and supportive mentorship helped me navigate the direction of this research and kept me on track when the path was unclear. I am equally grateful to my second supervisor, Ana Pereira Roders, for her attentive feedback and encouragement, which consistently pushed me to think more carefully about my work. I also wish to thank Giorgio, the co-reader of this thesis, for taking the time to review and assess my work and for his valuable feedback.

I would also like to thank the Allmaps team, particularly Jules Schoonman and Bert Spaan, whose work first inspired my admiration for historical maps and ultimately shaped the direction of this thesis. A substantial part of this research builds directly on the Allmaps platform and its underlying infrastructure, without which this work would not have been possible.

I would also like to thank my fellow Geomatics students – the geomagicians – for their companionship and support over the past two years, making this journey through the field so enjoyable.

I gratefully acknowledge the financial support of the Indonesia Endowment Fund for Education (LPDP), which made this master's study and thesis research possible.

Finally, my deepest appreciation goes to my wife, Sarah, who believed in me from the very first day and never stopped. Her unwavering support through the long library nights and difficult moments was the reason I could see this journey through. I am also deeply grateful to my family for their constant encouragement and belief in me.

Contents

Abstract	ii
1. Introduction	2
1.1. Background and motivation	2
1.2. Research Objectives	3
1.3. Scope	3
1.4. Thesis Outline	4
2. Related Work	6
2.1. Historical maps as sources for urban studies	6
2.2. International Image Interoperability Framework (IIIF) and Allmaps	7
2.3. Feature extraction from historical maps	8
2.4. Deep learning for historical map analysis	10
3. Methodology	12
3.1. Method 1: Patch-level classification	13
3.2. Method 2: Object-level classification	19
3.3. Urban Change Analysis	27
4. Implementation	32
4.1. Datasets	32
4.2. Study Area	35
4.3. Tools	36
4.4. Hardware	37
5. Result and Analysis	38
5.1. Method 1 - Patch level classification	38
5.2. Method 2 - Object-level classification	55
5.3. Visual Comparison of Method 1 and Method 2: Rotterdam	81
5.4. Change analysis	83
6. Conclusion	92
6.1. Research Questions	92
6.2. Discussion	94
6.3. Contribution	97
6.4. Future Work	98
A Declaration of AI/LLM usage	100
B Reproducibility self-assessment	102
B.1 Data availability	102
B.2 Code availability	102
B.3 Reproducibility rating	102
C Georeference Rotterdam 1897 historical maps	104
D Bonnebladen assessment	106
Bibliography	112

List of figures

Figure 2.1	The web application of <i>Topotijdreis</i> developed by the Kadaster	6
Figure 2.2	Conceptual diagram of IIIF Presentation API.	8
Figure 3.1	Diagram workflow for Method 1: Patch-level	12
Figure 3.2	Diagram workflow for Method 2: Object-level.	13
Figure 3.3	Workflow of the custom Bonnebladen download pipeline.	14
Figure 3.4	Patch extraction workflow in MapReader.	15
Figure 3.5	Annotation workflow in MapReader	16
Figure 3.6	Post-processing to generate raster.	18
Figure 3.7	Three sequential tiles with stride 50% overlap.	22
Figure 3.8	CycleGAN architecture.	22
Figure 3.9	Mask R-CNN framework for instance segmentation	24
Figure 3.10	Building regularisation.	27
Figure 3.11	Oriented bounding box illustration.	29
Figure 3.12	Intersection over Union (IoU) illustration.	29
Figure 4.1	Single sheet of Bonnebladen - sheet no 449 - Delft	32
Figure 4.2	Sample sheet of Rotterdam municipality map - sheet no 3	33
Figure 4.3	Fire boundary map of Rotterdam	33
Figure 4.4	Study area for Method 1: Province South Holland	35
Figure 4.5	Study area for Method 2: part of Rotterdam city	36
Figure 5.1	Spatial distribution of Bonnebladen edition metadata per sheet index.	40
Figure 5.2	Number of sheets published per edition rank.	41
Figure 5.3	Number of sheets published per year (cumulative).	41
Figure 5.4	Number of sheets published per year and edition grouped.	41
Figure 5.5	Ground control points on sheet's corner points for georeferencing (Sheet 440) ..	42
Figure 5.6	Distribution of South Holland map sheet publication per year for T1 and T2. . .	43
Figure 5.7	Example patch grid overlaid on a Bonnebladen map sheet.	44
Figure 5.8	Pixel intensities values of the patches.	45
Figure 5.9	Representative annotated patches per class.	46
Figure 5.10	Training curves for the weak classifier over 20 epochs.	47
Figure 5.11	Cartographic symbol for garden land with greenhouses, post-1930s Bonnebladen editions	50
Figure 5.12	Predicted patch classes overlaid on representative map sheets for T1 and T2. . .	51
Figure 5.13	Temporal alignment between the T1 dataset and HGN-1900.	52
Figure 5.14	Difference map for South Holland comparing T1 patch predictions against HGN-1900.	54
Figure 5.15	Before and after manual georeferencing of Rotterdam 1897 sheet 3.	56
Figure 5.16	Representative patches manually selected from the Rotterdam 1897 map sheets	58
Figure 5.17	CycleGAN training loss curves across epochs 1-30 (1/2)	60
Figure 5.18	CycleGAN training loss curves across epochs 1-30 (2/2)	62
Figure 5.19	CycleGAN epoch progression – Exhibit 1.	64
Figure 5.20	CycleGAN epoch progression – Exhibit 2.	65

Figure 5.21	Visual observation of CycleGAN translation quality on vegetation-dominant tiles.	66
Figure 5.22	Rotterdam 1897 historical map input and reconstructed CycleGAN output for the full study area.	67
Figure 5.23	Comparison of domain representations across the synthetic data pipeline.	67
Figure 5.24	CycleGAN synthetic tile and its corresponding BGT annotation mask.	68
Figure 5.25	Mask R-CNN training curves over 30 epochs.	69
Figure 5.26	Mask R-CNN individual training loss components across 30 epochs.	70
Figure 5.27	Prediction noise with lower confidence score.	72
Figure 5.28	Mask R-CNN prediction output at three levels of urban density, colored by confidence score.	73
Figure 5.29	Mask R-CNN prediction output for hatched buildings.	74
Figure 5.30	Tile edge effect on building prediction.	75
Figure 5.31	Slivers, gaps, and overlaps between adjacent predicted buildings.	76
Figure 5.32	Building polygons after post-processing and regularisation.	77
Figure 5.33	Distribution of confidence scores after post-processing (Sheet 3).	78
Figure 5.34	Reliability zones by confidence score and regularisation IoU (Sheet 3).	79
Figure 5.35	Spatial distribution of matched and unmatched buildings from BGT reference matching.	80
Figure 5.36	Spatial distribution of TP, FP, and FN detections (Sheet 3).	81
Figure 5.37	Input map sheets for Method 1 and Method 2 on the Rotterdam study area. ...	82
Figure 5.38	Overlay of Method 1 and Method 2 outputs on Rotterdam study area.	83
Figure 5.39	Change map for South Holland comparing T1 patch predictions against T2 predictions.	84
Figure 5.40	Patch grid mismatch between Bonnebladen editions.	86
Figure 5.41	Cartographic style difference between general purpose edition and secret military purposes.	87
Figure 5.42	Cartographic style difference between Bonnebladen editions before and after the 1930s.	88
Figure 5.43	Print quality difference between Bonnebladen editions.	89
Figure C.1	Georeferencing Rotterdam 1897 historical map.	104
Figure D.1	<i>Bonnebladen bladwijzer</i> , or sheet indices of bonnebladen map series	106
Figure D.2	Published date (<i>Uitgave</i>) of the first available edition per sheet.	107
Figure D.3	Published date (<i>Uitgave</i>) of the latest available edition per sheet.	108
Figure D.4	Timespan in years between the first available edition and the last edition per sheet.	109
Figure D.5	Total available editions per sheet.	110

List of tables

Table 1	Tile counts per dataset split for Mask R-CNN training.	25
Table 2	Change taxonomy for Method 1.	28
Table 3	Ground control points for sheet 440 and 459 from both edition T1 and T2.	42
Table 4	Downloaded Bonnebladen GeoTIFFs for the South Holland study area.	43
Table 5	Patchification summary for the South Holland T1 dataset.	43
Table 6	Annotated patch counts per class (weak classifier dataset).	45
Table 7	Dataset split for weak classifier training.	46
Table 8	Test set performance of the weak classifiers.	48
Table 9	Predicted patch counts per class for T1 and T2	48
Table 10	Difference map class counts for the South Holland study area, comparing T1 predictions against HGN-1900.	53
Table 11	ICP residuals after TPS georeferencing of Rotterdam 1897 sheet 3.	57
Table 12	Georeferencing accuracy summary for Rotterdam 1897 sheet 3	57
Table 13	Color taxonomy applied to BGT layers for cartographic style transfer.	58
Table 14	Patch counts per domain and split after tiling and content filtering.	59
Table 15	Epoch-averaged training loss values at selected checkpoints.	62
Table 16	FID scores per evaluated checkpoint epoch, computed against testB patches (732 images).	62
Table 17	Mask R-CNN training results per selected epoch (30 epochs total).	68
Table 18	Mask R-CNN training loss breakdown at selected epochs.	70
Table 19	Inference parameter comparison for sheet 3.	71
Table 20	Building-Regulariser parameter configurations.	77
Table 21	Polygon counts at each post-processing step for BR8 on Sheet 3.	77
Table 22	Reliability zone summary for post-processed building polygons.	79
Table 23	BGT reference matching results.	80
Table 24	Dataset comparison between Method 1 and Method 2 for Rotterdam.	82
Table 25	Change detection results for Method 1, South Holland.	83

1. Introduction

This chapter begins with an introduction to the problem statement and motivation behind this research, followed by the research objectives and questions. The scope of the research is defined. Ultimately, an outline of the thesis structure is provided to guide the reader through the subsequent chapters.

1.1. Background and motivation

Historical maps are widely recognized as valuable resources for understanding the transformation of geographical space over time, particularly in the fields of historical and urban studies. The rapid development of geodesy and cartography from the 18th century resulted in extensive production of topographic maps at various scales across the Western world and its colonial territories. Among these, urban maps are especially informative due to their high level of detail and often reliable geometric representation of spatial features (Chen et al., 2021). By integrating these maps within modern geospatial systems, researchers can gain valuable insights into the processes that shaped present-day urban configurations (Nobajas & Nadal, 2015).

According to urban studies conducted in the Netherlands (Rutte, 2023, p. 14) the underlying spatial structures of towns and cities such as street patterns, parcel layouts, and drainage networks, tend to persist over centuries, a phenomenon described as “inertia”, implying that present-day urban configurations cannot be properly understood without tracing their origins through historical map series. Such studies were made possible by the extensive availability of historical map collections in the Netherlands, such as the *Topografische en Militaire Kaart van het Koninkrijk der Nederlanden* (TMK) series (1850-1865), the *Chromotopografische kaart des Rijks* series (1884-1930), and numerous municipality-specific historical map series.

However, despite the rich availability of historical map collections and recent digitization efforts that have made large map collections browsable on screen, their contents remain largely inaccessible for computational analysis, because they are not yet machine-readable (Hosseini et al., 2022). Extracting geospatial information from these maps presents several challenges, including the heterogeneity and inconsistency in cartographic styles across different map series, the visual complexity introduced by overlapping text, symbols, and digitization artifacts, as well as the degradation of map quality over time (Chen et al., 2021). Furthermore, before any computational analysis can be performed, these maps must first be accurately georeferenced, a prerequisite step that is particularly demanding for maps with low planimetric accuracy, often requiring substantial manual effort (Vaienti et al., 2025). Moreover, the time-consuming nature of converting maps into machine-readable formats remained unfeasible over large geographical areas (Goodchild, 2018). The scalability limitation has historically constrained historical map analysis to case studies with smaller size of datasets, preventing computational analysis of larger areas needed to address complex spatiotemporal questions (Liu et al., 2024).

These limitations have long made computational analysis of historical maps across large geographic areas unfeasible. In recent years, computer-vision-based deep learning has emerged as a promising approach for automating feature extraction from satellite imagery and has been increasingly applied to historical map analysis, demonstrating good performance in detecting urban features, road networks, and land-cover classes (O’Hara et al., 2024; Xia et al., 2025). This

1.1. Background and motivation

shift has opened new possibilities for processing historical maps at a scale previously unfeasible through manual or semi-automated workflows (Hosseini et al., 2022).

However, deep learning methods are notoriously data-hungry, requiring large volume and high-quality annotated training data to achieve reliable performance. Producing such annotations is a costly and labor-intensive process, as each map sheet must be manually interpreted and labelled, a task that contradicts the very objective of automation (Arzoumanidis et al., 2025). This challenge is further amplified by the stylistic variability across different map series where different cartographic conventions, color palettes, and map quality across time periods and map series limit the ability of models trained on one collection to generalize to another (Arzoumanidis et al., 2025).

1.2. Research Objectives

Addressing the challenges and limitations in the previous Section 1.1 is essential for enabling reproducible urban change analysis from historical map series. Automated extraction of urban features from historical maps can be approached at different levels of granularity. At the broader level, patch-based methods classify map regions to delineate urban extent, enabling statistical-level analysis or urban growth across large geographic areas. At a finer-level, object-based methods aim to detect and delineate individual objects, particularly building objects, offering more detailed spatial information but at greater computational cost and data demand. Therefore, this thesis investigates and compares patch-level and object-level deep learning approaches for automated urban extent extraction from historical topographic maps, with the aim of assessing their respective effectiveness for enabling urban change analysis across multiple temporal periods.

The main research question of this thesis is: *“To what extent do patch-level and object-level deep learning approaches differ in their effectiveness for extracting urban extent from multi-temporal historical topographic maps for enabling urban change analysis?”*

To address the main question, the following sub-research questions are as follows:

1. How can patch-level deep learning classification be applied to extract urban extent from multi-temporal historical maps?
2. How can synthetic data generation be applied to enable object-level building footprint extraction from historical maps?
3. What urban change patterns can be identified by comparing automatically extracted urban extents from historical maps against reference data?
4. What are the relative strengths, limitations, and trade-offs between patch-level and object-level deep learning approaches for urban extent from historical maps?

1.3. Scope

The following clarifies the scope of this thesis:

- This study only uses historical maps that are already scanned, and digitally available to access online from authoritative sources. Besides this study only uses historical maps that are already georeferenced.
- The use of additional historical data sources (cadastral records, building registries, old photographs, paintings, or textual archives) is not included in this study.
- Urban extent in this study refers to the horizontal built-up footprint as depicted on the map surface, such as building blocks and individual building footprints. Vertical urban change such as building height or densification, and administration-based boundary definitions are not considered.

1.4. Thesis Outline

This thesis is structured into six main chapters, the outline of which is as follows:

- Chapter 1 gives the overview of the research, including the background and motivation, followed by research objectives, research questions, scope of research, and lastly, the outline to the thesis.
- Chapter 2 provides an overview of scientific related work and theoretical background related to the thesis topic. It begins with a review of previous studies on historical maps for urban studies. Followed by the use of IIIF in supporting access of historical map collections, existing methods of feature extraction from historical maps, and deep learning approaches.
- Chapter 3 provides the detailed description of the research methodology, including the approaches for data processing, model training, evaluation, and post-processing steps.
- Chapter 4 describes the datasets and study area used in this thesis. Followed by the description of programming tools, software, and hardware implemented to test the methodology.
- Chapter 5 presents the result and findings of the experiments and analyses conducted in this thesis.
- Chapter 6 discusses the main findings of the research, addresses the research questions, and provides recommendations for future work.

1.4. Thesis Outline

2. Related Work

In this chapter, an overview of the related work relevant to the thesis is provided. First section discusses the use of historical maps as sources for urban studies (Section 2.1). Then, the development and contribution of IIIF and Allmaps is given (Section 2.2). Lastly, existing feature extraction method from historical maps (Section 2.3), and relevant deep learning methods are described (Section 2.4).

2.1. Historical maps as sources for urban studies

The Netherlands has a particularly rich tradition of historical cartographic documentation, offering one of the most extensive national map series in Europe. The *Kraijenhoff* map (1798-1822), *Topographische en Militaire Kaart van het Koninkrijk der Nederlanden* (TMK, 1850-1865) and the *Chromotopografische Kaart des Rijks* (Bonnebladen, 1865-1949) together provide near-complete topographic coverage of the Netherlands across multiple decades, offering a unique multi-temporal record of the Dutch landscape. These series are further complemented by municipality-specific and thematic historical map series such as *Waterstaatkaart*, which offer considerably higher level of cartographic detail for specific regions and topics. The value of these collections for urban studies is well demonstrated by the *Historische Atlas van Nederland* (Rutte, 2023), which traces the spatial evolution of Dutch towns and cities from the medieval period to the twentieth century, documenting the dramatic expansion of built-up areas during the industrial era. More recently, institutional efforts have sought to make such collections digitally accessible. The Dutch Kadaster launched *Topotijdreis* (Kadaster, 2015)¹, an online platform providing access to over 200 years of topographic maps (illustrated in Figure 2.1), allowing users to explore Dutch landscape and how it has evolved over time. Despite these efforts, these digitized map collections remain in raster image format and are not yet machine-readable, limiting their use to visual inspection only and preventing direct computational querying or geospatial analysis.

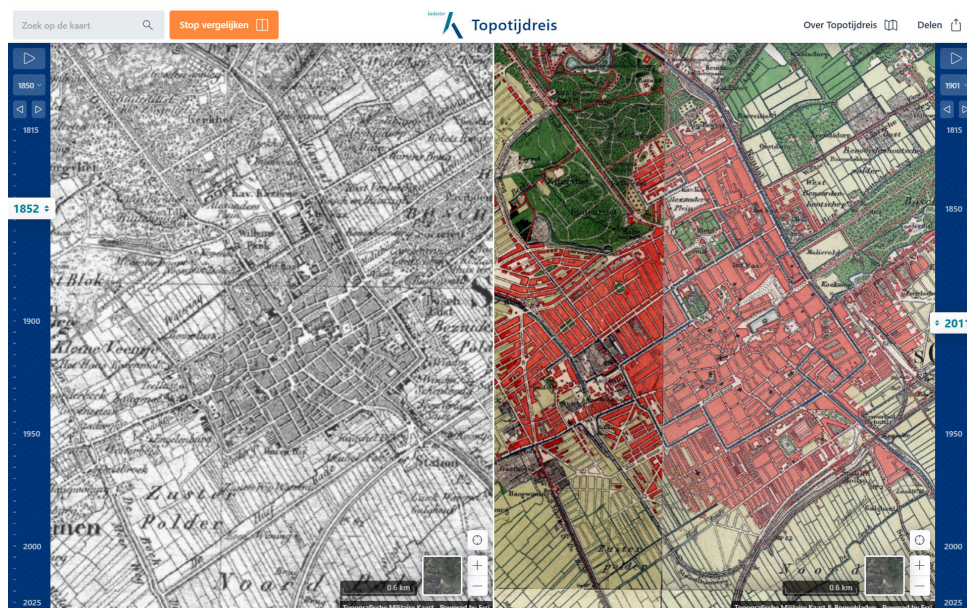


Figure 2.1: The web application of *Topotijdreis* developed by the Kadaster

¹Topotijdreis website: <https://topotijdreis.nl/>

2.1. Historical maps as sources for urban studies

Nevertheless, where such processing has been carried out, typically through manual digitization or automated workflows, historical map series have demonstrated considerable value for studying long-term urban growth and land use change at multiple scales. The *Historisch Grondgebruik Nederland* (HGN) project produced a nationwide digital land use dataset for the Netherlands around 1900, derived from historical topographic maps, providing one of the earliest spatially explicit records of Dutch land use before the remote sensing era (Knol et al., 2004). The HisGIS 1832 project digitized and georeferenced the first systematic cadastral survey of the Netherlands, producing a nationwide spatial database of land ownership and building footprints from 1832 (Humanities Cluster of the Royal Netherlands Academy of Arts and Sciences (KNAW), 2024). At the city level, studies such as those synthesized in the *Historische Atlas van Nederland* (Rutte, 2023) have quantified built-up area expansion for major Dutch cities between 1850 and 1900, revealing growth rates of several hundred percent in industrializing cities such as Rotterdam and Apeldoorn. These studies demonstrate that multi-temporal historical map series are capable of supporting statistical-level urban change analysis: measuring aggregate shifts in built-up extent, land use composition, and spatial distribution of urban development. However, such studies have largely been constrained to manually digitized datasets covering limited geographic extents, as the absence of scalable automated extraction methods has made nation-wide computational analysis infeasible.

2.2. International Image Interoperability Framework (IIIF) and Allmaps

A prerequisite for any large-scale computational analysis of historical maps is reliable programmatic access to digitized map collections, which are typically stored as high-resolution image files distributed across multiple institutional repositories. The International Image Interoperability Framework (IIIF) addresses this challenge by defining a set of open standards for delivering digital image objects online at scale, enabling consistent programmatic access through standardized APIs regardless of the hosting institution (Snydman et al., 2015). Cultural heritage institutions in the Netherlands, including the Dutch National Archives, have adopted IIIF to serve their digitized map collections, making thousands of historical map sheets retrievable through uniform URL-based requests.

The IIIF Presentation API (illustrated in Figure 2.2) organizes digital objects through a hierarchy of concepts: a *Collection* groups related *Manifests*, each *Manifest* describes a single digitized object (such as a map sheet) as a sequence of *Canvases*, and each *Canvas* can carry multiple *Annotations* pointing to image content or supplementary data such as text or georeferencing information (Meijers & Schoonman, 2025). The Georeference Extension builds directly on this annotation model, encoding a set of Ground Control Points (GCPs), an optional transformation algorithm (e.g. Thin Plate Spline, Polynomial), and a mask as a structured JSON-LD annotation attached to the relevant *Canvas* (IIIF Maps Community Group, 2023). The mask is defined as an SVG polygon tracing the cartographic area of the image, indicating which region is to be interpreted as map content and excluding surrounding non-cartographic material such as explanatory notes or decorative borders (Meijers & Schoonman, 2025).

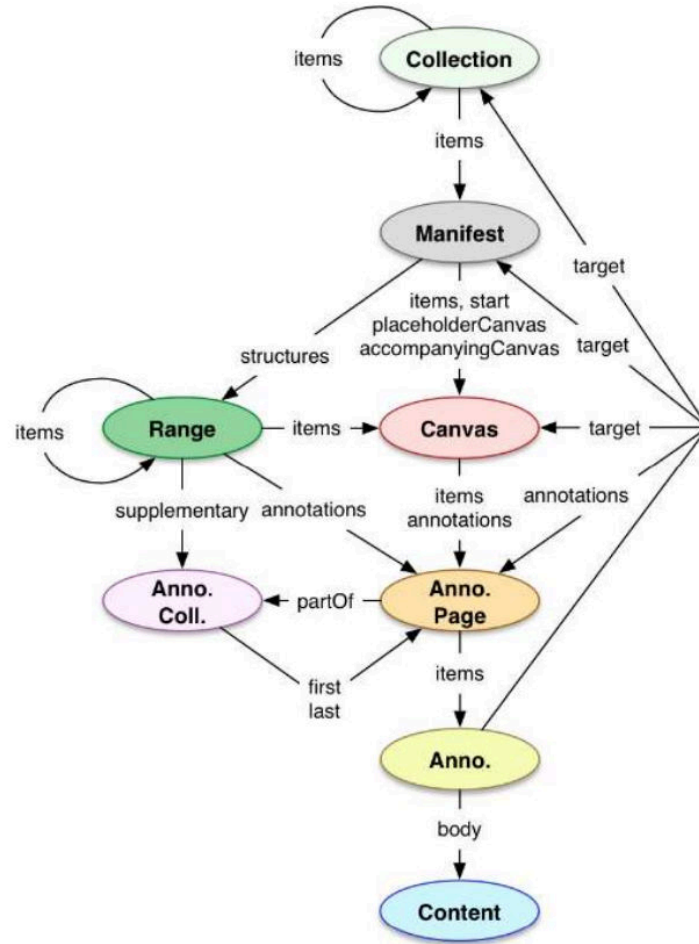


Figure 2.2: Conceptual diagram of IIIF Presentation API. Source: Meijers & Schoonman (2025).

Building on IIIF, the Allmaps² platform extends this infrastructure specifically for cartographic collections, providing tools to curate, georeference, and explore digitized maps from any IIIF-compliant repository without requiring complex GIS infrastructure (Hanlon et al., 2025). The resulting georeference annotation can be used to warp a map into a standard Coordinate Reference System (CRS), enabling direct client-side visualization through web mapping applications such as XYZ tile services, as well as integration into GIS tools such as QGIS for geospatial analysis workflows.

While IIIF and Allmaps resolve the distribution and access challenge, they do not address the semantic extraction problem. Maps served through IIIF remain raster images without machine-readable content, and extracting meaningful geospatial features from them still requires additional processing methods, which is the focus of the following sections.

2.3. Feature extraction from historical maps

Before historical maps can be used in any quantitative spatial analysis, they must undergo a multi-step processing chain to transform scanned raster images into machine-readable geodata. This chain typically involves digitization, georeferencing, image classification or segmentation, and post-processing to filter unwanted features as text and symbols (Gobbi et al., 2019). The output of this process can take two forms: raster data, where each pixel is assigned a thematic class, or vector data, where spatial features are represented as discrete geometric objects such as

²Allmaps website: <https://allmaps.org/>

2.3. Feature extraction from historical maps

polygons and lines. Raster outputs are well-suited for large-scale spatial analysis and land use change studies, while vector outputs enable more precise object-level queries and integration with modern GIS (Geographic Information System) datasets (Gobbi et al., 2019). Several main approaches of map digitization can be identified into four categories: manual, semi-automatic, machine learning and deep learning classification.

2.3.1. Manual digitization

The extraction of spatial information from historical maps has traditionally relied on manual digitization since the emergence of GIS technology. In this approach, trained operators visually interpret the map and manually trace geographic features such as building footprints, road networks, and land parcel boundaries, assigning class labels or attribute values to each delineated object (Gobbi et al., 2019). The approach is valued in digital humanities for the interpretive depth it enables: operators can resolve ambiguous features, apply contextual knowledge, and produce geometrically precise vector outputs that automated methods often struggle to match (Chen et al., 2021).

A prominent example in the Dutch context is the HisGIS project³, which manually digitized and vectorized the Napoleonic cadastral survey of 1832, producing a nationwide spatial database of approximately 3 million land parcels covering the entire Netherlands, with attributes including parcel size, owner and land use for each units (Humanities Cluster of the Royal Netherlands Academy of Arts and Sciences (KNAW), 2024). The scale and detail of HisGIS illustrates what manual digitization can achieve when organized as a sustained, multi-year collaborative effort. Collaborative approaches and large-scale crowdsourcing experiments have also been proposed to handle large map collections (Budig et al., 2016; Southall et al., 2017), distributing the annotation workload across many contributors. However, even with crowdsourcing, manual processes remain tedious, time-consuming, and non-reproducible (Chen et al., 2021).

2.3.2. Semi-automated classification

To reduce the reliance on manual labor, semi-automated methods have been developed that use image processing algorithms to automatically classify map pixels or regions, while still requiring human intervention to define parameters or provide reference samples. The main techniques developed for this purpose include histogram thresholding, color space clustering, edge detection, region growing, and Object-Based Image Analysis (Chiang et al., 2014; Gobbi et al., 2019). These methods exploit the cartographic properties of printed maps, particularly the use of distinct colors, line patterns, and spatial boundaries to encode different feature types, to separate target classes.

A notable application in the Dutch context is the *Historisch Grondgebruik Nederland 1900* (HGN1900) project (Knol et al., 2004), which produced a nationwide digital land use dataset for the Netherlands around 1900, derived from the historical topographic maps using color-based image classification and manual post-correction. While the result achieved an overall classification accuracy of 95.9% across 6,300 national validation points (Knol et al., 2004), a fundamental limitation of its color-based approach is the inability to distinguish between features, buildings and paved roads are both represented in red on the Bonne maps, resulting in a single merged class that conflates two distinct urban features.

Gobbi et al. (2019) proposed an OBIA-based workflow for semi-automatic digitization of heterogeneous map collections, combining region-based image segmentation with supervised

³HisGIS project website: <https://hisgis.nl/>

machine learning classifiers and an automated text and symbol removal module. Applied to three maps with very different characteristics, this approach achieved average classification of 97%, marking a significant improvement over manual methods.

Despite these advances, semi-automated methods carry important limitations. The correct parameters such as color thresholds, segmentation scales, or classifier training samples must be determined and tuned individually for each map or map series, as differences in printing quality, paper degradation, and cartographic style across map editions can significantly affect classification performance (Chen et al., 2021).

2.3.3. Machine learning and deep learning classification

The limitations of rule-based semi automated method have motivated the adoption of machine learning and, more recently, deep learning approaches, which learn feature representations directly from annotated training data rather than relying on manual defined rules. Compared to traditional machine learning classifiers such as Random Forest or Support Vector Machines, which require hand-crafted feature descriptors as input, deep learning models, particularly Convolutional Neural Network (CNN), can automatically learn hierarchical visual features from raw image patches, offering greater flexibility and robustness across varying cartographic styles (Heitzler & Hurni, 2020; Hosseini et al., 2022).

CNN-based methods have been increasingly applied within the historical maps domain. At the patch level, image classification approaches assign a class label to each image patch, and have been used to detect urban settlements and built-up areas from historical topographical map sheets (Hosseini et al., 2022; Uhl et al., 2017). At a finer level of detail, semantic segmentation models, most notably U-Net (Ronneberger et al., 2015), assign a class label to every pixel in the image and have been successfully applied to extract building footprints (Heitzler & Hurni, 2020), road networks, and wetland symbology (O'Hara et al., 2024). For tasks requiring individual object delineation, instance segmentation such as Mask R-CNN produce a separate mask for each detected object and have been applied to text detection and segmentation in historical maps, supporting downstream text removal and vectorization workflows (Dodda, 2024). These four task types, namely image classification, object detection, semantic segmentation, and instance segmentation, represent the main computer vision approaches used for historical map content extraction and form the methodological basis for the extraction approaches investigated in this thesis.

While these approaches have demonstrated promising results, their application to historical maps introduces a set of specific challenges that are discussed in the following section.

2.4. Deep learning for historical map analysis

Despite the promising results demonstrated by CNN-based approaches, deep learning methods share a common and fundamental requirement: large volumes of annotated training data. Producing such annotations is a costly and labor-intensive process, as each map sheet must be manually interpreted and labeled, which contradicts the very goal of automation (Arzoumanidis et al., 2025). This challenge is particularly significant for models developed for a specific map series or historical period, where the amount of available training data is limited in both geographic coverage and time span.

The problem of data scarcity is further compounded by the diversity of historical map styles. Models trained on a homogeneous corpus, such as a single national map series, tend to perform well within that corpus but struggle to generalize to maps with different cartographic

2.4. Deep learning for historical map analysis

styles, color palettes, or symbology (Petitpierre et al., 2021). Petitpierre et al. (2021) confirmed this through experiments across two diverse map corpora, showing that stylistic variability significantly degrades segmentation performance, and that developing a truly generic model for heterogeneous map collections remain an open challenge.

To address training data scarcity, image-to-image translation methods have emerged as a promising strategy for generating synthetic training data. Zhu et al. (2017) demonstrated image-to-image translation applied to several visual domain transfer tasks, including the translation of aerial photographs to Google Maps style and vice versa. In their work, they distinguished between two translation paradigms: paired translation using pix2pix framework, which requires matching input-output image pairs, and unpaired translation using CycleGAN, which learns to transfer styles between two domains without requiring any direct correspondence between images. For historical maps, the unpaired approach is particularly relevant, as paired correspondences between a historical map and its equivalent ground truth rarely exist. This approach offers a practical path toward reducing the annotation burden for historical map analysis and forms the basis for synthetic training data strategy adopted in this thesis.

A related limitation concerns the use of large pre-trained foundation models. Models such as the Segment Anything Model (SAM) and Vision Transformer (ViT)-based architectures have shown strong performance across general computer vision tasks, but their training on natural images creates a domain mismatch when applied to historical maps, leading to poor zero-shot performance on cartographic content. Xia et al. (2025) addressed this by adapting SAM specifically for historical map feature detection through domain-specific fine-tuning, demonstrating that generic foundation models require substantial adaptation before they can be reliably applied to historical maps.

Beyond the training data challenge, deep learning pipelines for historical map analysis, typically require multi-step post-processing workflows to convert pixel-level model outputs into usable vector geometries (Heitzler & Hurni, 2020). Each additional step in this pipeline introduces opportunities for error propagation, and the engineering effort required to design and maintain these steps for each specific map type reduce the overall degree of automation (Xia et al., 2024). This limitation underlines the need for extraction approaches that are not only accurate, but also practical and transferable across different map editions without requiring extensive map-specific tuning.

The review above identifies three gaps this thesis addresses: the lack of direct comparison between patch-level and object-level approaches on historical maps, the absence of scalable annotation strategies for national map series, and the need for reproducible post-processing pipelines. To address these, this thesis implements and compares a patch-level pipeline using MapReader applied to the Bonnebladen series, and an object-level pipeline using CycleGAN-based synthetic data and Mask R-CNN applied to the Rotterdam 1897 municipality maps. The methodology for both is described in Section 3.

3. Methodology

This chapter describes the methodology developed to automatically extract urban extent from historical topographic maps. In general, this thesis is divided into 2 main methodologies: (1) patch-based classification method and (2) object-based classification method. The first method is based on a patch-based classification approach, where the map is divided into small patches and each patch is classified as urban or non-urban using a deep learning model, hereafter referred to as Method 1. The second method is based on an object-based classification approach, where individual building footprints are extracted as objects alongside its geometry, which will be referred as Method 2. Both methods are implemented using deep learning techniques, specifically convolutional neural networks (CNN) based architectures. The methodology also includes a change analysis step to analyze the urban transformation across multiple temporal periods. The main difference of the 2 methodologies lies in their granularity of target object, where one focuses on patches and the other on individual buildings. Figure 3.1 and Figure 3.2 displays the general methodology workflow for each method.

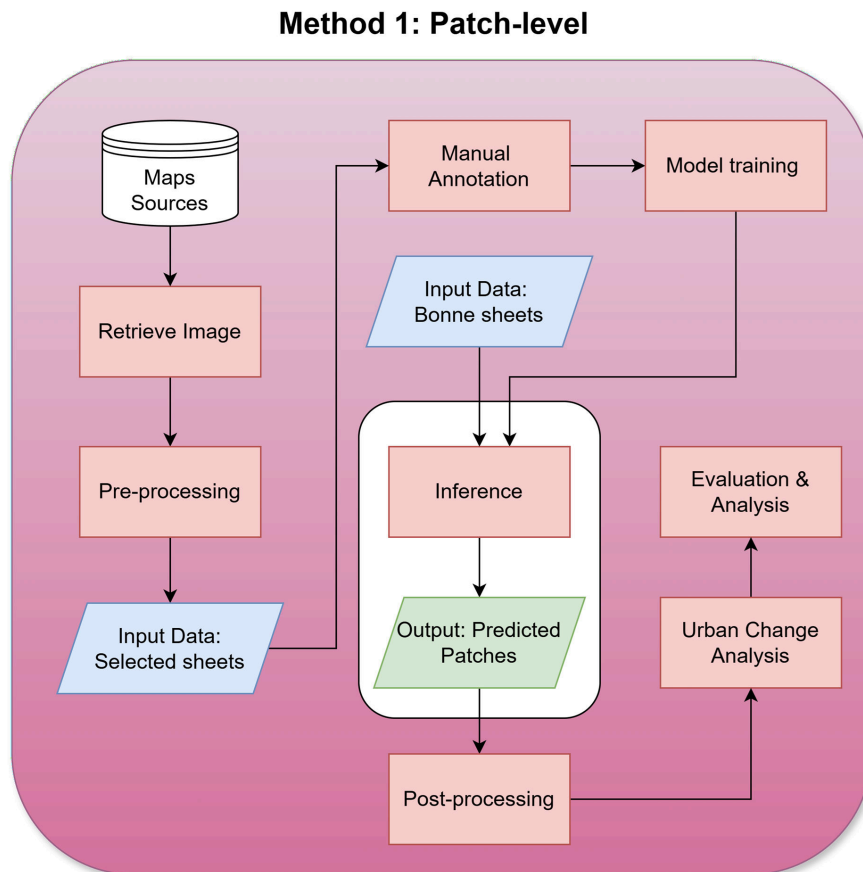


Figure 3.1: Diagram workflow for Method 1: Patch-level

3.1. Method 1: Patch-level classification

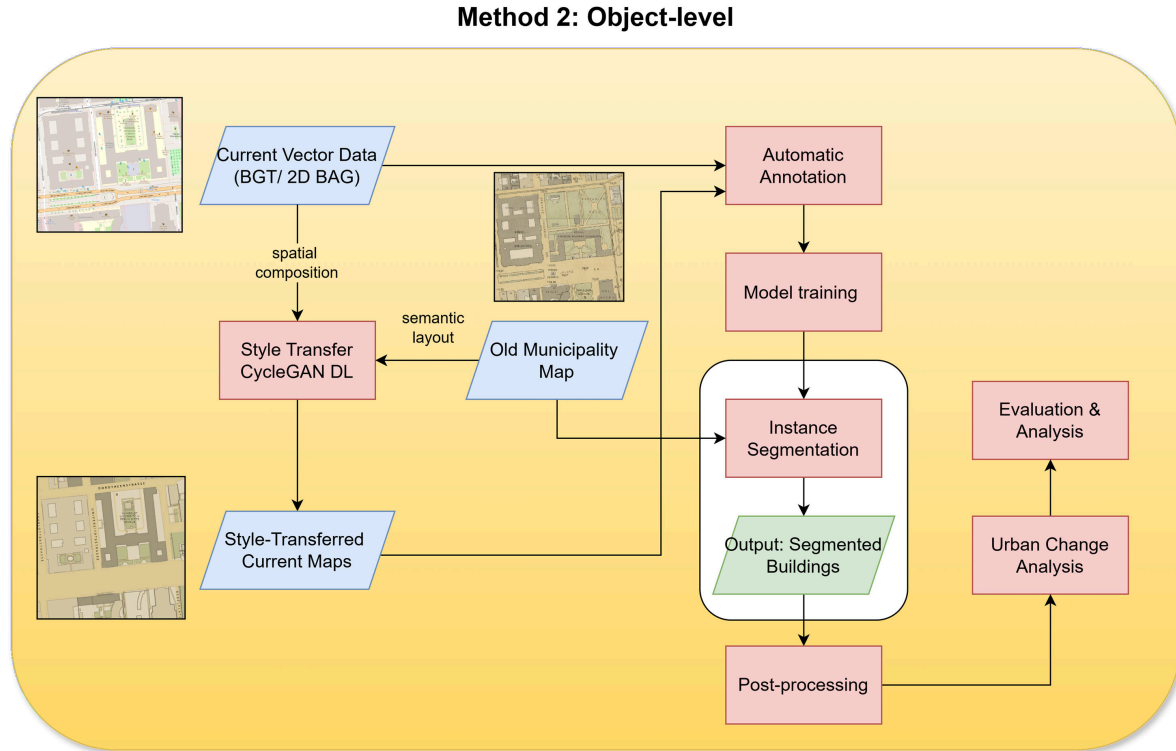


Figure 3.2: Diagram workflow for Method 2: Object-level, adapted from: Arzoumanidis et al. (2025)

The following sections provide a detailed description of each methodology.

3.1. Method 1: Patch-level classification

Method 1 extracts patches of classified urban extents as raster images from the Dutch Bonnebladen map series using a six-stage pipeline: (1) data acquisition and preparation, (2) patch creation, (3) manual annotation, (4) model training, (5) inference, and (6) post-processing. This study uses deep learning pipeline implementation from the MapReader⁴ framework established by Hosseini et al. (2022). In the following sections, the complete pipeline from Method 1 is described.

