# Modern vectorization and alignment of historical maps: An application to Paris Atlas (1789-1950)

Yizi Chen

Université Gustave Eiffel

# Modern vectorization and alignment of historical maps: An application to Paris Atlas (1789-1950)

**PhD thesis of the Gustave Eiffel University**

Doctoral school: MSTIC
Doctorate speciality: Geographical Information Sciences and technologies
Research unit: LASTIG

**Thesis presented and defended at the Gustave Eiffel University on March 22, 2023 by:**


## YIZI CHEN


### Jury composition

| | |
|---|---|
| **Nicole** VINCENT<br>Professor, Université Paris Cité, LIPADE | President |
| **Véronique** EGLIN<br>Professor, INSA Lyon, Imagine/LIRIS | Reviewer |
| **Lorenz** HURNI<br>Professor, ETH Zürich, IKG | Reviewer |
| **Mathieu** AUBRY<br>Senior researcher, ENPC, IMAGINE/LIGM | Examiner |
| **Stefan** LEYK<br>Professor, University of Colorado Boulder | Examiner |

### PhD supervision

| | |
|---|---|
| **Julien** PERRET<br>Senior researcher, LASTIG, Univ Gustave Eiffel, IGN-ENSG | Supervisor |
| **Joseph** CHAZALON<br>Lecturer, EPITA, LRE | Co-Supervisor |
| **Clément** MALLET<br>Senior researcher, LASTIG, Univ Gustave Eiffel, IGN-ENSG | Co-Supervisor |

# Contents

**Acknowledgements** v

**Abstract** vii

**Résumé** ix

**1 Introduction** 1
- 1.1 A brief introduction to historical map vectorization . . . . . 1
- 1.2 Related work . . . . . 2
- 1.3 Project SoDUCo . . . . . 3
- 1.4 Corpus of historical atlases . . . . . 3
- 1.5 Existing challenges in historical maps . . . . . 4
- 1.6 Problem statements . . . . . 6
- 1.7 Contributions . . . . . 6
- 1.8 Publications . . . . . 9

**2 Pipeline design for historical map vectorization** 11
- 2.1 Motivation . . . . . 11
- 2.2 Related work . . . . . 12
- 2.3 Vectorization pipelines under test . . . . . 17
- 2.4 Explaining our dataset (Atlas Municipal) . . . . . 18
- 2.5 Evaluation protocol . . . . . 18

**3 Learning edges through deep neural architectures** 23
- 3.1 Multi-scale deep neural network architecture . . . . . 23
- 3.2 Transformer architectures . . . . . 31
- 3.3 Deep watershed transform . . . . . 34
- 3.4 Data augmentations . . . . . 36

**4 Topology-aware loss functions** 43
- 4.1 Introduction to topology-awareness loss functions . . . . . 43
- 4.2 Enhancing pixel connectivity as a loss function . . . . . 54

**5 Improving model robustness of deep edge detectors** 59
- 5.1 Motivation . . . . . 59
- 5.2 Related work . . . . . 60
- 5.3 Method . . . . . 62
- 5.4 Experimental settings for segmenting historical maps . . . . . 64
- 5.5 Experimental results and analysis . . . . . 65

**6 Leveraging redundancies of historical maps** **69**
6.1 Motivation . . . . . 69
6.2 Related work . . . . . 71
6.3 Map image alignment method . . . . . 72
6.4 Experimental settings and results . . . . . 75
6.5 Perspectives . . . . . 80

**7 Conclusions and perspectives** **81**
7.1 Conclusion . . . . . 81
7.2 Perspectives . . . . . 82

**Appendices** **87**

**A Résumé substantiel** **89**
A.1 Introduction . . . . . 89
A.2 Conception d'une chaîne de traitement pour la vectorisation de cartes historiques . . . . . 90
A.3 Apprentissage des contours à travers des architectures de réseaux neuronaux profonds . . . . . 91
A.4 Fonctions de perte sensibles à la topologie . . . . . 92
A.5 Améliorer la robustesse du modèle des détecteurs de contours profonds . . . . . 93
A.6 Exploitation des redondances des cartes historiques . . . . . 93
A.7 Perspectives . . . . . 94

**Bibliography** **95**

# *Acknowledgements*

To begin with, I would like to thanks to the members of the committee, and in particular to the reviewers, for their detailed and insightful feedback.

I would like to thank Julien, my primary thesis supervisor, for offering me this Ph.D. position with trust and believe in my potential. His guidance has functioned in shaping my research and enhancing my understanding of open and reproducible practices that will undoubtedly benefit my future studies. Living in Paris as a foreigner can be quite challenging, particularly for someone like me who can not speak even a full French sentence. However, he has been an immense help to me, assisting me with various annoying administrative tasks that have significantly reduced the pressure on my daily life.

I would like to extend my appreciation to my co-supervisor, Joseph, who is not only a great and professional mentor but also one of the most compassionate individuals. He provided support during my challenging times, and has never hesitated to offer his assistance whenever I required it. Furthermore, Joseph has imparted an immense amount of knowledge to me, particularly in terms of producing high-quality and persuasive research outputs. I feel very lucky to have had the opportunity to work with him, not only as a supervisor but also as a close friend.

I would like to express my gratitude my another co-supervisor, Clément, who has provided me with precise, easy-to-follow, and promising research directions throughout my studies. His expertise in the field has been invaluable in guiding me toward achieving my research goals. Moreover, he has provided me with the professional assistance in producing high-quality research articles and presentations, which have been crucial in showcasing my research work. I am truly grateful for his support and contributions to my academic and professional growth.

I consider myself fortunate to have had the opportunity to work closely with my colleagues Minh, Zhou, and Baptiste during my Ph.D. studies. Together, we have accomplished significant milestones in enhancing the vectorization quality of historical maps, which would not have been possible without their contribution. Their passion for research has not only been a source of inspiration for me but has also influenced my future research aspirations positively. I am grateful for their support and look forward to collaborating with them on future projects.

I would like to express my appreciation to my friends, Yandong, Wenhao, Xingming, Sidi, and Zhou, for adding colors and texture to my life in Paris with their companionship.

Lastly, I would like to thank my family for their infinity support but without expecting anything in return.

# *Abstract*

Maps have been a unique source of knowledge for centuries. Such historical documents provide invaluable information for analyzing complex spatial transformations over important time frames. This is particularly true for urban areas that encompass multiple interleaved research domains: humanities, social sciences, etc. The large amount and significant diversity of map sources call for automatic image processing techniques in order to extract the relevant objects as vector features. The complexity of maps (text, noise, digitization artifacts, etc.) has hindered the capacity of proposing a versatile and efficient raster-to-vector approaches for decades.

In this thesis, we propose a learnable, reproducible, and reusable solution for the automatic transformation of raster maps into vector objects (building blocks, streets, rivers), focusing on the extraction of closed shapes. Our approach is built upon the complementary strengths of convolutional neural networks which excel at filtering edges while preserving poor topological properties for their outputs, and mathematical morphology, which offers solid guarantees regarding closed shape extraction while being very sensitive to noise.

In order to improve the robustness of deep edge filters to noise, we review several, and propose new topology-preserving loss functions which enable to improve the topological properties of the results. We also introduce a new contrast convolution (CConv) layer to investigate how architectural changes can impact such properties. Finally, we investigate the different approaches which can be used to implement each stage, and how to combine them in the most efficient way.

Thanks to a shape extraction pipeline, we propose a new alignment procedure for historical map images, and start to leverage the redundancies contained in map sheets with similar contents to propagate annotations, improve vectorization quality, and eventually detect evolution patterns for later analysis or to automatically assess vectorization quality.

To evaluate the performance of all methods mentioned above, we released a new dataset of annotated historical map images. It is the first public and open dataset targeting the task of historical map vectorization. We hope that thanks to our publications, public and open releases of datasets, codes and results, our work will benefit a wide range of historical map-related applications.

# *Résumé*

Les cartes sont une source unique de connaissances depuis des siècles. Ces documents historiques fournissent des informations inestimables pour analyser des transformations spatiales complexes sur des périodes importantes. Cela est particulièrement vrai pour les zones urbaines qui englobent de multiples domaines de recherche imbriqués : humanités, sciences sociales, etc. La complexité des cartes (texte, bruit, artefacts de numérisation, etc.) a entravé la capacité à proposer des approches de vectorisation polyvalentes et efficaces pendant des décennies.

Dans cette thèse, nous proposons une solution apprenable, reproductible et réutilisable pour la transformation automatique de cartes raster en objets vectoriels (îlots, rues, rivières), en nous focalisant sur le problème d'extraction de formes closes. Notre approche s'appuie sur la complémentarité des réseaux de neurones convolutifs qui excellent dans et de la morphologie mathématique, qui présente de solides garanties au regard de l'extraction de formes closes tout en étant très sensible au bruit.

Afin d'améliorer la robustesse au bruit des filtres convolutifs, nous comparons plusieurs fonctions de coût visant spécifiquement à préserver les propriétés topologiques des résultats, et en proposons de nouvelles. À cette fin, nous introduisons également un nouveau type de couche convolutive (CConv) exploitant le contraste des images, pour explorer les possibilités de telles améliorations à l'aide de transformations architecturales des réseaux. Finalement, nous comparons les différentes approches et architectures qui peuvent être utilisées pour implémenter chaque étape de notre chaîne de traitements, et comment combiner ces dernières de la meilleure façon possible.

Grâce à une chaîne de traitement fonctionnelle, nous proposons une nouvelle procédure d'alignement d'images de plans historiques, et commençons à tirer profit de la redondance des données extraites dans des images similaires pour propager des annotations, améliorer la qualité de la vectorisation, et éventuellement détecter des cas d'évolution en vue d'analyse thématique, ou encore l'estimation automatique de la qualité de la vectorisation.

Afin d'évaluer la performance des méthodes mentionnées précédemment, nous avons publié un nouveau jeu de données composé d'images de plans historiques annotées. C'est le premier jeu de données en libre accès dédié à la vectorisation de plans historiques. Nous espérons qu'au travers de nos publications, et de la diffusion ouverte et publique de nos résultats, sources et jeux de données, cette recherche pourra être utile à un large éventail d'applications liées aux cartes historiques.

# List of abbreviations

| | |
|---|---|
| **BDCN** | Bi-directional cascade network |
| **BCE** | Binary Cross Entropy |
| **BALoss** | Boundary-awareness Loss |
| **CSE** | Closed Shape Extraction |
| **ClDice** | Center Line Dice |
| **ConnNet** | Connectivity network |
| **CNN** | Convolutional Neural Network |
| **CConv** | Contrast Convolution |
| **CA** | Coarse Alignment |
| **FA** | Fine Alignment |
| **DEF** | Deep Edge Filtering |
| **DWS** | Deep Watershed |
| **ET** | Edge Thinning |
| **EPM** | Edge Probability Map |
| **HED** | Holistically Edge Detector |
| **MM** | Mathematical Morphology |
| **MWS** | Meyer Watershed Segmentation |
| **MBD** | Minimum Barrier Distance |
| **mPb** | Multiscale Probability Boundary |
| **MRF/CRF** | Markov / Conditional Random Field |
| **Pb** | Probability Boundary |
| **PQ** | **P**anoptic **Q**uality |
| **RQ** | **R**ecognition **Q**uality |
| **SQ** | **S**egmentation **Q**uality |
| **wl** | Watershed Lines |
| **ViT** | Vision Image Transformer |
| **PVT** | Pyramid Vision Transformer |
| **PCL** | Pixel Connectivity Loss |
| **sm** | Saliency Maps |
| **SOTA** | State Of The Art |
| **Topoloss** | Topology loss |
| **RANSAC** | RANdom SAmple Consensus |
| **SoDUCo** | Social Dynamics in Urban Context |

# Chapter 1

# Introduction

Historical maps contain rich information on quantitative study of urban morphogenesis which is the key to understanding the dynamics of cities.
In the introduction of this chapter (Section 1.1), we illustrate why vectorizing historical maps is important for understanding urban morphogenesis. The existing literature for vectorization of historical maps is explained in Section 1.2. The project SoDUCo and corpus of existing map documents are detailed in Section 1.3 and Section 1.4. The challenges in map resources are then inventoried in Section 1.5. We develop several research questions in order to improve the vectorization quality of historical maps in Section 1.6. Lastly, Section 1.7 and Section 1.8 list the contributions and publications over the course of my doctoral researches.
Parts of this chapter are extended and adapted from the contents of my publications [1–3].

## 1.1 A brief introduction to historical map vectorization

Historical maps are unique and powerful tools for understanding the transformations of the geographical space over significant time spans. They are invaluable inputs in historical and social sciences, architecture, and urban planning. The massive digitization of archival collection resources carried out by heritage institutions dramatically increases the amount of geospatial information available for certain areas of the world. In the western world, the rapid development of geodesy and cartography from the $18^{th}$ century resulted in massive production of topographic maps at various scales. City maps are of utter interest. They contain rich, detailed, and often geometrically accurate representations of numerous geographical entities. Maps also document the distribution in space and the topological relationship of the depicted entities, while legends and text labels provide semantic information, in particular about their functions [4, 5]. Recovering spatial and semantic information represented in old maps requires a so-called *vectorization* process.

Vectorizing maps consists in transforming rasterized graphical representations of geographic entities (often maps) into instanced geographic data (or vector data), that can be subsequently manipulated (using Geographic Information Systems, GIS). This is a key challenge today to better preserve, analyze and disseminate content for numerous spatial and spatio-temporal analysis purposes.

From an image processing and a document analysis perspective, vectorization can be illustrated in the following, often interleaved, problems:

1. Isolate the map content subregion on pictures of map sheets (leave out the legend, in particular);
2. Detect and separate the various layers of graphical content: points, lines, and shape objects, as well as symbols and text;
3. Classify / recognize each graphical object of interest (including text), while ensuring a topologically and geometrically consistent result (often considered as an *instance segmentation* problem [6]);
4. Georeference the geometries previously extracted; defined by Wade et al. [7] as **"Aligning geographic data to a known coordinate system so it can be viewed, queried and analyzed with other geographic data"**.

Currently, shape detection is usually performed manually, using GIS software. Such a costly and tedious process leads to heterogeneous data quality. The latest methodological developments in image processing enable to automatically build a significant number of geo-historical databases, and eventually benefit to multiple research areas. In this thesis, we focus on closed shape detection in produced over the course of the $19^{th}$ and early $20^{th}$-century historical map atlases of Paris (France).

## 1.2 Related work

The digitization of historical maps can be separated into three main categories: manual, automatic, and hybrid methods. As noted by Ostafin et al. [8], the manual approach is still a popular solution in digitizing maps when the dataset is small in coverage and time period. For larger datasets, collaborative approaches are used with possibly many contributors — so-called crowdsourcing experiments, as in the works of Budig et al. [9] and Southall et al. [10]— to speed up the digitization process. However, manual processes are still limited in time and quality and highly fluctuating through different contributors, which leads to non-reproducible results. To tackle such problem requires research on automatic and semi-automatic vectorization techniques. In this thesis, we particularly interested in automatic digitization techniques for historical maps.

An early attempt at automatic digitization from historical maps focuses on color. The color-based approach is widely used as a preprocessing stage to separate different object layers through specific thresholds [11–14]. This approach has been adopted in segmenting $19^{th}$ and $20^{th}$ colorized maps [15–18] which are relatively "modern" historical maps. Dating back to the $18^{th}$ century, historical maps rarely had affordable printing colors until the middle of $19^{th}$. Color-based approaches for map processing tasks tend to fail with the maps in gray or limited colors. Furthermore, segmentation approaches based on texture information have been proposed to tackle the issue of the limited colors in historical maps. These approaches either focus on the texture energy [11, 19–21] or hatched areas [22–24]. These texture-based techniques work well for the maps in regular textural patterns with manually tuned parameters, but those methods are designed for specific map applications. Therefore it is difficult to generalize between different datasets. Since the historical maps are mainly constructed using geometrical

shapes, morphological-based approaches are highly suitable for extracting geometries from historical maps with topological properties such as linear features [19], edges [1, 2] and closed shapes[1, 2, 25–29]. However, morphological-based methods are sensitive to image noises and overlaps between different map layers, therefore, some works dedicated to removing unrelated map overlays from the maps [30, 19, 31, 32], but these methods require prior knowledge such as size or shapes of the objects. Methods mentioned above exhibit two main drawbacks:

1. Prior knowledge of color and object shapes narrows down the versatility power of the methods.
2. No focus is made on extracting multiple closed shapes under a learning paradigm.