3.1.1. Data acquisition and preparation

The Bonnebladen sheets are accessed through the IIIF infrastructure maintained by the TU Delft Library team and georeferenced via the Allmaps platform. The georeference annotations, sheet catalog, and associated metadata used in this research are sourced from a TU Delft Library Observable notebook⁵. Two existing approaches were considered for retrieving the sheets as georeferenced GeoTIFFs.

The first is the Allmaps CLI, which applies a georeference annotation to export a warped GeoTIFF per sheet. While functional for individual sheets, it has no mechanism for batch processing or filtering by edition, year, or sheet number. The second is MapReader's built-in IIIFDownloader (Hosseini et al., 2022), which downloads georeferenced maps from a IIIF server to local storage as GeoTIFFs. However, it requires a IIIF Manifest compliant with the Presentation API as input. The Bonnebladen annotations are structured as flat AnnotationPages (one per edition)

⁴<https://github.com/maps-as-data/MapReader>

⁵TU Delft Library Observable notebook: <https://observablehq.com/d/ded63244d9f6625d>

rather than Manifests, making this approach incompatible without additional preprocessing. Furthermore, neither approach supports joining IIIF image identifiers – which are opaque hash-based strings – to human-readable sheet metadata (number, name, year, edition), a step required for temporal attribution in subsequent pipeline stages.

A custom Python script is therefore developed to address these requirements (Figure 3.3). The script joins the sheet catalog with the annotation metadata to resolve image identifiers to sheet numbers and years, applies edition-aware filtering, and downloads each selected sheet as a JPEG via the IIIF Image API. Map margins are then cropped using the edge mask polygons stored in the annotations. Each sheet is georeferenced by deriving an affine geotransform from the GCPs and assigning coordinates in EPSG:28992. The output per sheet is a tiled, LZW-compressed GeoTIFF together with a catalog CSV recording sheet number, name, year, edition, and file paths. This catalog serves directly as metadata input to MapReader’s `add_metadata()` step in the subsequent pipeline stages.

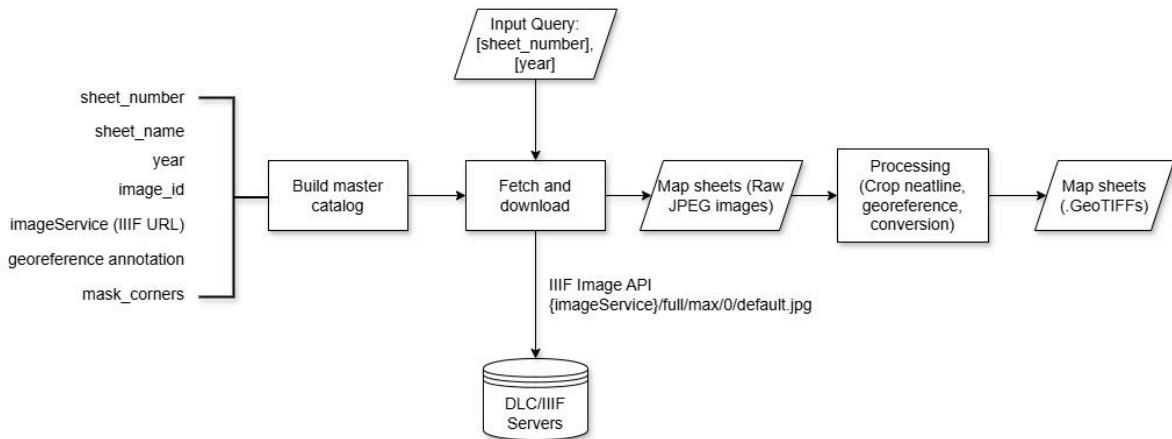


Figure 3.3: Workflow of the custom Bonnebladen download pipeline.

3.1.2. Patch creation

The georeferenced map sheets are loaded into a `MapImages` object provided by the MapReader library, which manages the collection of image paths and their associated geographic metadata. Prior to patchification, geographic information is extracted from each GeoTIFF and registered to the `MapImages` object. This step is a prerequisite for defining patch size in metric units: MapReader supports patch size specification either in pixels or in meters, and the latter requires the geographic coordinates of the bounding box corners to be available in the image metadata (Hosseini et al., 2022).

The map sheets are then split into square patches (as illustrated in Figure 3.4) with a size specified in meters. Patch size is a critical design parameter in the MapReader pipeline: it must be large enough that the target feature occupies a sufficient portion of the patch to be identifiable by both the human annotator and the classifier, yet small enough to retain meaningful spatial resolution in the output (Hosseini et al., 2022).

3.1. Method 1: Patch-level classification

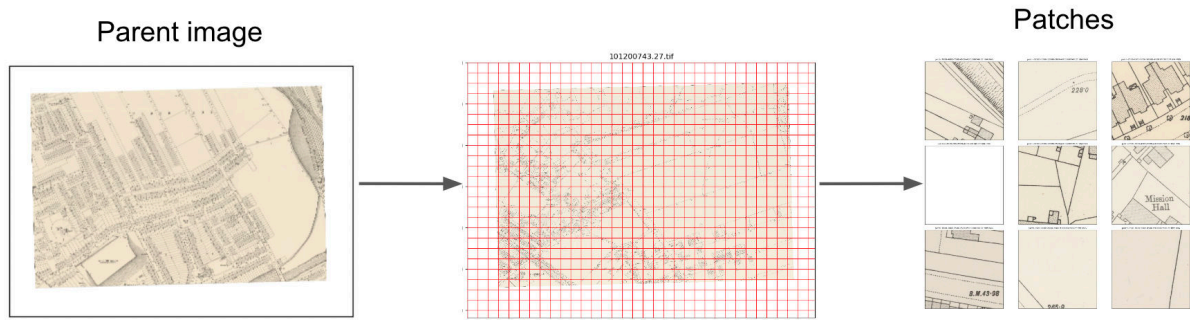


Figure 3.4: Patch extraction workflow in MapReader. Source: Hosseini et al. (2022), adapted from the MapReader documentation.

The patch size is determined through a comparative visual inspection of candidate sizes on a representative subset of map sheets; the selected value and the rationale behind it are reported in Section 5. Specifying patch size in metric units ensures that each patch represents a consistent physical footprint on the ground, independent of the pixel resolution or scan characteristics of individual sheets, which is particularly relevant for the Bonnebladen collection where sheets from different survey years may vary in scan resolution. Each patch is assigned geographic coordinates derived from the affine geotransform of its parent sheet, making the patch dataset spatially referenced and directly compatible with GIS workflows.

Per-patch pixel statistics, specifically the mean and standard deviation of pixel intensities per RGB channel, are computed and stored as patch attributes. These statistics serve as a proxy for visual content and are used in the subsequent annotation stage to pre-filter candidate patches by color response, as described in Section 3.1.3. The resulting patch metadata is exported as a CSV file alongside the patch image files, which together form the input to subsequent stages of the pipeline.

3.1.3. Manual annotation

Annotation is performed using MapReader’s interactive Annotator tool, which presents each patch alongside a context window showing the surrounding area of the parent sheet to aid interpretation of ambiguous cases, illustrated in Figure 3.5.

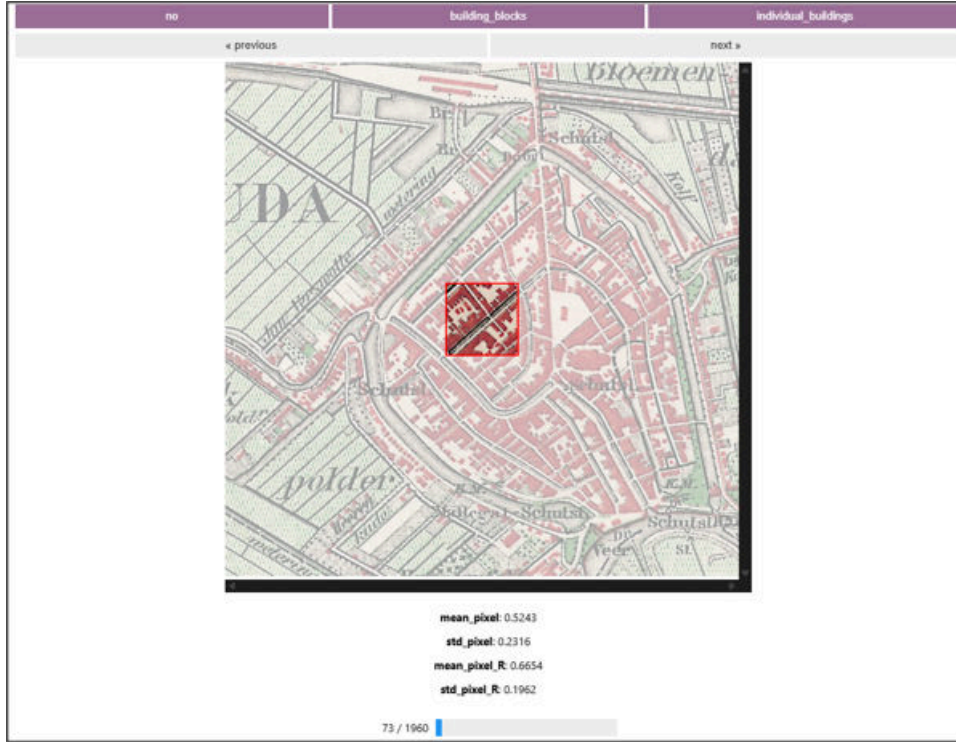


Figure 3.5: Annotation workflow in MapReader. Source: Hosseini et al. (2022).

Three label classes are defined: **“no”** (no significant urban content), **“building_blocks”** (dense building clusters characteristic of urban fabric), and **“individual_buildings”** (isolated or dispersed buildings in urban or rural areas). This taxonomy distinguishes continuous urban fabric from scattered settlement, a distinction that is clearly visible on the Bonnebladen series due to the consistent red fill used to render buildings across the collection.

To focus annotation effort on informative patches, a pixel-based pre-filtering step is applied before presenting patches to the annotator. Patches are filtered by thresholds on the mean and standard deviation of the red channel intensity, retaining only those whose color statistics fall within the range characteristic of building content. This step reduces the proportion of uninformative background patches in the annotation queue, following the accelerated annotation strategy of Hosseini et al. (2022). A label review pass is conducted after the initial annotation to verify consistency, with particular attention to **“individual_buildings”**, which shares visual characteristics with both **“no”** and **“building_blocks”** in edge cases.

3.1.4. Model training

The annotated patches are loaded using MapReader’s AnnotationsLoader. To split the dataset into training, validation, and test subsets, stratified sampling is used. Stratified sampling is a method that ensures each subset contains the same proportion of each label class as the full dataset (Pedregosa et al., 2018). This is important here because the **no** class is expected to dominate across most map sheets, and without stratification the minority classes could be underrepresented in training. The dataset is partitioned into 70% training, 15% validation, and 15% test.

Before training, each patch undergoes resizing, which rescales the patch to the fixed pixel dimensions required by the model architecture, and normalization, which rescales raw pixel intensity values to a small centered range to ensure stable and efficient training. These default image transforms (i.e. normalization and resizing) are applied at training time. Dataloaders are

3.1. Method 1: Patch-level classification

constructed with a batch size of 16. Batch size defines how many training samples are processed in a single forward-backward pass; a smaller batch size introduces more gradient noise but reduces memory requirements.

MapReader supports multiple options for the classifier model: a pretrained model from torchvision (Paszke et al., 2019), a custom-defined architecture, or a locally saved model. In this research, a pretrained ResNet-101 model is selected and fine-tuned on the annotated training set. Fine-tuning a pretrained model – rather than training from scratch – is preferred because pretraining on large natural image datasets provides a strong initialization of low-level visual features such as edges and textures, which transfer effectively to the historical map domain and substantially improve classification performance (Hosseini et al., 2022; Larochelle et al., 2009).

The cross-entropy loss is used as the optimization criterion. It measures the difference between the predicted class probabilities and the true labels across all classes:

$$\mathcal{L}_{CE} = - \sum_{c=1}^C y_c \log(\hat{y}_c) \quad (3.1)$$

where C is the number of classes, $y_c \in \{0, 1\}$ is the ground-truth indicator for class c , and $\hat{y}_c$ is the predicted probability for class c .

The model is optimized using AdamW (Loshchilov & Hutter, 2019), an optimizer that combines adaptive learning rates with decoupled weight decay regularization, and has been shown to generalize well across a range of image classification tasks. A layer-wise learning rate strategy is applied, where each layer in the network is assigned a different learning rate. Rates are distributed using geometric spacing between a minimum of 1×10^{-4} and a maximum of 1×10^{-3} , with lower rates assigned to earlier layers and higher rates to later layers. This preserves the general visual features learned during pretraining in the earlier layers, while allowing the later layers to adapt more strongly to the historical map domain (Hosseini et al., 2022). A learning rate scheduler progressively reduces the learning rate during training, preventing large weight updates in later epochs when the model is close to convergence. The model checkpoint with the lowest validation loss is selected for inference.

3.1.5. Inference

The fine-tuned model is applied to the complete patch set covering all loaded map sheets. A patch dataset is constructed from the patch metadata CSV, and test-time transforms are applied consistently with those used during training. The ClassifierContainer produces a predicted label and a confidence score for each patch. Predictions are saved to a CSV file and joined to the spatial patch metadata, producing a georeferenced patch dataset in which each patch polygon carries its predicted class label as an attribute. The output is exported as a GeoJSON file to enable spatial analysis and visualization in GIS software.

3.1.6. Post-processing

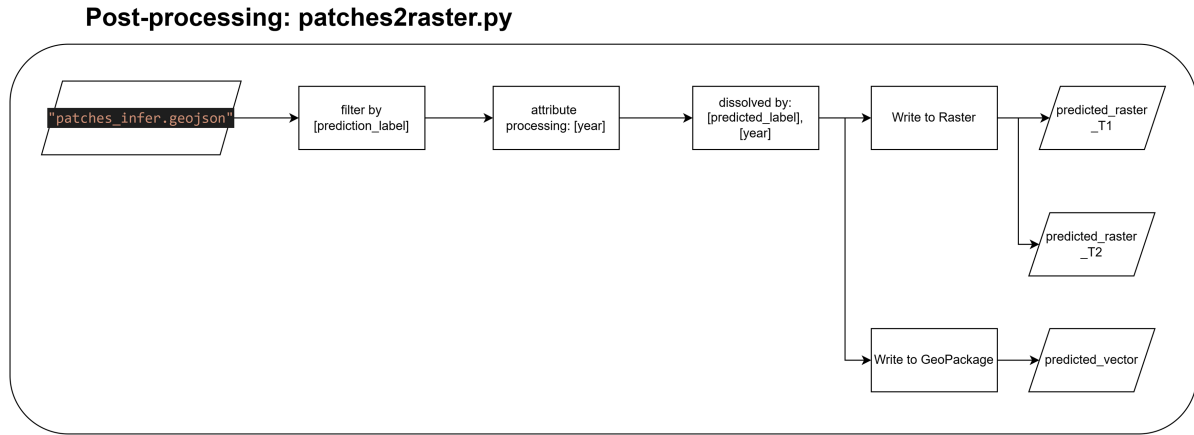


Figure 3.6: Post-processing to generate raster.

The inference output GeoJSON is processed through a custom post-processing script to produce classified raster outputs per temporal period, suitable for post-classification change detection – an approach in which independently produced classified maps from two temporal periods are compared to identify land cover change (Singh, 1989).

First, patches predicted as background (**no**) are discarded, retaining only patches classified as **building_blocks** or **individual_buildings**. An optional confidence threshold can be applied at this stage to discard low-confidence predictions before any spatial processing is performed. Second, a survey year and edition number are extracted from each patch’s parent sheet filename, enabling temporal attribution without a separate metadata lookup. Third, the filtered patches are dissolved by predicted label and survey year, merging spatially adjacent patches that share the same class and temporal period into unified polygons. The dissolved output is saved as a GeoPackage for vector-based inspection.

Finally, the dissolved polygons are rasterized to a GeoTIFF at a resolution of 200 meters in EPSG:28992, with pixel values encoding the predicted class directly. Producing consistent raster outputs with the same class schema and encoding across temporal periods is a prerequisite for reliable post-classification comparison (Hussain et al., 2013). This process is applied twice, once for the first edition of each sheet (T1) and once for the latest available edition (T2), following the Bonnebladen’s multi-edition structure in which individual sheets were resurveyed and revised independently over time. Rather than selecting a fixed calendar year, this edition-based approach ensures that T1 and T2 each represent the earliest and most recent available survey state per location, respectively. The two resulting rasters serve as the primary input to the change analysis described in

3.1.7. Evaluation

Model performance is evaluated on the held-out test set using precision, recall, and F1-score, computed per class and as macro-and micro-averages across all classes, following the evaluation protocol of Hosseini et al. (2022). Accuracy is not used as a primary metric because it has limited expressiveness under class imbalance, tending to reflect the performance of the dominant class rather than the minority classes of interest (Heitzler & Hurni, 2020).

Precision and recall measure the correctness and completeness of positive predictions for each class c , respectively:

3.1. Method 1: Patch-level classification

$$P_c = \frac{TP_c}{TP_c + FP_c} \quad (3.2)$$

$$R_c = \frac{TP_c}{TP_c + FN_c} \quad (3.3)$$

where TP_c , FP_c , and FN_c denote the true positives, false positives, and false negatives for class c . The F1-score combines precision and recall as their harmonic mean:

$$F_{1,c} = 2 \cdot \frac{P_c \cdot R_c}{P_c + R_c} \quad (3.4)$$

The macro-averaged F1-score is the unweighted mean of per-class F1-scores across all C classes:

$$F_{1,\text{macro}} = \frac{1}{C} \sum_{c=1}^C F_{1,c} \quad (3.5)$$

The micro-averaged F1-score aggregates contributions globally before computing the ratio:

$$F_{1,\text{micro}} = \frac{2 \sum_c TP_c}{2 \sum_c TP_c + \sum_c FP_c + \sum_c FN_c} \quad (3.6)$$

Macro-averaging treats all classes equally regardless of their support, making it sensitive to performance on minority classes such as **individual_buildings**. Micro-averaging weights each instance equally and therefore reflects the dominant **no** class more strongly. Both averages are reported to provide a complete characterization of model behavior across the imbalanced class distribution.

All metrics range from 0 to 1, where 1 indicates perfect performance. A high precision means few false positives; a high recall means few missed detections. The F1-score reflects their balance. Due to class imbalance, the macro-averaged F1 is the primary indicator of performance across all classes, while the micro-averaged F1 is expected to be higher as it is dominated by the **no** class and should not be interpreted in isolation.

Beyond quantitative metrics, a qualitative inspection of inference outputs is conducted by overlaying predicted patch polygons on the source map sheets in QGIS, following the visual inspection approach described by Heitzler & Hurni (2020). This assessment evaluates the spatial coherence of predictions and identifies systematic errors related to cartographic symbols that visually resemble buildings. As an additional external validation, predicted urban patches are compared against a reference HGN1900dataset to assess spatial correspondence beyond the annotated test set.

3.2. Method 2: Object-level classification

Method 2 extracts individual building footprints as vector polygons from the Rotterdam 1897 municipality map series using a four-stage pipeline: (1) data acquisition and preparation, (2) synthetic training data generation via CycleGAN, (3) instance segmentation via Mask R-CNN, and (4) post-processing. In the following sections, the complete pipeline from Method 2 is described.

3.2.1. Data acquisition and preparation

Method 2 uses two primary datasets: the Rotterdam 1897 municipality maps and the current BGT vector data, both described in Section 4.1.

The historical map sheets are retrieved using the Allmaps CLI⁶, which exports each sheet as a georeferenced GeoTIFF by applying the georeference annotations stored on the Allmaps platform (Meijers & Schoonman, 2025). Of the 10 sheets in the series, only sheets 2, 3, and 4 are used. These three sheets cover areas with dense building content, making them more suitable for training a building segmentation model. The remaining sheets are dominated by water bodies and open land.

BGT data are downloaded via the BGT Downloader plugin in QGIS. The relevant layers selected for this research are:

1. *Pand* (buildings)
2. *Wegdeel* (road surfaces)
3. *Waterdeel* (water bodies)
4. *BegroeidTerreindeel* (vegetated areas)
5. *OnbegroeidTerreindeel* (bare surfaces)

While the primary segmentation target is buildings, the background layers are included because they form the visual context that looks similar to the historical map sheets. Then, all downloaded data are inspected in QGIS to verify spatial coverage, alignment with the historical map extent, and layer completeness. The Allmaps XYZ Tile server is additionally used as an alternative to directly fetch IIF resources as reference background in QGIS, without any preprocessing steps.

3.2.2. Georeference

The Rotterdam 1897 map sheets are available as georeferenced GeoTIFFs produced by the Allmaps CLI, using IIF georeferencing annotation maintained by the TU Delft Library. The warping procedure derives a geotransform from the stored ground control points (GCPs), assigning real-world coordinates in EPSG:3857 to each pixel of the map scan.

While the maps are already georeferenced, the accuracy of the existing annotation is not known. For urban change analysis at the building level, precise spatial alignment between the historical map and current reference data is critical. A misalignment of even a few meters can cause a building footprint from 1897 to appear shifted relative to the BGT, producing false positives or false negatives in the change detection step.

A visual inspection is therefore conducted in QGIS by overlaying the georeferenced sheets with the BGT *Pand* layer, filtered to buildings constructed before 1897 using *bouwjaar* attribute field. This filter isolates buildings likely to appear in both the historical map and the current BGT dataset, providing reliable visual anchors for alignment assessment. The qualitative result of this inspection is reported in Section 5.2.1.

Based on the visual inspection, improvement to the georeference quality is applied manually to sheet 3, which covers the Rotterdam city center and the northern peri-urban area and is the target sheet for urban change analysis.

Ground control points are collected from the filtered BGT *Pand* layer. For each GCP, a visually identifiable feature is selected on the historical map, primarily building corners at street intersections, and matched to the corresponding location in the BGT.

Thin Plate Spline (TPS) is selected as the transformation model. TPS is a locally adaptive method that passes exactly through every GCP, interpolating smoothly between control points.

⁶<https://github.com/allmaps/allmaps/tree/main/apps/cli>

3.2. Method 2: Object-level classification

This behavior makes it well-suited for historical maps, which often contain local geometric distortions that cannot be corrected by global polynomial transformations (Herrault et al., 2013). Because TPS residuals at GCP locations are mathematically zero by definition, they cannot be used as an accuracy metric. Independent Check Point (ICPs) provide meaningful quantitative measure of georeferencing accuracy, reported in Section 5.2.1.

3.2.3. Synthetic data generation (CycleGAN)

Instance segmentation models require annotated training data in the form of pixel-level masks paired with source images. Manual annotation of individual building footprints is time-consuming and impractical at scale. To address this, Method 2 adopts a synthetic data bootstrapping approach following Arzoumanidis et al. (2025), where modern vector data is style-transferred to visually resemble the historical maps, generating a synthetic training dataset with automatically aligned annotation masks. The pipeline consists of two main steps: (1) cartographic style transfer, which renders the BGT vector data to approximate the visual appearance of the historical maps, and (2) unpaired image-to-image translation using CycleGAN (Zhu et al., 2017), which adds realistic cartographic degradation effects to the style-transferred images. The BGT vector geometry simultaneously serves as the annotation mask, eliminating the need for manual labelling.

Cartographic style transfer

The BGT layers are rendered in a cartographic style approximating the Rotterdam 1897 maps. To determine the target fill and stroke colors per class, a set of class-representative image patches is manually selected from visually clean areas of the historical sheets for each feature class, avoiding regions with scan artifacts, text labels, or class boundaries. The mean RGB value per class is computed from these patches using a custom Python script, following the visual inspection approach described by Arzoumanidis et al. (2025). The resulting color values (in hexcodes) are applied as fill and stroke colors to the corresponding BGT layers in QGIS.

The Rotterdam 1897 map series distinguishes two building types: (1) **regular buildings**, rendered with a solid red fill, and (2) **hatched buildings**, rendered with a diagonal hatching pattern over the same red fill. Visual inspection of the historical sheets indicates that hatched buildings consistently correspond to public and civic functions, including churches, schools, markets, prisons, and government offices. Example of the text overlays in Dutch are as follows: *Cellulaire Gevangenis, O.S (Public School), O.B.S (Nursery School), Societeit Harmonie (Doele), Zuider kerk, Post en Telegraaf kantoor, Zeevischmarkt, Kapel, Barakbesrn: zieken, Exercitie-loods, Openbaar Slaachthuis, Goedernloods*. To represent this distinction in the synthetic training data, the BGT *Pand* layer is split into two subsets using a SQL filter on the *gebruiksdoe1* attribute, which classifies each building by its intended use. Buildings with a meeting, industrial, or office function are assigned the hatching style; all remaining buildings receive the solid fill style.

The styled BGT map is exported from QGIS using PyQGIS export script as a PNG image covering the spatial extent of each selected sheet.

Data preprocessing and patch tiling

The styled synthetic PNGs and the historical GeoTIFFs are sliced into fixed-size patches using a custom tiling script. Patches are extracted at 512 x 512 pixels with a stride of 256 pixels, producing a 50% spatial overlap between adjacent patches (Figure 3.7). This overlap strategy increases dataset diversity and ensures that features near patch boundaries appear in multiple contexts during training.

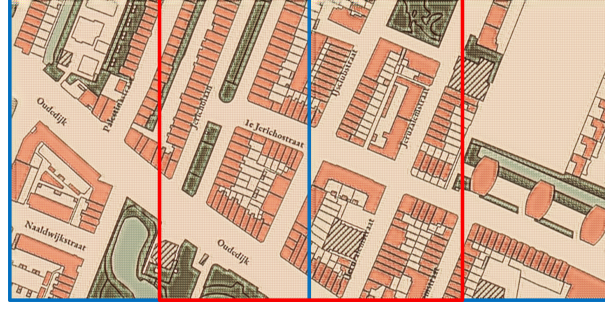


Figure 3.7: Three sequential tiles with size of 512x512 pixel and stride 256 pixel (50% overlap).

The dataset is split by sheet: sheets 2 and 3 form the training set (trainA: synthetic, trainB: historical), and sheet 4 forms the test set (testA: synthetic, testB: historical). This spatial split ensures evaluation on a geographically unseen area.

CycleGAN image translation

CycleGAN (Zhu et al., 2017) is an unpaired image-to-image translation framework that learns mappings between two visual domains X and Y without requiring paired training examples. The framework consists of two generators, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and two discriminators, D_X and D_Y . In this research, domain X corresponds to the style-transferred BGT patches and domain Y corresponds to the historical Rotterdam 1897 map patches. The translation $G(x)$ produces a synthetic historical map patch from a styled BGT input, adding cartographic aging and degradation effects while preserving building geometry.

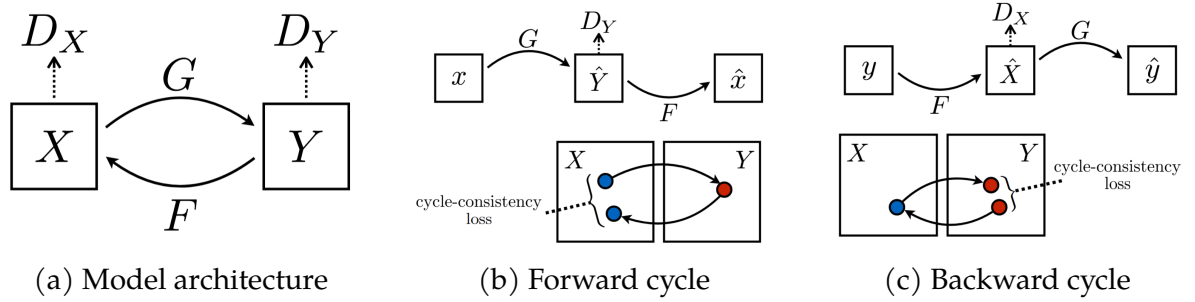


Figure 3.8: CycleGAN architecture (Zhu et al., 2017).

Figure 3.8a illustrates two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ with discriminators D_Y and D_X , where X denotes the style-transferred BGT domain and Y denotes the historical Rotterdam 1897 map domain. Figure 3.8b shows the forward cycle-consistency: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. Figure 3.8c shows the backward cycle-consistency: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.

The training objective combines two loss terms, following the formulation of Zhu et al. (2017). The adversarial loss encourages each generator to produce outputs indistinguishable from the target domain:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_y[\log D_Y(y)] + \mathbb{E}_x[\log(1 - D_Y(G(x)))] \quad (3.7)$$

The cycle consistency loss enforces that translating an image to the target domain and back recovers the original:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_x[\|F(G(x)) - x\|_1] + \mathbb{E}_y[\|G(F(y)) - y\|_1] \quad (3.8)$$

The full training objective is:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \quad (3.9)$$

3.2. Method 2: Object-level classification

where $\lambda = 10$ controls the relative weight of the cycle consistency term. The model is trained using the official PyTorch implementation of Zhu et al. with a ResNet-based generator (9 residual blocks), instance normalization, and a PatchGAN discriminator. Input patches are cropped to 256×256 pixels from the 512×512 source patches at training time. Training uses the Adam optimizer with a batch size of 1, a constant learning rate of 0.0002 for the first phase, followed by linear decay to zero (Zhu et al., 2017).

Evaluation of synthetic data quality

The quality of the translated synthetic images is evaluated using the Frechet Inception Distance or FID (Heusel et al., 2018), a widely adopted metric for assessing the realism of generative model outputs. FID measures the distance between the feature distributions of two image sets by fitting multivariate Gaussian distributions to the activations of the final pooling layer of a pre-trained Inception-v3 network. A lower FID indicates greater similarity between the synthetic and reference distributions. The score is computed as:

$$\text{FID} = \|\mu_X - \mu_Y\|^2 + \text{Tr}\left(\Sigma_X + \Sigma_Y - 2(\Sigma_X \Sigma_Y)^{\frac{1}{2}}\right) \quad (3.10)$$

where μ_X, Σ_X and μ_Y, Σ_Y are the mean and covariance of the feature embeddings of the synthetic and real image sets respectively (Heusel et al., 2018).

A key consideration specific to this research is the choice of training checkpoint for inference. As noted by Arzoumanidis et al. (2025), early training epochs introduce cartographic degradation effects at the texture and color level without inducing geometric distortions of building boundaries, which would corrupt the alignment between synthetic image and annotation mask. To identify the optimal checkpoint, inference is run at multiple saved epochs and each output is evaluated using FID against the historical test patches (`testB`), supplemented by visual inspection of building geometry preservation. The checkpoint yielding the best balance between FID score and geometric fidelity is selected for generating the final synthetic training dataset. Results of this evaluation are reported in Chapter 5.

Post-processing and output

CycleGAN inference produces individual translated patches (`fake_B`) as PNG files. These are stitched into a full-sheet composite image using a custom Python script, which places each patch at its pixel-origin position encoded in the filename and averages overlapping regions to reduce seam artifacts. The stitched composite is georeferenced using the CRS and affine geotransform of the corresponding historical GeoTIFF for the same sheet, producing a spatially referenced GeoTIFF per sheet. These files serve as input to the Mask R-CNN pipeline described in Section 3.2.4.

3.2.4. Instance segmentation (Mask R-CNN)

Mask R-CNN (He et al., 2018) is an instance segmentation framework that extends Faster R-CNN by adding a third parallel branch to the network head. While the original Faster R-CNN produces a class label and a bounding-box offset for each candidate object, Mask R-CNN adds a fully convolutional mask branch that simultaneously predicts a binary pixel-level mask for each detected instance as illustrated in Figure 3.9. Feature extraction uses a ResNet-FPN (Feature Pyramid Network) backbone, which builds a multi-scale feature representation from a single input image, and RoIAlign, which resolves the spatial misalignment of the earlier RoIPool operation by preserving exact pixel coordinates of each region of interest.

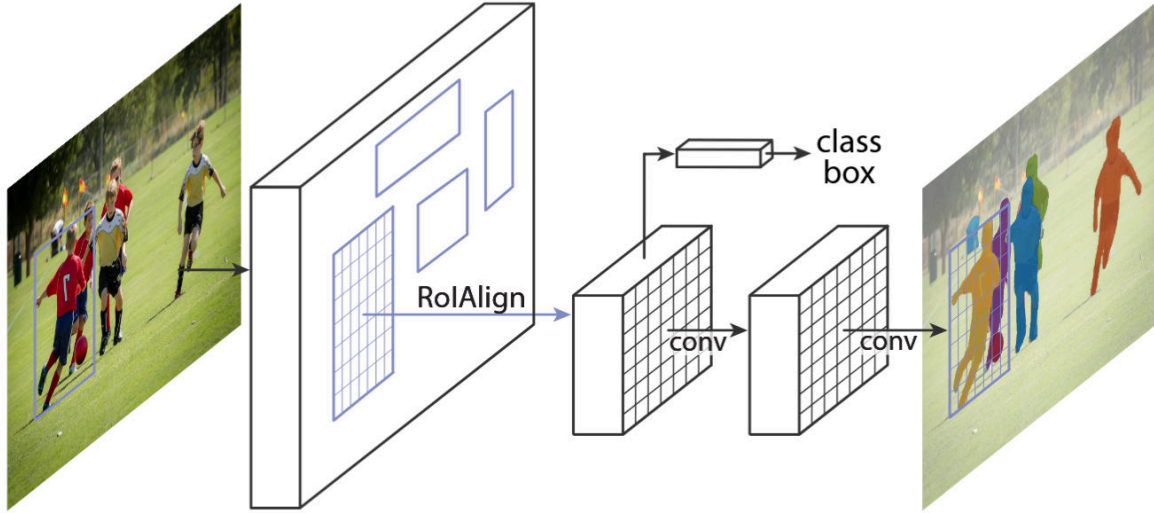


Figure 3.9: Mask R-CNN framework for instance segmentation, source: (He et al., 2018)

Instance segmentation is a particularly suitable approach for extracting individual building footprints from the Rotterdam 1897 maps. Unlike semantic segmentation, which assigns a class label to each pixel and requires a subsequent vectorization step to convert the raster output into usable polygon geometries, Mask R-CNN produces a separate binary mask per detected instance in a single forward pass (He et al., 2018). These masks are directly convertible to vector polygons, reducing the post-processing overhead that typically follows pixel-level classification methods applied to historical maps (Heitzler & Hurni, 2020).

The full Mask R-CNN pipeline for this research consists of four steps, described in the following sub-sections: data preparation, model training and evaluation, inference, and post-processing.

Data preparation

The input to the Mask R-CNN pipeline consists of two datasets: (1) the CycleGAN-translated synthetic GeoTIFFs (*fake_B*, sheets 2, 3, and 4) produced as the source images, and (2) the BGT *Pand* polygon layer as the source of annotation masks. All data are kept in EPSG:3857 throughout the preparation pipeline. Converting raster data between coordinate systems requires pixel resampling (e.g. bilinear interpolation), which blurs sharp ink edges typical of printed map sheets. Keeping a single CRS avoids this artifact.

The BGT *Pand* GPKG, originally in EPSG:28992 (RD New), is reprojected to EPSG:3857 in QGIS. The three CycleGAN GeoTIFFs are then merged into a single mosaic and clipped to the Rotterdam bounding box. This produces a single spatially consistent raster covering the study area.

Patches of 512 x 512 pixels are extracted from the clipped mosaic using a custom tiling script. A spatial split by sheet is applied, following the same strategy used for CycleGAN: sheets 2 and 3 form the training and validation set, and sheet 4 forms the test set. This ensures that the test tiles cover a geographically distinct area from the training data. Tiles containing no building instances are excluded. The resulting tile counts per split are given in Table 1.

Table 1: Tile counts per dataset split for Mask R-CNN training.

Split	Sheets	Tiles
Train	2, 3	291
Validation	2, 3	62
Test	4	63

The BGT building polygons are converted to COCO-format annotation JSON files using a custom script. COCO (Common Objects in Context) is a widely used annotation standard that stores instance-level labels as a JSON file containing image metadata, bounding boxes, class labels, and per-instance polygon masks (Lin et al., 2015). Each tile receives an annotation file listing all building polygons that intersect that tile, represented as instance masks. Two building classes are used, consistent with the distinction defined in Section 3.2.4 as building (class label 1) and building_hatching (class label 2). Class label 0 is reserved for background.

```
{
  "info": {"description": "Rotterdam 1897 buildings - test"},
  "categories": [
    {"id": 1, "name": "building", "supercategory": "building"},
    {"id": 2, "name": "building_hatching", "supercategory": "building"}
  ],
  "images": [
    {"id": 15, "file_name": "test/tile_01536_12800.tif",
     "width": 512, "height": 512}
  ],
  "annotations": [
    {
      "id": 3, "image_id": 15, "category_id": 1,
      "segmentation": [[250.58, 391.89, 251.36, 389.98, ..., 250.58, 391.89]],
      "bbox": [245.24, 389.98, 23.58, 22.13],
      "area": 304.98, "iscrowd": 0
    },
    {
      "id": 1, "image_id": 15, "category_id": 2,
      "segmentation": [[186.34, 509.95, 187.95, 505.73, ..., 186.34, 509.95]],
      "bbox": [178.25, 483.07, 48.05, 28.93],
      "area": 854.15, "iscrowd": 0
    }
  ]
}
```

Listing 1: Each annotation stores the image reference (image_id), class label (category_id), instance polygon (segmentation), bounding box (bbox), and pixel area. Two classes are defined: building (id: 1) and building_hatching (id: 2). Segmentation coordinates abbreviated.

Model training and evaluation

The model used is maskrcnn_resnet50_fpn, a Mask R-CNN with a ResNet-50 backbone and FPN (Feature Pyramid Network) head. This configuration provides a good balance between detection accuracy and computational cost, and is the standard implementation available in the PyTorch Torchvision library. The model is initialized with weights pretrained on the COCO dataset, applying transfer learning to accelerate convergence on the small Rotterdam training set. The classification and mask prediction heads are replaced to output num_classes = 3 (background, building, building_hatching).

The Region Proposal Network (RPN) is the first stage of Mask R-CNN, which scans the feature map and proposes candidate bounding boxes (called region proposals) that may contain

objects. These proposals are defined at pre-set anchor sizes, which are reference bounding box dimensions at multiple scales placed at each spatial position in the feature map. The default COCO anchor sizes are optimized for large objects in natural images and are too large for the small, dense buildings in the Rotterdam maps. The anchor sizes are therefore reduced to (8, 16, 32, 64, 128) pixels to improve proposal recall for small building instances. The maximum number of detections per image is set to 512 to provide sufficient capacity for dense tile areas.

The model is trained using Stochastic Gradient Descent (SGD), an optimization algorithm that updates model weights by computing the gradient of the loss over a mini-batch of training samples. A linear warmup is applied for the initial iterations to stabilize early training, followed by a StepLR decay schedule that reduces the learning rate at fixed epoch intervals to refine the model as training progresses.

During training, the model is evaluated on the validation set after each epoch using mAP50 (mean Average Precision at IoU threshold 0.5). IoU (Intersection over Union) measures the spatial overlap between a predicted bounding box and the ground truth box as the ratio of their intersection area to their union area. mAP50 computes the mean of the per-class Average Precision scores at an IoU threshold of 0.5, summarizing the model’s precision-recall trade-off across all classes. The best model checkpoint is saved when the validation mAP50 improves. Training progress is logged using Weights and Biases (WandB). The quantitative training results are reported in Section 5.2.3.

Inference

Inference is applied to the historical Rotterdam 1897 map sheets (sheets 2, 3, and 4) using the best model checkpoint selected during training. The historical GeoTIFFs are used as input at this stage, not the CycleGAN synthetic images, as the goal is to detect buildings in the original historical map scans.

A sliding window approach is used to process each full-sheet GeoTIFF as a sequence of partially overlapping tiles. Each tile is 512 × 512 pixels with a 128-pixel overlap between adjacent tiles. The overlap reduces the number of buildings split at tile boundaries, which would otherwise produce partial predictions.

For each tile, the model outputs bounding boxes, class labels, confidence scores, and per-instance binary masks. Predictions below a confidence score threshold are discarded. Non-maximum suppression (NMS) is then applied to remove duplicate predictions from overlapping tiles. NMS retains the highest-confidence prediction among any group of predictions with IoU above a set threshold, discarding the lower-confidence duplicates. Box-level NMS (using `torchvision.ops.nms`) is used rather than polygon-level NMS for computational efficiency. The predicted binary masks are converted to polygon geometries and saved as a GeoPackage in EPSG:28992. The inference results are reported in Section 5.2.3.

Post-processing

The post-processing pipeline applies the following steps in order. Parameter values are subject to tuning and are reported in Section 5.2.3.

Area filtering. Polygons smaller than a minimum area threshold are removed to eliminate sliver polygons from map text characters (e.g. circular letters such as ‘O’ and ‘a’). Polygons larger than a maximum area threshold are removed to discard large false positives caused by predictions at the edges of the map scan.

Duplicate removal. Duplicate predictions from overlapping tiles are removed using IoU-based NMS. When two predictions have an IoU above 0.3, the smaller polygon is discarded. A

3.2. Method 2: Object-level classification

secondary IoS (Intersection over Smaller) cleanup step is then applied: a small polygon that has a large fraction of its area contained within a larger polygon is discarded, addressing cases where a sub-building fragment is predicted inside a correctly detected building.

Shape regularization. The Building-Regulariser tool⁷ is applied to orthogonalize the jagged pixel-traced polygon boundaries into rectilinear or circular building shapes, as illustrated in Figure Figure 3.10. Final parameter values are subject to tuning and are reported in Section 5.2.3.

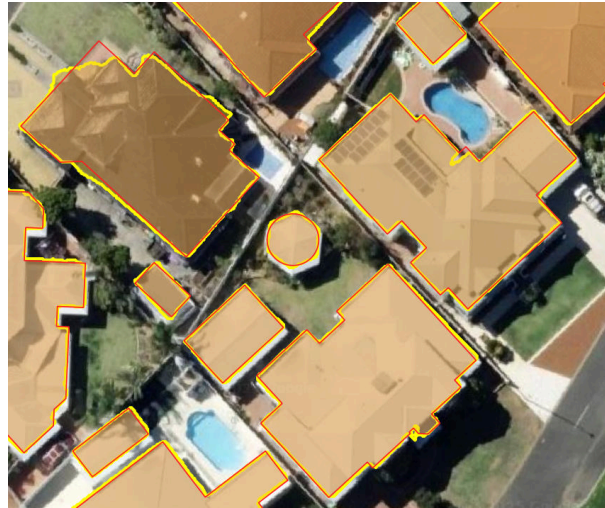


Figure 3.10: Before (yellow line) and after regularization (red line). Source: DPIRD-DMA (2024).

An external quantitative evaluation of the post-processed output is performed by comparing predictions against a BGT reference dataset. The evaluation methodology is defined as part of the urban change analysis in Section 3.3.2.

3.3. Urban Change Analysis

Urban change is analyzed by comparing the urban extents extracted from T1 and T2 for each method. For Method 1, T1 represents the earliest available Bonnebladen edition (editions 1869–1916); T2 represents the most recent edition (editions 1910–1943). For Method 2, T1 represents the 1897, the published date of Rotterdam municipality map; T2 represent the recent building vector dataset from BGT.

Two approaches are applied, one per method: a raster-based post-classification comparison for Method 1 (Section 3.3.1), and a vector overlay approach for Method 2 (Section 3.3.2).

3.3.1. Method 1: Post-classification change analysis

Method 1 applies post-classification comparison (PCC) to detect urban change between T1 and T2 (Hussain et al., 2013; Singh, 1989). In PCC, two independently classified rasters are compared pixel by pixel to produce a “from-to” change map. This approach is suited here because T1 and T2 were classified separately from different map editions. PCC does not require normalization of radiometric or stylistic differences between editions before comparison (Singh, 1989).

A known limitation of PCC is error propagation: classification errors in either edition produce false change signals. The overall change map accuracy is approximately the product of the two individual classification accuracies (Singh, 1989). This is discussed further in Section 5.4.1.

⁷Building-Regulariser: <https://github.com/DPIRD-DMA/Building-Regulariser>

Change taxonomy. Each pixel is assigned a change class based on its T1 and T2 class values. Table 2 defines the six output classes.

Table 2: Change taxonomy for Method 1.

Output class	Label	T1 -> T2	Description
0	Background	0 -> 0	Non-urban in both editions
1	No change	1->1 / 2->2	Urban in both editions, same class
2	Expansion 1	0 -> 1	New scattered buildings
3	Expansion 2	0 -> 2	New building blocks
4	Densification	1 -> 2	Scattered buildings upgraded to blocks
5	Uncertainty / Demolition	1/2 -> 0	Urban class disappeared
6	Uncertainty / Downgrade	2 -> 1	Building blocks reduced to scattered

Classes 5 and 6 are flagged as uncertain. Large-scale demolition or systematic downgrading of building density is not expected between T1 and T2 at this spatial scale. These signals are more likely caused by classification errors in one or both editions. They are reported separately in Section 5.4.1 and excluded from the main urban growth analysis.

3.3.2. Method 2: Urban persistence analysis

Object-based change detection approaches compare detected building objects between two time periods to identify spatial changes (Chen et al., 2012; Hussain et al., 2013). A common building change taxonomy includes new construction, demolition, persistence, and geometric transformation (Matikainen et al., 2010).

Applying a full change taxonomy is not feasible in this context. Geometric transformation analysis requires comparable shape accuracy between T1 and T2, which is not satisfied here given the difference in geometric precision between a hand-surveyed historical map and a current national cadastral dataset. New building counts can be derived more reliably from cadastral “*bouwjaar*” attributes directly, without depending on building predictions that carry detection uncertainty. This analysis therefore applies a narrower scope: **urban persistence analysis**.

Definition. A T1 predicted building is defined as persistent if it can be spatially matched to a confirmed pre-existing building in T2. The persistence rate is the proportion of confirmed pre-existing buildings that are matched by a T1 prediction.

Matching procedure. Matching is performed as a one-to-one greedy spatial assignment. First, both datasets are filtered to near-rectangular buildings using an Oriented Bounding Box (OBB) fit ratio threshold. The OBB fit ratio is defined as:

$$\text{OBB fit ratio} = \frac{A_{\text{OBB}}}{A_{\text{polygon}}} \quad (3.11)$$

where A_{OBB} is the area of the minimum rotated bounding rectangle and A_{polygon} is the area of the original polygon. A ratio close to 1.0 indicates a near-rectangular shape. Higher values indicate complex shapes such as L- or T-shaped footprints, for which IoU-based matching is less reliable. Figure 3.11 illustrates the distinction between regular horizontal bounding box and oriented bounding box.

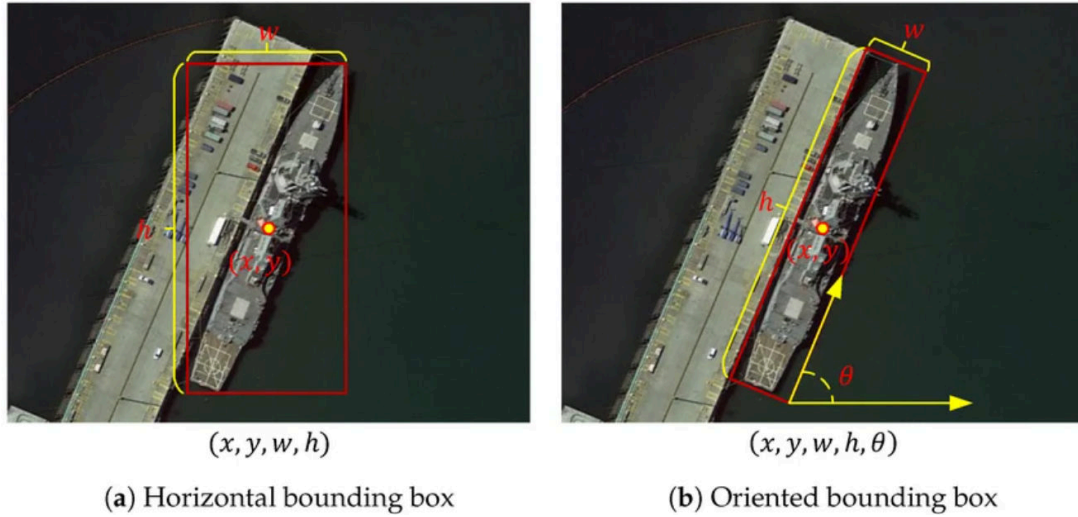


Figure 3.11: Illustration of oriented bounding box on ship detection use case, source: Gao et al. (2023)

For each T2 reference building, candidate T1 predictions within a centroid distance threshold are identified. For each candidate pair, the Intersection over Union (IoU) is computed as a shape similarity measure:

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.12)$$

where A and B are the two polygon geometries, $A \cap B$ is their intersection, and $A \cup B$ is their union. Figure 3.12 illustrates the concept. A relaxed IoU threshold is applied to account for positional offset from georeferencing uncertainty. All candidate pairs are sorted by IoU in descending order and assigned greedily: the highest-IoU pair is matched first, and each building can be matched at most once.

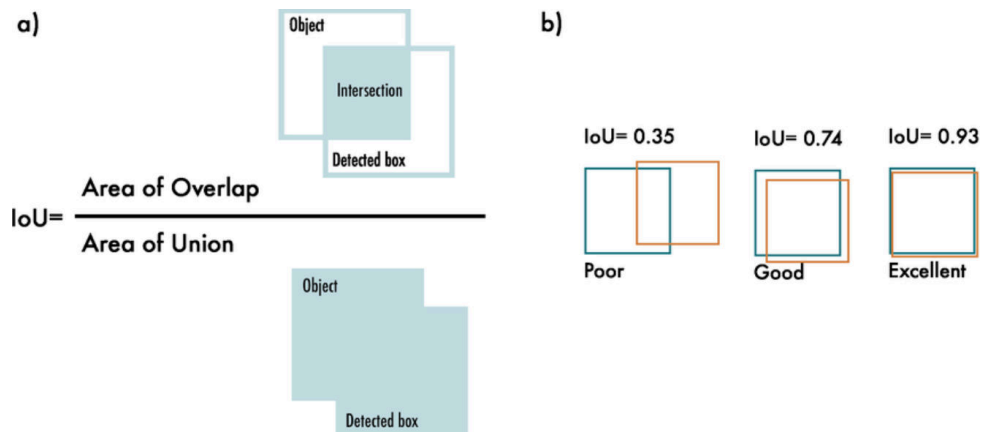


Figure 3.12: Illustration of Intersection over Union (IoU). (a) Definition of IoU, (b) Examples of three different IoU values. Source: Terven et al. (2023)

Output. Two classes are assigned based on the matching result:

- **Persistent:** a T1 prediction matched to a confirmed pre-existing T2 building.
- **Not detected:** a confirmed pre-existing T2 building with no matching T1 prediction.

Limitations. The matching cannot distinguish between a genuine demolition and a false positive prediction. A building demolished between T1 and T2 and rebuilt at the same location

may produce a false persistent match if the new footprint overlaps the prediction. This is a known limitation of comparing predictions against a current reference dataset.

3.3. *Urban Change Analysis*

4. Implementation

This chapter begins by outlining all the datasets required for conducting this research, including the historical maps and supporting datasets as described in Section 4.1. Section 4.2 describes the study area for each method. In addition, the tools and hardware used to implement and test the methodologies are also described in Section 4.3 and Section 4.4 respectively.

4.1. Datasets

As mentioned in the previous chapter, this research involves two main methodologies, each method requires different datasets. For Method 1, multi-temporal Dutch historical topographic maps or the *Bonnebladen* map series is used as the main dataset. Subsequently, historical land use data (*Historisch Grondgebruik Nederland* - HGN1900) are used as reference data for evaluation.

For Method 2, historical municipality maps (i.e. *Gemeente Rotterdam Plattegrond 1897* and *Brandgrens kaart*) and current building footprint data from *Basisregistratie Grootchalige Topografie* (BGT) are used as the main datasets.

Additionally, several supporting datasets are also used for the implementation of both methods, such as current administrative boundaries and topographic base maps. All datasets used in this research originate from their respective sources and are available through open-access means. The following sections provide a detailed description of each dataset.

4.1.1. Bonnebladen map series

Bonnebladen map series is the chromo-topographical map of the Kingdom of the Netherlands on the scale of 1:25,000, which was originally produced by the former *Topografisch Bureau* between 1865 to 1949 (Kadaster et al., 1940). The map series can be accessed from the Data Station Physical and Technical Sciences (DANS)⁸ repository in JPEG scans and GeoTIFF formats. Relevant materials such as data catalogue, map legends, and metadata are also available on the website. To access the dataset, users are required to create a DANS account and request access to the dataset. Example of a single map sheet of Delft area is shown in the Figure 4.1 below.

Alternatively, the same datasets are also available in the form of IIIF resources, which are organized and maintained by the TU Delft Library⁹.

Bonnebladen is selected as the primary dataset for Method 1 due to its nationwide coverage, multi-temporal structure, and consistent cartographic style. Further details on the sheet selection and temporal coverage used in this research are reported in Section 5.

4.1.2. HGN1900 (Historisch Grondgebruik Nederland)

The *Historisch Grondgebruik Nederland* (HGN1900)¹⁰ dataset is a historical land use dataset that provides information about the land use in the Netherlands around the year 1900 (Knol et al., 2004). The dataset is available in raster format with resolution 50 meters per pixel and georeferenced with the RD coordinate system. The final product of the dataset is a land use map with 10 classes, which are derived from a semi-automatic classification from the original

⁸<https://phys-techsciences.datastations.nl/dataset.xhtml?persistentId=doi:10.17026/dans-xqf-r4xf>

⁹<https://observablehq.com/@tudelft>

¹⁰<https://lgn.nl/bestanden>

4.1. Datasets

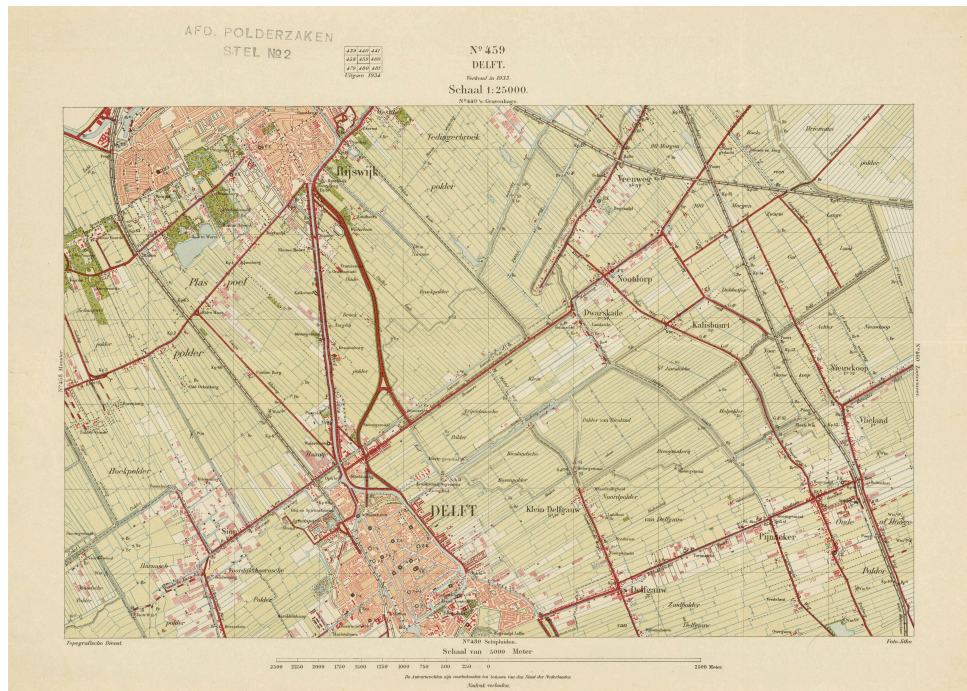


Figure 4.1: Single sheet of Bonnebladen - sheet no 449 - Delft

scanned Bonnebladen maps. The dataset is selected as reference data for Method 1 because it provides a comprehensive land use map of the Netherlands around the same period as the Bonnebladen maps, which allows for evaluation of the results.

4.1.3. Historical municipality maps of Rotterdam

Map of Rotterdam (1897) sheets are the topographical maps series (1:5000 map scale) obtained from the Rotterdam City Archives¹¹. The series contains 10 sheets covering the city of Rotterdam, published around 1897.

Additionally, these map series has been processed and georeferenced by the TU Delft Library¹² and AllMaps team, and made available as IIIF resources.

Following this, another historical maps from the city archives, the *Brandgrens kaart* (1940) or fire boundary map with 1:5000 map scale, is also used as a supporting dataset for Method 2. The map provides information about the extent of the fire that occurred in Rotterdam during World War II, particularly the bombing of May 14th 1940, which can be used to analyze the urban transformation before and after the war. Figure 4.2 and Figure 4.3 show the sample sheets of the Rotterdam municipality map and the fire boundary map respectively.

The dataset is selected as the main dataset for Method 2 because of its detailed representation of the city of Rotterdam in the late 19th century, which allows for the extraction of individual building footprints as objects, instead of building blocks as in other historical maps with lower map scale.

¹¹<https://stadsarchief.rotterdam.nl/zoek-alles?mivast=184&mizig=299&miadt=184&miview=ldt&milang=nl&micode=4201&minr=39521873&miaet=14>

¹²<https://observablehq.com/@allmaps/rtm-historical-atlas>

¹³<https://api.pdok.nl/lv/bgt/ogc/v1/>



Figure 4.2: Sample sheet of Rotterdam municipality map - sheet no 3

4.1.4. Basisregistratie Grootchalige Topografie (BGT)

The *Basisregistratie Grootchalige Topografie* (BGT)¹³ is a large-scale topographic dataset that provides detailed information about the built environment in the Netherlands. The dataset includes information about buildings, roads, water bodies, and other topographic features. The building footprints in the BGT dataset are represented as polygons with relevant attributes to



Figure 4.3: Fire boundary map of Rotterdam

4.1. Datasets

the research such as function and construction year. Another method to download the data is by using BGT Downloader plugin¹⁴ in QGIS.

4.2. Study Area

The province of South Holland is selected as the study area for Method 1 (Figure 4.4) due to the availability of a dense Bonnebladen collection covering the full province across multiple editions, providing sufficient temporal and spatial coverage for urban change analysis. Rotterdam is selected as the study area for Method 2 (Figure 4.5) due to the availability of the 1897 municipality map series at 1:5,000 scale, which provides the geometric detail required for building-level extraction. The two study areas overlap spatially and temporally, allowing a direct comparison of the two methods over the same geographic region. Both areas have also undergone significant urban transformation, making them well-suited for testing the urban change analysis component of the methodology.

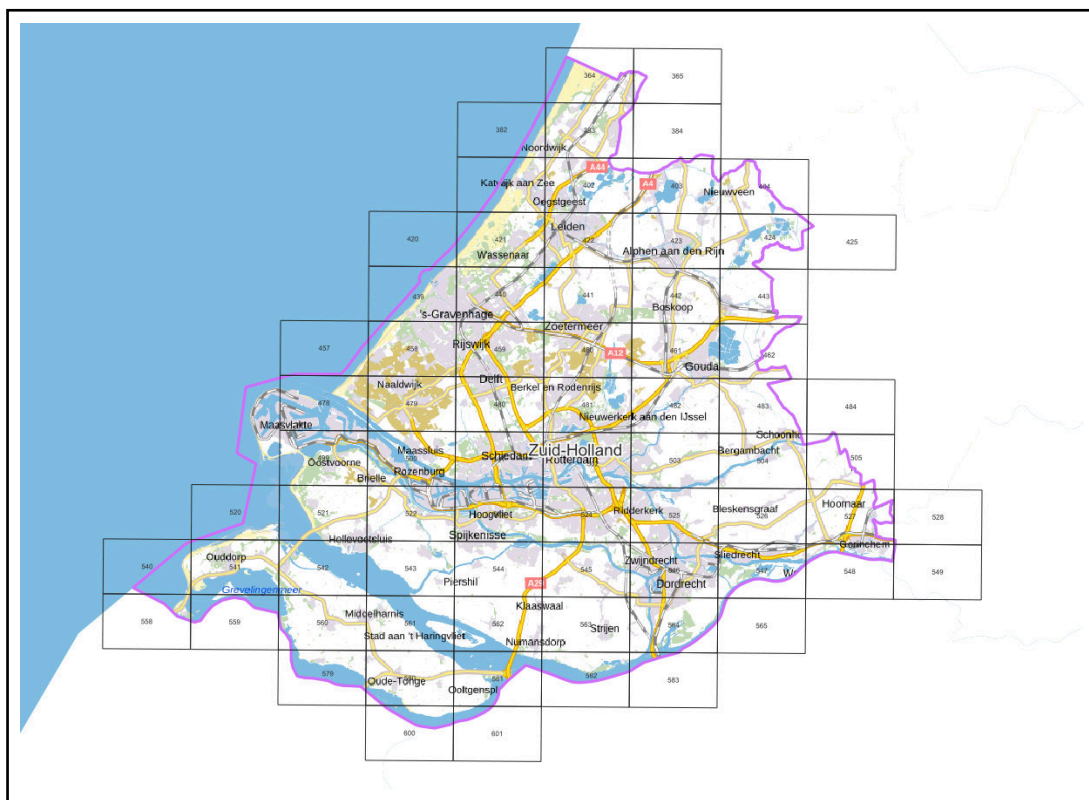


Figure 4.4: Study area for Method 1: Province South Holland

¹⁴https://plugins.qgis.org/plugins/bgt_downloader/

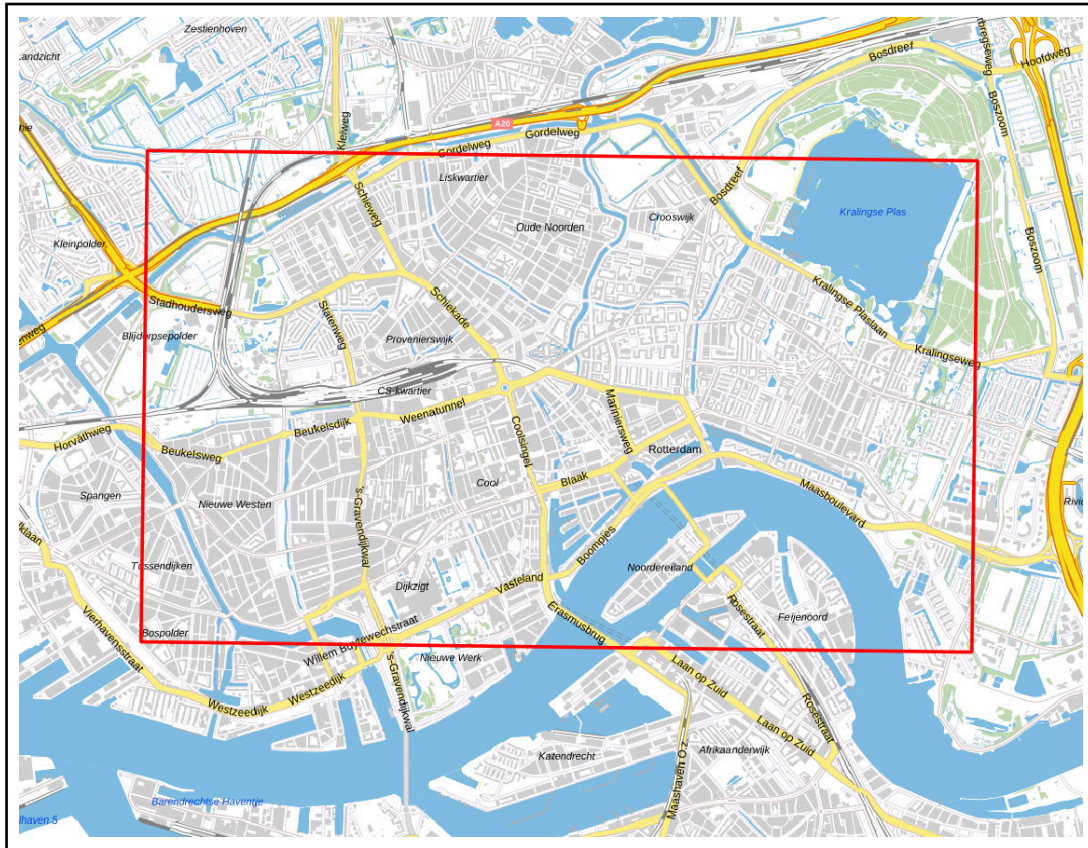


Figure 4.5: Study area for Method 2: part of Rotterdam city

4.3. Tools

The implementation in this research is written primarily in Python. It covers data preparation, model training, evaluation, and post-processing. The following sections provide a detailed description of the tools used in this research, including software, essential libraries and frameworks, and hardware.

4.3.1. Essential libraries and frameworks

- GDAL is an open-source geospatial library for reading, writing, and transforming raster and vector geospatial data formats.
- Rasterio is a Python library for reading and writing geospatial raster data, built on top of GDAL, used for GeoTIFF handling and spatial metadata access.
- Geopandas is a Python library extending Pandas for working with geospatial vector data, used for spatial operations, reprojections, and saving GeoPackage outputs.
- MapReader¹⁵ is a Python library for patch-based classification pipeline for large map collections, used for data loading, patch extraction, model training, inference, and evaluation.
- PyTorch is an open-source deep learning framework used to implement and train the CycleGAN image-to-image translation model and the Mask R-CNN instance segmentation model.
- Torchvision is a PyTorch companion library providing pre-trained models and utilities used here to load the Mask R-CNN architecture and apply non-maximum suppression.

¹⁵<https://github.com/maps-as-data/MapReader>

4.3. Tools

- pycocotools is a Python library for COCO dataset evaluation, used here to compute MAP50 metrics during Mask R-CNN training.
- Building-Regulariser¹⁶ is a post-processing tool used to regularize the building footprints extracted from the instance segmentation model.
- Weights and Biases (WandB)¹⁷ is a machine learning experiment tracking tool used to log and visualize the training process of the deep learning models, used here to monitor the training of both CycleGAN and Mask R-CNN models.
- AllMaps CLI¹⁸ is a command-line interface tool for programmatically accessing and manipulating the IIIF resources such as georeference annotations of the historical mapsheets. AllMaps XYZ Tiles API¹⁹ is also used to access the historical maps in the form of XYZ tiles for visualization and exploration purposes.

4.3.2. Software

- QGIS used for spatial data inspection, visualization, and manual verification of results. Additional plugins are also used for specific tasks including PDOK plugins for accessing BGT data.
- GIMP is an open-source raster image editor used for visual inspection of the map images.

4.4. Hardware

All implementation is performed on a laptop computer with the following hardware specifications:

- CPU Intel Core with 14 physical cores (20 logical cores)
- GPU NVidia Geforce RTX 4050 with 6GB VRAM, CUDA 12.7
- RAM 16GB
- OS Windows 11

¹⁶<https://github.com/DPIRD-DMA/Building-Regulariser>

¹⁷<https://wandb.ai/>

¹⁸<https://github.com/allmaps/allmaps/tree/develop/apps/cli>

¹⁹<https://observablehq.com/@allmaps/allmaps-tile-server>

5. Result and Analysis

This chapter presents the results of both classification methods and the subsequent change analysis. Section 5.1 covers Method 1, the patch-level classification applied to the Bonnebladen map series, including dataset assessment, pipeline execution, and evaluation. Section 5.2 covers Method 2, the object-level building footprint extraction applied to historical municipality maps. Urban change analysis derived from both methods is presented in Section 5.4.

5.1. Method 1 - Patch level classification

The following sections document the implementation and results of the patch-level pipeline using the MapReader framework. The assessment of the Bonnebladen dataset is reported first, as several dataset characteristics directly influenced design decisions in subsequent pipeline stages.

5.1.1. Assessment of bonnebladen map series

According to the sheet index (*bladwijzer*), the series comprises 776 sheet indices covering the entire Dutch territory as illustrated in Figure D.1. Of these, 709 indices correspond to actual map sheets; the remaining 67 cover areas that were still open water or sea at the time of survey, most notably the present-day province of Flevoland, for which no sheets were produced. Multiple editions were published for most sheets as content was resurveyed and revised over time. The combined catalog used in this research, derived from the TU Delft Library and Allmaps IIIF metadata, contains 2940 sheet editions in total across those 709 sheet indices.

Each sheet is identified by a numeric sheet number, a sheet name corresponding to the primary toponym in that area, and a publication year. In the catalog, year refers to the *Uitgave* date, that is, the date on which the edition was officially published. This is distinct from the *Verkend* date, which records when the field survey was conducted and can predate publication by years. The practical implication is that the year attribute used in this thesis reflects when each edition was made available, not necessarily when the depicted landscape was observed on the ground. This distinction is relevant when comparing results against external datasets: for instance, the HGN1900 land use dataset uses the *Verkend* date as the reference year for temporal attribution (Knol et al., 2004), which is discussed in detail in Section 5.1.7.

The filename convention for sheets downloaded via the custom pipeline follows the pattern `sheet_{NNNN}_{NAME}_{YEAR}.tif`, where NNNN is the zero-padded four-digit sheet number, NAME is the sheet toponym, and YEAR is the *Uitgave* year.

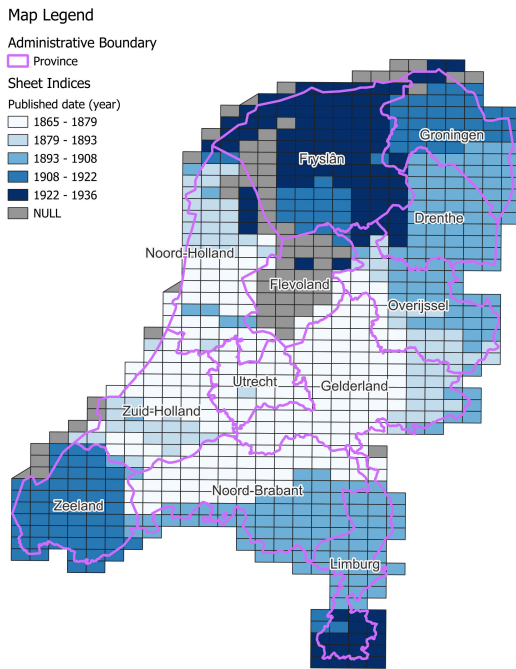
Temporal structure and editions

The term *edition* in the Bonnebladen context does not refer to a single revision of the entire map series, but to an individual resurvey of a specific sheet. According to the catalog, one sheet could be revised up to approximately 11 times across the production period, with each revision published as a new dated edition. The temporal coverage therefore varies per sheet: sheets in urbanising or strategically significant areas were revised more frequently than those in stable rural regions.

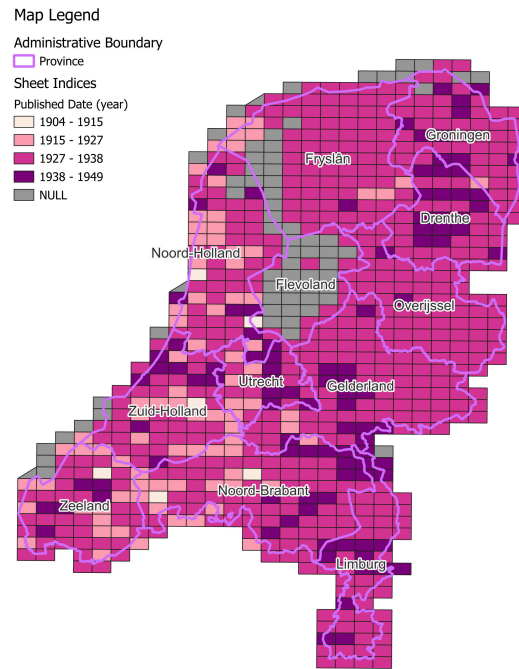
Four map sheet index figures are included below, showing for each sheet location across the Netherlands: the *Uitgave* year of the first available edition, the *Uitgave* year of the last available edition, the timespan between the two in years, and the total edition count per sheet. Three bar

5.1. Method 1 - Patch level classification

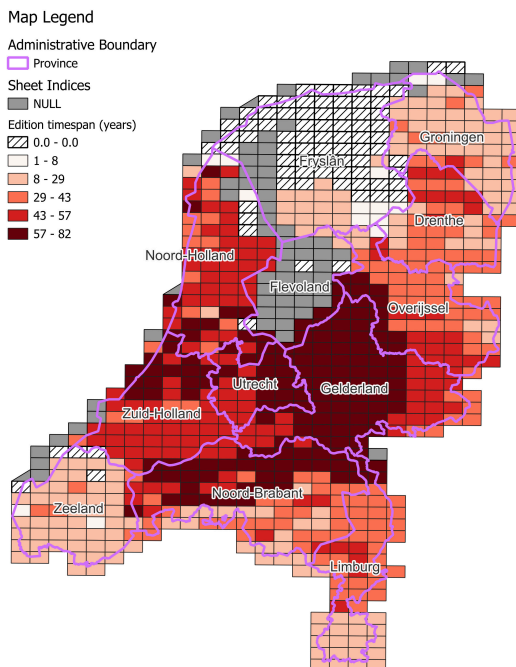
charts are additionally included summarising the number of sheets published per edition rank (Figure 5.2), cumulative sheet output per year (Figure 5.3), and publication counts grouped by edition (Figure 5.4).



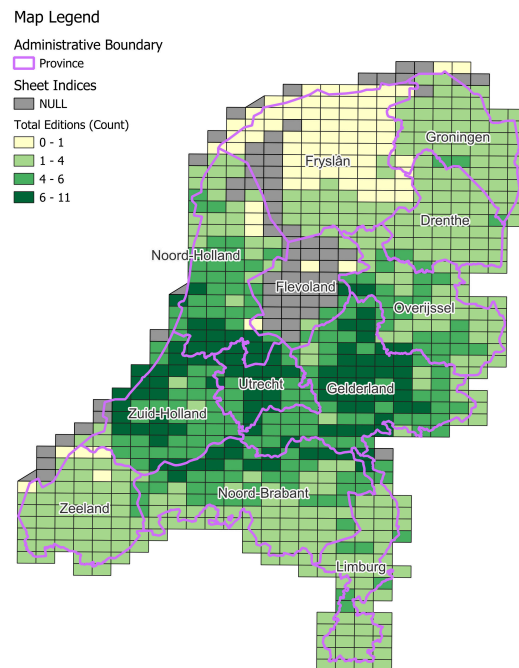
(a) Published date (*Uitgave*) of the first available edition per sheet.



(b) Published date (*Uitgave*) of the latest available edition per sheet.



(c) Timespan in years between the first and last available edition per sheet.



(d) Total available editions per sheet.

Figure 5.1: Spatial distribution of Bonnebladen edition metadata per sheet index.

5.1. Method 1 - Patch level classification

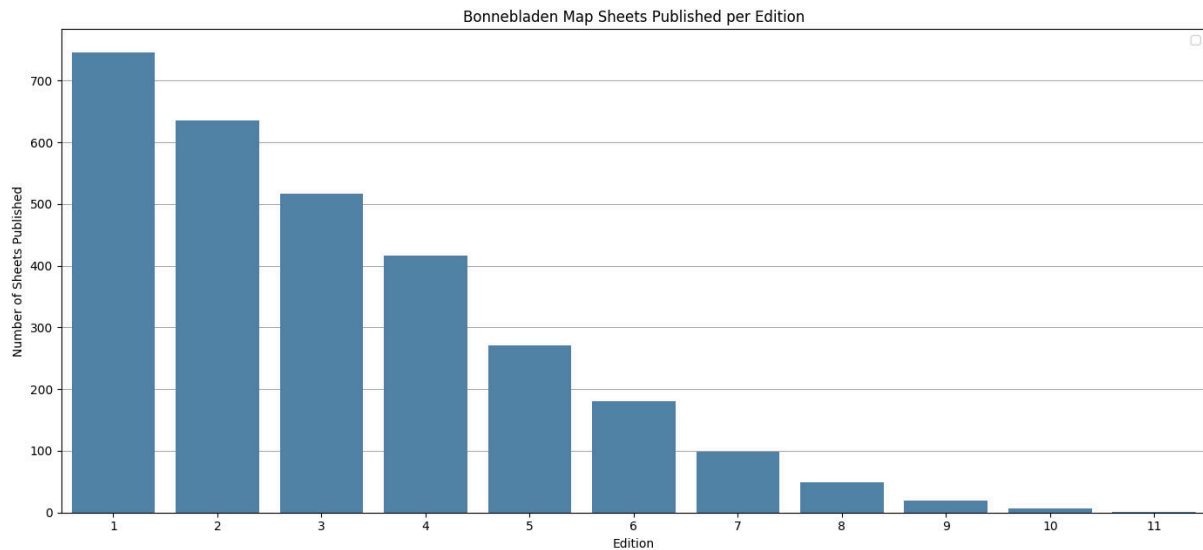


Figure 5.2: Number of sheets published per edition rank.

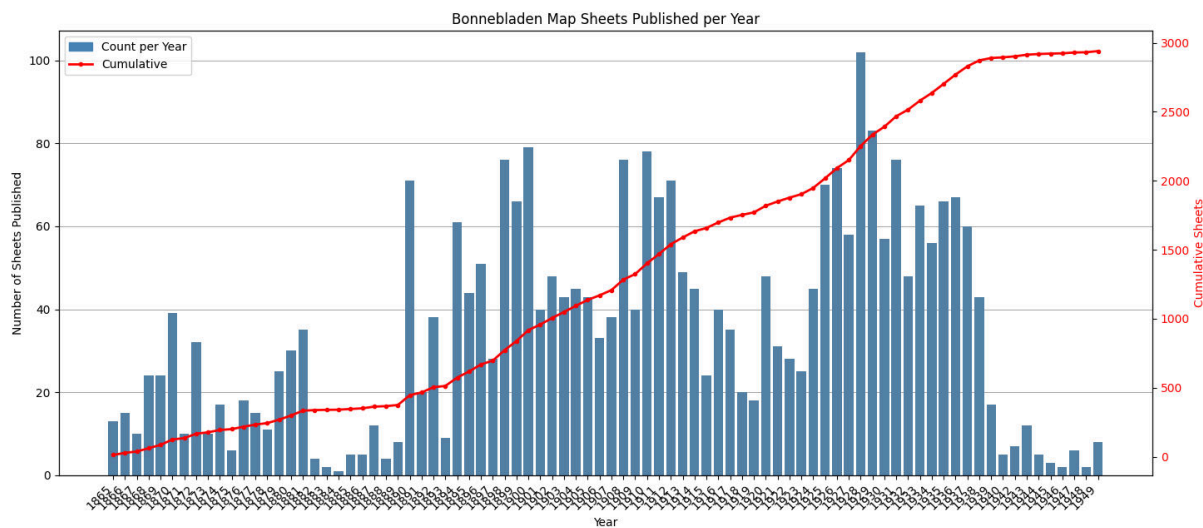


Figure 5.3: Number of sheets published per year (cumulative).