With the development of the theory of neural networks, many deep-based methods have been used to better extract and separate layers from historical maps. Liu et al. [33] combine the fully convolutional network (FCN) with the integer programming for maintaining topologically and geometrically corrected vector of the floor plan; this network (FCN) also achieves great success in semantic understanding documents [34] and maps [34–38]. However, extracting objects by using semantic segmentation [39] might not be always sufficient particularly when the historical maps have limited colors and textures. Rather than using information on colors and textures for object extraction from historical maps, Oliveira et al. [34] propose the combinations of the detection of boundaries of objects with region growing algorithms watershed for better separating and removing map layers while maintaining closed shapes. Nonetheless, this pipeline has not been comprehensively studied yet in the regime of modern computer vision and a more general and learnable pipeline for the automatic digitization of high-scale historical maps is still in high demand. In this thesis, we propose a supervised framework that appears to be the best solution for correctly fostering information extraction from existing samples in an automatic way by combining deep learning with mathematical morphology for closed-shape extraction.

## 1.3 Project SoDUCo

The project **So**cial **D**ynamics in **U**rban **Co**ntext (**SoDUCo**) aims at developing approaches, models, and usable tools to study the evolution of urban spatial structure concerning the social and professional practices of the population. To reach this goal, we conduct our study based on the reconstitution of the evolution of Paris from 1789 to 1950 with two specific sets of sources, 16 master maps with cadastral maps of Paris and its suburbs, as well as trade directories. We then need to extract useful information from both resources; maps and directories. In this thesis, we focus on developing reliable models and tools for extracting closed shapes from historical maps.

## 1.4 Corpus of historical atlases

The initial corpus of historical maps identified by the researchers of the SoDUCo project [40] (see Figure 1.1) contains a number of maps relevant to

| Short title | Spatial footprint | Temporel footprint | Volume (# of sheets) | Scanned | Georeferenced | State of vectorisation | Streets | City blocks | Parcels | Street numbers |
|---|---|---|---|---|---|---|---|---|---|---|
| Atlas de Verniquet | City boundary & adjacent suburbs bef. 1860 | 1789-1799 | 72 | ● | ● | ■■■■ | ● | ● | ○ | ○ |
| Minutes de l'atlas de Verniquet | | 1783-1789 | ~ 1000 | ● | ○ | □□□□ | ○ | ○ | ● | ○ |
| Plan de Maire | | 1808 | 20 | ● | ◐ | □□□□ | ● | ● | ○ | ○ |
| Atlas de Jacoubet | | 1827-1836 | 54 | ● | ● | ■■■■ | ● | ● | ◐ | ● |
| Carte de Paris et de ses fortifications | | 1845-1849 | 1 | ● | ● | ■■■■ | ● | ○ | ○ | ○ |
| Cadastre de Vasserot [1] | City boundary bef. 1860 | 1810-1836 | 910 | ● | ◐ | ■■■■ | ● | ● | ● | ● |
| Vasserot&Bellanger cadastre | Paris right bank bef. 1860 | 1830-1850 | 155 | ● | ○ | □□□□ | ● | ● | ● | ● |
| Municipal atlas | Paris boundary after 1860 | 1868 | 16 | ● | ● | ■■■□ | ● | ● | ○ | ◐ |
| Mun. atlas & roadworks | | 1871 | 1 | ● | ● | ■■□□ | ● | ● | ○ | ◐ |
| Reconstructed city cadastre | | 1880-1910 | 1 | ● | ◐ | ■□□□ | ○ | ● | ● | ● |
| Lanée map | | 1879 | 1 | ● | ○ | □□□□ | ● | ○ | ○ | ○ |
| Municipal atlas updated | | 1888 | 16 | ● | ● | ■■■■ | ● | ● | ○ | ◐ |
| Hachette map | | 1900 | 1 | ● | ○ | □□□□ | ● | ○ | ○ | ○ |
| Municipal atlas updated | | 1905 | 16 | ○ | ○ | □□□□ | ● | ● | ○ | ◐ |
| Taride map | | 1907 | 1 | ● | ○ | □□□□ | ● | ○ | ○ | ○ |
| Municipal atlas updated | | 1920 | 16 | ● | ○ | □□□□ | ● | ● | ○ | ◐ |
| Guilmin map | | 1926 | 1 | ● | ○ | □□□□ | ● | ○ | ○ | ○ |
| Municipal atlas updated | | 1929 | 16 | ○ | ○ | □□□□ | ● | ● | ○ | ◐ |
| Dufrénoy map | | 1937 | 1 | ○ | ○ | □□□□ | ● | ○ | ○ | ◐ |
| Municipal atlas of Paris | | 1948 | 23 | ○ | ○ | □□□□ | ● | ● | ○ | ◐ |
| City cadastre | | 1999 | n.c | ● | ● | □□□□ | ● | ● | ● | ● |
| Napoleonic cadastre | Adj. suburbs bef. 1860 | 1808-1825 | n.c | ● | ◐ | ■□□□ | ○ | ● | ● | ○ |
| Napoleonic cadastre updated | | 1830-1850 | n.c | ● | ○ | □□□□ | ○ | ● | ● | ○ |

FIGURE 1.1: Corpus of documents listed in Social Dynamics in Urban Context *SoDUCo* Project [40]. Short title represents the name of Atlases.

study the morphogenesis of the city of Paris since the late $18^{th}$ century. Out of all these sources, certain ones have been the object of specific interest for researchers [41–44]. The *Atlas Municipal*, to the best of our knowledge, had received little interest and only one edition (1888) had been exploited for its house numbers in the context of historical geocoding [45]. After a more thorough search, it appeared that the *Bibliothèque de l'hôtel de Ville de Paris* held 24 different versions of the *Atlas Municipal*, ranging from 1866 to 1937. Given that these updated versions were made in order to keep track of the different road works happening at the time, the interest of this map series for the study of the morphogenesis of Paris during this very specific time period became obvious. Furthermore, this maps series presents certain properties that make them especially challenging for map object extraction: thin boundary of objects (building blocks in particular) as well as very little textures and color.

## 1.5 Existing challenges in historical maps

Historical maps provide a very valuable resource for historians and a rich body of scientific challenges for the document analysis and recognition (DAR) community: map-related challenges (Figure 1.3, *left*) and document-related ones (Figure 1.3, *right*).

We detail three map-related challenges in this thesis. Firstly, unlike modern computer-generated maps which follow roughly the same semiotic rules, these maps vary in terms of legend, level of generalization, type of geographic features and text fonts [4]. They also usually lack texture information, which creates ambiguities in the detection of objects. Popular semantic [39, 46, 47] and instance [48–50] image segmentation algorithms detect


FIGURE 1.2: Sample map sheet. *Atlas municipal des vingt arrondissements de Paris*. 1937. Bibliothèque de l'Hôtel de Ville. City of Paris. France. Original size: 11136×7711 pixels.

objects based on textures and are prone to fail in our context. Secondly, color is not a relevant cue either: the palette is usually highly restricted due to the technical limitations and financial constraints of their production. Thirdly, objects in maps are often overlapping, some are thus partially hidden and hardly separable. Occlusion happens with overlaid textual and carto-geodetic information in particular (Figure 1.4, rectangles (1) and (2)). We aim to accelerate the detection of core city structures (building blocks, rivers, street networks), as well as the georeferencing process while keeping both very accurate.