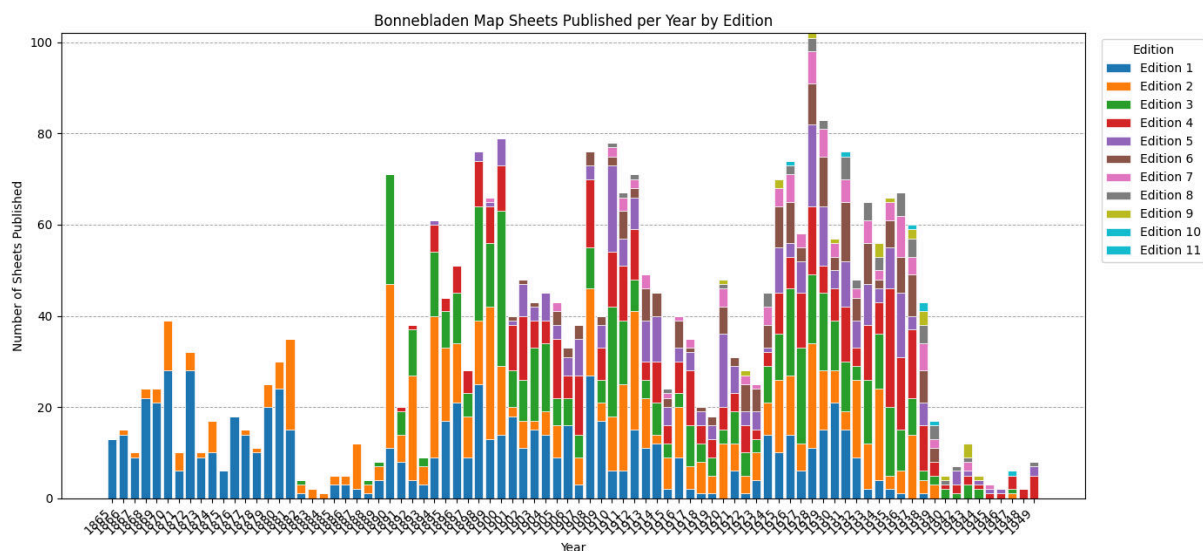


Figure 5.4: Number of sheets published per year and edition grouped.

Georeference

The Bonnebladen sheets are georeferenced programmatically using the map neatline corner technique, consistent with Meijers & Schoonman (2025). Each annotation encodes four GCPs at the neatline corners, from which an affine geotransform is derived.

To verify spatial consistency across editions, sheets 440 and 459 are inspected in both T1 and T2. As reported in Table 3, the corner coordinates are identical across editions for each sheet, confirming that T1 and T2 share the same GCP set and will produce consistent patch grids in the MapReader pipeline. Figure 5.5 illustrates the GCP placement on sheet 440.

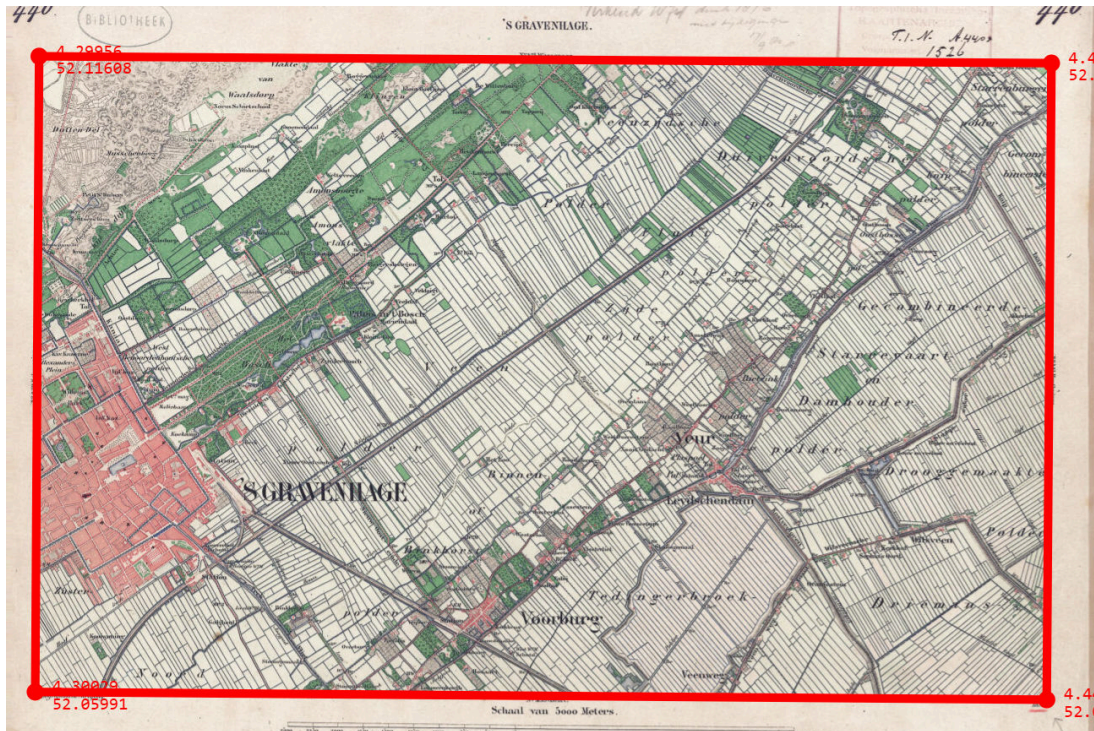


Figure 5.5: Ground control points on sheet's corner points for georeferencing (Sheet 440)

Table 3: Ground control points for sheet 440 and 459 from both edition T1 and T2.

Sheet Name	Longitude max	Latitude max	Longitude min	Latitude min
Sheet459_T2	4.446641	52.060532	4.30029	52.003736
Sheet459_T1	4.446641	52.060532	4.30029	52.003736
Sheet440_T1	4.446095	52.116704	4.299559	52.059908
Sheet440_T2	4.446095	52.116704	4.299559	52.059908

5.1.2. Data preparation

The study area for Method 1 is the province of South Holland. As shown in Figure 5.1, South Holland is among the best-represented provinces in the Bonnebladen catalog in terms of edition count, temporal range, and coverage continuity, making it the most suitable candidate for a multi-temporal analysis. The province also offers a range of settlement types relevant to the classification task, from the dense historic urban regions of Rotterdam and The Hague to smaller towns and dispersed rural settlement.

A total of 74 sheet numbers covering the province were identified and downloaded via the custom pipeline. The pipeline was executed twice: once selecting the first available edition per sheet (T1) and once selecting the last available edition per sheet (T2), as summarised in Table 4.

5.1. Method 1 - Patch level classification

The resulting two sets of georeferenced GeoTIFFs form the primary input for all subsequent pipeline stages, and are used for the urban change analysis described in Section 5.4.

Table 4: Downloaded Bonnebladen GeoTIFFs for the South Holland study area.

Temporal set	Number of sheets	Data volume
T1 (first edition)	74	2.51 GB
T2 (last edition)	74	2.26 GB

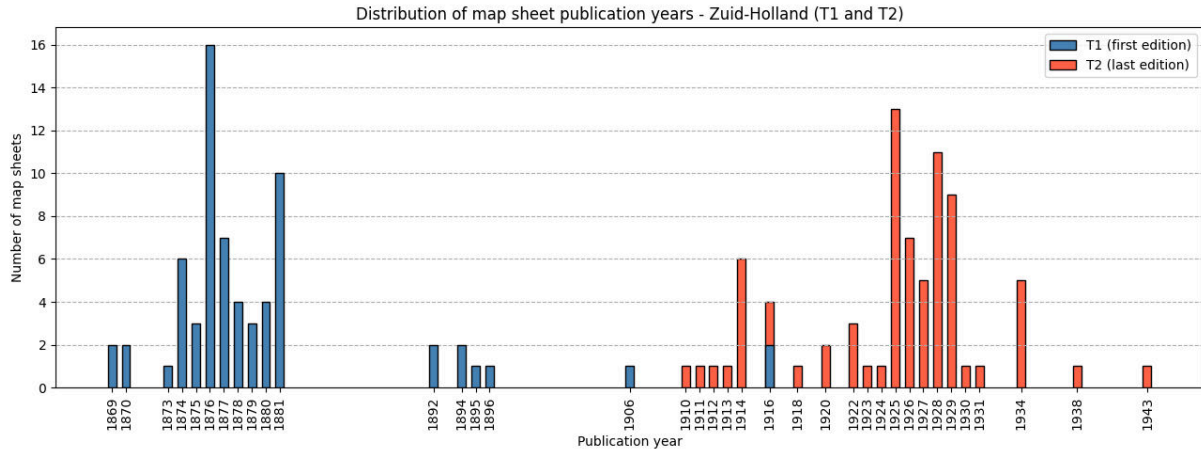


Figure 5.6: Distribution of South Holland map sheet publication per year for T1 and T2.

As shown in Figure 5.6, the T1 set spans from 1869 to 1916, with sheets originating from a wide range of publication years. This temporal spread reflects the gradual surveying and revision of the province over several decades, and implies that the training data captures variation in cartographic style and print quality rather than representing a single homogeneous campaign. The T2 set spans from 1910 to 1943, with the majority of sheets concentrated between 1924 and 1929. Together, these distributions confirm that both T1 and T2 include sheets from stylistically distinct periods, which is relevant for assessing how well the model generalises across cartographic variation.

5.1.3. Patch creation

The 74 downloaded GeoTIFFs are split into square patches of 200 meters, as described in Section 3.1.2. The patch size is selected to ensure that dense building clusters are fully enclosed within a single patch while retaining sufficient spatial detail to capture individual buildings. The patchification results are summarised in Table 5. Figure 5.7 shows an example of the resulting patch grid overlaid on a map sheet for visual inspection.

Table 5: Patchification summary for the South Holland T1 dataset.

Metric	Value
Input sheets	74
Total patches	127,468
Patches per second	68.2
Total duration	31.2 min

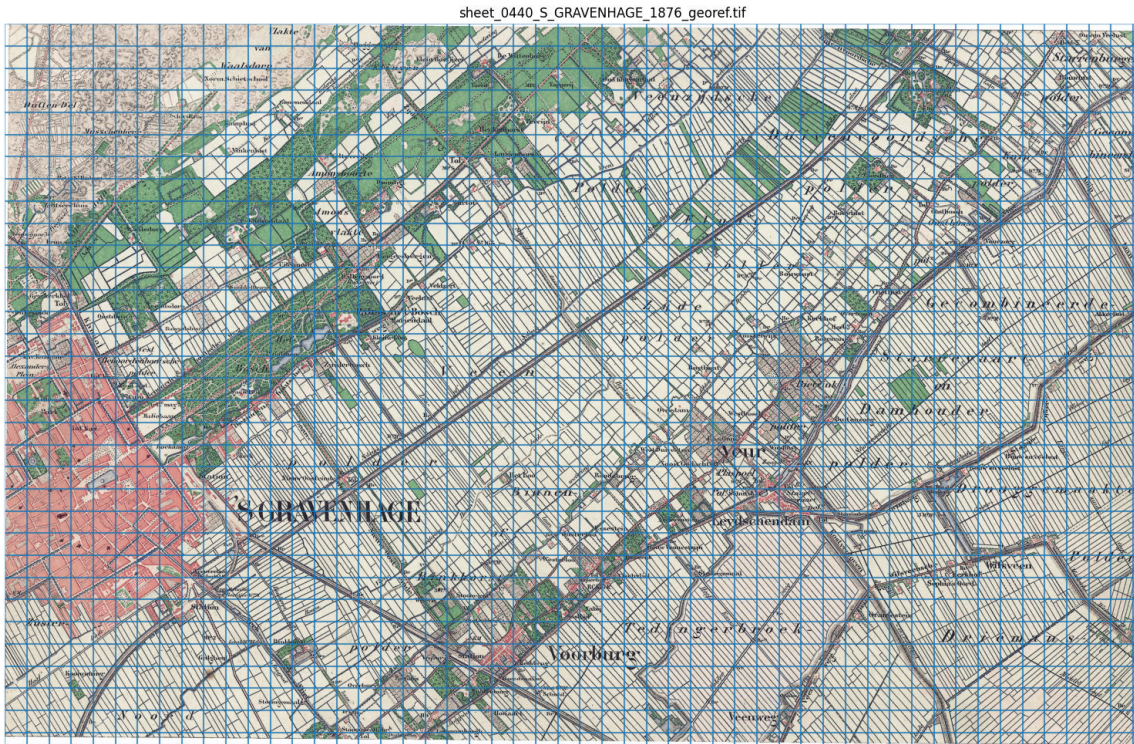


Figure 5.7: Example patch grid overlaid on a Bonnebladen map sheet.

5.1.4. Manual annotation

Manual annotation is performed across patches drawn from the full South Holland map series, distributed randomly across all 74 sheets rather than concentrated on specific regions or sheets. This ensures the annotated set captures the range of visual variation present across the collection.

Given the large total patch count of 127,468, annotation effort is focused using the per-patch pixel statistics computed in the previous stage. Patches are pre-filtered by the standard deviation of the red channel (`std_pixel_r`), retaining only those whose values fall within the range characteristic of building content (Figure 5.8). This is effective because buildings are rendered with a consistent red fill across the Bonnebladen series, making the red channel a reliable proxy for identifying candidate patches before presenting them to the annotator, as described in Section 3.1.3. The annotation step for the weak classifier training dataset required only approximately 2 hours.

5.1. Method 1 - Patch level classification

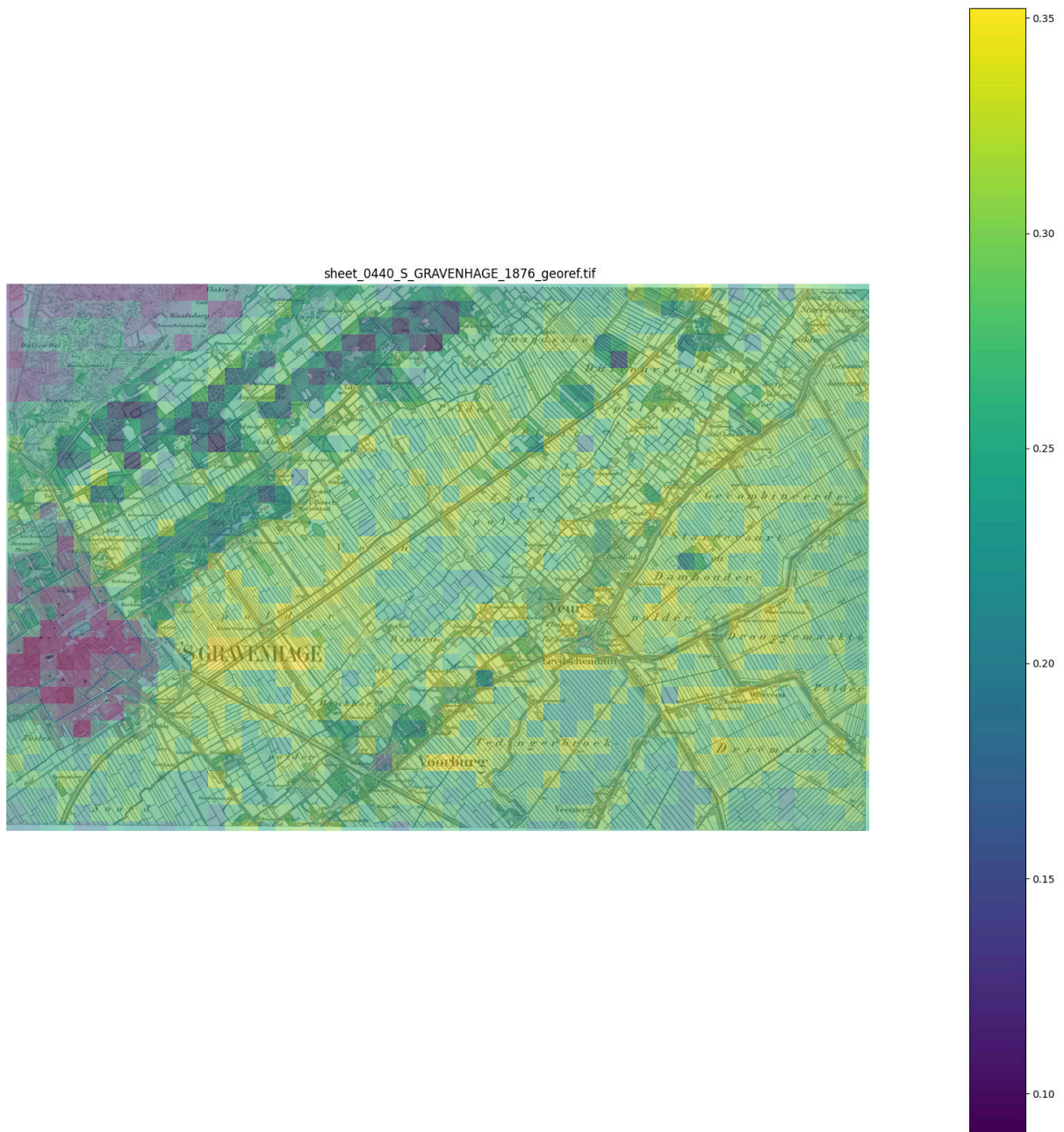


Figure 5.8: Pixel intensities values of the patches.

The resulting annotated dataset, referred to as the weak classifier dataset, contains 1,768 labeled patches across three classes, as summarised in Table 6. The class distribution is strongly imbalanced, with the **no** class dominating, which is expected given that non-urban patches constitute the majority of the map area. Representative examples of each class are shown in Figure 5.9.

Table 6: Annotated patch counts per class (weak classifier dataset).

Label	Number of patches
no	1,500
individual_buildings	163
building_blocks	105
Total	1,768

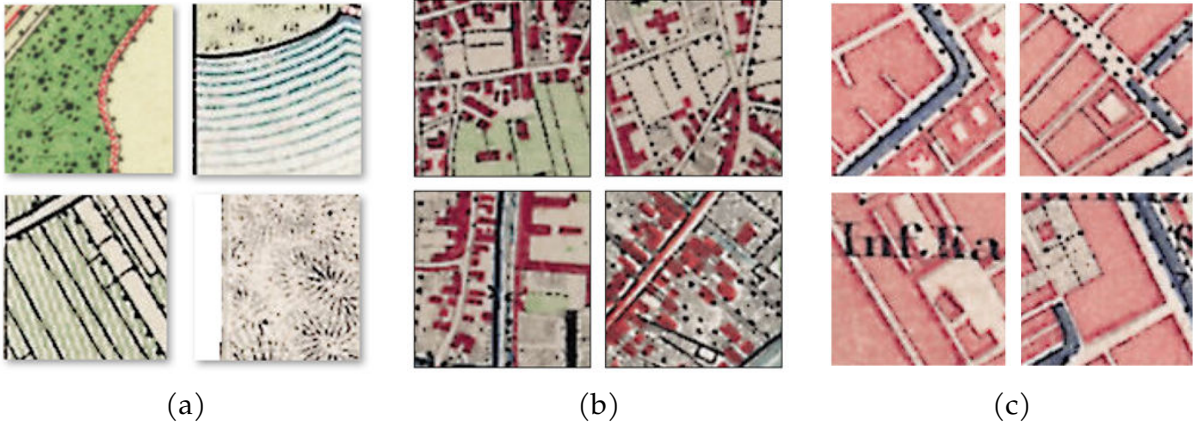


Figure 5.9: (a) no. (b) individual_buildings. (c) building_blocks.

5.1.5. Model training and evaluation

The classification model is trained in a single stage due to time constraints. The planned second-stage fine-tuning, in which low-confidence predictions from the weak classifier would be reviewed and used to extend the training set, was not completed. All results in this section therefore reflect the weak classifier only. The performance and inference results should be interpreted accordingly: the fine-tuning stage was designed to improve generalisation across cartographic styles, and its absence is a likely contributor to the T2 performance degradation discussed in Section 5.1.6.

Weak classifier

The annotated dataset of 1,768 patches is split into training, validation, and test subsets using stratified sampling, as summarised in Table 7. A pretrained ResNet-101 model is fine-tuned for 20 epochs using the training configuration described in Section 3.1.4. The model checkpoint with the lowest validation loss is selected; this occurs at epoch 7 with a validation loss of 0.0699. Training runs on a CUDA-enabled GPU and completes in approximately 7 minutes, with a peak VRAM usage of 4.42 GB out of 6.0 GB available.

Table 7: Dataset split for weak classifier training.

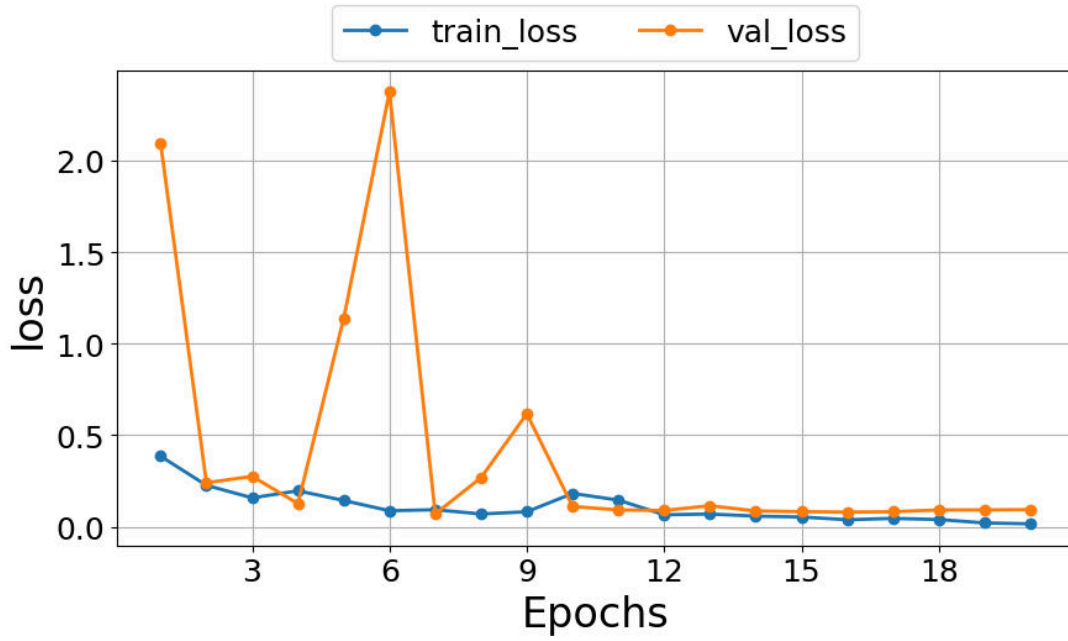
Subset	Patches	Share (%)
Train	1,237	70
Val	265	15
Test	266	15
Total	1,768	100

The loss curves are shown in Figure 5.10a. Training loss decreases smoothly throughout, reaching near zero by epoch 20. Validation loss is volatile in the first 10 epochs, with pronounced spikes at epochs 6 and 9, before stabilising around 0.07-0.10 from epoch 12 onward. The instability in the early validation loss is consistent with the small dataset size, where individual batches have a disproportionate effect on the computed loss. The model checkpoint with the lowest validation loss is selected at epoch 7 (val loss = 0.0699), falling in the brief dip between the two spikes.

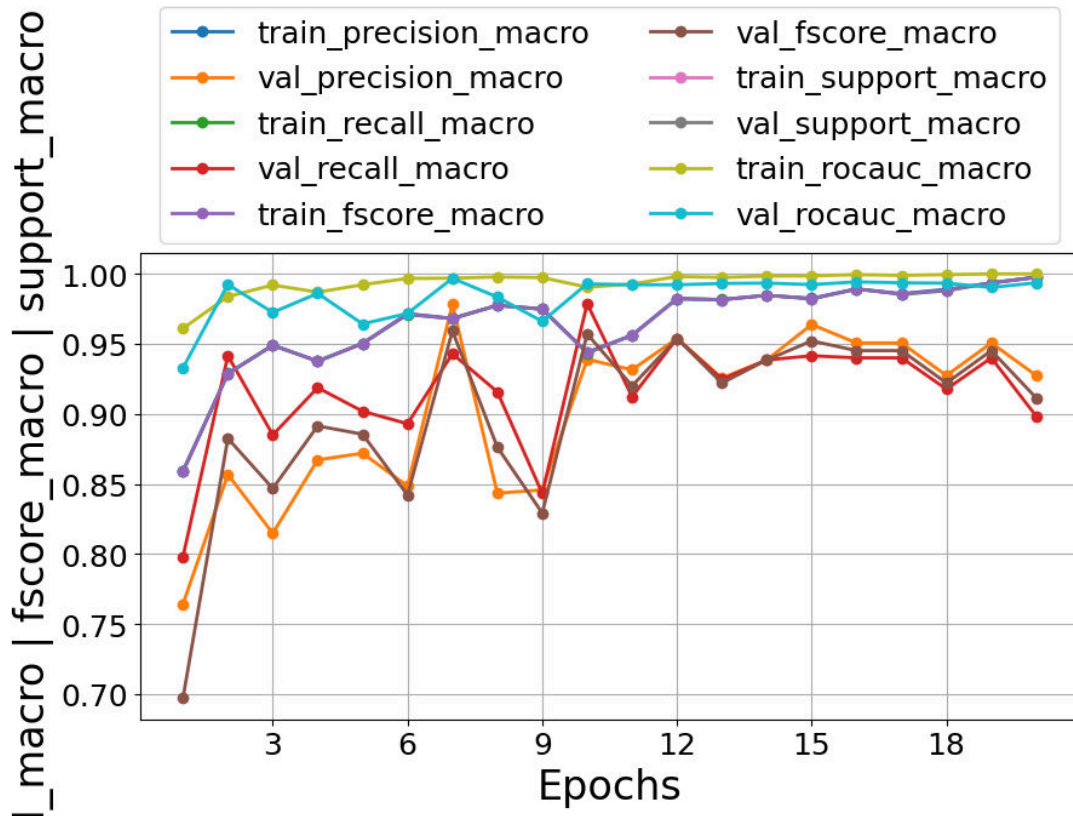
The macro-averaged performance metrics per epoch are shown in Figure 5.10b. Training metrics saturate near 1.0 by approximately epoch 8 and remain flat for the remainder of training. Validation ROC-AUC macro stabilises quickly and stays consistently high throughout, while validation F-macro fluctuates between epochs 1 and 10 before settling in the 0.90-0.95 range

5.1. Method 1 - Patch level classification

from epoch 12 onward. The persistent gap between training and validation F-macro is a mild overfitting signal, attributable to the limited size of the annotated dataset, and is one motivation for the planned fine-tuning stage.



(a)



(b)

Figure 5.10: (a) Training and validation loss. (b) Macro-averaged precision, recall, F1-score, and ROC-AUC.

Test set performance is reported in Table 8. The **no** class dominates the test set (225 out of 266 patches, 84.6%) and achieves near-perfect F1 (0.984), which inflates the micro-averaged F1 (0.962) relative to the macro-averaged F1 (0.900). The macro average is therefore the more informative metric for this imbalanced distribution. Among the minority classes, **building_blocks** achieves F1 = 0.909 despite having only 16 test samples, suggesting the model learned this visually distinct class well. The weakest class is **individual_buildings** (F1 = 0.808, 25 samples), which is expected given its visual overlap with both **no** and **building_blocks** in edge cases. The precision-recall balance differs between the two minority classes: **building_blocks** has higher recall than precision (0.938 vs 0.882), while **individual_buildings** shows the inverse pattern (recall 0.840, precision 0.778), indicating the model is more prone to over-predicting **individual_buildings** than to missing instances of it.

Table 8: Test set performance of the weak classifiers.

Metric	Class	Value
Precision	no	0.991
Precision	individual_buildings	0.7778
Precision	building_blocks	0.8824
Recall	no	0.9778
Recall	individual_buildings	0.84
Recall	building_blocks	0.9375
F1	no	0.9843
F1	individual_buildings	0.8077
F1	building_blocks	0.9091
Support	no	225
Support	individual_buildings	25
Support	building_blocks	16
F1-macro	-	0.9004
F1-micro	-	0.9624
ROC-AUC macro	-	0.9837

5.1.6. Inference results

Inference is applied to the complete T1 and T2 patch sets using the trained weak classifier. Each patch receives a predicted label and a confidence score. Results for a fine-tuned classifier are not reported as this stage was not completed. The predicted class distribution across both temporal sets is summarised in Table 9. A qualitative inspection of the results is conducted by overlaying predicted patches on the source map sheets in QGIS.

Table 9: Predicted patch counts per class for T1 and T2

Class	Annotation	T1 edition	T2 edition
no	1500	126076	126065
individual_buildings	163	952	686
building_blocks	105	440	1607
individual_buildings (conf < 0.5)	n/a	15	20
building_blocks (conf < 0.5)	n/a	6	27

T1 inference

T1 inference produces 952 patches predicted as **individual_buildings** and 440 as **building_blocks**, with very few low-confidence predictions (15 and 6 respectively). Qualitative inspection in QGIS confirms that the spatial distribution of predicted urban patches is coherent: dense urban cores such as The Hague and Rotterdam are correctly identified as **building_blocks**, and smaller settlements and dispersed rural buildings are captured under **individual_buildings**. The model shows no systematic spatial bias across the province.

T2 inference

T2 inference with the same weak classifier produces a shifted class distribution relative to T1. **individual_buildings** predictions decrease from 952 to 686 (approximately 28%), while **building_blocks** predictions increase from 440 to 1,607 (approximately 265%). Low-confidence predictions also increase, particularly for **building_blocks** (6 to 27). Total patch counts differ marginally between T1 and T2 (127,468 versus 128,358).²⁰

Interpreting the T1-to-T2 shift requires accounting for two competing effects. The temporal gap between editions spans several decades for most sheets, during which genuine urban growth is expected. An increase in **building_blocks** and **individual_buildings** predictions is therefore partly consistent with actual land cover change. At the same time, cartographic style changes in post-1930s editions alter the visual signal the model was trained on.

Separating these two effects from prediction counts alone is not straightforward. However, the near-identical **no** class count between T1 and T2 (126,076 versus 126,065) suggests the model detects almost no net increase in urban patches despite decades of expected development. The decrease in **individual_buildings** predictions also contradicts the expected direction of change. Qualitative inspection in QGIS confirms omission errors on T2 sheets where clearly visible building features are not detected (Figure 5.12).

The sharp rise in **building_blocks** predictions combined with elevated low-confidence counts points to false positive predictions in T2. One specific source identified during visual inspection is the cartographic symbol for *bouwland of tuingrond met kassen* (arable land or garden land with greenhouses), illustrated in Figure 5.11. This symbol, introduced in post-1930s editions, has a visual pattern that resembles the building fill of dense urban fabric, making it a likely trigger for false positive **building_blocks** predictions.

²⁰Both T1 and T2 are derived from the same 74 sheet indices and should in theory produce identical patch counts. The discrepancy of less than 0.7% is likely attributable to small differences in georeferenced sheet extents between editions and does not affect the analysis.

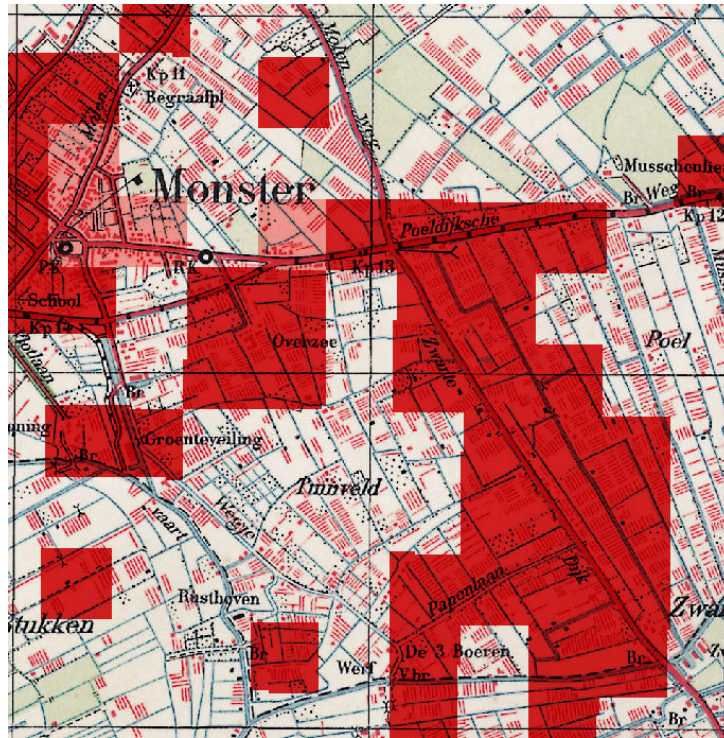
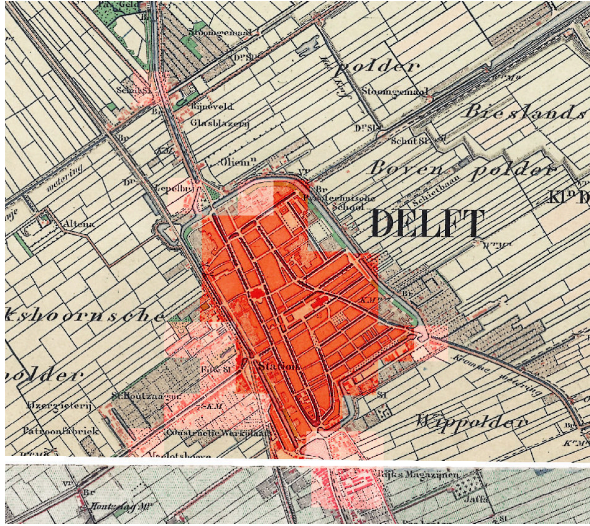


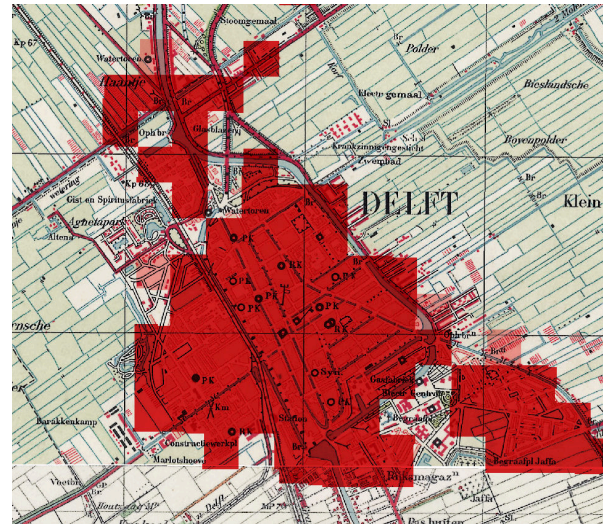
Figure 5.11: Cartographic symbol for garden land with greenhouses as depicted in post-1930s Bonnebladen editions, illustrating its visual similarity to dense building fill patterns.

The T2 results therefore reflect a mixture of genuine urban growth signal and model degradation driven by cartographic style shift. A comparison against reference data is provided in Section 5.1.7. The intended strategy to address this style-induced degradation is to train a single model on annotation examples drawn from both pre- and post-1930s editions, so that the classifier learns to handle both cartographic styles without requiring separate models per temporal period. This is preferable to training edition-specific models because it avoids the need to pre-classify each sheet by style before inference, and reduces the total annotation effort. The fine-tuning stage was not completed in this research, this remains a limitation of the current results and is discussed further in Chapter 6.

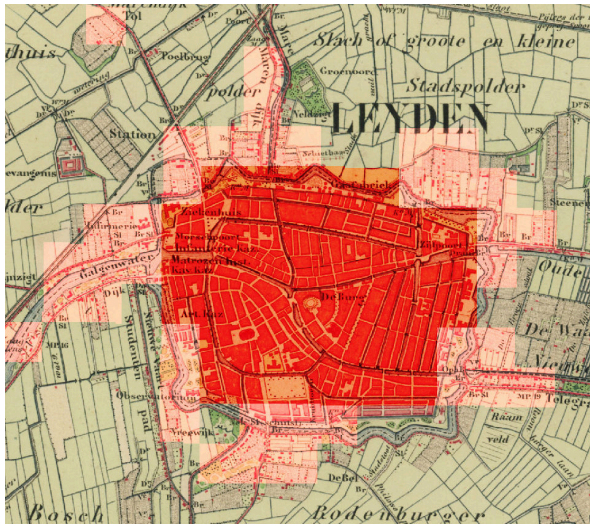
5.1. Method 1 - Patch level classification



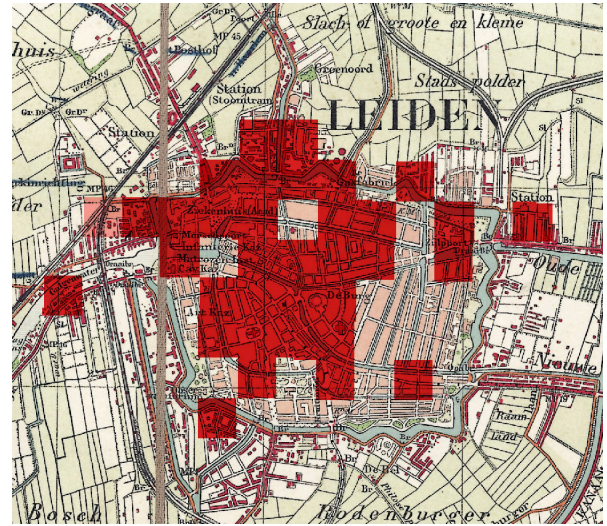
(a)



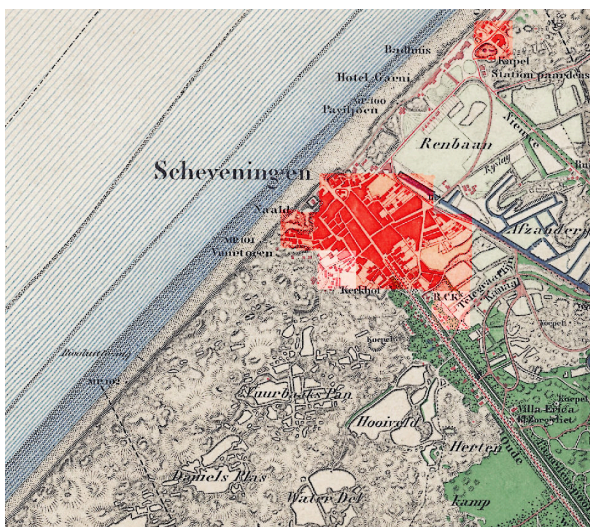
(b)



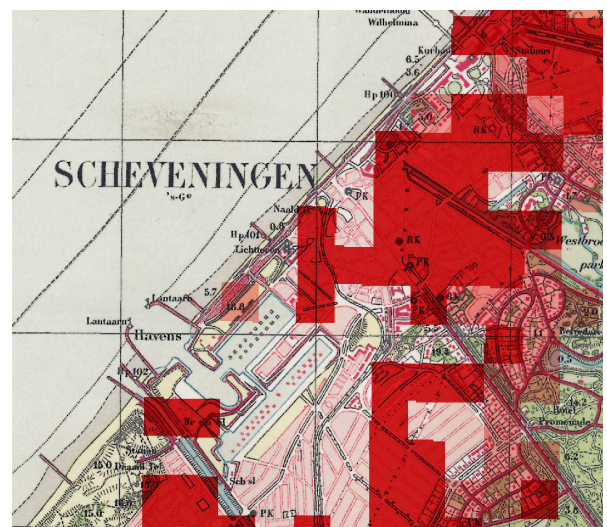
(c)



(d)



(e)



(f)

Figure 5.12: (a),(c),(e) T1 inference results. (b),(d),(f) T2 inference results. Colors indicate predicted class: **no** (transparent), **individual_buildings** (red), and **building_blocks** (darker red).

5.1.7. Comparison with HGN-1900

Justification and temporal alignment

The HGN-1900 dataset is used as an external spatial reference to assess the correspondence between T1 predictions and an independently produced historical land use map of the Netherlands. HGN-1900 is a nationwide raster dataset at 50 meter resolution, derived from a color-based classification of the same Bonnebladen map series (Knol et al., 2004). As noted in Section 4.1, HGN-1900 uses the *Verkend* date for temporal attribution, while the T1 dataset in this research uses the *Uitgave* date. To verify that both works are nonetheless using the same physical map editions, two representative sheets (440 *S'Gravenhage* and 502 *Rotterdam*) are inspected by examining the map margin text, which records both dates. The inspection confirms that the editions used in both datasets are identical for these sheets, supporting the use of HGN-1900 as a valid spatial reference for T1. Figure 5.13 illustrates this alignment by comparing the sheet index maps from both datasets alongside the map margin scans of the two inspected sheets.