Historical maps also exhibit general document-related challenges. Damage paper (Figure 1.4, rectangles (3)), non-straight lines (Fig. 1.3, *right*), and image compression create image inconsistency, and missing information and the change in the topological properties in historical map images leads to the difficulties of the map digitization process. Moreover, the style of handwritten texts has inconsistent representations (different font, size, and rotations) across different maps, where traditional text-related algorithms might succeed in one map but fail in others.

## 1.6 Problem statements

To extract reliable closed shapes from historical maps with the existing challenges, we propose to focus our research on the following questions:

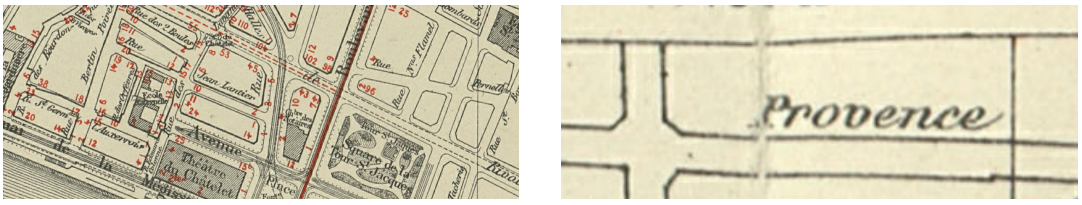



FIGURE 1.3: Some map-related challenges (*left*): visual polysemy, planimetric overlap, text overlap... and some document-related challenges (*right*): damaged paper, non-straight lines, image compression, handwritten text...For more detail of document-related challenges please refer to Figure 1.4.

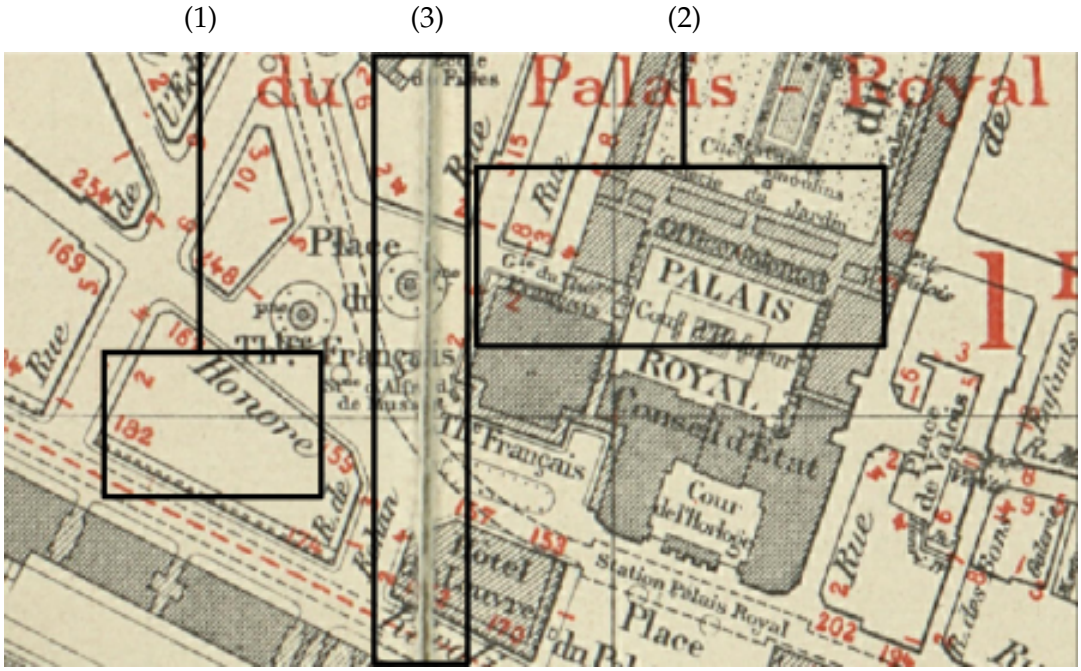



FIGURE 1.4: Contents of a 1925 urban topographic map along with an overview of their challenging properties for automatic feature extraction. Challenges in historical maps: (1) planimetric overlap, (2) text overlap, (3) paper folds.

**Our main research question is how to automatically extract high-quality closed shapes from historical maps on a large scale. Then we decompose the main problem into auxiliary questions which are complementary research directions and it will in turn be addressed in dedicated chapters.**

1. How to design a pipeline that can reliably extract closed shapes from map images? (Chapter 2)
2. How to better filter historical map images? (Chapter 3)
3. How to guarantee topological properties in the predictions? (Chapter 4)
4. How to improve the model robustness in different scanning conditions? (Chapter 5)
5. How to leverage the redundancies of historical maps? (Chapter 6)

## 1.7 Contributions

**A two-stage pipeline combining deep learning and mathematical morphology for historical maps vectorization task.** In this thesis, we answer the first research question by proposing a universal pipeline for vectorizing historical maps. The digitization of historical maps enables the study of ancient, fragile, unique, and hardly accessible information sources. Main map features can be retrieved and tracked through the time for subsequent thematic analysis. The goal of this work is the vectorization step, i.e., the extraction of vector shapes of the objects of interest from raster images of maps. We are particularly interested in closed shape detection such as buildings, building blocks, gardens, rivers, etc. in order to monitor their temporal evolution. This information of temporal evolution is important for studying urban morphogenesis.

Historical map images present significant pattern recognition challenges. The extraction of closed shapes by using traditional Mathematical Morphology (MM) is highly challenging due to the overlapping of multiple map features and texts. Moreover, state-of-the-art Convolutional Neural Networks (CNN) are perfectly designed for content image filtering but provide no guarantee about closed shape detection. Also, the lack of textural and color information of historical maps makes it hard for CNN to detect shapes that are represented by only their boundaries. Our contribution is a pipeline that combines the strengths of CNN (efficient edge detection and filtering) and MM (guaranteed extraction of closed shapes) in order to achieve such a task. The evaluation of our approach on a public dataset shows its effectiveness for extracting the closed boundaries of objects in historical maps. This work is explained in Chapter 2.

**A benchmark for historical maps vectorization tasks.** In this thesis, we study the general solution for shape vectorization which is a key stage of the digitization of high-scale historical maps, especially city maps. It relies on probable multiple methodological choices that hamper the vectorization performances in terms of accuracy and completeness. Investigating the optimal solution of vectorized historical maps is mandatory. The following contributions are introduced: a public dataset over which an extensive benchmark is performed; (i) a comparison of the performance of state-of-the-art deep edge detectors with topology-preserving loss functions, among which vision transformers and with deep or classical watershed approaches; (ii) a joint optimization of the edge detection and shape extraction stages; (iii) a study of the effects of augmentation techniques. This work is explained in Chapter 3.

**New loss functions for topology-oriented deep image segmentation** In this thesis, we propose two new topology-oriented loss functions for topological properties for segmenting historical maps. Most contemporary supervised image segmentation methods do not preserve the initial topology of the given input (like the closeness of the contours). One can generally remark that edge points have been inserted or removed when the binary prediction and the ground truth are compared. This can be critical when accurate localization of multiple interconnected objects is required.

We present a new loss function, called, Boundary-Aware loss (*BALoss*), based on the Minimum Barrier Distance [51] (*MBD*) cut algorithm. It is able to locate what we call the *leakage pixels* and to encode the boundary information coming from the given ground truth. Thanks to this adapted loss, we are able to significantly refine the quality of the predicted boundaries during the learning procedure. Furthermore, our loss function is backpropable and can be applied to any kind of neural network used in image processing. We apply this loss function on the standard U-Net [47] architectures on the historical map datasets. They are well-known to be challenging due to its high noise level, thin boundary and to the close or even connected objects covering the image space.

However, Boundary-Aware loss (*BALoss*) can not locate the boundaries with edge pixel which equals to zero. To tackle this issue, we propose a new topology-preserving deep image segmentation method which relies on a new leakage loss: the Pathloss. Our method is an upgrade solution of the BALoss [52], in which we want to improve the leakage detection for better recovering the closeness property of the image segmentation. This loss allows us to correctly localize and fix the critical points (a leakage in the boundaries, whether the value of pixels is zero or not) that could occur in the predictions, and is based on a shortest-path search algorithm. This way, loss minimization enforces connectivity only where it is necessary and finally provides a good localization of the boundaries of the objects in the image. Moreover, according to our research, our Pathloss learns to preserve stronger elongated structure compared to methods without using topology-preserving loss. Training with our topological loss function, our method outperforms state-of-the-art topology-aware methods on our historical maps. This work is explained in Chapter 4.

**Designing a contrast convolution block for historical map segmentation task** In this thesis, we propose a novel contrast convolution block for improving model robustness in the vectorization task. Detecting curvilinear structures is a pivotal low-level task in multiple image analysis challenges. Such structures can be abundantly found in nature and various data sources, e.g., vessels in medical images, roads in remote sensing images, and buildings in historical maps. Several solutions have been proposed but often fail to propose a unified framework for multiple object recognition. In parallel, with the development of complex neural networks architectures, existing networks can achieve segmentation results that satisfy pixel-level accuracy, but the correctness of the curvilinear structure cannot be guaranteed, hence, we propose a novel unified solution called ***contrast convolution***, which learns the gradient information for every pixel to improve curvilinear structure correctness. We use such contrast convolution to build higher-level modules named ***contrast blocks*** that add extra information in the network to enhance the curvilinear feature while training the network. By simply stacking our contrast blocks in front of different architectures, we evaluate our methods on our historical map dataset and prove the module effectiveness to maintain the high segmentation accuracy in curvilinear structure segmentation tasks. Surprisingly, these modules have a large potential for model robustness without significantly increasing the parameters of the models. This work is explained in

Chapter 5.

**Leveraging the redundancies of historical maps** In this thesis, we align historical maps in an unsupervised fashion to unlock the redundancies of historical maps. To be able to analyze, extract or leverage the redundancies and changes of historical maps, it is necessary to align maps in different time period. However, most of the existing historical maps are not or poorly aligned and therefore the redundancies cannot be used for any application such as improving map geo-referencing or map vectorization. To tackle this issue, we propose a geometric alignment framework with the help of edges, so-called *edge-guided geometric alignment* network where it leverages edge image (learning the edges are explained in Chapter 3) to guide the alignment of the original historical map images to minimize the false matches due to the unrelated information (such as texts or textures) in the historical map images. This work is explained in Chapter 6.

## 1.8 Publications

Here is the list of publications and contributions we made over the course of my doctoral researches:

### 1.8.1 Journal papers

1. Ngoc M Ô V*, **Chen Y***, Boutry N, et al. BuyTheDips: PathLoss for improved topology-preserving deep learning-based image segmentation. (Under review)
2. **Chen Y** et al. Automatic Vectorization of Historical Maps: a Benchmark. International Journal of Geographical Information Science, 2022. (Under review)

### 1.8.2 Conference papers

1. **Chen Y**, Carlinet E, Chazalon J, et al. Combining deep learning and mathematical morphology for historical map segmentation, International Conference on Discrete Geometry and Mathematical Morphology. Springer, Cham, 2021: 79-92.
2. **Chen Y**, Carlinet E, Chazalon J, et al. Vectorization of historical maps using deep edge filtering and closed shape extraction, International Conference on document analysis and recognition. Springer, Cham, 2021: 510-525.
3. Ngoc M Ô V*, **Chen Y***, Boutry N, et al. Introducing the Boundary-Aware loss for deep image segmentation, British Machine Vision Conference (BMVC) 2021.
4. Chazalon J, Carlinet E, **Chen Y**, et al. ICDAR 2021 competition on historical map segmentation, International Conference on Document Analysis and Recognition. Springer, Cham, 2021: 693-707.

5. **Chen Y**, Zhao Z, Ngoc M Ô V, Géraud T, Mallet C: Rethinking the Pixel Contrast in Curvilinear Structure Segmentation, submit to CVPR 2023. (Under review)

### 1.8.3 Published Dataset

1. Historical map segmentation:
   https://zenodo.org/record/4817662
   Authors: Joseph Chazalon, Edwin Carlinet, **Yizi Chen**, Julien Perret, Bertrand Duménieu, Clément Mallet and Thierry Géraud.

### 1.8.4 Other contribution

1. Competition for historical map segmentation:
   https://icdar21-mapseg.github.io/
   Organizers: Joseph Chazalon, Edwin Carlinet, **Yizi Chen**, Julien Perret, Bertrand Duménieu, Clément Mallet, Thierry Géraud.

### 1.8.5 Published Codes

1. Code for "Combining deep learning and mathematical morphology for historical map segmentation":
   https://github.com/soduco/paper-dgmm2021.git
   Contributors: **Yizi Chen**, Joseph Chazalon, Edwin Carlinet
2. Code for "Vectorization of historical maps using deep edge filtering and closed shape extraction":
   https://github.com/soduco/ICDAR-2021-Vectorization.git
   Contributors: **Yizi Chen**, Joseph Chazalon, Edwin Carlinet
3. Code for "Automatic vectorization of historical maps: a benchmark.":
   https://github.com/soduco/Benchmark_historical_map_vectorization.git
   Contributors: **Yizi Chen**, Joseph Chazalon, Edwin Carlinet
4. Code for "Introducing the boundary-Aware loss for deep image segmentation":
   https://github.com/onvungocminh/MBD_BAL
   Contributors: **Yizi Chen**, Minh On Vu Ngoc
5. Code for "BuyTheDips: pathLoss for improved topology-preserving deep learning-based image segmentation":
   https://github.com/onvungocminh/PathLoss.git
   Contributors: **Yizi Chen**, Minh On Vu Ngoc

In this chapter, we presented the background information for the *vectorization* of historical maps. We are particularly interest in the *Paris Atlas Municipal* which contains both document analysis retrieval and map-related challenges that exist in the source of historical maps. We begin with the main goal of this thesis which aims at facilitating the manual annotation effort by automating the process of vectorization of historical maps. This main goal is then divided into five research questions to push the performance of historical map vectorization as far as possible.

# Chapter 2

# Pipeline design for historical map vectorization

To answer our first research question i.e. **how to design a pipeline that can reliably extract closed shapes from map images**, we proposed a two-stage pipeline. It combines a deep edge detector with mathematical morphology to extract the closed shapes from historical maps. The deep edge detector is good at filtering edges while watershed can extract closed shapes from edge probability maps (or likelihood maps).
In Section 2.1, we explain the advantages of a two-stage pipeline and why it is effective for extracting closed shapes compared to a single stage pipeline. In Section 2.2, we introduce the background study for the two-stage pipeline. In Section 2.3, we summarize the test variants of our benchmark. In Section 2.4, we present the historical map dataset for the vectorization task. In Section 2.5, we illustrate the protocols for evaluating the performance of our proposed methods.
This chapter is an extended and adapted version of the contents of our publication [1].

## 2.1 Motivation

We target to recover geometric structures from scans of historical maps. As mentioned in Section 1.5, due to the limited texture and color content of such data sources, traditional semantic segmentation approaches of the literature would fail for most cases. Instead, we cast our problem as a vectorization challenge that can be turned into a region-based contour extraction task. Such a problem is traditionally solved through a two-step approach: the detection of edges or local primitives (lines, corners) followed by the retrieval of structures based on global constraints as proposed by Zhang et al.[53]. Recent works have shown the relevance of a coupled solution [54]. It remains tractable and efficient only for a limited number of structures. Region-based methods (e.g., based on probability density estimations (PDEs) [55]) may lead to oversimplified results and will not be further analyzed here.