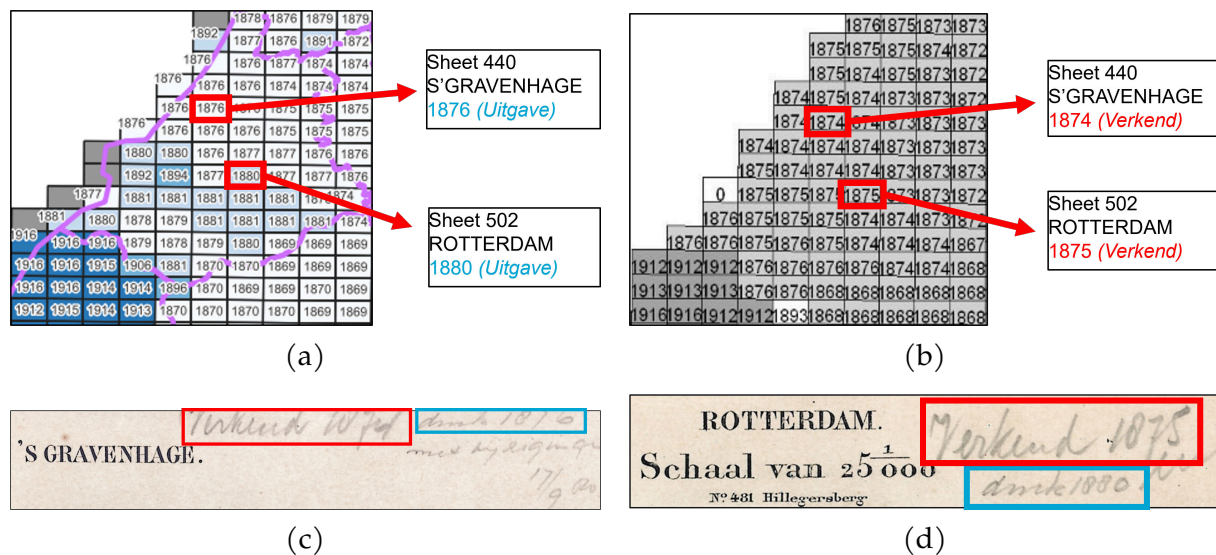


Figure 5.13: (a) Sheet index of Province of South Holland from this research (T1, *Uitgave* date). (b) Sheet index of Province of South Holland reconstructed from HGN-1900 (*Verkend* date), source: Knol et al. (2004). (c) Map margin of sheet 440 (*S'Gravenhage*) showing *Verkend* and *Uitgave* dates. (d) Map margin of sheet 502 (*Rotterdam*), source: Kadaster et al. (1940).

HGN-1900 covers the entire Netherlands, but for this comparison only the South Holland region is extracted to match the study area of Method 1.

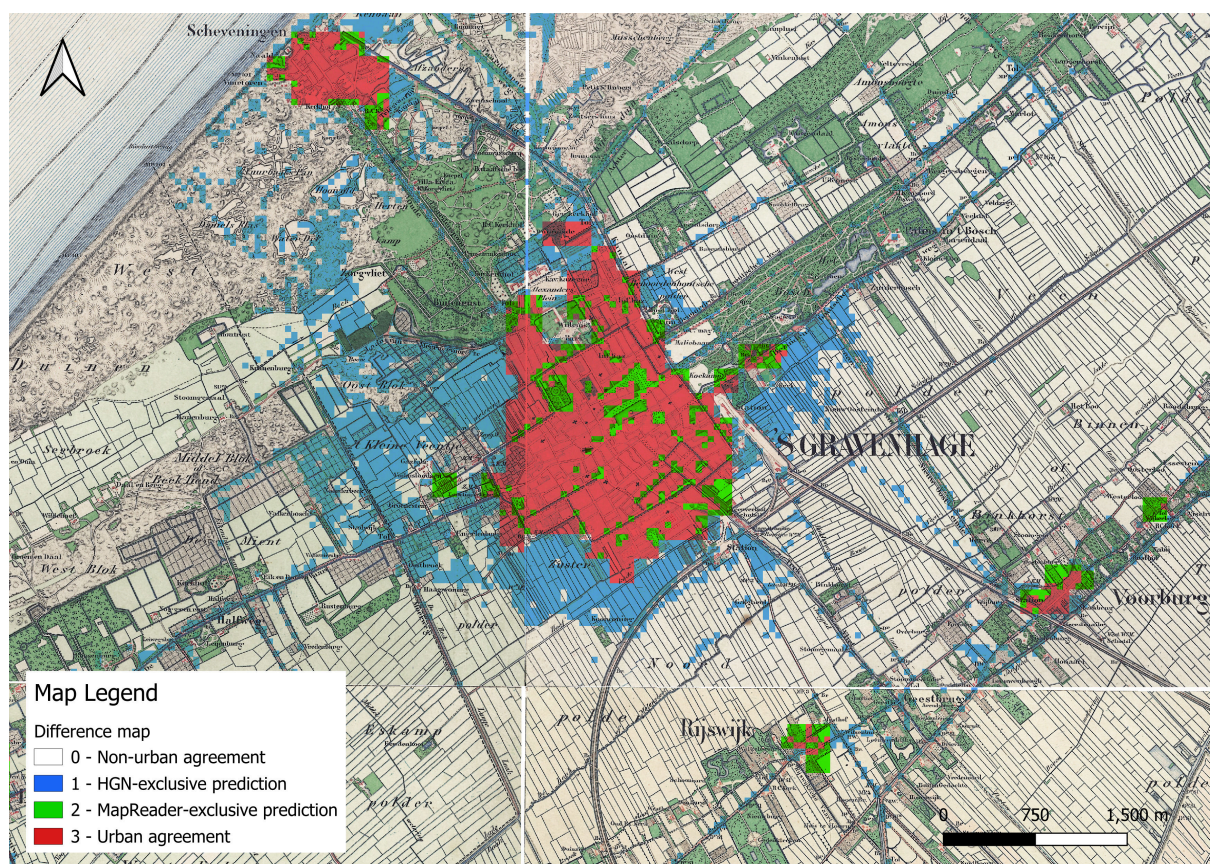
Two methodological differences between the datasets are noted before interpreting the comparison. First, HGN-1900 classifies built-up areas and paved roads together as a single class (in HGN identifies as class 6 - *bebouwd en wegen*), because both features are rendered in red on the Bonnebladen and cannot be separated by color alone (Knol et al., 2004). In rural areas in particular, the built-up/roads class contains more paved roads than buildings. This means HGN-1900 likely overestimates the spatial extent of built-up areas relative to a building-only classification. Second, HGN-1900 operates at 50 meter pixel resolution, while the patches in this research are 200 meters. The 4:1 size difference means that a single patch in this dataset corresponds to a 4x4 block of HGN pixels, and fine-grained spatial patterns below 200 meters are not captured in the patch predictions.

Difference map analysis

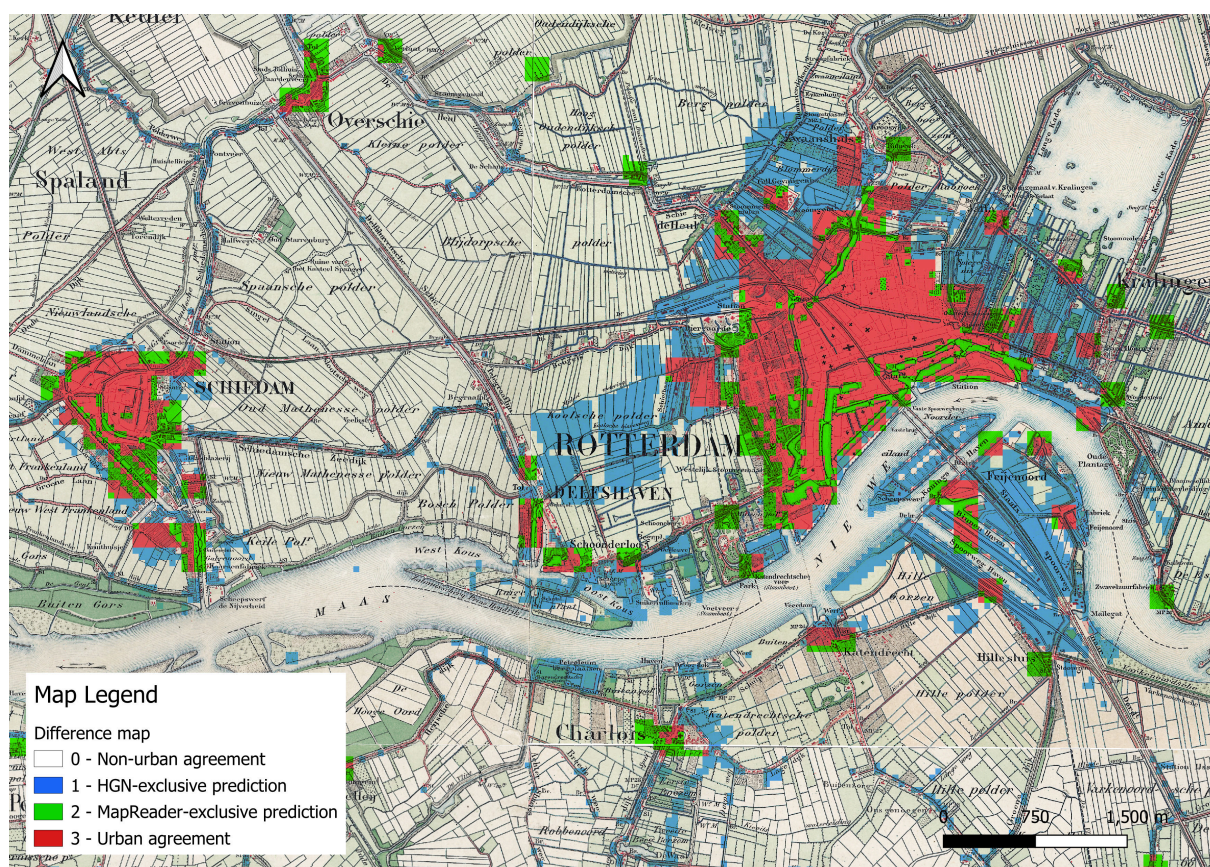
A difference map is produced by comparing the T1 prediction raster against the HGN-1900 built-up class raster for South Holland. Both rasters are resampled to a common resolution and co-registered before comparison. The resulting map assigns each cell to one of four classes, as summarised in Table 10 and illustrated in Figure 5.14.

Table 10: Difference map class counts for the South Holland study area, comparing T1 predictions against HGN-1900.

Value	Class	Pixel count	Area (ha)	Share (%)
0	Non-urban agreement	2 549 755	637 642.17	97.36%
1	HGN-exclusive detection	47 802	11 954.31	1.83%
2	MapReader-exclusive detection	13 462	3 366.57	0.51%
3	Urban agreement	7 924	1 981.63	0.30%



(a)



(b)

Figure 5.14: Difference map (a) S'GRAVENHAGE sheet. (b) ROTTERDAM sheet.

5.1. Method 1 - Patch level classification

The vast majority of the study area (97.36%) falls under non-urban agreement, where both datasets assign a non-urban classification. Urban agreement accounts for 0.30% of the total area (1,982 ha), representing locations where both methods detect built-up content. HGN-exclusive detection (1.83%, 11,954 ha) is substantially larger than MapReader-exclusive detection (0.51%, 3,367 ha).

Visual inspection by overlaying the difference map with the Bonnebladen sheets in QGIS reveals several patterns worth noting. Neither method produces consistent missed detections: clearly visible urban fabric on the map is detected by both methods in all inspected areas. HGN-exclusive detection, however, shows spatially clustered false positive patterns in several locations. A prominent example occurs in the outskirts of S'Gravenhage, where a large cluster of HGN-predicted urban pixels wraps around the city center and extends northward toward Scheveningen (Figure 5.14). Smaller but similar clusters are observed around Rotterdam and Vlaardingen, while cities such as Leiden and Delft do not show this pattern.

Crucially, these clusters appear as contiguous regions rather than elongated or linear shapes, making road misclassification an unlikely explanation. An alternative explanation is that a different map edition was used for those specific sheets in the HGN analysis, introducing content differences not visible in the sheet index comparison. This cannot be confirmed without direct access to the scans used in the original HGN classification.

The larger footprint of HGN-exclusive detection is also partly expected due to the roads-and-built-up conflation in HGN-1900: paved roads produce red pixels that are classified as built-up in HGN but do not trigger building predictions in the patch classifier. The 200 meter patch size further means that isolated or small buildings may fall within a patch that is predominantly non-urban, causing a non-urban prediction for that patch even where HGN's finer 50 meter resolution detects built-up content.

MapReader-exclusive detection (0.51%, 3,367 ha) is considerably smaller in spatial extent. Visual inspection shows that these pixels are predominantly found within or at the immediate edges of urban fabric, such as parks, canals, and open spaces enclosed within city centers, or in the corners of urban clusters. This pattern is consistent with the resolution difference between the two methods: a 200 meter patch that contains a mix of urban and non-urban land cover is classified as urban by MapReader based on its dominant content, while HGN's 50 meter resolution can resolve the non-urban elements within the same area individually. MapReader-exclusive detection is therefore largely a consequence of spatial generalisation at the patch level rather than a systematic over-prediction of urban extent.

HGN-1900 is not treated as ground truth in this comparison. It was produced from the same source maps using color-based classification without independent field reference, and carries its own classification limitations, most notably the built-up/roads conflation (Knol et al., 2004). The visual inspection further confirms that HGN itself produces spatially clustered false positive detections in South Holland. The comparison is therefore treated as a spatial plausibility check rather than a formal accuracy assessment.

5.2. Method 2 - Object-level classification

5.2.1. Georeferencing

The IIF georeferencing annotation for sheet 3 uses 14 GCPs distributed across the sheet with a Polynomial Order 1 (affine) transformation. A Polynomial Order 1 transformation applies a single global correction and cannot account for local geometric distortions within the map.

Visual inspection in QGIS confirms that the sheet is roughly aligned with the BGT at a global level. However, at the building level, a clear spatial offset is visible between building outlines in the historical map and their corresponding BGT footprints. This misalignment is not uniform across the sheet, which is consistent with local distortion in the original map that a global affine transformation cannot correct.

Figure 5.15 shows the alignment of sheet 3 with the BGT *Pand* layer before and after improving the placement of GCPs. After applying the Thin Plate Spline (TPS) transformation with 100 GCPs, building outlines in the historical map align closely with the BGT footprints across the sheet. The spatial distribution of all GCPs and ICPs is shown in the appendix (Section C).



(a) Before: Allmaps annotation. (b) After: Improving the placement of GCPs.
Figure 5.15: Overlay of Rotterdam 1897 sheet 3 with BGT *Pand* layer (buildings with *bouwjaar* < 1897, shown in yellow outline, buildings with *bouwjaar* > 1897, shown in green outline) before and after improving placement of GCPs.

Table 11 reports the residuals for all 15 ICPs after TPS transformation. Residuals are reported in both pixels and meters, using the output pixel size of 0.56292 m/pixel. ICPs with total error > 5 m, located near the western and south-eastern sheet edges.

Table 11: ICP residuals after TPS georeferencing of Rotterdam 1897 sheet 3.

ICP	dX (px)	dY (px)	Total (px)	dX (m)	dY (m)	Total (m)
1	+0.50	+0.53	0.72	+0.28	+0.30	0.41
2	−0.48	+6.16	6.17	−0.27	+3.47	3.48
3	−1.42	−3.06	3.37	−0.80	−1.72	1.90
4	+0.98	−0.38	1.05	+0.55	−0.21	0.59
5	−2.12	−1.14	2.40	−1.19	−0.64	1.35
6	−1.19	+11.64	11.70	−0.67	+6.55	6.58*
7	−0.10	−2.63	2.64	−0.06	−1.48	1.48
8	−0.71	+1.24	1.43	−0.40	+0.70	0.81
9	−1.00	+2.38	2.58	−0.56	+1.34	1.45
10	−1.86	+1.42	2.34	−1.05	+0.80	1.31
11	+0.93	−0.06	0.94	+0.53	−0.03	0.53
12	+1.84	−0.06	1.84	+1.04	−0.03	1.04
13	+0.69	−0.48	0.84	+0.39	−0.27	0.48
14	+4.83	−11.53	12.51	+2.72	−6.49	7.04*
15	+6.05	−7.14	9.36	+3.41	−4.02	5.27*

Table 12: Georeferencing accuracy summary for Rotterdam 1897 sheet 3

Metric	All 15 ICPs	11 ICPs (outliers excluded)
RMSE X (m)	1.292	–
RMSE Y (m)	2.851	–
RMSE Total (m)	3.130	1.137
Min error (m)	0.408	–
Mean error (m)	2.248	–
Max error (m)	7.040	–
% of map diagonal	0.062%	0.023%

Three ICPs (6, 14, 15) show total errors exceeding 5 m and are classified as outliers. All three are located near the western and south-eastern boundaries of sheet 3, away from the Rotterdam city center. The likely cause is increased local geometric distortion near the sheet edges, where fewer identifiable building features are available for precise GCP placement. Their large dY components suggest a systematic northing offset in those areas that the TPS interpolation could not fully correct.

Excluding these three outliers, the RMSE is 1.137 m (0.023% of the map diagonal, map diagonal is 5016 m). This is well within the 1% diagonal threshold commonly used as a benchmark for historical map georeferencing accuracy (Vaianti et al., 2025). At the building scale, where individual building footprints are typically 10–30 m wide, a sub-2 m RMSE is considered acceptable for the urban change analysis performed in Section 5.4.

5.2.2. Synthetic data generation using CycleGAN image translation

This section reports the results of the CycleGAN synthetic data generation pipeline described in Section 3.2.3. The pipeline produces a spatially referenced synthetic training dataset used as

input to the Mask R-CNN instance segmentation model (Section 3.2.4). Results are reported in four parts: cartographic style transfer, data preprocessing and patch tiling, model training and loss analysis, and synthetic data quality evaluation using FID.

Cartographic style transfer

The BGT layers are rendered in five feature classes matching the cartographic conventions of the Rotterdam 1897 maps: regular buildings, hatched buildings, water bodies, vegetation, and bare surfaces. Target fill and stroke colors per class are derived from mean RGB values computed from manually selected patches. The resulting color parameters are summarized in Table 13. Figure 5.16 shows representative source patches selected from the historical sheets for each class. Text overlays are rendered in EB Garamond²¹, matching the font style of the historical maps. The styled BGT output, the CycleGAN-translated result, and the corresponding original map sheet are compared in Figure 5.23.

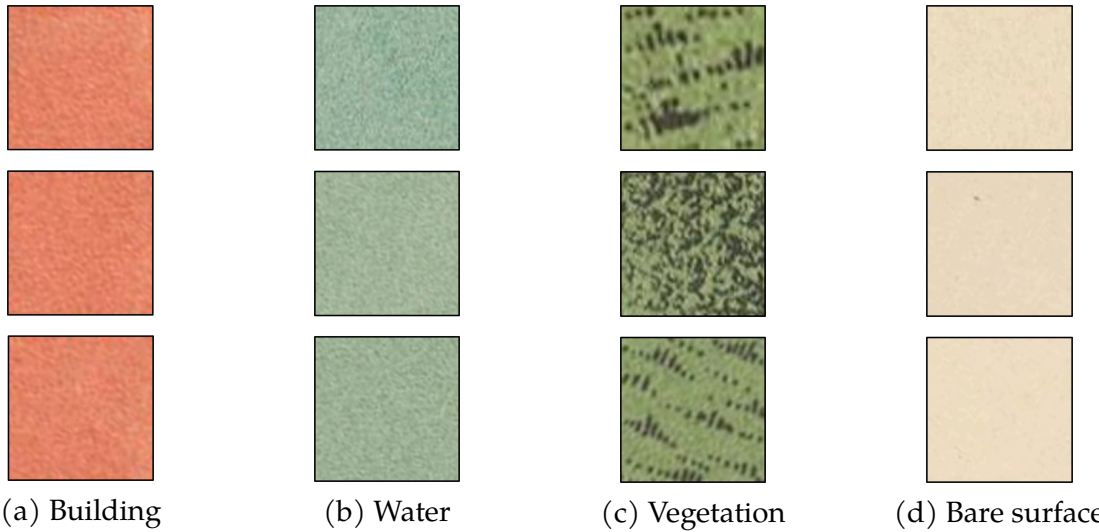


Figure 5.16: Representative patches manually selected from the Rotterdam 1897 map sheets for each feature class, used to compute mean RGB color values.

Table 13: Color taxonomy applied to BGT layers for cartographic style transfer.

class	hex_mean	mean_R	mean_G	mean_B	std_R	std_G	std_B	min_R	min_G	min_B	max_R	max_G	max_B	total pixels
pand	#E68669	229.97	134.14	105.46	6.02	6.58	6.61	203	109	74	255	171	140	16875
pand hatched	#C36D58	195.12	108.98	88.1	28.41	18.92	19.73	103	46	27	249	210	177	4900
water-deel	#9FB49A	158.79	180.45	154.37	7.8	6.85	7.44	118	145	120	183	207	185	30000
begroeid terrein-deel	#7C865E	123.65	133.76	94.1	27.57	29.35	24.68	40	48	9	185	194	161	15525
back-ground	#ECDCC1	236.12	220.36	193.16	2.73	2.8	2.69	181	165	139	247	230	202	30000

Data preprocessing and patch tiling

The synthetic BGT PNGs and historical GeoTIFFs are tiled into 512 × 512 pixel patches with a stride of 256 pixels, producing a 50% spatial overlap between adjacent patches.

The dataset is split by sheet to ensure spatial separation between training and test sets. Patches from sheets 2 and 3 form the training set, and patches from sheet 4 form the test set. Table 14 summarizes the resulting patch counts per domain and split.

²¹<https://googlefonts.github.io/ebgaramond-specimen/>

Table 14: Patch counts per domain and split after tiling and content filtering.

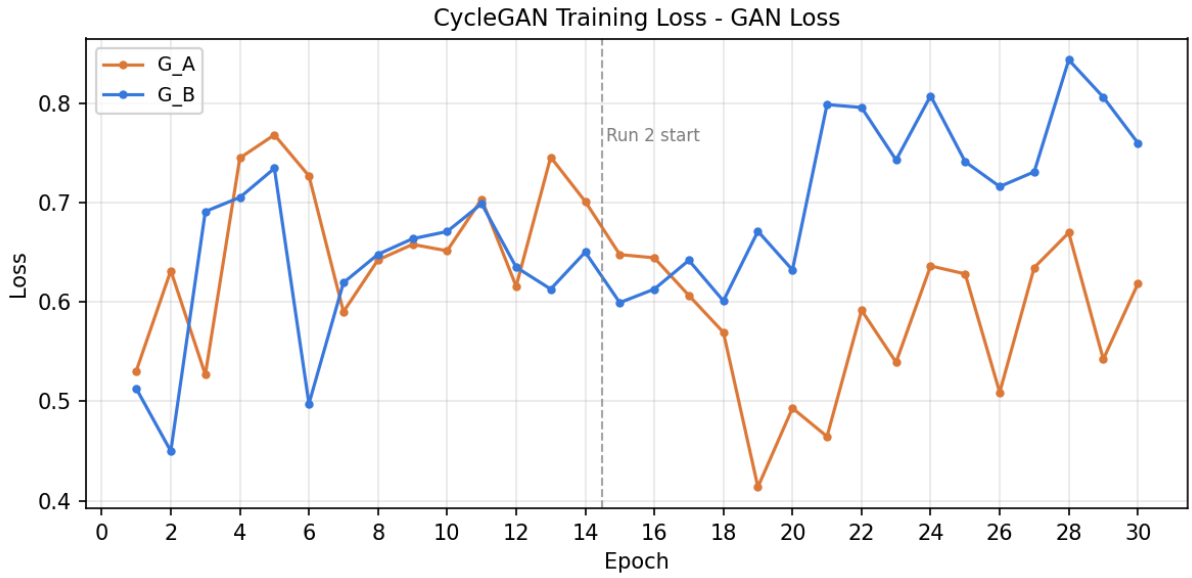
Split	Domain A (Synthetic BGT)	Domain B (Historical GeoTIFF)
Train set	1876	1146
Test set	610	732
Total	2486	1878

Model training

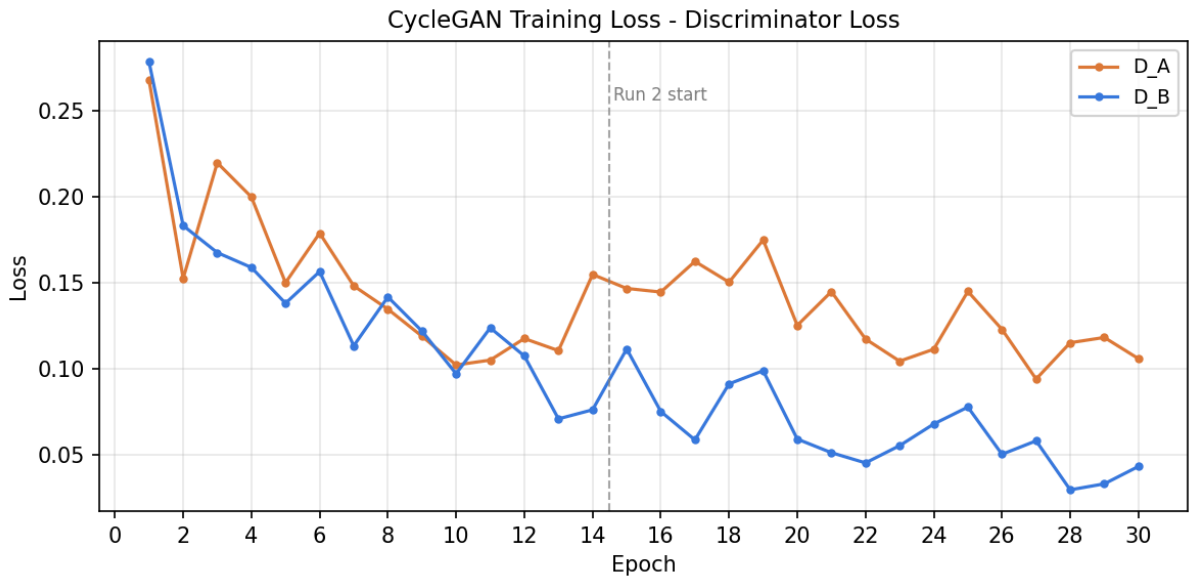
Training was conducted in two runs on an NVIDIA GeForce RTX 4050 Laptop GPU (6 GB VRAM). Run 1 covered epochs 1 to 14 and was stopped manually before completion. Run 2 resumed from the epoch 14 checkpoint using the `--continue_train` flag, extending training through epoch 30. Each run used the same hyperparameter configuration as described in Section 3.2.3. Checkpoints were saved every 2 epochs.

Run 1 completed in 3 hours 47 min and Run 2 in 3 hours 26 min. Each epoch processes 1,876 iterations at a batch size of 1, with a per-iteration time of approximately 0.4-0.9 seconds.

Figure 5.17 and Figure 5.18 shows the epoch-averaged training loss curves across both runs. The cycle consistency losses (`cycle_A`, `cycle_B`) drop sharply within the first two epochs, from approximately 1.6 to below 0.5, indicating the model quickly learns a structure-preserving mapping between domains. The GAN losses (`G_A`, `G_B`) oscillate throughout training, which is expected behavior reflecting the adversarial competition between generator and discriminator. By epoch 30, the discriminator loss `D_B` reaches 0.013, indicating the generator produces outputs largely indistinguishable from real historical patches. Final loss values at epoch 30 are summarized in Table 15.



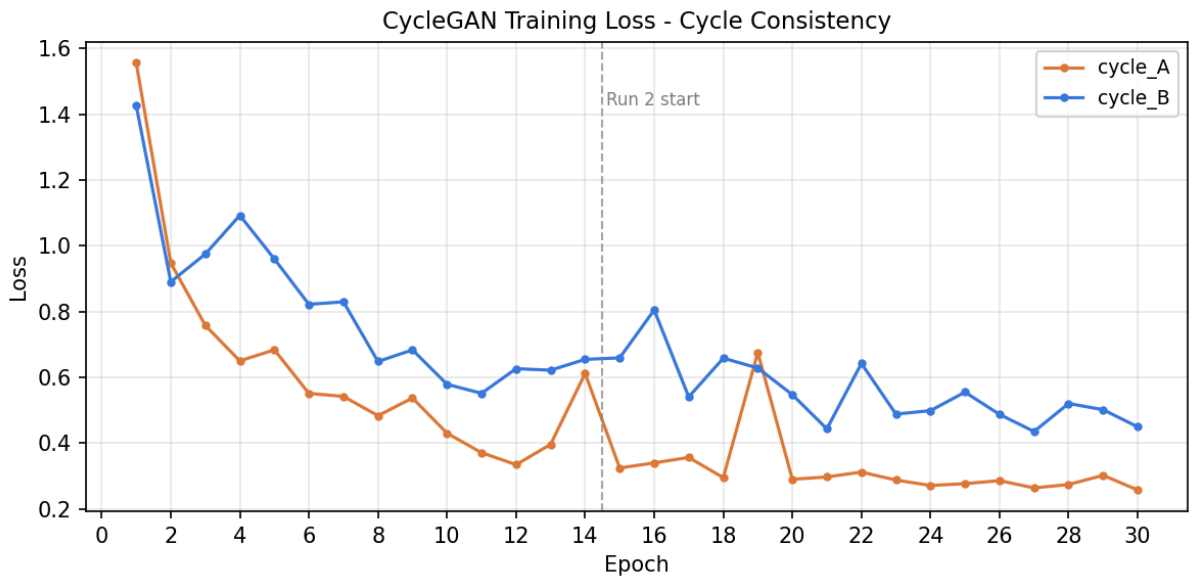
(a) GAN loss



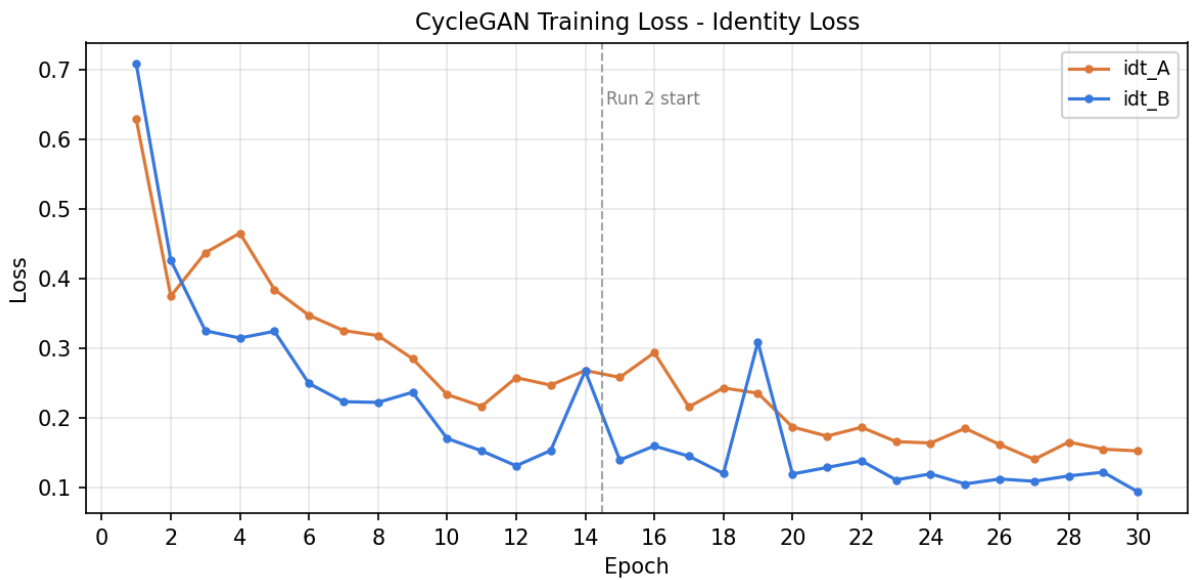
(b) Discriminator loss

Figure 5.17: GAN and discriminator losses. The dashed vertical line marks the boundary between Run 1 (epochs 1-14) and Run 2 (epochs 15-30).

5.2. Method 2 - Object-level classification



(c) Cycle consistency loss



(d) Identity loss

Figure 5.18: Cycle consistency and identity losses. Loss values represent epoch averages across all iterations.

Table 15: Epoch-averaged training loss values at selected checkpoints.

Epoch	D_A	G_A	cycle_A	D_B	G_B	cycle_B
2	0.153	0.631	0.947	0.183	0.450	0.890
4	0.200	0.745	0.650	0.159	0.705	1.092
6	0.179	0.727	0.551	0.157	0.498	0.822
7	0.148	0.590	0.542	0.113	0.619	0.829
8	0.135	0.642	0.483	0.142	0.648	0.648
14	0.155	0.701	0.612	0.076	0.650	0.655
30	0.106	0.618	0.257	0.043	0.760	0.449

Synthetic data quality (FID evaluation)

FID scores are computed for checkpoints at epochs 4, 6, 7, 8, 14, and 30 by comparing each set of translated `fake_B` patches against the `testB` reference patches. Table 16 summarizes the results. Epoch 6 yields the lowest FID among epochs 4-14 (151.08), while epoch 30 achieves a marginally lower overall score (149.94).

Table 16: FID scores per evaluated checkpoint epoch, computed against `testB` patches (732 images).

Epoch	FID (vs. <code>testB</code>)
4	158.93
6	151.08
7	159.64
8	163.94
14	156.48
30	149.94

However, FID provides only an aggregate measure of distributional similarity and does not capture class-specific translation quality. Furthermore, the `testB` reference set contains 732 patches, which falls below the commonly recommended minimum of 2,048 images for statistically stable FID scores (Heusel et al., 2018), so the reported scores should be interpreted as indicative rather than definitive. Visual inspection of `fake_B` outputs at each epoch is therefore conducted to complement the quantitative evaluation, shown in Figure 5.19 and Figure 5.20.

In tiles containing sparse building distributions with significant inter-building spacing, epoch 30 produces convincing vegetation textures that closely resemble the brushstroke-like mix of grass and trees visible in the Rotterdam 1897 maps, as shown in Figure 5.21. However, in denser tiles, the epoch 30 output shows clear degradation of building-class features. Hatched buildings (*pand hatching*) frequently break down at their boundaries, blending into the adjacent road surface (*wegdeel*). Narrow roads running between dense building clusters, particularly those overlaid with text labels, also lose structural definition.

Epoch 6 outputs, by contrast, are visually stable across all target classes. Building and hatched building boundaries are consistently preserved, and the translated patches show clear resemblance to the cartographic style and degradation effects of the historical domain without introducing geometric distortions to the annotation mask.

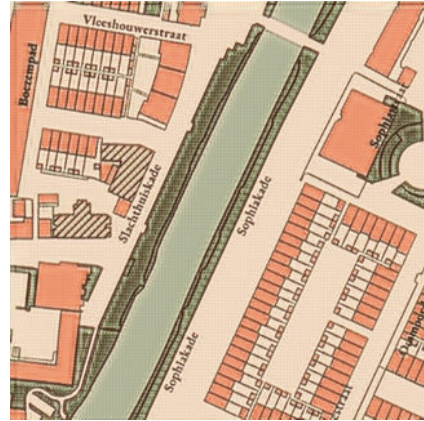
5.2. Method 2 - Object-level classification

This observation is consistent with the early-stopping strategy described by Arzoumanidis et al. (2025), where extended CycleGAN training introduces cumulative geometric distortions that corrupt the spatial alignment between the synthetic image and its paired annotation mask, undermining the utility of the generated training data.

Considering both FID scores and visual observation, epoch 6 offers the best balance between visual similarity to the historical domain and geometric fidelity of building outlines, and is therefore selected as the checkpoint for synthetic training data generation.



Cartographic style transfer (styled BGT input)



Epoch 4



Epoch 6



Epoch 8



Epoch 14



Epoch 30



Rotterdam 1897 historical map patch

Figure 5.19: From top to bottom: cartographic style transfer input, translated outputs at epochs 4, 6, 8, 14, and 30, and the real historical map patch as reference.

5.2. Method 2 - Object-level classification



Cartographic style transfer (styled BGT input)



Epoch 4



Epoch 6



Epoch 8



Epoch 14

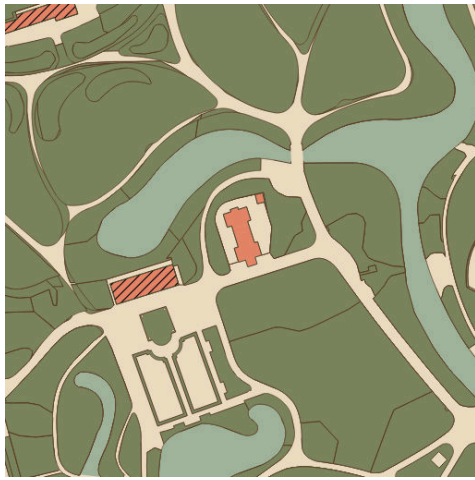


Epoch 30

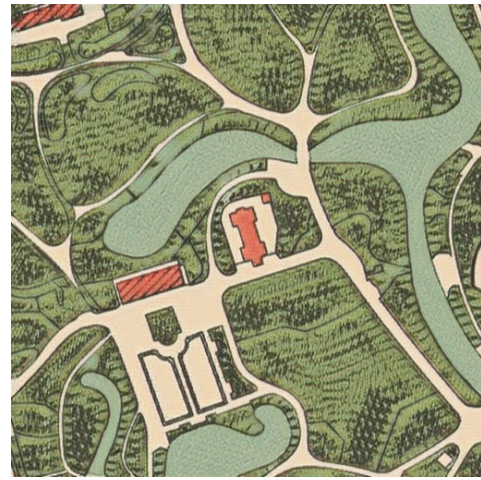


Real Rotterdam 1897 historical map patch

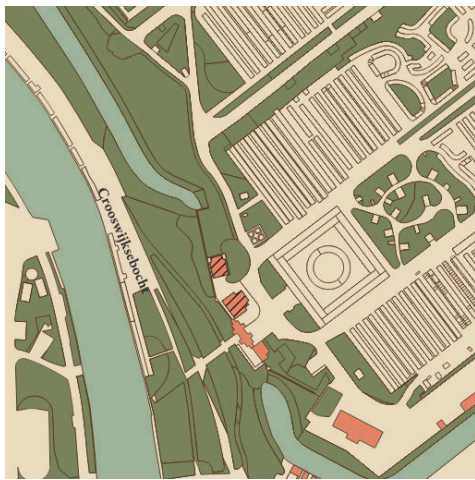
Figure 5.20: From top to bottom: cartographic style transfer input, translated outputs at epochs 4, 6, 8, 14, and 30, and the real Rotterdam 1897 map patch as reference.



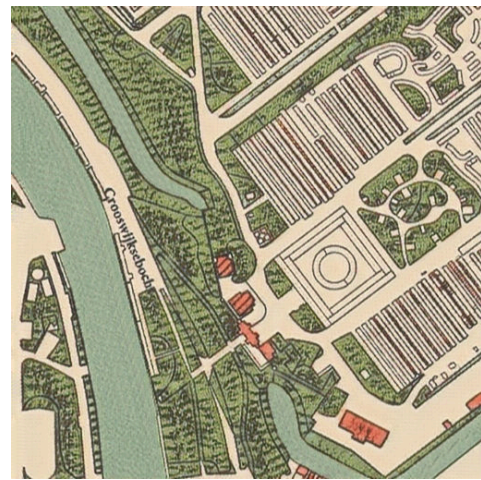
(a) Styled BGT input – sample 1



(b) CycleGAN output (epoch 30) – sample 1



(c) Styled BGT input – sample 2



(d) CycleGAN output (epoch 30) – sample 2

Figure 5.21: **(a, c)** Styled BGT input patches. **(b, d)** Corresponding CycleGAN-translated outputs at epoch 30. Both samples contain sparse buildings with significant inter-building spacing. The translated outputs show convincing vegetation texture resembling the brushstroke-like appearance of the Rotterdam 1897 maps, while building geometry remains spatially intact.