The main issue of two-step solutions is the edge detection step. This low-level task is achieved by measuring local pixel gradients. Due to the amount of noise (overlapping objects, map deformation), this would result in many tiny and spurious elements that any global solution would manage connecting. Instead, we focus on boundary detection, i.e., a middle-level image task that separates objects at the semantic level according to different geometric

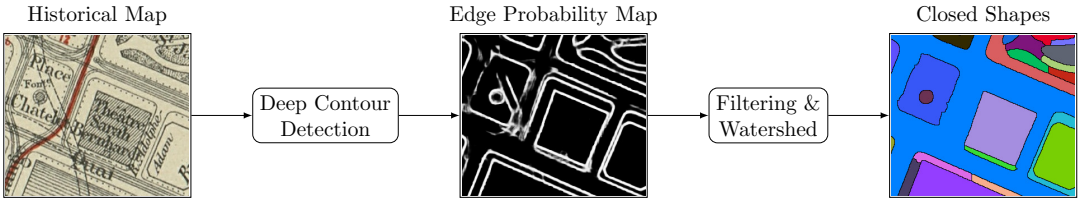



FIGURE 2.1: We combine an efficient edge detection and filtering stage using a deep network with a fast closed shape extraction using mathematical morphology tools.

properties of images. This offers two main advantages: (i) a limited sensitivity to noise in maps and (ii) the provision of more salient and robust primitives for the subsequent object extraction step.

Recently, among the vast amount of literature, CNN have shown a high level of performance for boundary detection [56, 57]. However, they only provide probability edge maps. Without topological constraints, image partitioning is not ensured. Conversely, watershed segmentation techniques in mathematical morphology can directly extract closed contours. They run fast for such a generation, but may lead to many false-positive results. Indeed, using only low-level image features such as image gradients, watershed techniques may not efficiently maintain useful boundary information [58]. Consequently, we propose here to merge the CNN-based and watershed image segmentation methods in order to benefit from the strengths of both strategies [59]. A supervised approach is conceivable since we both have access to reference vectorized maps and CNN architectures pre-trained with natural image.

## 2.2 Related work

### 2.2.1 Image vectorization approaches

Image vectorization approaches can be separated into following three types:

**Contour detection with polygon post-processing:** Images can be vectorized through combining contour extraction (marching cubes [60], Grabcut [61]) with polygon simplification method (Douglas-Peucker [62] or simply delaunay triangulations [63, 64]). However, this type of vectorization approach has three main drawbacks. Firstly, the quality of polygons is highly influenced according to the instability prediction of classification maps by using existing deep segmentation methods. Secondly, achieving high-quality vectorized output requires manual refinement processes [65, 66] which are expensive and complex. At last, these approaches can not be applied to applications that require detecting complex polygons such as in historical maps.

**End to end polygon detection:** To speed up annotation speed for polygonization images, Castrejon et al. [67] and Acuna et al. [68] propose Polygon-RNN and Polygon-RNN++, which can annotate polygons in semi-automatically fashion. Although these two networks produce polygons directly from images, it still creates invalid polygons due to vertexes self-intersections and

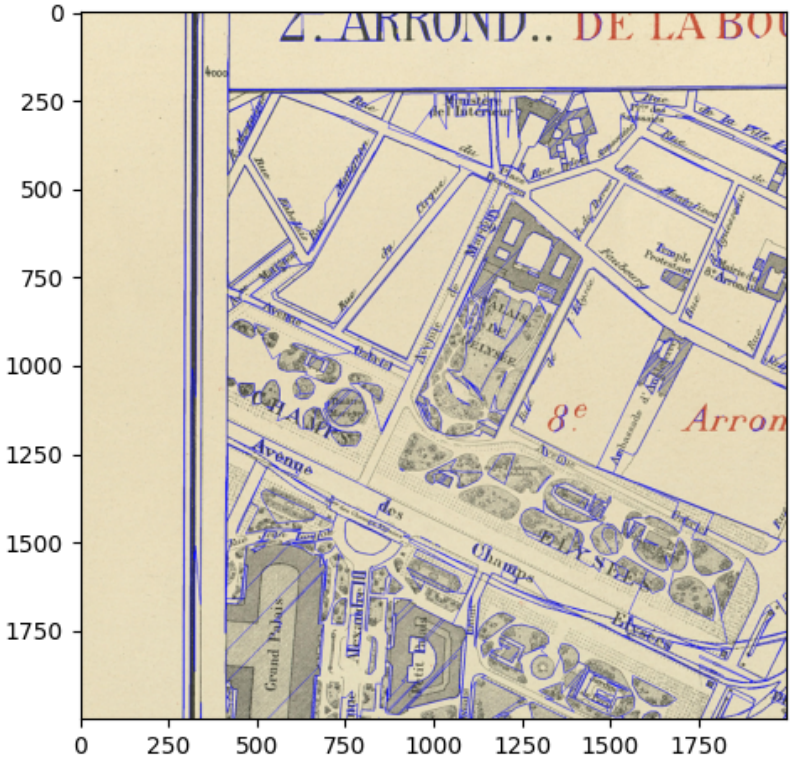



FIGURE 2.2: KIPPI results for historical map vectorization. We can see that wrong partitions are created due to miss detected lines, mainly because of text regions.

line overlaps. To eliminate vertex self-intersections and line overlaps, Girard et al. [69] propose Poly-CNN, but those polygons are limited with only four vertices which makes it difficult to apply to scenarios requiring complex polygons with more vertices. PolyMapper [70] uses RNN similar to Polygon-RNN and Polygon-RNN++, still, it is limited to produce high-quality complex polygons. Moreover, the methods mentioned above highly rely on images with rich textures and polygons are normally have a low percentage of shared borders, whereas it might not be sufficient to apply to our historical map images which have limited colors and share neighbors of objects.

**Polygon partitions:** Line segment detector (LSD) proposed by Von et al. [71] is a tool to detect line segment vector from images. However, these line segments do not guarantee to structure of geometrical partitions or polygons. To be able to use those line segments for the purpose of creating geometrical partitions, Duan et al. [72] construct a Voronoi diagram based on those line segments to produce shapes with strong geometric properties. Bauchet et al. [73] propose Kinetic polygonal partitioning (KIPPI) to create image partitions by progressively extending line segments until they meet each other. However, this technique heavily relies on the quality of line segments and the wrong parameter setting could lead to too many falsely detected lines which results in image over-segmentation shown in Figure 2.2. Furthermore, line segments are not sufficient to polygonized objects with curve boundaries, which is another limitation of line segments related polygon partition approaches. As

a results, these methods for producing high quality polygons might not be sufficient enough for the purpose of vectorization historical maps.

### 2.2.2 Two-stage pipeline

Therefore, we use a two-stage pipeline by combining the strength of deep edge filters (good at filtering edge images) with watershed segmentation (which offer strong geometric guaranties for closed shape extraction).

Watershed segmentation techniques are typically applied on gradient images to extract closed shapes from natural images. The input image of the watershed is a single channel image of boundaries' activation which is also called *probabilistic boundaries (Pb)* [74]. Getting the boundaries' activation through image features (image gradient) have been comprehensively studied by Martin et al. [74], where the authors use hand-crafted image features such as brightness, color, and texture gradients to localize the gradient of the image. Following the idea of *Pb*, Arbelaez et al. [75] combine different scales of *Pb* local cues into *multiscale oriented probabilistic boundaries (mPb)*. Moreover, applying watershed in *Pb* proposed by Hanbury et al. [76] and *mPb* proposed by Arbelaez et al. [75] are the two early attempts of two stage pipelines to extract closed shapes from edge images. Due to their effectiveness of two stages pipeline for extracting closed shapes from images, the methods have been adapted to the historical map segmentation (vectorization) tasks. Ares et al. [77] use ridge detection to detect edges objects followed by a flood fill algorithm to extract the buildings from cadastral maps. Similarly to this work, we propose to combine deep edge detector with watershed to extract closed shapes in historical maps. The two-stage pipeline is proved to be more effective compared to one-stage pipeline which are shown in quantitatively *COCO-PQ* evaluation scores.

### 2.2.3 Learning probabilistic boundaries

Learning probabilistic boundaries (or edge probability maps) is a long-term studied topic in the field of computer vision. Defining a source color image $I \in \mathbb{R}^3$, the probabilistic mapping function for boundaries, $f$ transfers the source image into the target image of probabilistic boundaries $y$, where $f : I \mapsto y; f : \mathbb{R}^3 \mapsto \mathbb{R}^2$. The value of a pixel close to 1 means that the pixel is more likely to be classified as a boundary pixel.

Early probabilistic boundaries are detected through hand-crafted features. Martin et al. [74] use gradient operators for brightness, color, and textures (called **Cue combination**) with a logistic regression classifier to determine the probability of every pixel whether it should belong to edges or not. To enhance the image properties with scales, Arbelaez et al. [75] upgrade the **Cue combination** with multiscale image combination to predict better probabilistic boundaries. Overcoming the limitations of hand-crafted features, Dollar et al. [78] invent **Boosted Edge Learning (BEL)** which selects and combines a large number of computed features in different scales thanks to **probabilistic boosting-tree** invented by Tu [79]. Similar to the work in Dollar et al. [78], Lim et al. [80] and Dollar et al. [81] uses a random forest classifier to determine edge patches. However, these methods are generated through

hand-crafted or pre-computed image features which have limited visual representations that can not be easily adapted in an end-to-end system.

To tackle this problem, Xie et al. [56] develop an end-to-end edge detection system based on CNN (so-called **Holistically-nested Edge Detection (HED)**) which can automatically learn the abstract image features to resolve the ambiguity in the edges of a natural image. Moreover, several publications are proposed to boost the architectures of the HED. Liu et al. [82] design a coarse-to-fine edge detection network with relaxed labels to guide the HED network. Xu et al. [83] design an **Attention-Gated Conditional Random Field (AG-CRF)** to refine and robustly fuse the intermediate edge representations in different scales. He et al. [84] proposed **Bi-directional cascade network (BDCN)** which combines a cascaded network architecture with the **Scale Enhancement Module (SEM)** to efficiently learn the multiscale representations of edges in the network. Since detected edges have very strong spatial correlations between neighboring pixels, Su et al. [85] proposed a **Pixel Difference Network (PiDiNet)** that adds traditional edge detection operators into a CNN architecture which decrease the complexity of the edge detection models (<0.1M parameters) and also surpass the recorded result of human perception in BSDS500 datasets [75]. Currently, vision image transformers show the greatest performance in wide range of computer vision problems, including edge detection. Pu et al. [86] proposed **Edge Detection with Transformer (DETER)** with global and local transformer architecture to capture high-resolution fine-grained features and long-range global context in the image, becoming the current state-of-the-art in BSDS500 and NYUDv2 datasets.

Traditional VGG- [87] or ResNet-based [88] architectures normally require a large amount of annotated data to train. These architectures with a high number of parameters cannot be easily adapted to our historical map dataset. We annotated our historical maps with limited in image size comparing with natural image. To train more efficiently for small datasets, Ronneberger et al. [47] invented **U-Net** which is a high-performance architecture that uses a contracting path to capture context and a symmetric expanding path to maintain a fine spatial accuracy for the prediction.

### 2.2.4 Watershed segmentation techniques in general

In Mathematical Morphology, the Watershed Transform [89] is a *de facto* standard approach for image segmentation. It has been used in many applications and has been widely studied in terms of topological properties [90, 91], in terms of algorithms and in terms on computation speed [91, 92].

There are two well-known issues in general watershed segmentation techniques: the over-segmentation due to the high number of minima, and the gradient leakage that merges regions. There is a third general issue with the watershed that concerns the separation of overlapping or touching objects, but this is not a problem in our case since the map components do not overlap for a given layer.

The over-segmentation problem is generally solved by filtering the minima first. Soille et al.[93], the $h$-minima characterize the importance of each local minimum through their *dynamic*. When flooding a basin, the dynamic

actually refers to the water elevation required to merge with another basin. Attribute filters, filters by reconstruction [94] also allow to eliminate some minima based on their algebraic properties: size, shape, volume... Another efficient approach consists in first ordering the way the basins merge to create a hierarchy of partitions and then performing a cut in the hierarchy to get a segmentation with non-meaningful basins removed [95–97].

The early leakage problem lies in the quality of the gradient. It has been noted [98], that (hierarchical) watersheds have better results on non-local supervised gradient estimators. The idea of combining the watershed with high performance contour detector dates back to Arbelaez et al. [75]. The relevance of a simple closing by area and dynamic on the edge map produced by our deep-learning edge detector combined with the watershed for this application lies in three points.

First, the minimum size of the components is known. Indeed, the document represents a physical size, and regions whose area is below a certain threshold. Thus, we have a strong *a priori* knowledge we want to inject in the process, the minimum size of the regions (in pixels). This type of constrain is hard to infer in a deep-learning system, and we cannot have such guarantees from its output. Having hard guarantees about the shapes and their size is at the foundation of the granulometry in Mathematical Morphology. Moreover, the connected (area) filter used for filtering the edge image ensure that we do not distort the signal at the boundaries of the meaningful regions.

Second, the watershed segmentation method does not rely on the strength of the gradients to select the regions. Even if the edge response is low (i.e, the gradient is weak), the watershed is able to consider this weak response and closes the contour of the region. We do not depend on the strength of the edge response from a deep edge filter, which is difficult to calibrate and normalize.

Last but not least, not only the watershed outputs a segmentation, but some implementations also produce watershed lines between regions. In our application, watershed lines are even more important than regions because we need to extract polygons for each meaningful shape. Event if we could extract boundaries from regions, it avoids an extra processing step. The watershed lines produced by the algorithm are one-pixel-large and are located where the edges are the strongest, i.e., where the network has the strongest response on thick edges. The watershed lines form closed boundaries around regions which is a guarantee we cannot have from the output of a network.

### 2.2.5 Meyer and end-to-end deep watershed segmentation

The watershed segmentation is, indeed, a very powerful tool which can leverage a global image context, succeed even in the presence of low contrast, and present strong topological guarantees, like the production of closed shapes exclusively. Our preliminary works [1, 2] restricted to the use of the **Meyer Watershed** [89] and showed that its sensitivity to noise could be mitigated thanks to the deep edge filtering stage.

The Meyer watershed detects the catchment basins of the minima in the gradients of images. The watershed process consists in flooding "water" from each catchment basin (also called regional minima) until regions merge,

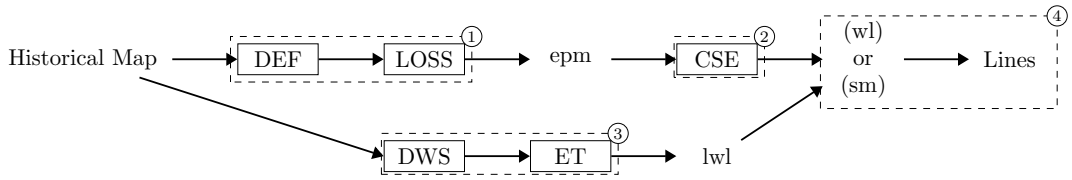



FIGURE 2.3: Our proposed pipeline; DEF: deep edge filter; LOSS: binary cross entropy loss or topology-oriented losses; DWS: deep watershed; CSE: component labelling (CC) or watershed segmentation (WS) (Meyer Watershed); ET: edge thinning; LINES: vector output; epm: edge probability map(image of likelihood); lwl: learned watershed levels; (wl) or (sm): watershed lines or saliency maps; The number indicates in the top right of the blocks shows the stages of the processes from 1-4. (Capital letters with box represents the processing steps and lowercase ones represent intermediate results.)

creating watershed lines. The strength of watershed lines depends on the height at which basins get connected. The resulting image is called the *saliency map*, and shape properties can be subsequently computed for extra filtering. We used two criterions to perform this shape filtering stage: shape area and edge dynamic, which is the difference between some basin's minimum and the height of the lowest point on its boundary. Such closed shape extraction stage can reweigh weak edges, filter some weak or small shapes, but cannot recover lost edges (for which the deep edge filter predicted very low edge probabilities).

Although Meyer watershed is a powerful tool to extract closed shapes from edge images, the *area* and *dynamic* values are still required to be set manually and the optimal setting of these two parameters can vary a lot with different input images. To avoid manual parameter setting based on the prior knowledge of shapes in traditional watershed transform, Bai et al. [50] invented the so-called deep watershed which learns the watershed transform in an end-to-end fashion.