Post-processing output

Individual `fake_b` patches produced at epoch 6 are stitched into a full-sheet composite image. Each patch is placed at its pixel-origin position encoded in the filename. To avoid digital seam artifacts caused by blending overlapping regions, only odd-indexed patches are used during stitching, effectively skipping every second patch in the overlap sequence.

The stitched composite is georeferenced by assigning the CRS and affine geotransform of the corresponding historical GeoTIFF for the same sheet, producing a spatially referenced GeoTIFF per sheet in EPSG:3857. Figure 5.22 shows the stitched output for one sheet. Ultimately, the CycleGAN translation technique successfully produced 32,084 buildings and 1,007 hatched buildings within the study area. These files serve as input to the Mask R-CNN instance segmentation pipeline described in Section 3.2.4.

5.2. Method 2 - Object-level classification



(a) Rotterdam 1897 input sheets 2, 3, and 4

(b) Reconstructed CycleGAN output

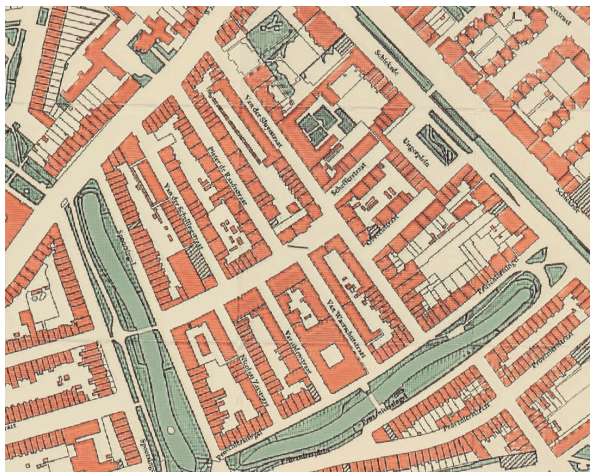
Figure 5.22: (a) Rotterdam 1897 municipality map sheets 2, 3, and 4 used as model input. (b) Reconstructed full-sheet synthetic output produced by stitching translated `fake_B` patches from the selected CycleGAN checkpoint.



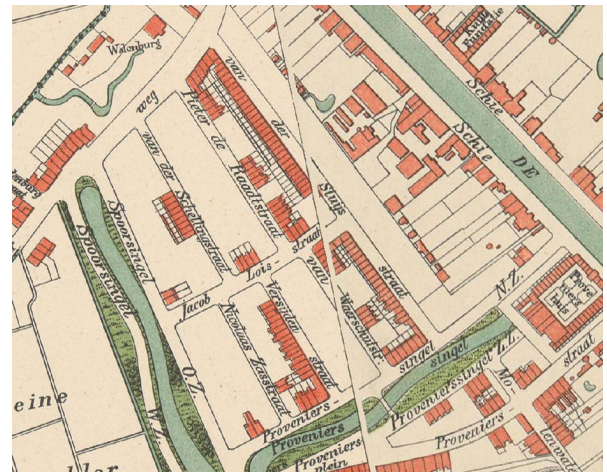
(a) BGT vector data



(b) Cartographic style transfer



(c) CycleGAN translated output



(d) Rotterdam 1897 historical map

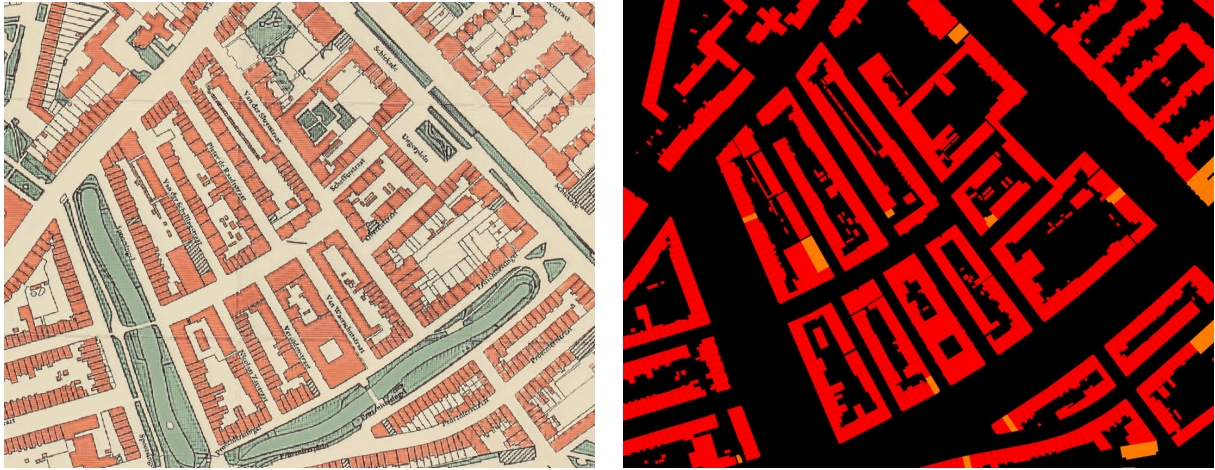
Figure 5.23: (a) Raw BGT vector data (source domain). (b) Cartographic style transfer approximating the historical map appearance. (c) CycleGAN-translated output adding cartographic degradation effects. (d) Real Rotterdam 1897 historical map patch (target domain) shown as reference.

5.2.3. Instance segmentation (Mask R-CNN)

This section reports the results of the four-stage Mask R-CNN pipeline described in Section 3.2.4. The pipeline is applied to the Rotterdam 1897 municipality map sheets 2, 3, and 4.

Data preparation

Tiling the merged CycleGAN GeoTIFF mosaic produced 416 tiles. None were discarded by the nodata filter, as all tiles contained sufficient image content. After splitting by sheet, the tile counts per split are given in Table 1 (train=291, val=62, test=63). Converting the tiles to COCO annotation format confirmed that 288 of the 291 training tiles contained at least one annotated building instance, the remaining 3 were empty and excluded from the training set. A sample set of tiles with corresponding annotation masks is shown in Figure 5.24.



(a) CycleGAN-translated synthetic tile

(b) Corresponding BGT annotation mask

Figure 5.24: (a) A set of CycleGAN-translated synthetic tiles used as model input. (b) The paired BGT annotation mask for the same tile sets, where each color encodes a class: background (class 0), building (class 1), and building_hatching (class 2).

Model training and evaluation

The model is trained for 30 epochs. Total training time is approximately 50 minutes on the available GPU (see Section 4.4). Selected epoch results are reported in Table 17.

Table 17: Mask R-CNN training results per selected epoch (30 epochs total).

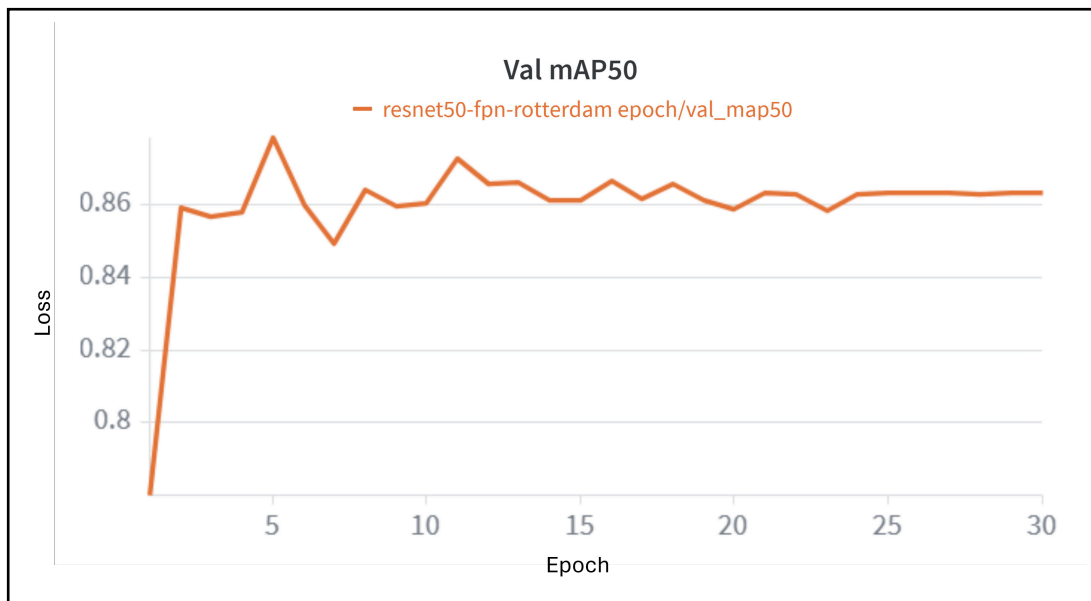
Epoch	Train Loss	Val mAP50	LR	Note
1	1.2234	0.7801	0.005000	Starting point
2	0.6239	0.8593	0.005000	New best; transfer learning effect
3	0.5305	0.8566	0.005000	
4	0.4810	0.8581	0.005000	
5	0.4571	0.8784	0.005000	Best checkpoint
6	0.4327	0.8601	0.005000	mAP50 begins to drop
7	0.4061	0.8493	0.005000	
10	0.37	0.874	0.000500	LR decay applied
24	0.2686	0.8629	0.000050	
25	0.2686	0.8631	0.000050	
30	0.2655	0.8631	0.000050	Final epoch

5.2. Method 2 - Object-level classification

Figure 5.25 shows the training loss and validation mAP50 across all 30 epochs, logged via WandB.



(a) Training loss



(b) Validation mAP50

Figure 5.25: (a) Total training loss per epoch. (b) Validation mAP50 per epoch. The best checkpoint at epoch 5 (mAP50 = 0.8784) and the StepLR decay is implemented at epoch 10.

The training follows three distinct phases. In the first two epochs, the total loss drops sharply from 1.2234 to 0.6239 and mAP50 rises from 0.7801 to 0.8593. This rapid improvement reflects the effect of transfer learning from COCO pretrained weights, which provide a strong starting point for general object detection before adaptation to the Rotterdam map domain. From epoch 3 to 5, the loss continues to decrease and mAP50 reaches its peak of 0.8784 at epoch 5, at which point the best checkpoint is saved. From epoch 6 onward, validation mAP50 oscillates and slightly decreases while training loss continues to fall, indicating mild overfitting to the synthetic training tiles. The StepLR decay at epoch 10 reduces the learning rate from 0.005 to 0.0005, after which the training loss decreases more slowly and validation mAP50 stabilizes in

the range 0.862-0.863 for the remaining epochs. The best checkpoint at epoch 5 is used for all subsequent inference.

Figure 5.26 shows the five individual loss components across all epochs. Five loss components plotted across 30 epochs: classification (`loss_classifier`), bounding-box regression (`loss_box_reg`), binary mask (`loss_mask`), objectness (`loss_objectness`), and RPN regression (`loss_rpn_box_reg`). `loss_mask` shows the slowest convergence and remains the largest single component at epoch 30 (0.1252), while `loss_classifier` and `loss_objectness` approach near-zero, indicating confident building-background separation. The StepLR decay is implemented at epoch 10.

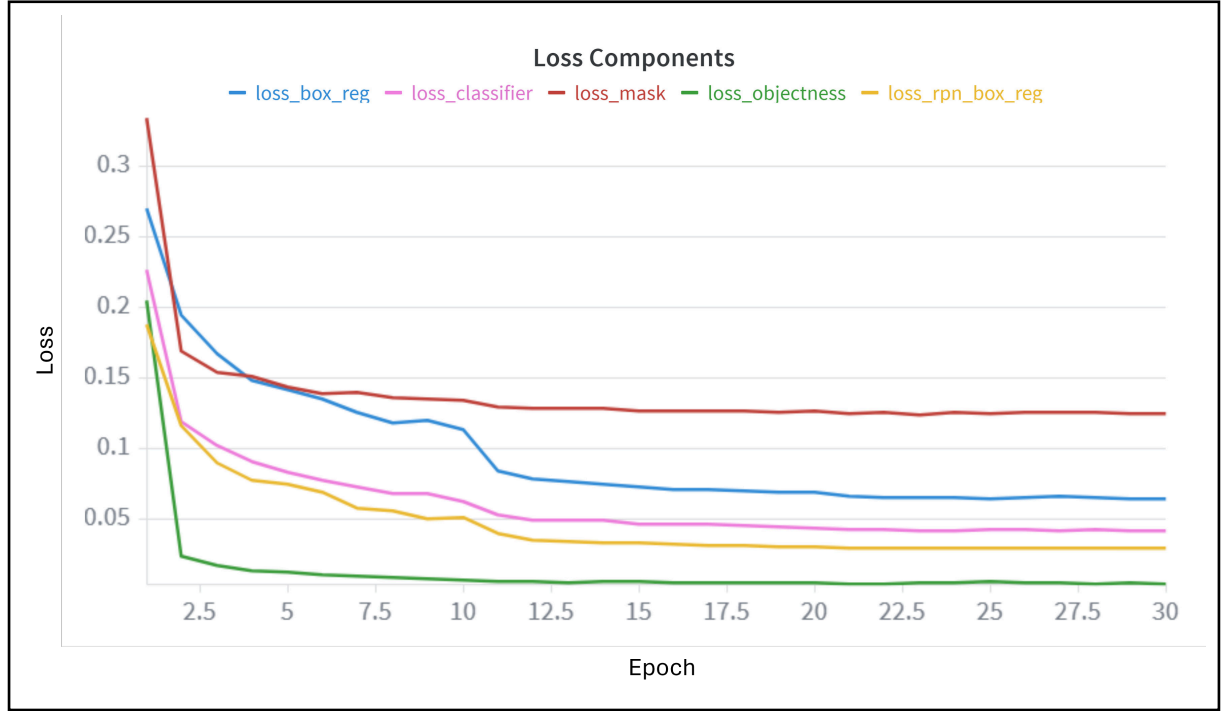


Figure 5.26: Mask R-CNN individual training loss components across 30 epochs.

Table 18 reports the loss breakdown at selected epochs for reference: epochs 1, 5 (best checkpoint), and 30 (final epoch).

Table 18: Mask R-CNN training loss breakdown at selected epochs.

Epoch	Total	loss_cls	loss_box_reg	loss_mask	loss_obj	loss_rpn_box_reg
1	1.2234	0.2269	0.2699	0.3339	0.2049	0.1879
5	0.4571	0.0838	0.1419	0.1435	0.0126	0.0754
30	0.2655	0.0417	0.0646	0.1252	0.0046	0.0294

The mask loss is the largest single component at epoch 30 (0.1252) and shows the smallest relative reduction between epoch 5 and epoch 30 compared to the other components. The classification and objectness losses both approach near-zero by epoch 30, indicating that the model has learned to confidently distinguish buildings from background. The mask branch, which must predict a precise per-pixel boundary for each detected instance, remains the most difficult sub-task, consistent with the complex and irregular building boundaries in the Rotterdam 1897 maps.

Inference

Inference is applied to the historical Rotterdam 1897 map sheets 2, 3, and 4 using the best model checkpoint from epoch 5. The V3 inference configuration (tile=512 px, overlap=128 px, confidence threshold=0.5, NMS IoU=0.3) is used for all sheets. An additional parameter comparison (V1 and V2) was performed on sheet 3 only, to determine the final V3 configuration.

Parameter comparison

Three inference configurations are tested on sheet 3. The results are summarized in Table 19.

Table 19: Inference parameter comparison for sheet 3.

Parameter / Metric	V1	V2	V3
Overlap (px)	64	128	128
Conf. threshold	0.5	0.3	0.5
NMS IoU threshold	0.3	0.2	0.3
Total predictions	13,499	14,491	15,659
building	13,232	14,098	15,266
building_hatching	267	393	393
Mean conf. score	0.941	0.882	0.922
Median area (m ²)	177	164	171
Max area (m ²)	145,778	157,681	157,681

V1 uses a small tile overlap (64 px), which reduces the number of processed tiles but increases the number of buildings split at tile boundaries. V2 increases the overlap to 128 px and lowers the confidence threshold to 0.3 to capture more detections. However, the lower threshold introduces more noise, visible in the reduced mean confidence score (0.882 vs 0.941 in V1), and also illustrated in Figure 5.27, where text labels are predicted as false positive. V3 restores the confidence threshold to 0.5 while keeping the 128 px overlap, combining the coverage benefit of V2 with the prediction quality of V1. V3 is selected as the final configuration.

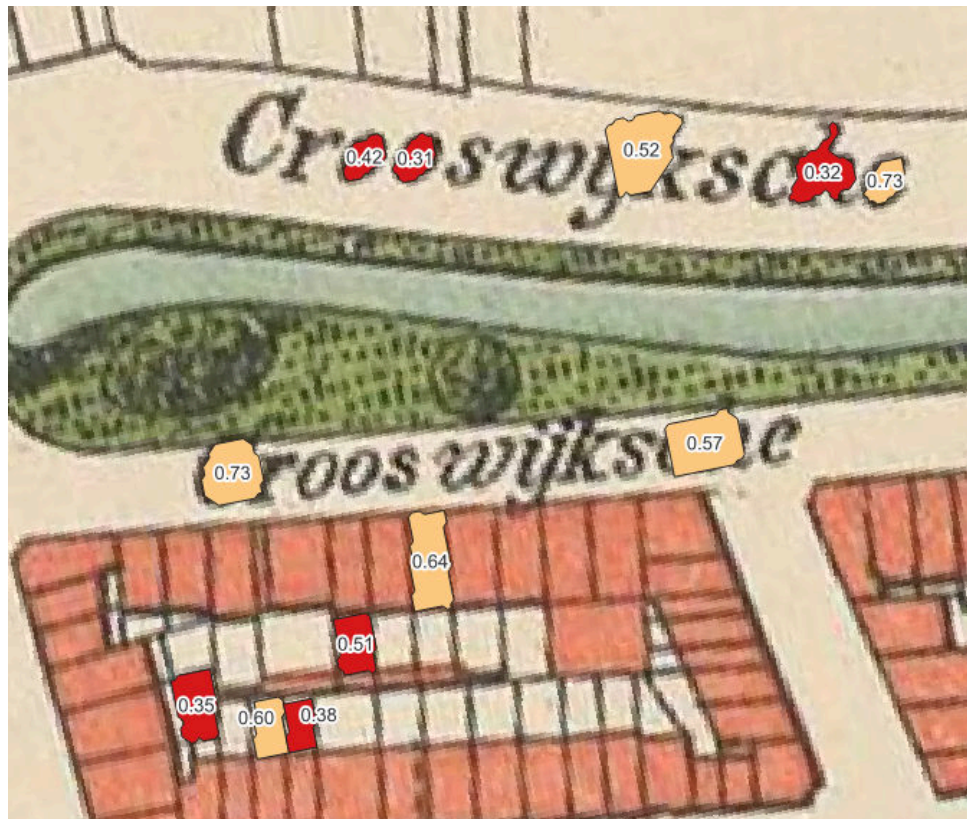


Figure 5.27: Prediction noise with lower confidence score: text labels and backyards as false positive.

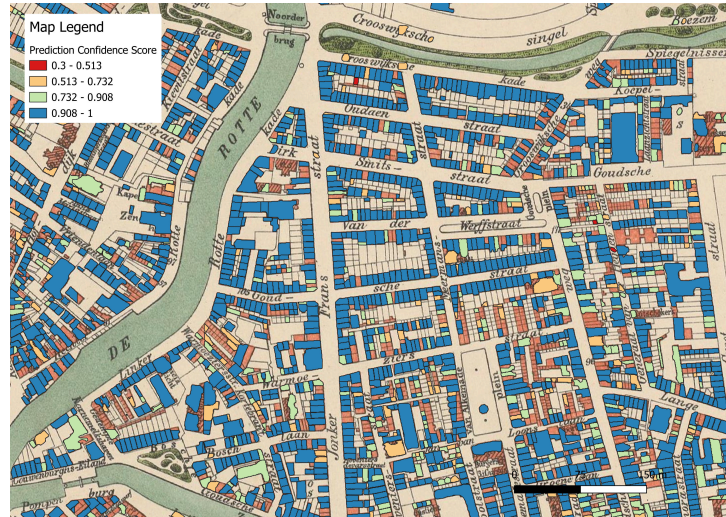
Visual observation

Building detection across urban density levels. Figure 5.28 shows prediction output in three areas of different urban density, colored by confidence score. In sparse areas, individual buildings are detected with high confidence (dark blue). In medium-density areas, predictions remain mostly complete with fewer missed buildings. In dense urban fabric such as the Rotterdam city center, detection quality declines: confidence scores are lower (yellow to red range), individual building boundaries are harder to distinguish, and instances are more likely to be merged or fragmented. This pattern is consistent with the training data: the synthetic BGT tiles provide clear, well-separated building boundaries, while densely packed historic buildings with shared walls or no visible gaps between them are underrepresented.

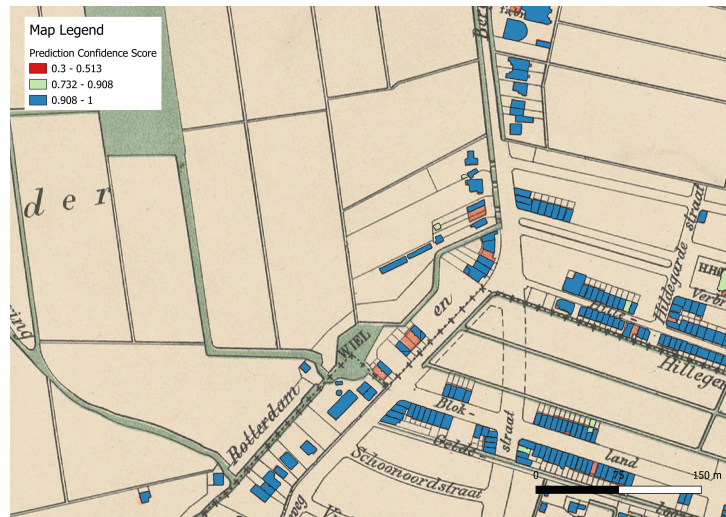
5.2. Method 2 - Object-level classification



(a) Dense urban fabric



(b) Medium density



(c) Sparse area

Figure 5.28: Mask R-CNN prediction output colored by confidence score: dark blue (> 0.908), pastel green (0.732–0.908), yellow (0.513–0.732), red (0.300–0.513). (a) Dense urban fabric (Rotterdam centrum): lower confidence and more fragmented predictions. (b) Medium density area: cleaner and more complete predictions. (c) Sparse area: high-confidence individual building detections.

Detection of hatched buildings (building_hatching class). Figure 5.29 illustrates prediction quality for the hatching class under two conditions. Where hatched buildings are free of text overlays, the model predicts the correct class and footprint shape. Where text labels or street names overlaid the hatched building symbol, predictions either fail entirely or capture only part of the building footprint. This failure mode is expected: the synthetic training data does not fully replicate the density and style of Dutch handwritten text labels present in the real 1897 map scans.



(a) Hatched buildings without text overlays

(b) Hatched buildings with text overlays

Figure 5.29: (a) Hatched buildings without text overlays: correctly predicted with accurate footprints and class labels, shown in blue outline. Regular buildings shown in yellow outline. (b) Hatched buildings with text overlays: predictions are either completely missed or only partially detected.

Prediction splitting at tile boundaries. Large and elongated buildings that span two adjacent tiles are susceptible to being predicted as two separate polygons rather than one continuous footprint. Figure 5.30 shows an example of this artifact. Increasing the tile overlap from 64 px (V1) to 128 px (V3) reduced the frequency of this artifact by ensuring that more of the building area appears in at least one complete tile. However, the artifact persists for buildings whose width or length exceeds the overlap margin, particularly elongated industrial or civic structures.

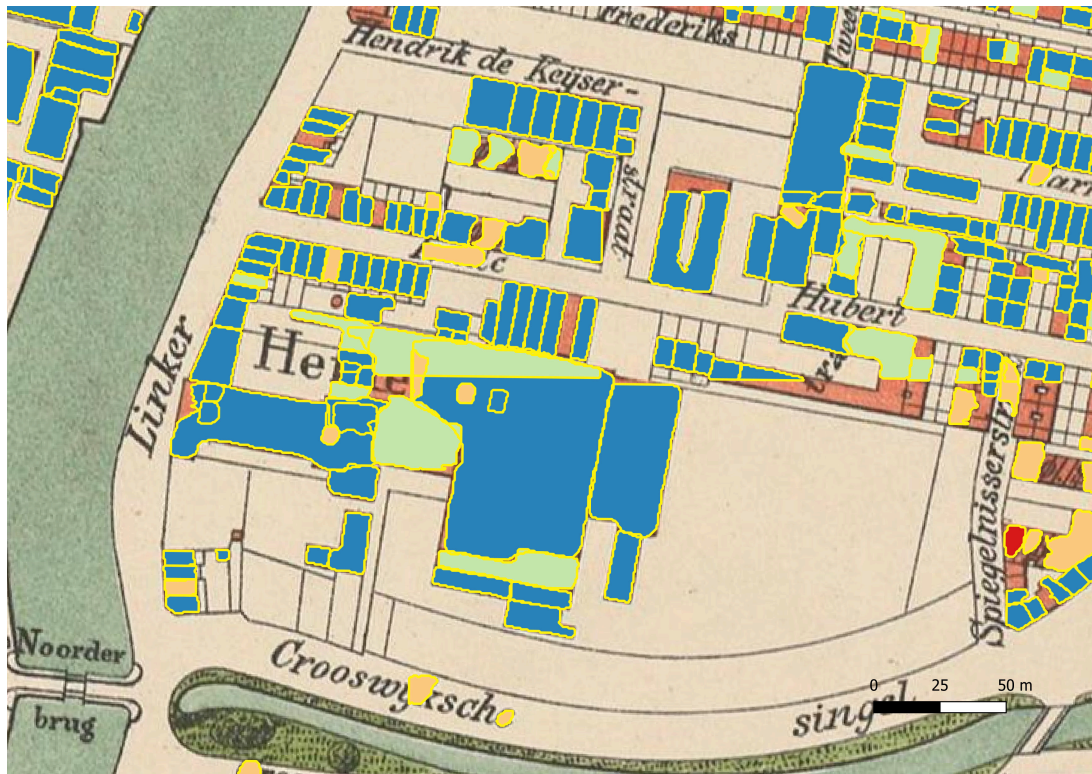


Figure 5.30: A large building predicted as two separate polygons due to a tile boundary crossing the building footprint. Increasing overlap from 64 px to 128 px reduced this artifact but did not eliminate it for wide or elongated buildings.

Irregular polygon boundaries and inter-building gaps. Raw Mask R-CNN predictions trace pixel-level contours from binary instance masks, producing jagged polygon boundaries that do not reflect the rectilinear shapes of real buildings. Figure 5.31 shows related artifacts that result from this: small slivers between adjacent buildings where the predicted boundaries do not meet exactly, and partial overlaps where predicted polygons slightly intersect. Both artifacts arise from the 0.563 m pixel resolution of the map scan images, which limits the geometric precision of the predicted masks. Shape regularization in the post-processing step addresses these artifacts by orthogonalizing the polygon boundaries.

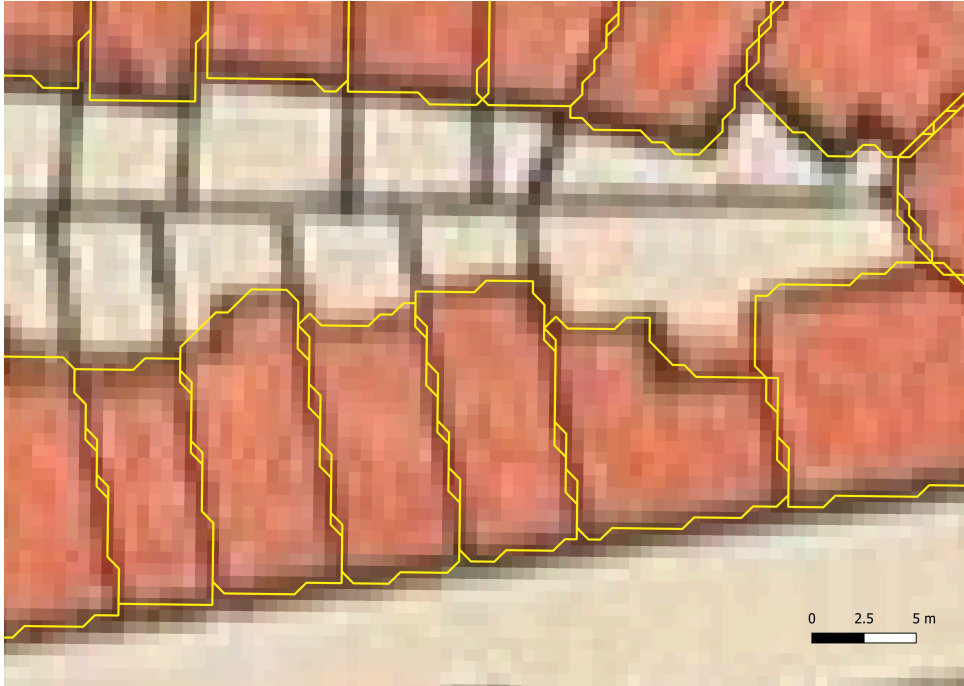


Figure 5.31: Slivers, gaps, and overlaps between adjacent predicted buildings

Post-processing

The raw inference output contains several artifact types that must be addressed before the predicted polygons can be used as building footprints. These artifacts arise from four sources: (1) text characters on the map that visually resemble small building shapes, producing false positive detections; (2) duplicate predictions from overlapping tiles; (3) jagged polygon boundaries that trace pixel contours rather than the rectilinear building shapes expected in the output; and (4) tile edge effects that split large buildings into multiple polygons. The post-processing pipeline applies a set of filters and regularization steps as described in Section 3.2.4.

The raw V3 inference output on sheet 3 is post-processed using the pipeline described in Section 3.2.4. This section reports the parameter configuration selected after tuning and the output statistics for Sheet 3.

Parameter tuning. Eight configurations were tested in total by visual inspection in QGIS to understand each parameter to the result. The best two configurations are BR7 and BR8. Both differ in three parameters: `simplify_tolerance`, `parallel_threshold`, and `snap_tolerance`. BR8 produced more rectangular building shapes with cleaner neighbour alignment and was selected as the final configuration. Table 20 lists the full parameter values for both configurations.

Table 20: Building-Regulariser parameter configurations.

Parameter	BR7	BR8 (selected)
Simplify tolerance (m)	1.5	2.25
Parallel threshold (m)	1.0	1.5
Snap tolerance (m)	1.0	0.5
IoU NMS threshold	0.5	0.5
IoS cleanup threshold	0.6	0.6
Score filter	0.65	0.65
Min area (m ²)	25	25
Max area (m ²)	10,000	10,000
Neighbour max rotation (deg)	10	10
Neighbour search distance (m)	30	30

Filtering and regularisation results. Table 21 reports the polygon count at each pipeline step for the BR8 configuration.

Table 21: Polygon counts at each post-processing step for BR8 on Sheet 3.

Step	Input	Removed	Remaining
Raw inference	-	-	14,035
Area filter (25 – 10,000 m ²)	14,035	635	13,400
IoU NMS (threshold = 0.5)	13,400	2	13,398
IoS cleanup (threshold = 0.6)	13,398	253	13,145
Score filter (≥ 0.65)	13,145	382	12,763
Snap cleanup + regularisation	12,763	—	12,766



Figure 5.32: Building polygons after post-processing and regularisation, shown in green. Raw Mask R-CNN predictions shown in red for comparison.

Internal evaluation – Reliability zone classification

After regularisation, polygons smaller than 25 m² are removed, as regularisation may reduce polygon area below the minimum threshold. This yields 12,299 polygons for analysis.

Each polygon is assigned to one of three reliability zones based on confidence score and regularisation IoU. The regularisation IoU measures how closely the regularised polygon matches the original predicted shape. A low value indicates that the predicted shape was irregular and the regulariser significantly modified it.

- **Reliable:** score ≥ 0.80 and IoU ≥ 0.60
- **Acceptable:** all other combinations
- **Poor:** score < 0.80 and IoU < 0.60

Figure 5.33 shows the distribution of confidence scores across all 12,299 polygons. Figure 5.34 shows the joint distribution of score and regularisation IoU by zone. Table 22 reports the counts and descriptive statistics per zone.

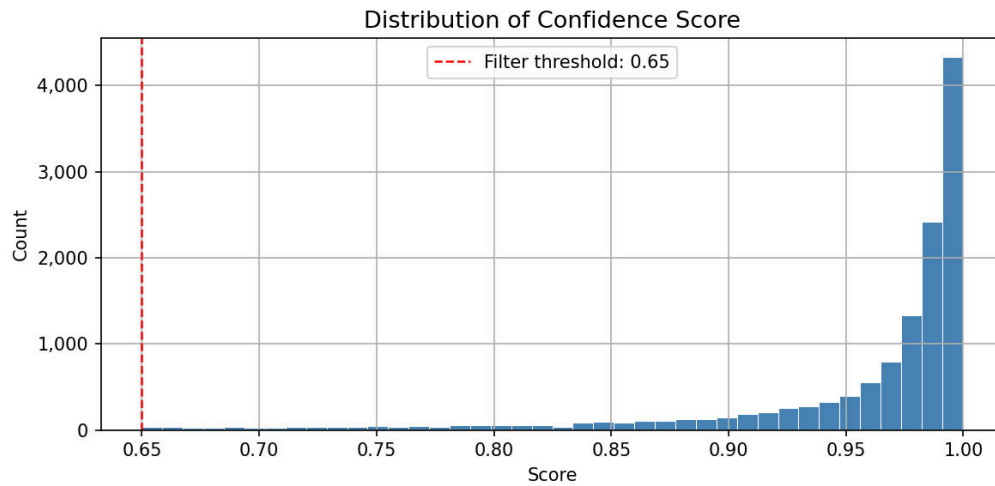


Figure 5.33: Distribution of confidence scores after post-processing. The dashed line marks the pre-filter boundary at score = 0.65.

5.2. Method 2 - Object-level classification

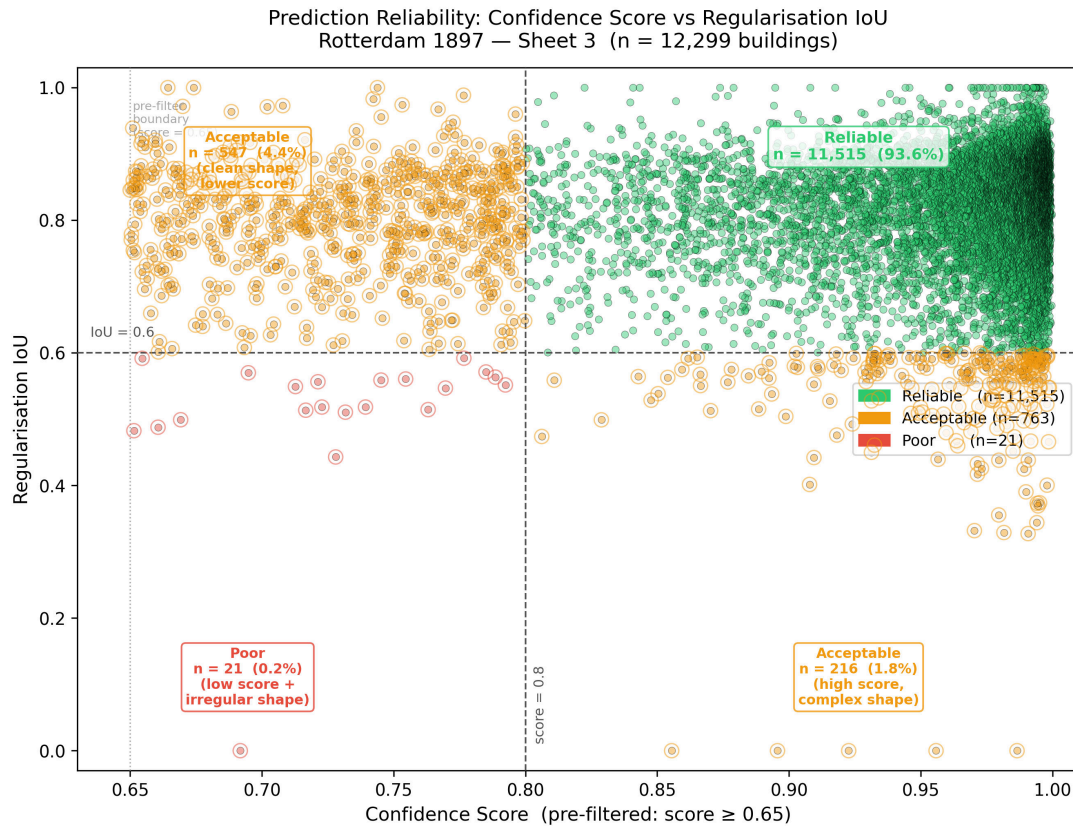


Figure 5.34: Joint distribution of confidence score and regularisation IoU for post-processed building polygons. Dashed lines mark zone boundaries at score = 0.80 and IoU = 0.60. The dotted vertical line marks the pre-filter boundary at score = 0.65.

The majority of polygons fall in the Reliable zone (93.6%), indicating that most predicted buildings have both high detection confidence and well-regularised shapes. The Acceptable zone contains 6.2%; these are buildings where either confidence or shape quality is lower. Only 21 polygons (0.2%) fall in the Poor zone. Table 22 summarises the zone statistics.

Table 22: Reliability zone summary for post-processed building polygons.

Zone	Count	% Total	Mean Score	Mean IoU	Median Area (m ²)
Reliable	11,515	93.6%	0.969	0.824	65.7
Acceptable	763	6.2%	0.796	0.725	49.2
Poor	21	0.2%	0.727	0.509	38.1
Total	12,299	100.0%	0.958	0.817	64.7

Visual inspection in QGIS reveals that some Reliable-classified polygons still have irregular shapes. These are buildings where the original predicted mask was affected by pixel-traced contours or tile-edge effects. The confidence score and regularisation IoU do not directly measure geometric shape quality. A polygon can score high on both metrics if the model was confident and the regulariser made minimal changes, yet the underlying predicted shape remains irregular. The zone classification is therefore a first-order quality indicator, not a guarantee of geometric accuracy.

External evaluation – BGT reference matching

The internal evaluation does not use an external reference dataset. To complement it, the post-processed predictions are compared against BGT pand buildings with bouwjaar < 1897 within the Sheet 3 boundary. These are buildings confirmed to have existed before 1897 and still present in the current BGT. The reference set contains 1,701 buildings.

The matching procedure follows the urban persistence methodology defined in Section 3.3.2. The same procedure is interpreted as an urban change analysis in Section 5.4.2. Parameters applied here: OBB fit ratio threshold = 1.25, centroid distance threshold = 8 m, IoU threshold = 0.30.

Recall is the primary metric. A high recall means the model detects most confirmed pre-1897 buildings. Precision is reported separately with a caveat: unmatched predictions may be correctly detected 1897 buildings that were demolished after the map was made and are absent from the current BGT. Table 23 reports the results. Figure 5.35 illustrates the spatial distribution of matched and unmatched buildings for both the BGT reference set and the Mask R-CNN predictions.

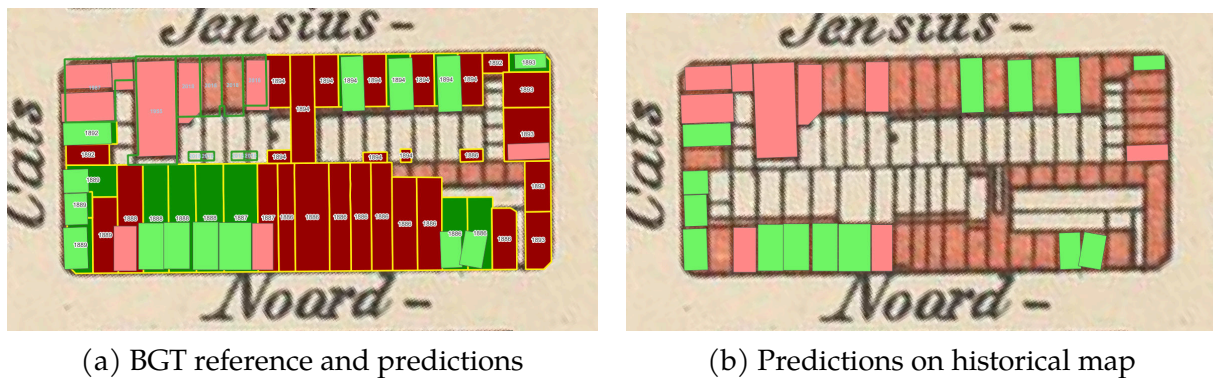


Figure 5.35: Spatial distribution of BGT reference matching results. (a) BGT reference buildings and Mask R-CNN predictions: dark green with yellow outline = matched BGT (*bouwjaar* label shown); dark red with yellow outline = unmatched BGT; light green = matched prediction; light red = unmatched prediction. (b) Mask R-CNN predictions overlaid on the Rotterdam 1897 municipality map: light green = matched prediction (persistent building); light red = unmatched prediction.

Table 23: BGT reference matching results.

Metric	Value	Note
True Positive (TP)	789	Predicted building matched to BGT reference
False Negative (FN)	1,116	BGT reference building not detected
False Positive (FP)	10,663	Unmatched prediction
Recall (primary)	41.4%	TP / (TP + FN)
Precision	6.9%	Interpret with caution (see below)
F1	0.118	

The recall of 41.4% means that of the confirmed surviving pre-1897 buildings, 789 were detected by the model. The low precision (6.9%) reflects the urban history of Rotterdam rather than model performance. The city centre was largely destroyed in the May 1940 bombing and subsequently rebuilt. Most predicted buildings are correctly detected 1897 buildings that

5.2. Method 2 - Object-level classification

no longer exist in the current BGT and therefore appear as false positives. Precision is not a meaningful metric in this context and is reported only for completeness.

The mean centroid distance between matched pairs is 1.85 m (median 1.47 m). This is consistent with the georeferencing accuracy reported in Table 12: the 11-ICP RMSE of 1.137 m represents georeferencing error alone, while prediction uncertainty raises the observed offset to 1.85 m. The maximum matched centroid distance of 7.38 m corresponds to the elevated residuals at the sheet edges reported for ICPs 6, 14, and 15.



Figure 5.36: Spatial distribution of matching results across Sheet 3. Green: true positive predictions matched to a BGT reference building. Orange: unmatched predictions (FP). Red outline: BGT reference buildings not detected (FN).

5.3. Visual Comparison of Method 1 and Method 2: Rotterdam

This section compares the outputs of Method 1 and Method 2 on the same study area: Rotterdam, around 1897. The comparison is based on visual inspection, as the two methods

produce outputs at different spatial resolutions and granularity levels that do not allow direct quantitative comparison.

Table 24 summarises the key differences between the two datasets used in this comparison.

Table 24: Dataset comparison between Method 1 and Method 2 for Rotterdam.

Attribute	Method 1	Method 2
Map series	Bonnebladen	Rotterdam municipality
Sheet	502	Sheets 2, 3, 4
Year	1901 (publication)	1897 (survey/publication unknown)
Map scale	1:25,000	1:5,000
Georeferencing	4 corner points	TPS, 100 GCPs
Output type	Classified patch grids	Building polygons

There is a temporal gap of approximately four years between the two datasets. The Bonnebladen sheet used in Method 1 is the 1901 edition, while the municipality map used in Method 2 is the 1897 edition. Figure 5.37 shows both input map sheets.

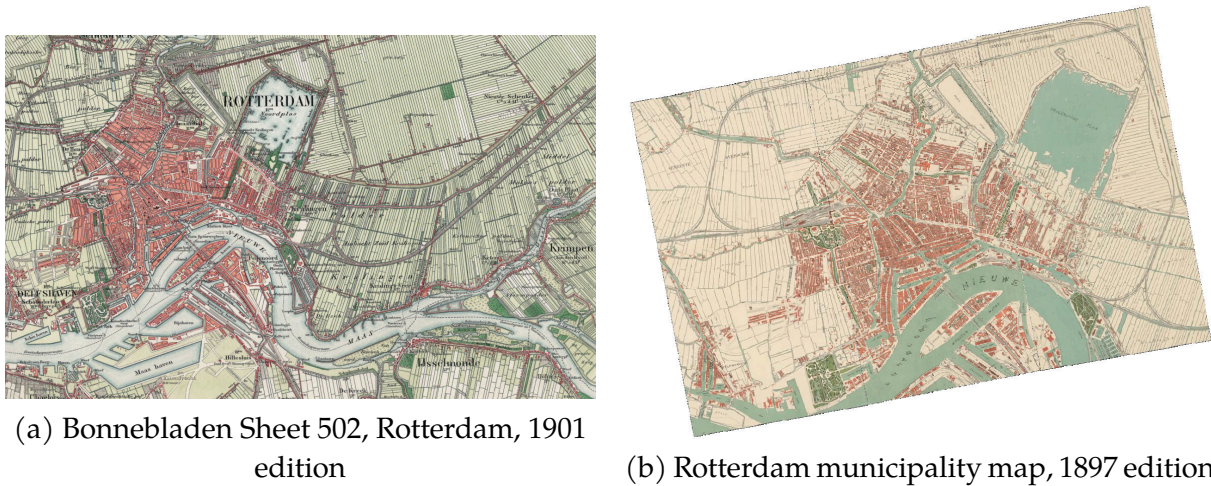


Figure 5.37: Input map sheets for Method 1 and Method 2 on the Rotterdam study area. (a) Bonnebladen Sheet 502, 1901 edition, used in Method 1. (b) Rotterdam municipality map, 1897 edition, used in Method 2. The two sheets cover a similar spatial extent but differ in map scale and cartographic style.

Figure 5.38 shows the spatial overlay of both outputs on the Bonnebladen 1901 sheet. Red patch grids show the Method 1 output. Yellow building polygons show the Method 2 output.

At the overview level, the two methods show broad spatial agreement. The building polygons extracted from the municipality map in Method 2 follow the same block patterns visible in the Bonnebladen sheet used in Method 1. Urban areas identified by Method 1 generally correspond to areas where Method 2 also detects buildings.

At the local level, some disagreements are visible. These are expected given the differences in map scale, cartographic style, temporal gap, and the performance of each deep learning model. A slight positional offset between the two outputs is also visible. Method 2 used a more precise georeferencing approach (TPS transformation with approximately 100 GCPs), while the Bonnebladen dataset used for Method 1 was georeferenced on four map corner points only.

5.3. Visual Comparison of Method 1 and Method 2: Rotterdam

The positional offset between the two outputs may therefore partly reflect this difference in georeferencing precision.

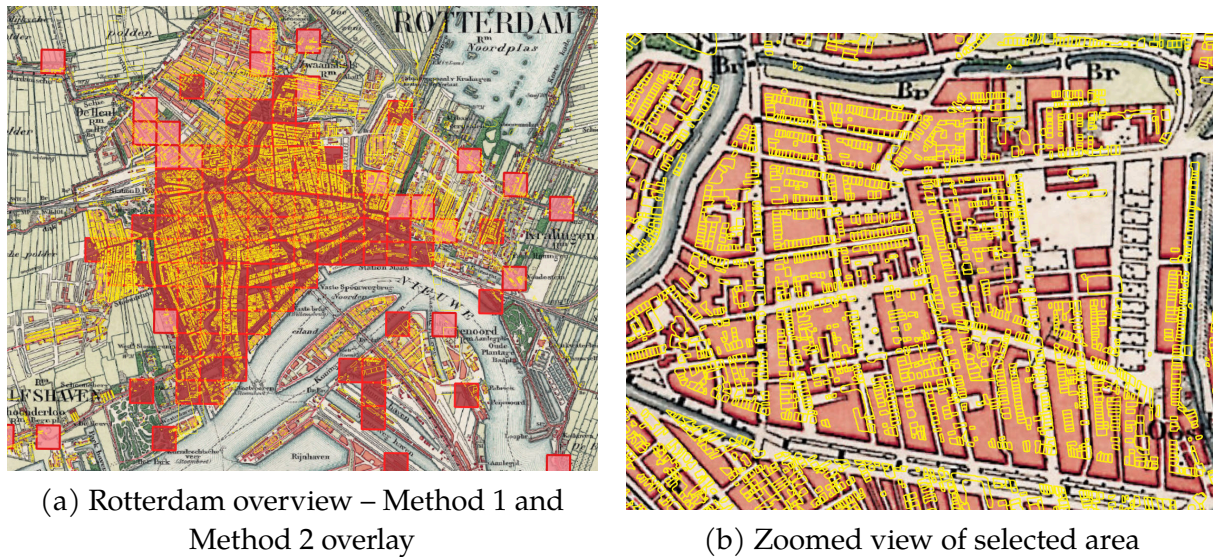


Figure 5.38: Overlay of Method 1 and Method 2 outputs on the Rotterdam study area. Background: Bonnebladen Sheet 502, 1901 edition. Red grids: classified patches from Method 1. Yellow polygons: predicted building footprints from Method 2. (a) Full Rotterdam extent. (b) Zoomed view of a selected area showing local spatial agreement and disagreement between the two outputs.

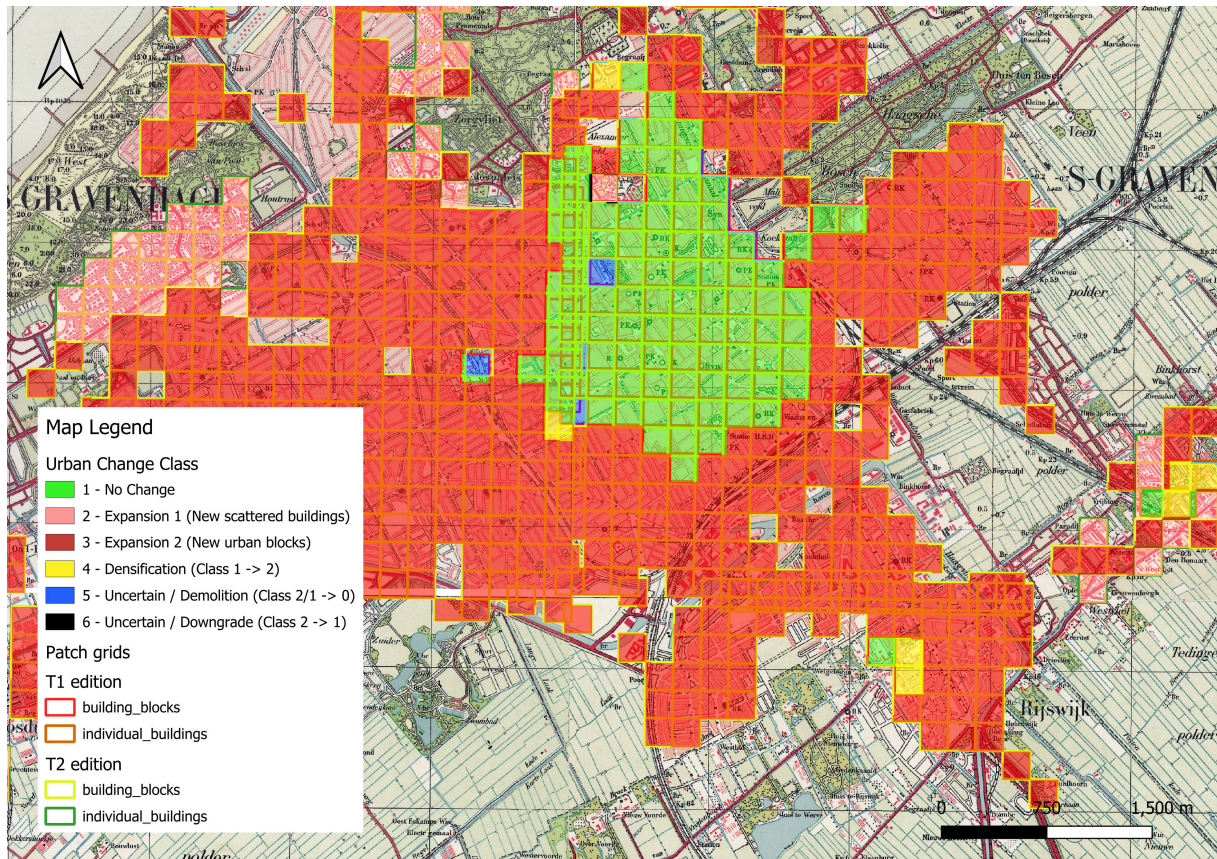
5.4. Change analysis

5.4.1. Method 1: Post-classification change analysis

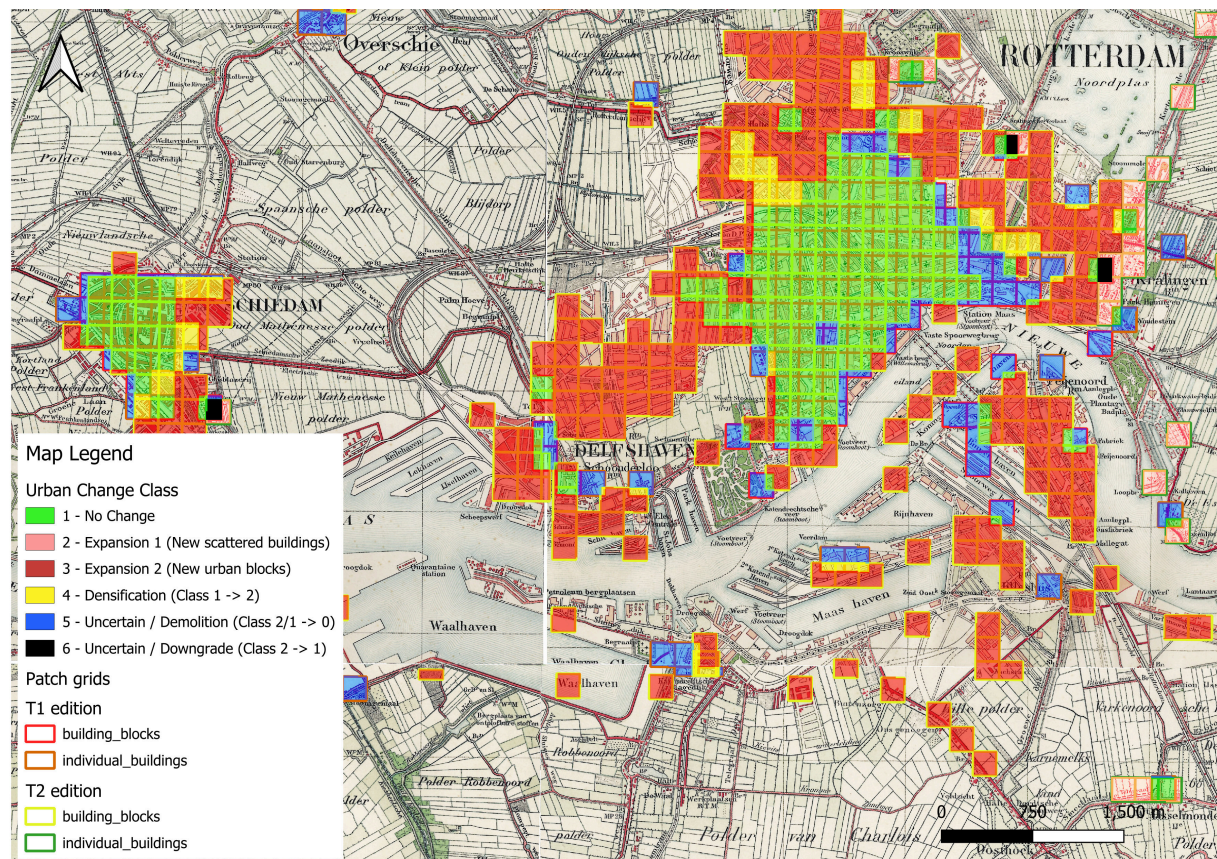
Table 25 presents the full change detection results for South Holland, comparing T1 (1869–1916) and T2 (1910–1943). Urban-related classes (1–6) cover a combined area of approximately 113 km², representing about 1.6% of the total valid provincial area (~7,263 km²). Figure 5.39 shows the spatial distribution of change classes on the S'GRAVENHAGE and ROTTERDAM sheets.

Table 25: Change detection results for Method 1, South Holland.

Output class	Label	T1 -> T2	Pixel count	Area (km2)
0	Background	0 -> 0	71 492 357	7149.7
1	No Change	1->1 / 2->2	211 511	21.2
2	Expansion 1	0 -> 1	181 686	18.2
3	Expansion 2	0 -> 2	419 433	41.9
4	Densification	1 -> 2	52 350	5.2
5	Uncertainty / Demolition	1/2 -> 0	262 189	26.2
6	Uncertainty / Downgrade	2 -> 1	6 265	0.6



(a)



(b)

Figure 5.39: Change map for South Holland comparing T1 patch predictions against T2 predictions. (a) S'GRAVENHAGE sheet. (b) ROTTERDAM sheet.

5.4. Change analysis

Interpretation caveats. Before interpreting the results, two limitations of this analysis must be stated. First, both T1 and T2 rasters are produced using the weak classifier, which was trained exclusively on T1 patches. As discussed in Section 5.1.6, the weak classifier shows reduced performance on T2 editions due to cartographic style changes introduced in post-1930s sheets (Section 5.4). This primarily causes false negative predictions in T2, where buildings are present but not detected. Second, a spatial misalignment between T1 and T2 patch grids is observed during visual inspection. Despite covering the same geographic area, patches from different editions show a positional shift of up to several dozens of pixels. This produces narrow boundary artifacts in the change raster, where pixels along patch edges are assigned a change class due to misalignment rather than real land cover change. Both issues affect the interpretation of all change classes and are discussed in detail below.

Urban extent estimates. The T1 urban extent, estimated by summing the classes that were urban in T1 (No Change, Densification, Uncertainty / Demolition, and Uncertainty / Downgrade), is approximately 53 km². The T2 urban extent, estimated from classes that are urban in T2 (No Change, Expansion 1, Expansion 2, Densification, and Uncertainty / Downgrade), is approximately 87 km². However, the T2 estimate is likely underestimated. As discussed below, the Uncertainty / Demolition class (~26 km²) is largely attributed to T2 false negatives rather than real demolition. If those pixels are considered persistent urban, the corrected T2 urban extent approaches approximately 113 km².

Urban growth. Total detected urban growth, summing Expansion 1, Expansion 2, and Densification, is approximately 65 km². Expansion 2 (new building blocks, ~42 km²) is the dominant growth type, accounting for approximately 64% of total detected growth. Expansion 1 (new scattered buildings, ~18 km²) accounts for approximately 28%. Densification (~5 km²) accounts for the remaining 8%, indicating local intensification where scattered buildings transitioned to block-level density. These figures should be interpreted as an upper bound: grid/patch misalignment between T1 and T2 produces narrow boundary artifacts that may inflate all change categories, including the Expansion classes.

Persistent urban area. The No Change class covers approximately 21 km². These pixels were classified as urban in both T1 and T2 with the same class label. This is likely an underestimate of true stable urban area, as T2 false negatives cause some persistent urban pixels to be reclassified into the Uncertainty / Demolition category instead.

Uncertainty / Demolition. The Uncertainty / Demolition class (~26 km²) is the most prominent signal that requires reinterpretation. This class is larger than Expansion 1 (~18 km²), which is inconsistent with expectations for South Holland in this period. Visual inspection of the change map overlaid on T2 map sheets confirms that the majority of Demolition pixels correspond to areas where buildings are clearly visible in T2 but were not detected by the weak classifier. This is consistent with the false negative pattern documented in Section 5.1.6, where the cartographic style shift in post-1930s editions (building outline color change, grid overlay introduction; Section 5.4) degrades model performance on T2 patches. The Demolition class therefore does not reflect real physical demolition. It reflects model failure on T2. This class is excluded from urban growth totals and its area should be treated as undetected persistent urban rather than lost urban extent. The Uncertainty / Downgrade class (~0.6 km²) is negligible.

Patch grid mismatch artifacts. Visual comparison of T1 and T2 patch grids in the same geographic area reveals a consistent positional shift of up to several dozens of pixels between

editions. This misalignment is attributed to several compounding factors: differences in how cartographers drew each edition despite a shared map scale; georeferencing based solely on map margin neatlines, which does not guarantee sub-pixel accuracy (Meijers & Schoonman, 2025); distortion introduced during reprojection from the original Bonne projection to EPSG:28992; and potential differences in the pixel coordinates of patch boundaries caused by tilted scans or inaccurate image masks in the IIF georeference annotations. In the change raster, this misalignment manifests as narrow bands of change pixels along patch boundaries, present across all change categories. These artifacts cannot be separated from genuine change signals at the current processing stage and represent a known source of overestimation in all change class areas.

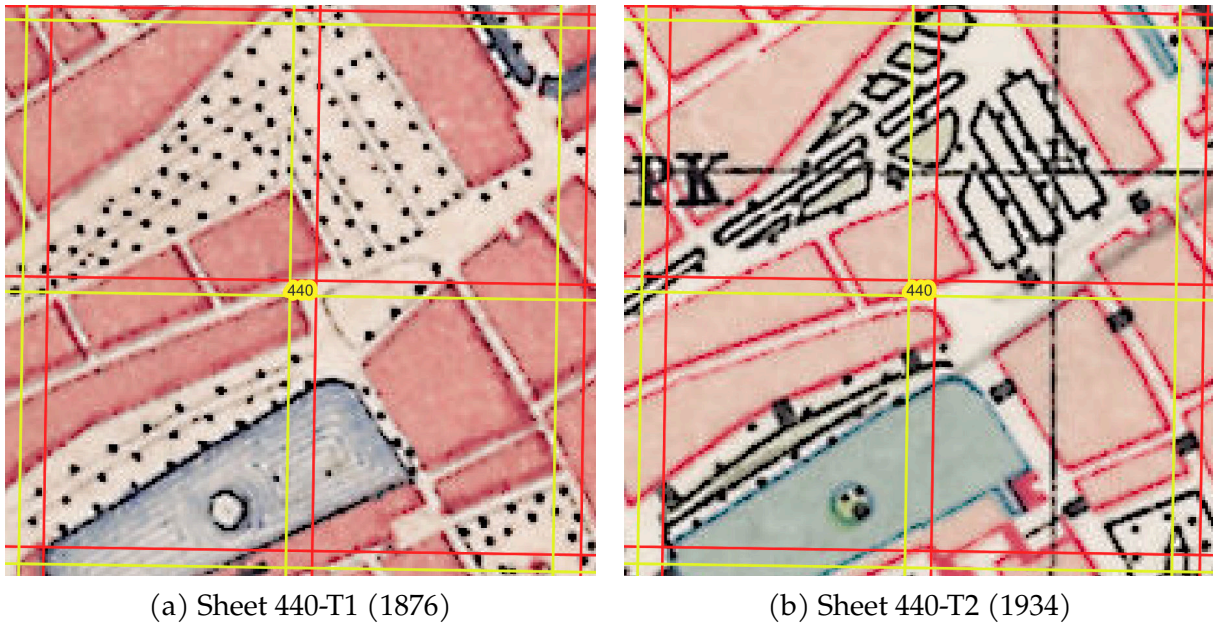


Figure 5.40: Patch grid mismatch between Bonnebladen editions (a) Sheet 440-T1 (b) Sheet 440-T2. Both sheet are overlaid with patch gridlines of T1 (yellow grids) and T2 (red grids). Location: Binnenhof, The Hague.

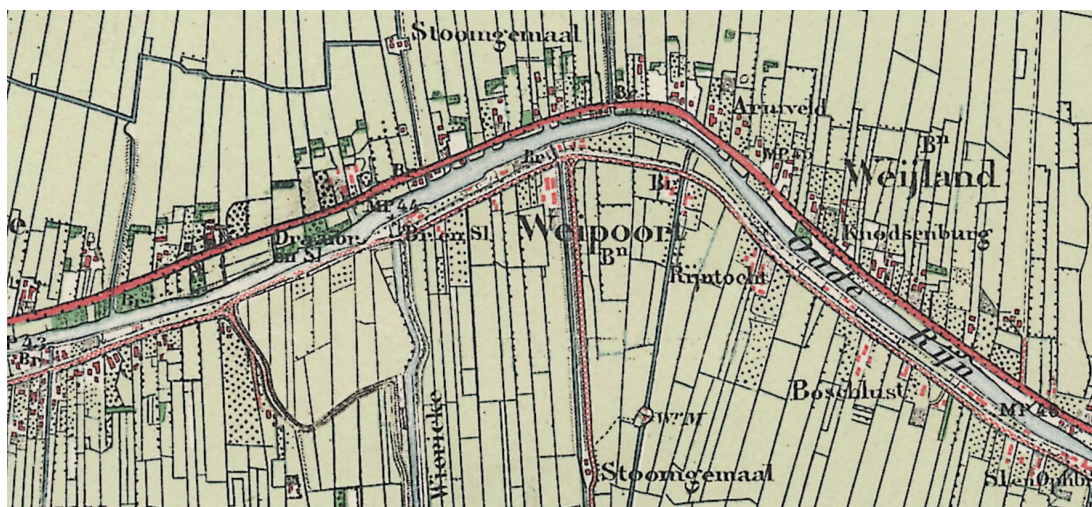
Edition types: general and secret military

Two edition types exist in the catalog. General-purpose editions constitute the majority of the dataset. A subset of sheets also has a secret military edition, flagged in the DANS catalog as *OOK GEHEIME UITGAVE* (also secret edition). These secret editions include depictions of military fortifications that were omitted from, or intentionally obscured in, the corresponding general-purpose edition. Unlike what the flag description implies, both edition types are accessible through the publicly available IIF resources maintained by TU Delft Library, and both are included in this research.

Sheet 443 (*Bodegraven*) illustrates the difference explicitly: in its first edition, the Wierickerschans Fort is clearly depicted in Figure 5.41a, while in a later edition the same area is rendered without military structures and turned into an open land Figure 5.41b. This raises a consideration for the change analysis: if T1 and T2 for a given sheet correspond to editions of different types (one general-purpose, one secret military), observed differences in urban content may partly reflect cartographic omission rather than genuine land cover change.



(a)



(b)

Figure 5.41: (a) Secret military (*Geheim*) edition. (b) General edition.

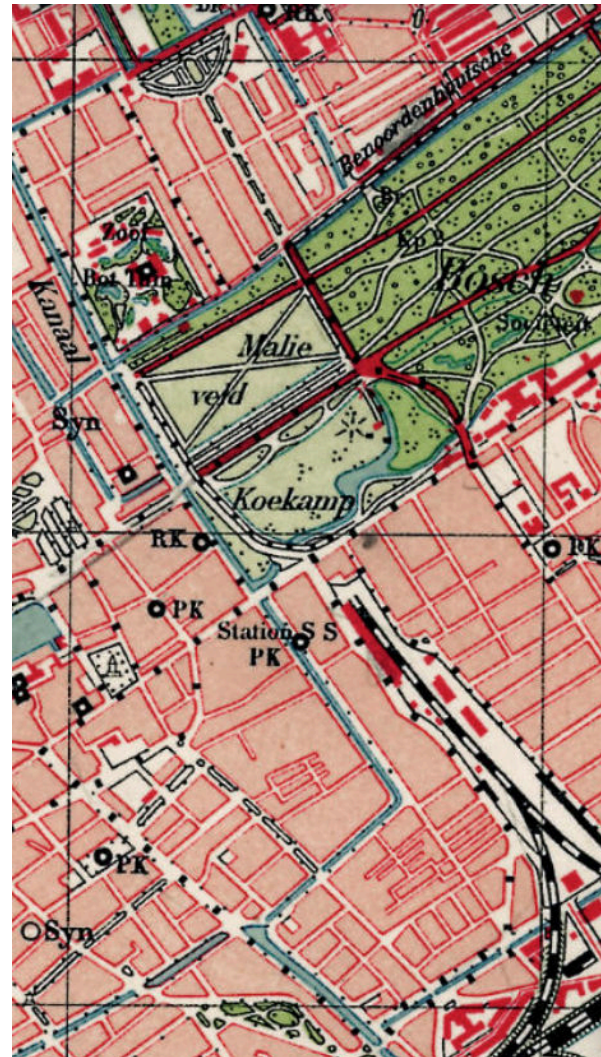
Cartographic style changes and print/scan quality inconsistencies

A shift in cartographic style occurs in editions published after approximately 1930, consistent with observations reported in the HGN methodology (Knol et al., 2004). Visual comparison of sheets before and after this period, conducted in QGIS using the pipeline-produced GeoTIFFs, reveals several changes, as illustrated in Figure 5.42.

- **Map grid introduction.** Editions after the 1930s include a kilometer grid overlay. Measurements in QGIS confirm a grid spacing of 1×1 km.
- **Building outline color.** The building outline stroke shifts from a near-black color (approximately RGB 26, 20, 19) to a clear red (approximately RGB 194, 29, 43). This change is unambiguous and directly affects the inference result. (Section 5.1.6).
- **Railway symbology.** The graphical representation of railways changes between early and late editions.



(a)



(b)

Figure 5.42: (a) Pre-1930s edition showing dark building outlines and no map grid. (b) Post-1930s edition showing red building outlines and 1 x 1 km grid overlay.



Figure 5.43: (a) 1874 edition. (b) 1914 edition. (c) 1938 edition.

These stylistic differences introduce potential inconsistencies when applying a single trained model across sheets from different periods.

5.4.2. Method 2: Urban Persistence Analysis

The matching procedure, detection metrics, and spatial results are reported in Section 5.2.3 (external evaluation). This section presents the urban change interpretation of those results.

Persistence rate. Of the 1,701 confirmed pre-1897 BGT buildings within Sheet 3, 789 were matched to a post-processed Mask R-CNN prediction. This corresponds to a persistence rate of 41.4%. These buildings existed in Rotterdam in 1897 and remain in the current cadastre, indicating physical continuity of the built environment at those locations over more than 125 years.

Urban loss since 1897. The remaining 58.6% of confirmed pre-1897 buildings were not matched. This reflects the substantial transformation of Rotterdam's urban fabric since 1897. The primary cause is the destruction of the Rotterdam city centre during the German bombing on 14 May 1940, which destroyed the majority of the pre-war built environment. Subsequent post-war reconstruction replaced most of the historic building stock with new construction, leaving few surviving pre-1897 structures in the city centre area covered by Sheet 3.

Lower bound interpretation. The persistence rate of 41.4% is a lower bound estimate. A fraction of the unmatched BGT reference buildings are genuine persistent buildings that the model failed to detect, classified as false negatives in Table 23. The true persistence rate is therefore higher than 41.4%. The model’s detection failures are concentrated in dense urban fabric and buildings with irregular or overlapping predicted shapes, as discussed in Section 5.2.3.

BGT *bouwjaar* (construction year) context. Within the Sheet 3 boundary, the BGT *Pand* layer contains approximately 13,293 buildings in total:

- buildings constructed before 1897: 1,701 buildings – confirmed pre-1897 stock, consistent with the number of features used in the matching analysis,
- buildings constructed in 1897: 367 buildings – ambiguous; may or may not have existed at the time of the survey,
- buildings constructed after 1897: 11,202 buildings (84% of current stock) – constructed after the map was made.

The scale of post-1897 construction confirms the substantial urban renewal of Rotterdam since 1897. The spatial distribution of persistent and undetected buildings is shown in Figure 5.36.

5.4. *Change analysis*

6. Conclusion

6.1. Research Questions

This section addresses the main research question of this thesis: *“To what extent do patch-level and object-level deep learning approaches differ in their effectiveness for extracting urban extent from multi-temporal historical topographic maps for enabling urban change analysis?”* This is done by answering each sub-research question defined in Section 1.2, based on the results and analysis presented in Chapter 5.

Sub-research question 1: How can patch-level deep learning classification be applied to extract urban extent from multi-temporal historical maps?

Patch-level classification was applied to 74 Bonnebladen sheets covering the Province of South Holland using the MapReader pipeline (Section 3.1). Map sheets were divided into 200-meter square patches and classified into three classes – *no*, *individual_buildings*, and *building_blocks* – defined based on the visual characteristics of the red-fill building symbology on the Bonnebladen series. Manual annotation was performed on a subset of 1,768 patches, assisted by a red-channel pixel statistics pre-filter to reduce the annotation queue to informative patches only. The full annotation was completed in under 2 hours, demonstrating that the approach is feasible even for large map collections.

A ResNet-101 model was fine-tuned for 20 epochs; the best checkpoint was selected at epoch 7. The weak classifier achieved a macro-averaged F1 of 0.900 on the held-out test set, with *building_blocks* performing best (F1 = 0.909) and *individual_buildings* weakest (F1 = 0.808), consistent with the visual ambiguity of that class. Inference was applied to both T1 and T2 sheet sets – covering 127,468 and 128,358 patches respectively – producing georeferenced patch datasets that served directly as input to the post-classification change analysis at provincial scale (Section 5.4). The classifier generalizes to unseen sheets, but performance degraded on T2 editions: false negatives are observed in areas where post-1930s cartographic style changes altered the visual signal the model was trained on (Section 5.4).

The T1 inference results are compared with the HGN-1900 dataset as a spatial plausibility check (Section 5.1.7). This comparison is justified because both datasets are derived from the same physical map editions. The results show broad spatial agreement in non-urban areas and built-up areas, particularly in city cores. Discrepancies are partly explained by the roads-and-built-up conflation in HGN-1900, the resolution difference between the two methods (50 m vs 200 m), and potential edition inconsistencies in the HGN classification.

Sub-research question 2: How can CycleGAN-based synthetic data generation be applied to enable object-level building footprint extraction from historical maps?

The original IIIF georeference annotation used an affine transformation with 14 GCPs, which was insufficient for building-level alignment. Manual improvement of the georeference was conducted using a TPS transformation with 100 GCPs, achieving an RMSE of 1.137 m, within the accepted accuracy threshold for historical map georeferencing (Table 12). This step was a prerequisite for reliable spatial matching between predicted building polygons and reference data. Due to limited identifiable features, only sheet 3 completed the full pipeline; sheets 2 and 4 were excluded.

6.1. Research Questions

BGT vector data was styled to approximate the cartographic conventions of the Rotterdam 1897 maps, then translated using CycleGAN to produce synthetic training images with realistic cartographic aging effects. The epoch 6 checkpoint was selected based on FID evaluation (FID = 151.08) and visual inspection, as later epochs introduced geometric distortions that corrupt the alignment between the synthetic image and its annotation mask (Section 5.2.2). The synthetic GeoTIFFs – containing 32,084 buildings and 1,007 hatched buildings – paired with BGT building polygons as annotation masks, served as training input for Mask R-CNN without any manual annotation of individual building footprints.

The Mask R-CNN model trained on this data achieved a best validation mAP50 of 0.8784 at epoch 5. External evaluation against BGT buildings confirmed to predate 1897 produced a recall of 41.4% (Table 23); the interpretation of this figure in the context of Rotterdam's urban history is discussed in Section 6.2.3. Post-processing was required to filter false positives and regularise irregular polygon boundaries, adding considerable complexity to the pipeline.

Sub-research question 3: What urban change patterns can be identified by comparing automatically extracted urban extents from historical maps against reference data?

For Method 1, post-classification comparison between T1 (1869–1916) and T2 (1910–1943) detected approximately 65 km² of total urban growth across South Holland (Section 5.4.1). New building blocks (~42 km², 64% of total growth) was the dominant pattern, followed by new scattered buildings (~18 km², 28%) and densification (~5 km², 8%). These area estimates should be interpreted as coarse indicators rather than precise measurements. The T1 urban extent is estimated at approximately 53 km², and the corrected T2 urban extent approaches approximately 113 km² when the Uncertainty / Demolition class (~26 km²) is reattributed as undetected persistent urban rather than actual demolition. This reattribution is supported by visual inspection confirming that the Demolition signal is primarily caused by false negatives in T2, where post-1930s cartographic style changes altered the visual signal the weak classifier was trained on (Section 5.4). Patch grid misalignment between T1 and T2 editions further inflates all change class areas and cannot be separated from genuine change signals at this stage.

For Method 2, urban persistence analysis of Rotterdam 1897 sheet 3 identified 789 matched buildings out of 1,701 confirmed pre-1897 BGT buildings, yielding a persistence rate of 41.4% (Section 5.4.2). This figure is a lower bound, as a fraction of unmatched reference buildings reflect model detection failures. These results confirm that the Method 2 vector output supports building-level urban persistence analysis, a type of analysis not possible with the patch-level output of Method 1.

Sub-research question 4: What are the relative strengths, limitations, and trade-offs between patch-level and object-level deep learning approaches for urban extent extraction from historical maps?

Method 1 produces classified raster patches at 200 m resolution, scales to provincial level with limited annotation effort and short training time, and is suited for statistical-level urban change analysis over large areas. Its main limitations are sensitivity to cartographic style shifts across temporal editions and patch grid misalignment artifacts in post-classification comparison. Method 2 produces individual building footprint polygons as vector output, enabling building-level queries and urban persistence analysis not possible with patch-level output, but requires a more complex multi-stage pipeline, is constrained by georeferencing quality, and depends on synthetic training data that may underrepresent the densest historic urban configurations. Both methods benefit from IIIF infrastructure for programmatic map access, and both were

fully implemented on a consumer-grade laptop (RTX 4050, 6 GB VRAM), confirming that the approaches are accessible without HPC infrastructure.

6.2. Discussion

6.2.1. Method 1: Patch-level classification

Annotation efficiency and classifier performance. The annotation pre-filtering step based on red channel pixel statistics proved highly effective because buildings on the Bonnebladen series are rendered with a consistent red fill across all editions, making the red channel a reliable proxy for building content before any model is involved. This design choice is specific to the Bonnebladen collection and would need to be reconsidered for map series that use different color conventions for buildings. Beyond efficiency, the MapReader annotator presents patches drawn randomly from the full sheet set rather than from a single area, ensuring the annotated training set captures cartographic style variation and print quality differences across the collection. This diversity benefits model generalisation and is a deliberate design of the MapReader pipeline (Hosseini et al., 2022).

The strong performance of the weak classifier (Table 8) despite a small training set is partly attributable to the visual distinctiveness of buildings on the Bonnebladen series: the red fill combined with rectilinear building shapes shares few characteristics with other map features. Transfer learning from a ResNet-101 pretrained on ImageNet further accelerates convergence by initialising the model with low-level visual features – edges, textures, color gradients – that transfer effectively to the historical map domain (Hosseini et al., 2022; Larochelle et al., 2009).

Challenges: cartographic style shift and the two-stage training strategy. As documented in Section 5.4, a cartographic style shift occurs in post-1930s Bonnebladen editions: building outlines change from near-black to red, a kilometer grid is introduced, and a greenhouse symbol resembling dense building fill appears. A model trained exclusively on pre-1930s patches does not encounter these signals and therefore degrades on T2 sheets, as observed in Section 5.1.6.

The MapReader pipeline addresses this through a two-stage training strategy. The weak classifier – the first stage, reported in this thesis – is trained on a small manually annotated set. The second stage extends the training set using low-confidence predictions from the full inference output, producing a fine-tuned classifier with broader generalisation. This second stage was not completed due to time constraints.

Challenges: data storage at scale. A practical concern for provincial or nationwide application is data storage. Scaling to the full first-edition collection would require roughly 10 times the storage used in this thesis (30 GB for sheets, 60 GB for patches). This requires deliberate pipeline design: patch files should be generated on demand rather than pre-generated in full, and cloud or institutional storage should be considered for nationwide runs.

Limitations. The 200-meter patch size is a deliberate design choice but also an inherent limitation. It is large enough to identify dominant urban content reliably, but too coarse to resolve precise building boundaries. The output is therefore best understood as a land use classification: each patch records whether urban content is dominant, not where buildings begin and end. This places Method 1 closer to statistical-level urban analysis than to precise geospatial object extraction. For studies requiring building-level accuracy, this resolution is insufficient.

6.2.2. Method 2: Object-level classification

Synthetic data strategy. The CycleGAN-based synthetic data strategy successfully eliminates the need for manual annotation of individual building footprints. This is possible because the BGT provides geometrically accurate building polygons that can serve simultaneously as style transfer input and annotation mask. The unpaired translation approach is particularly relevant for historical maps, where paired correspondences between a historical scan and its equivalent ground truth do not exist. This distinguishes the CycleGAN approach from supervised methods that require matched image pairs, such as pix-2-pix (Zhu et al., 2017).

The approach is not limited to the Netherlands. The BGT plays the role of the reference geodata in this thesis, but any authoritative or open building dataset (including OpenStreetMap) could fulfil the same role for other countries or regions. The reference data does not need to spatially overlap with the historical map sheet: any area with a similar urban typology and building geometry can provide the source domain for CycleGAN training. In this thesis, the Rotterdam study area exhibits row terrace houses, L-shaped buildings, and perimeter block urban planning patterns that are representative of late 19th-century Dutch cities.

Challenges: domain gap and geometric mismatch. The most fundamental challenge is the domain gap between the synthetic training data and the real historical map scans. The BGT represents the current built environment, where buildings are geometrically accurate, well-separated, and spatially distinct. The Rotterdam 1897 map depicts a 19th-century city where buildings in the downtown area are extremely narrow, densely packed, and share walls with no visible gap between them. These configurations are absent from the modern BGT and therefore absent from the synthetic training set. The model was never trained on these shapes, which directly causes missed detections in the densest parts of the study area.

Beyond training data gaps, a geometric mismatch exists at the prediction level. Even where buildings are detected, the predicted polygon boundaries do not align precisely with the painted ink strokes on the historical map. Buildings on historical maps are rendered by hand: a dark outline stroke defines the boundary, and a solid color fill occupies the interior. This is a visually direct representation, but it is not structured as a bounding-box-defined object. The Mask R-CNN pipeline defines candidate objects via bounding boxes first through the RPN, which is not well-suited for densely packed, touching, or hand-drawn shapes where object boundaries are ambiguous at the pixel level. This raises the question of whether edge-detection-based deep learning approaches, such as the deep edge filtering method proposed by Chen et al. (2021), may be a more natural fit for historical map building extraction, as they directly target ink boundaries rather than first proposing bounding boxes.

The consequence is that the post-processing pipeline becomes complex and requires extensive parameter tuning to handle the resulting artifacts: text false positives, jagged pixel-traced boundaries, duplicate predictions from tile overlaps, and fragmented predictions at tile edges. The tile edge effect – where large buildings spanning two adjacent tiles are split into two partial predictions – is a known limitation of sliding-window Mask R-CNN instance segmentation and is not fully resolvable by increasing tile overlap alone (Kirillov et al., 2020).

Challenges: text overlays. Historical map sheets contain dense text overlays: street names, toponym labels, and point-of-interest markers for public buildings. In this thesis, street name text was approximated in the synthetic data. However, the dense point-of-interest labels that typically overlie the footprints of public and civic buildings were not replicated in the synthetic training data. The model therefore fails to detect hatched buildings in areas where these labels

cover the building fill symbol, as confirmed by visual inspection. This is not a fundamental limitation of the CycleGAN approach, but a gap in the current synthetic data generation step that could be addressed with more complete text overlay simulation.

Limitations. The georeferencing accuracy of the historical map sheet is a hard constraint on the entire Method 2 pipeline. The original IIF annotation was insufficient for building-level spatial matching, requiring improvement of the georeference with 100 GCPs and a TPS transformation (Section 5.2.1). For map sheets where few identifiable building features are available for GCP placement (such as open land-dominated or heavily destroyed areas) this step becomes infeasible, as demonstrated by the exclusion of sheets 2 and 4 in this thesis. Application of this pipeline to significantly older maps with lower planimetric accuracy, or to maps from regions without reliable reference geodata, would compound both the georeferencing difficulty and the domain gap, yielding substantially less reliable results.

Relation to existing work. The synthetic data bootstrapping strategy directly extends the approach of Arzoumanidis et al. (2025), who demonstrated CycleGAN-based synthetic data generation for historical map segmentation. This thesis applies that strategy to a Dutch municipality map series and couples it with instance segmentation rather than semantic segmentation, producing vector polygon outputs rather than pixel-level class maps. Compared to the building extraction approach of Heitzler & Hurni (2020), which requires manually annotated training data from the target map series, the synthetic data strategy offers a practical path toward annotation-free building extraction. The post-processing complexity observed in this thesis is consistent with the broader finding in the literature that deep learning pipelines for historical maps require substantial post-processing to convert pixel-level outputs into usable vector geometries (Xia et al., 2024).

6.2.3. On urban change analysis

The extraction of urban extent from historical maps enables quantitative analysis of how cities transformed over time. The results of both methods demonstrate complementary forms of this analysis: Method 1 produces a statistical-level change map at provincial scale, while Method 2 enables building-level urban persistence analysis at city scale.

Method 1: growth signal and its caveats. The post-classification change map produces a directionally plausible urban growth signal broadly consistent with the known industrialisation-era expansion of Dutch cities (Rutte, 2023); the specific figures are reported in Section 6.1. The Uncertainty / Demolition class requires reattribution as undetected persistent urban rather than real demolition. This is consistent with the concept of urban inertia described by Rutte (2023, p. 14) : once urban spatial structures are established, they tend to persist over centuries, making large-scale demolition between the 1870s and 1940s implausible as a real-world signal at provincial scale. Beyond this reattribution, PCC carries a structural accuracy limitation: the change map accuracy is approximately the product of the two individual classification accuracies (Singh, 1989), meaning T2 classification errors compound directly into the change signal.

The patch grid misalignment documented in Section 5.4.1 further inflates all change class areas by introducing narrow boundary artifacts along patch edges. The underlying cause – tilted scans producing variable patch grid origins between editions – is structural and cannot be resolved by post-processing the existing raster output. A practical mitigation is to adopt a vector-based one-to-one patch matching approach: dissolving predicted patches into polygon objects and matching T1 to T2 by IoU or centroid proximity. This would be insensitive to sub-patch positional offsets and is more appropriate for studies where boundary precision matters.

At provincial scale, the offset is acceptable given the 200-meter patch size, but it should be addressed before any finer-grained temporal analysis is attempted.

Method 2: urban persistence analysis. The numbers and primary results are reported in Section 6.1. The low precision (6.9%) is not a meaningful model performance indicator in this context: most unmatched predictions are correctly detected 1897 buildings that no longer exist in the current BGT due to the 1940 bombing. The persistence rate therefore measures the survival fraction of the pre-1897 built environment, which is a directly meaningful urban history indicator rather than a model accuracy metric.

However, even this survival fraction is an underestimate, not only due to model detection failures but also due to a fundamental cartographic representation gap between the historical map and the current BGT. In the Rotterdam 1897 maps, buildings in traditional perimeter block urban layouts are drawn as solid red footprints facing the road, with the backyard depicted as background color behind the building. Two houses facing opposite roads share a backyard boundary at the centre of the block (Figure 5.35). In the current BGT, the building polygon covers the full parcel extent, including the backyard area. As a result, the centroid of a BGT building is positioned further inward from the road than the centroid of the corresponding historical map prediction, which traces only the front facade footprint. This systematic offset frequently exceeds the 8 m centroid matching threshold, causing correct predictions to be classified as unmatched and artificially deflating recall. This is not a failure of the deep learning model: it is a consequence of fundamentally different cartographic conventions between a hand-drawn 19th-century municipality map and a modern cadastral dataset. This challenge would persist even if building polygons were manually and perfectly digitized from the historical map, because the spatial definition of a building footprint differs between the two data sources.

This observation points to a broader methodological challenge for historical map urban change analysis beyond the deep learning aspect: extracting meaningful building-level change conclusions requires not only accurate feature extraction but also a clear understanding of how cartographic conventions map onto modern spatial definitions. Method 2's vector output enables building-level spatial queries, but the interpretation of those queries depends on resolving the representational gap between historical and modern geodata.

6.3. Contribution

This thesis produces the following contributions:

- **Custom Bonnebladen download pipeline.** A Python script that joins IIF image identifiers to human-readable sheet metadata, supports edition-aware filtering, and exports georeferenced GeoTIFFs with a catalog CSV. This step also demonstrated that IIF and Allmaps as infrastructure support geospatial deep learning pipelines.
- **Temporal dataset assessment of the Bonnebladen map series.** A documented characterization of the temporal structure, edition counts, and cartographic style changes across 2,940 sheet editions, providing a reusable reference for future computational work on this collection.
- **Trained weak classifier for the Bonnebladen map series.** A fine-tuned ResNet-101 patch classifier (macro F1 = 0.900) trained on the Bonnebladen series, saved as a reusable model checkpoint, covering the Province of South Holland, accessible on Github (Appendix B).

- **Patch-level urban extent extraction at provincial scale.** Application of the MapReader pipeline to 74 South Holland sheets, producing georeferenced and classified patch datasets.
- **A trained Mask R-CNN instance segmentation model,** producing 12,299 regularized building polygon predictions from sheet 3 of the Rotterdam 1897 series, evaluated against BGT reference data with validation mAP = 0.874 and recall of 41.4%.

6.4. Future Work

The following are identified for future research based on the findings and limitations of this thesis.

- **Fine-tuned classifier for Method 1.** Complete the second training stage of the MapReader pipeline by reviewing low-confidence predictions from the full T1 and T2 inference output, extending the training set with both pre- and post-1930s examples, and retraining. A fine-tuned classifier is expected to reduce style-induced errors on T2 editions and improve the reliability of the change map.
- **Nationwide Bonnebladen classification.** With a fine-tuned classifier that generalizes across all editions, extend the pipeline to the full 2940-edition Bonnebladen collection, covering the entire Netherlands. Results could be published as open data to support nationwide historical land use research.
- **Patch size experiments.** Investigate smaller patch sizes (25 m, 50 m, 100 m) to assess whether finer spatial granularity improves urban extent precision at 1:25.000 map scale without prohibitive annotation cost. Alternatively, test inferencing with smaller patches on a model trained at 200 m, to explore finer output detail without retraining.
- **Text removal pre-processing for Method 2.** Implement a text detection and removal step before Mask R-CNN inference to reduce false positives from map text characters and improve detection of hatched public buildings whose footprints are covered by point-of-interest labels.
- **Alternative object-level extraction method.** Investigate alternative deep learning approaches for building footprint extraction from historical maps. Deep edge filtering, as proposed by Chen et al. (2021), directly targets the ink boundary strokes that define building outlines and may be more naturally suited to the hand-drawn visual structure of historical maps than bounding-box-first instance segmentation. Separately, semantic segmentation produces pixel-level class maps that identify building regions. Combining these two methods could yield both accurate geometric shapes and correct class labels, addressing limitations observed in the Mask R-CNN pipeline.
- **Improved synthetic training data for dense urban areas.** Address the domain gap caused by narrow, touching building footprints in dense 19th-century city centres, which are absent from the current BGT. One approach is to apply semi-automated geometric intervention to the BGT data, such as splitting buildings polygons. Another direction is to extend the training dataset by combining translated outputs from multiple CycleGAN checkpoints simultaneously. For instance, training on patches from both epoch 6 and epoch 14, rather than a single checkpoint, would expose the model to a wider range of cartographic degradation effects and visual styles. Earlier checkpoints contribute geometric fidelity; later checkpoints contribute richer texture.

6.4. *Future Work*

A Declaration of AI/LLM usage

Some parts of this thesis benefited from the use of Claude AI (web version) throughout the research and writing process. The tool was used in the following ways:

- Programming and troubleshooting: AI was used regularly to support the implementation of Python scripts, debugging, and explaining error messages.
- Learning and literature orientation: AI was used to help explain theoretical concepts, brainstorming, and support the initial orientation through related literature.
- Writing refinement: AI was used to refine grammar, fix typos, and improve sentence clarity throughout the thesis. It was also used to assist with Typst formatting, including tables, figures, and template elements.
- Translation: AI was used to translate Dutch-language documentation into English, including the HGN land use report, selected BGT technical documentation, and several documentation relevant to Dutch historical map collections to support accurate citation and interpretation of those sources.

All research results, analyses, interpretations, and conclusions presented in this thesis are my own. I verified that no AI-generated content was used to substitute the original analysis, and that no plagiarised content introduced through the use of the tools.

B Reproducibility self-assessment

B.1 Data availability

The Bonnebladen dataset used in Method 1 is accessible via the DANS Data Station repository²² (Kadaster et al., 1940). Access requires registration and a data access request through the DANS website. The Rotterdam 1897 municipality maps used in Method 2 are openly accessible via the Rotterdam City Archives.²³ The BGT dataset is publicly available via PDOK. The HGN-1900 dataset is publicly available via the Wageningen University repository (Knol et al., 2004).

B.2 Code availability

All analysis code is openly available on GitHub across four repositories:

- Bonnebladen download pipeline: https://github.com/ahhutama/download_bonnebladen
- Method 1 – MapReader pipeline: https://github.com/ahhutama/mapreader_bonnebladen
- Method 2 – CycleGAN pipeline: https://github.com/ahhutama/cyclegan_historical_maps
- Method 2 – Mask R-CNN pipeline: https://github.com/ahhutama/mask_rcnn_buildings

Each repository includes a README with instructions for environment setup and pipeline execution. Some intermediate steps, particularly data preparation and parameter tuning, may require additional clarification beyond what is documented.

B.3 Reproducibility rating

Moderate. All source code and publicly accessible datasets are available. The Bonnebladen dataset requires a DANS access request. Basic documentation is provided for all pipelines, though some steps may require clarification to reproduce exactly.

²²<https://phys-techsciences.datastations.nl/dataset.xhtml?persistentId=doi:10.17026/dans-xqf-r4xf>

²³<https://stadsarchief.rotterdam.nl/zoek-alles?mivast=184&mizig=299&miadt=184&miview=ldt&milang=nl&micode=4201&minr=39521873&miaet=14>

C Georeference Rotterdam 1897 historical maps



Figure C.1: One hundred ground control points and fifteen independent check points distributed across Rotterdam 1897 map – sheet 3.

D Bonnebladen assessment

Map Legend

Administrative Boundary

Province

Sheet Indices

BLAAD_NR

NULL

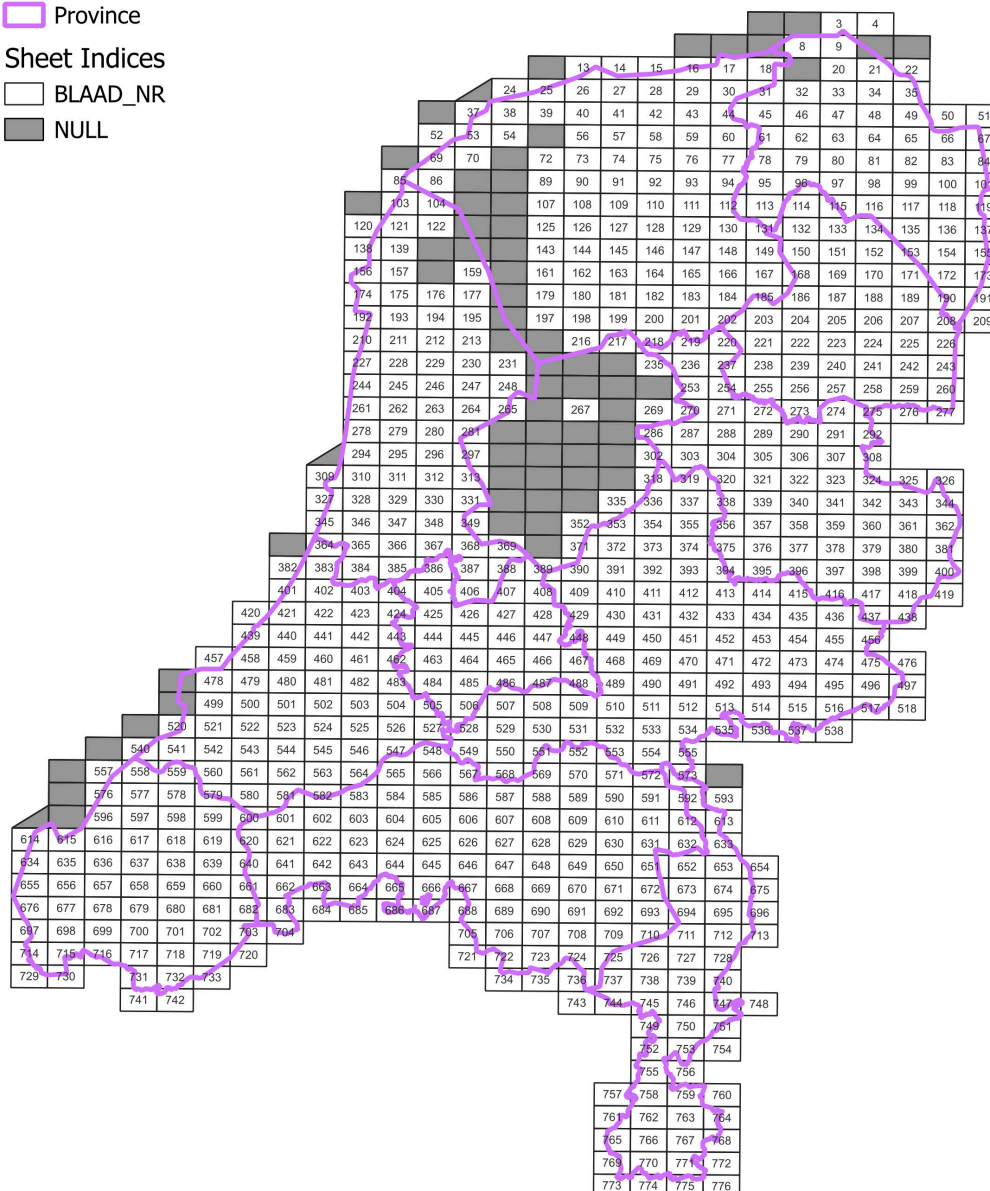


Figure D.1: *Bonnebladen bladwijzer*, or sheet indices of bonnebladen map series

Map Legend

Administrative Boundary

Province

Sheet Indices

Published Date (year)

1904 - 1915

1915 - 1927

1927 - 1938

1938 - 1949

NULL

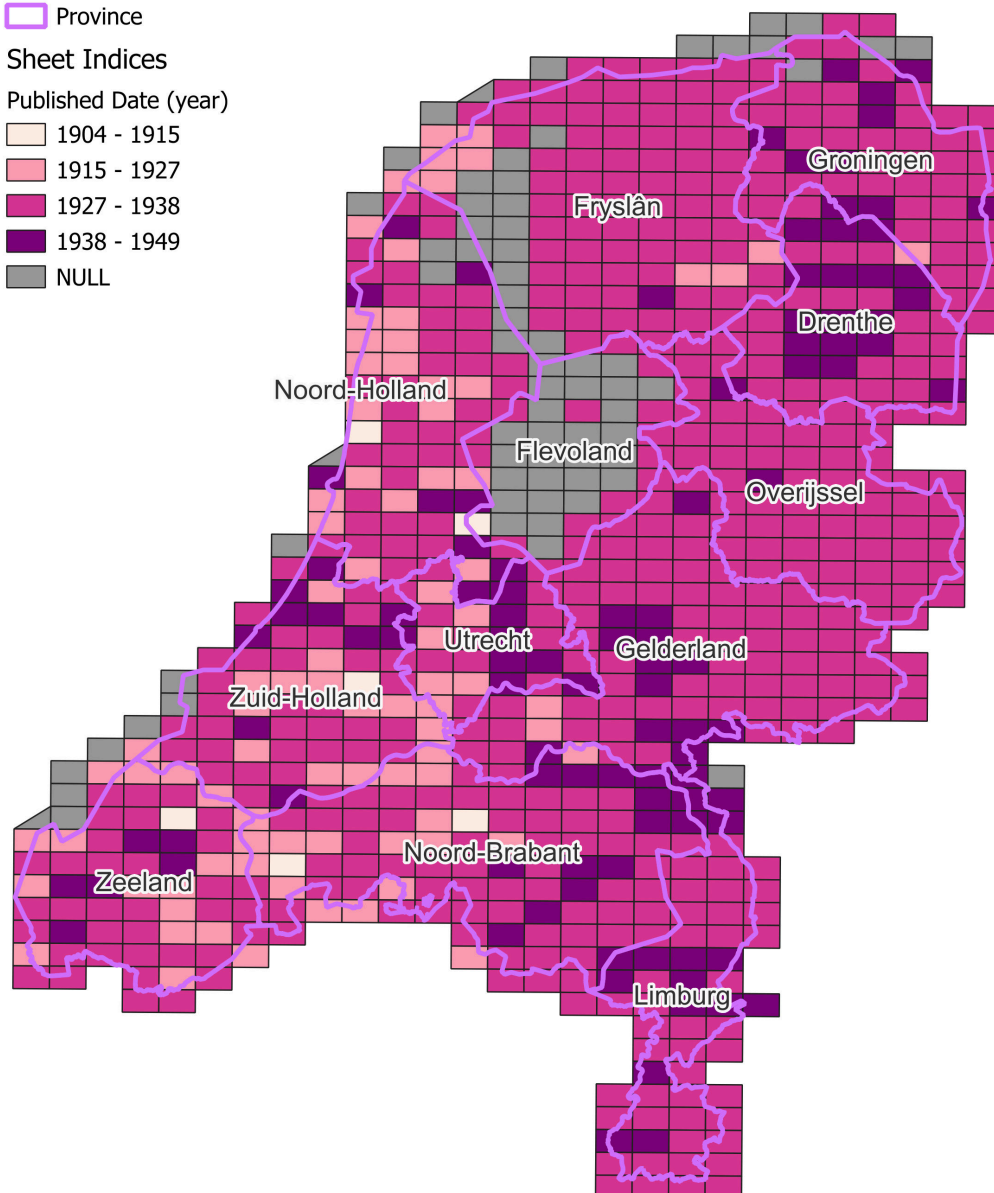


Figure D.3: Published date (*Uitgave*) of the latest available edition per sheet.

Map Legend

Administrative Boundary

Province

Sheet Indices

NULL

Edition timespan (years)

0.0 - 0.0

1 - 8

8 - 29

29 - 43

43 - 57

57 - 82

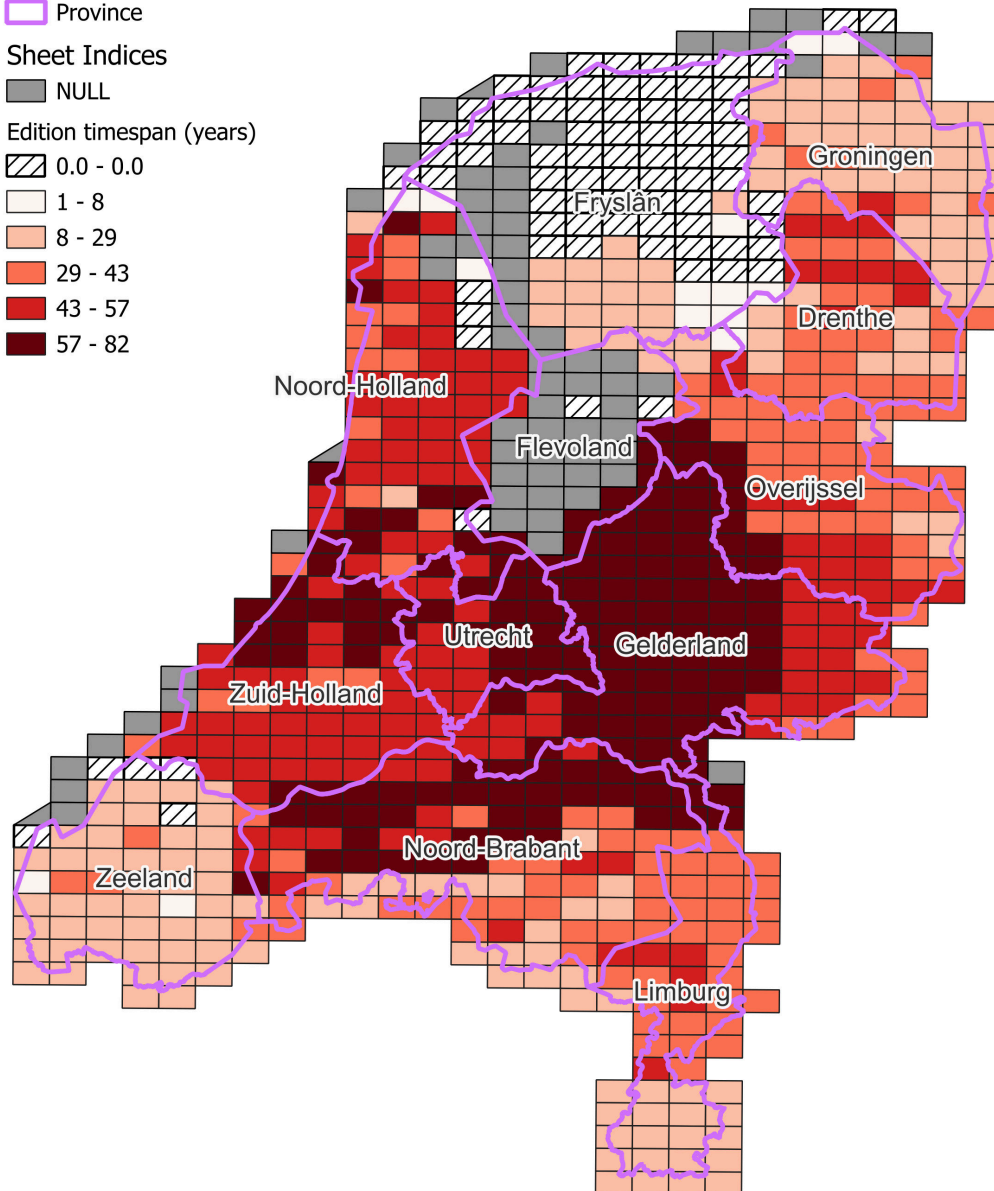


Figure D.4: Timespan in years between the first available edition and the last edition per sheet.

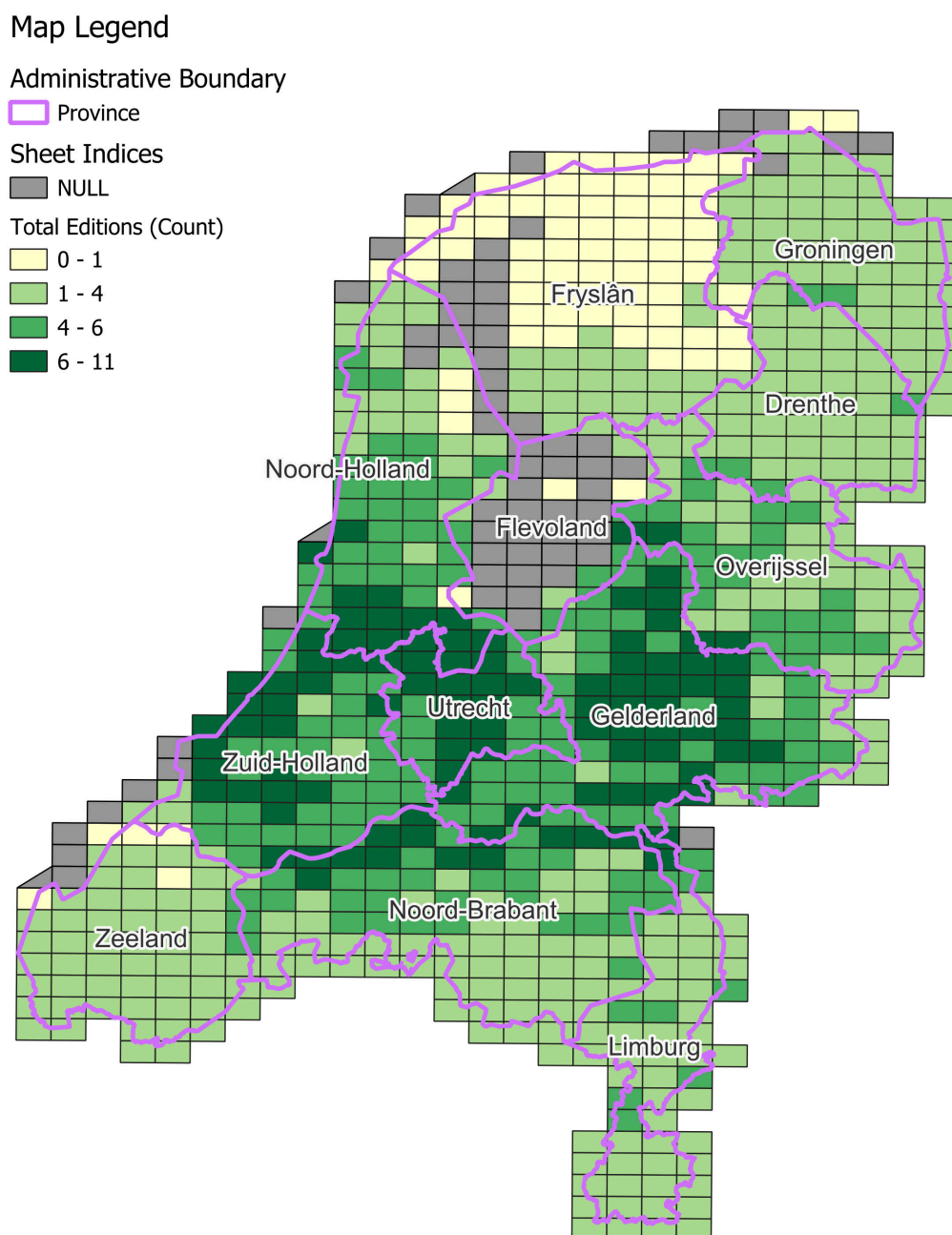


Figure D.5: Total available editions per sheet.

Bibliography

- Arzoumanidis, L., Knechtel, J., Haunert, J.-H., & Dehbi, Y. (2025). *Automatic Uncertainty-Aware Synthetic Data Bootstrapping for Historical Map Segmentation*. arXiv. <https://doi.org/10.48550/ARXIV.2511.15875>
- Budig, B., Dijk, T. C. van, Feitsch, F., & Arteaga, M. G. (2016). Polygon consensus: smart crowdsourcing for extracting building footprints from historical maps. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Sigspatial'16*, 1–4. <https://doi.org/10.1145/2996913.2996951>
- Chen, G., Hay, G. J., Carvalho, L. M. T., & Wulder, M. A. (2012). Object-based change detection. *International Journal of Remote Sensing*, 33(14), 4434–4457. <https://doi.org/10.1080/01431161.2011.648285>
- Chen, Y., Carlinet, E., Chazalon, J., Mallet, C., Duménieu, B., & Perret, J. (2021). Vectorization of Historical Maps Using Deep Edge Filtering and Closed Shape Extraction. In J. Lladós, D. Lopresti, & S. Uchida (Eds.), *Document Analysis and Recognition – ICDAR 2021: Document Analysis and Recognition – ICDAR 2021*. https://link.springer.com/chapter/10.1007/978-3-030-86337-1_34
- Chiang, Y.-Y., Leyk, S., & Knoblock, C. A. (2014). A Survey of Digital Map Processing Techniques. *ACM Computing Surveys*, 47(1), 1–44. <https://doi.org/10.1145/2557423>
- Dodda, S. (2024). Automated Text Recognition and Segmentation for Historic Map Vectorization: A Mask R-CNN and UNet Approach. *Journal of Electrical Systems*, 20(7s), 635–649. <https://doi.org/10.52783/jes.3413>
- DPIRD-DMA. (2024). *Building-Regulariser*. <https://github.com/DPIRD-DMA/Building-Regulariser> GitHub repository, accessed 2026-06-05
- Gao, L., Gao, H., Wang, Y., Liu, D., & Momanyi, B. M. (2023). Center-Ness and Repulsion: Constraints to Improve Remote Sensing Object Detection via RepPoints. *Remote Sensing*, 15(6), 1479. <https://doi.org/10.3390/rs15061479>
- Gobbi, S., Ciolli, M., La Porta, N., Rocchini, D., Tattoni, C., & Zatelli, P. (2019). New Tools for the Classification and Filtering of Historical Maps. *ISPRS International Journal of Geo-Information*, 8(10), 455. <https://doi.org/10.3390/ijgi8100455>
- Goodchild, M. F. (2018). Reimagining the history of GIS. *Annals of GIS*, 24(1), 1–8. <https://doi.org/10.1080/19475683.2018.1424737>
- Hanlon, A., Spangler, I., Appel, S., Schoonman, J., & Spaan, B. (2025). The Social Life of Allmaps: Reflecting on Software Communities in Sustaining the Digital Geohumanities. *Journal of Map & Geography Libraries*, 21(3), 156–163. <https://doi.org/10.1080/15420353.2025.2574896>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). *Mask R-CNN*. <https://arxiv.org/abs/1703.06870>

- Heitzler, M., & Hurni, L. (2020). Cartographic reconstruction of building footprints from historical maps: A study on the Swiss Siegfried map. *Transactions in GIS*, 24(2), 442–461. <https://doi.org/10.1111/tgis.12610>
- Herrault, P. A., Sheeren, D., Fauvel, M., Monteil, C., & Paegelow, M. (2013). A comparative study of geometric transformation models for the historical "map of france" registration. *Geographia Technica*, (1), 34–46. <https://hal.science/hal-01416127>
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2018). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. <https://arxiv.org/abs/1706.08500>
- Hosseini, K., Wilson, D. C. S., Beelen, K., & McDonough, K. (2022). MapReader: a computer vision pipeline for the semantic exploration of maps at scale. *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Geospatial Humanities, SIGSPATIAL '22*, 8–19. <https://doi.org/10.1145/3557919.3565812>
- Humanities Cluster of the Royal Netherlands Academy of Arts and Sciences (KNAW). (2024). *HisGIS: Historical Geographic Information System*. <https://hisgis.nl/>
- Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80, 91–106. <https://doi.org/10.1016/j.isprsjprs.2013.03.006>
- IIIF Maps Community Group. (2023). *Georeference Extension*. <https://iiif.io/api/extension/georef/> An approved extension to the IIIF Presentation API for storing georeferencing metadata
- Kadaster. (2015). *Topotijdreis*. <https://www.topotijdreis.nl/> Online platform
- Kadaster, Topografische Dienst, & Topographisch Bureau. (1940). *Chromo-Topographische Kaart des Rijks (1:25.000). Bonnebladen [geo-tiffs]*. DANS Data Station Physical, Technical Sciences. <https://doi.org/10.17026/DANS-XQF-R4XF>
- Kadaster, Topografische Dienst, & Topographisch Bureau. (1940). *Chromo-Topographische Kaart des Rijks (1:25.000). Bonnebladen [scans]*. DANS Data Station Physical, Technical Sciences. <https://doi.org/10.17026/DANS-XMD-R6P4>
- Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). PointRend: Image Segmentation As Rendering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9796–9805. <https://doi.org/10.1109/cvpr42600.2020.00982>
- Knol, W., Kramer, H., & Gijsbertse, H. (2004). *Historisch grondgebruik Nederland: een landelijke reconstructie van het grondgebruik rond 1900* (Issue 573). Alterra. <https://edepot.wur.nl/38370>
- Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring Strategies for Training Deep Neural Networks. *Journal of Machine Learning Research*, 10(1), 1–40. <http://jmlr.org/papers/v10/larochelle09a.html>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). *Microsoft COCO: Common Objects in Context*. <https://arxiv.org/abs/1405.0312>

- Liu, B., Wu, C., Xu, W., Shen, Y., & Tang, F. (2024). Emerging trends in GIS application on cultural heritage conservation: a review. *Heritage Science*, 12(1). <https://doi.org/10.1186/s40494-024-01265-7>
- Loshchilov, I., & Hutter, F. (2019). *Decoupled Weight Decay Regularization*. <https://arxiv.org/abs/1711.05101>
- Matikainen, L., Hyypä, J., Ahokas, E., Markelin, L., & Kaartinen, H. (2010). Automatic Detection of Buildings and Changes in Buildings for Updating of Maps. *Remote Sensing*, 2(5), 1217–1248. <https://doi.org/10.3390/rs2051217>
- Meijers, M., & Schoonman, J. (2025). Mapping the Edge: A Novel Approach to Georeferencing Historical Map Series. *E-Perimtron*, 20(1), 12–24. http://www.e-perimtron.org/Vol_20_1/Meijers_et_al.pdf
- Nobajas, A., & Nadal, F. (2015). From historical map to online 3D recreation: the 1861 cadastral map of Horta (Barcelona). *Cartography and Geographic Information Science*, 42(3), 211–223. <https://doi.org/10.1080/15230406.2014.998285>
- O'Hara, R., Marwaha, R., Zimmermann, J., Saunders, M., & Green, S. (2024). Unleashing the power of old maps: Extracting symbology from nineteenth century maps using convolutional neural networks to quantify modern land use on historic wetlands. *Ecological Indicators*, 158, 111363. <https://doi.org/10.1016/j.ecolind.2023.111363>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. <https://arxiv.org/abs/1912.01703>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018). *Scikit-learn: Machine Learning in Python*. <https://arxiv.org/abs/1201.0490>
- Petitpierre, R., Kaplan, F., & Lenardo, I. di. (2021). *Generic Semantic Segmentation of Historical Maps*. https://www.researchgate.net/publication/355711778_Generic_Semantic_Segmentation_of_Historical_Maps
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv. <https://doi.org/10.48550/ARXIV.1505.04597>
- Rutte, R. (2023). *The Making of the Netherlands: Landscape, Cities and Architecture* (p. 159). Uitgeverij Thoth. English edition. Supplementary text by Everhard Korthals Altes; cartography by Yvonne van Mil and Pieter van der Weele. Translation of: Historische atlas van Nederland: hoe ons land in 2000 jaar is ingericht (Bussum: Thoth, 2023)
- Singh, A. (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>
- Snydman, S., Sanderson, R., & Cramer, T. (2015). The International Image Interoperability Framework (IIIF): A community & technology approach for web-based images. *Archiving Conference*, 12(1), 16–21. <https://doi.org/10.2352/issn.2168-3204.2015.12.1.art00005>

- Southall, H., Aucott, P., Fleet, C., Pert, T., & Stoner, M. (2017). GB1900: Engaging the Public in Very Large Scale Gazetteer Construction from the Ordnance Survey “County Series” 1:10, 560 Mapping of Great Britain. *Journal of Map & Geography Libraries*, 13(1), 7–28. <https://doi.org/10.1080/15420353.2017.1307305>
- Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., Chavez-Urbiola, E. A., & Romero-Gonzalez, J. A. (2023). *Loss Functions and Metrics in Deep Learning*. arXiv. <https://doi.org/10.48550/ARXIV.2307.02694>
- Uhl, J., Leyk, S., Chiang, Y.-Y., Duan, W., & Knoblock, C. (2017). Extracting Human Settlement Footprint from Historical Topographic Map Series Using Context-Based Machine Learning. *8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, 15 (6 .)–15 (6 .). <https://doi.org/10.1049/cp.2017.0144>
- Vaienti, B., Lenardo, I. di, & Kaplan, F. (2025). Georeferencing historical maps using local feature matching and Delaunay consistency. *Cartography and Geographic Information Science*, 1–23. <https://doi.org/10.1080/15230406.2025.2566789>
- Xia, X., Zhang, D., Song, W., Huang, W., & Hurni, L. (2025). MapSAM: adapting segment anything model for automated feature detection in historical maps. *Giscience & Remote Sensing*, 62(1). <https://doi.org/10.1080/15481603.2025.2494883>
- Xia, X., Zhang, T., Heitzler, M., & Hurni, L. (2024). Vectorizing historical maps with topological consistency: A hybrid approach using transformers and contour-based instance segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 129, 103837. <https://doi.org/10.1016/j.jag.2024.103837>
- Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. arXiv. <https://doi.org/10.48550/ARXIV.1703.10593>