### 2.2.6 Joint optimization

In our previous work [2], the optimal parameters for each deep edge filtering network is to maximize the *COCO-PQ* metrics for a simplified pipeline composed of each deep edge filter followed by a threshold of the resulting Edge Probability Map is at a fixed value ($P = 0.5$), on a validation set. However, it is possible to perform a global, joint optimization of the parameters of both stages. There is no guarantee that the combination of independently optimized stage is globally optimal. To address this issue, we propose a global optimization procedure.

## 2.3 Vectorization pipelines under test

In this thesis, we test different vectorization pipelines which are summarized in Figure 2.3. Chapter 3 demonstrates about the connected component labeling and joint optimization of the deep edge filtering and closed shape extraction stages. Chapter 4 introduces extra topological loss functions during

TABLE 2.1: Summary of the training, validation and test sets used in this study.

| Subset | *image size* | *Num. of closed shapes* |
|---|---|---|
| train | 4,500px × 9,000px | 3343 inst. |
| val | 3,000px × 9,000px | 2183 inst. |
| test | 6,000px × 5,500px | 2836 inst. |

the training of the deep edge filters. Chapter 5 introduces the a novelty modules for improving robustness of models during the training of the deep edge filters.

## 2.4 Explaining our dataset (Atlas Municipal)

The dataset used in this thesis was published and publicly release in the context of a publication at the International Conference on Document Analysis and Recognition in 2021 [2]. This dataset is built using excerpt from the corpus of historical maps we introduced in Section 1.4, the *Paris Atlas Municipal*. The performance of the different pipelines under test is assessed using the protocol of the ICDAR 2021 competition on historical map segmentation [99]. In particular, we follow the protocol of task 1 (*Building blocks detection from historical maps*), but use a different dataset, containing fewer images and for which all closed shapes were annotated — not only building blocks. The dataset contains 2 large map images, extracted from a series of Paris Atlases dating from 1898[1] and 1926[2].

Each map image was manually annotated to create 8,362 polygons in total — one for each closed shape. Such annotation procedure makes it possible to generate the target *Edge Probability Map* using the boundaries of the polygons, or to assess the final performance of the vectorization process. Figure 2.4 shows an excerpt of some input image and the associated shape annotations.

The dataset was split into the subsets summarized in Table 2.1. The training set is an excerpt from the top of the first sheet of the 1926 edition of the *Atlas Municipal*, while the validation set is built using the lower part of this particular sheet and is used in the early stopping mechanism to prevent overfitting of the networks. The test set is built using the third sheet of the 1898 edition to test the generalizability of the networks for other historical maps with different scanning conditions.

## 2.5 Evaluation protocol

We mainly follow the evaluation protocol used in our previous publications [1, 2], which relies on the *COCO-PQ* metric proposed by Kirillov et al.[100]. Indeed, such metric effectively focuses on the number of shapes which are correctly detected (from the point of view of the ground truth) or predicted

[1] Atlas municipal des vingt arrondissements de Paris. 1898. Bibliothèque de l'Hôtel de Ville. Ville de Paris. http://bibliotheques-specialisees.paris.fr/ark:/73873/pf0000935524

[2] Atlas municipal des vingt arrondissements de Paris. 1926. Bibliothèque de l'Hôtel de Ville. Ville de Paris. http://bibliotheques-specialisees.paris.fr/ark:/73873/pf0000935524

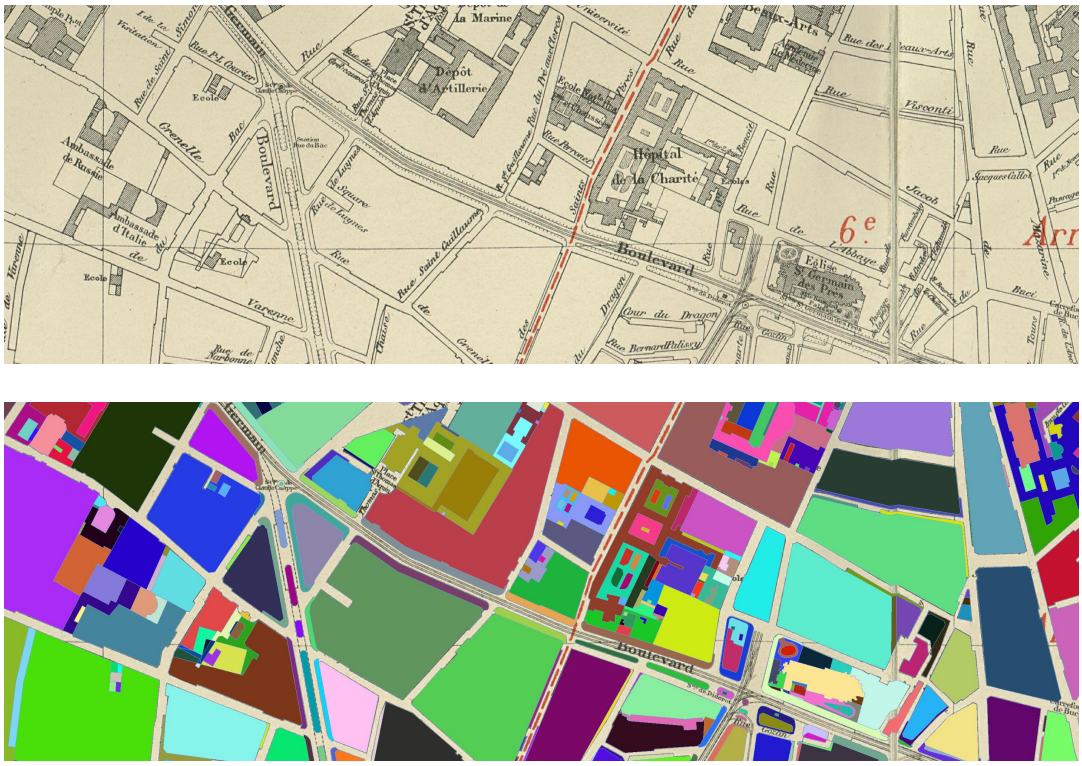



FIGURE 2.4: Excerpt from the 1926 edition *"Atlas municipal"* (Scale: 1: 5,000) (*top*) and manually-labelled closed shapes (*bottom*) that we aim at extracting automatically and it is shown in random colors.

(from the point of view of the prediction), leaving the relative size of the shapes as an optional extra indication which may or may not be considered. The *COCO-PQ* term is to measure the quality of intersection of union (IoU) between detected and ground truth. $t_i$ is the target shapes, $p_j$ is the predicted shapes, $TP$ is true positive detected shapes, $FP$ is false positive detected shapes and $FN$ is false negative detected shapes.

$$PQ = \frac{\sum_{(t_i,p_j)\in TP} IoU(t_i, p_j)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{2.1}$$

where $FP$ means the predicted instances have smaller overlap with the ground truth; and $FN$ means the ground truth instances does not pair with any predicted instances. The term $PQ$ can also be represented as the product of segmentation quality $SQ$ and recognition quality $RQ$ where:

$$SQ = \frac{\sum_{(t_i,p_j)\in TP} IoU(t_i, p_j)}{|TP|}, \quad RQ = \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}. \tag{2.2}$$

While this *COCO-PQ* indicator is fairly sufficient for our study, we propose to consider several extra indicators to provide a more accurate view of the performance of each architecture, as well as to exhibit the counter-intuitive results of other metrics. To these ends, we will consider: extensions of the *COCO-PQ* metric which enable more qualitative analysis, and pixel-level metrics which focus on shape boundaries, and also on some extra

topology-based indicator.

When appropriate, we will show Precision and Recall maps computed using the same IoU values as the *COCO-PQ* scores, as introduced by Chazalon et al. [101]. Precision maps will show, for each predicted shape, the value of the highest possible IoU between this predicted shape and every ground truth shape, using a color scale. Recall maps will conversely show, for each ground truth shape, the value of the highest possible IoU between this expected shape and every predicted shape, using the same color scale. These two qualitative indicators, complemented by a study of the detection quality against shape size, can provide deeper insights about the performance of the segmentation systems we are studying. The evaluation procedure is illustrated in Figure 2.5.

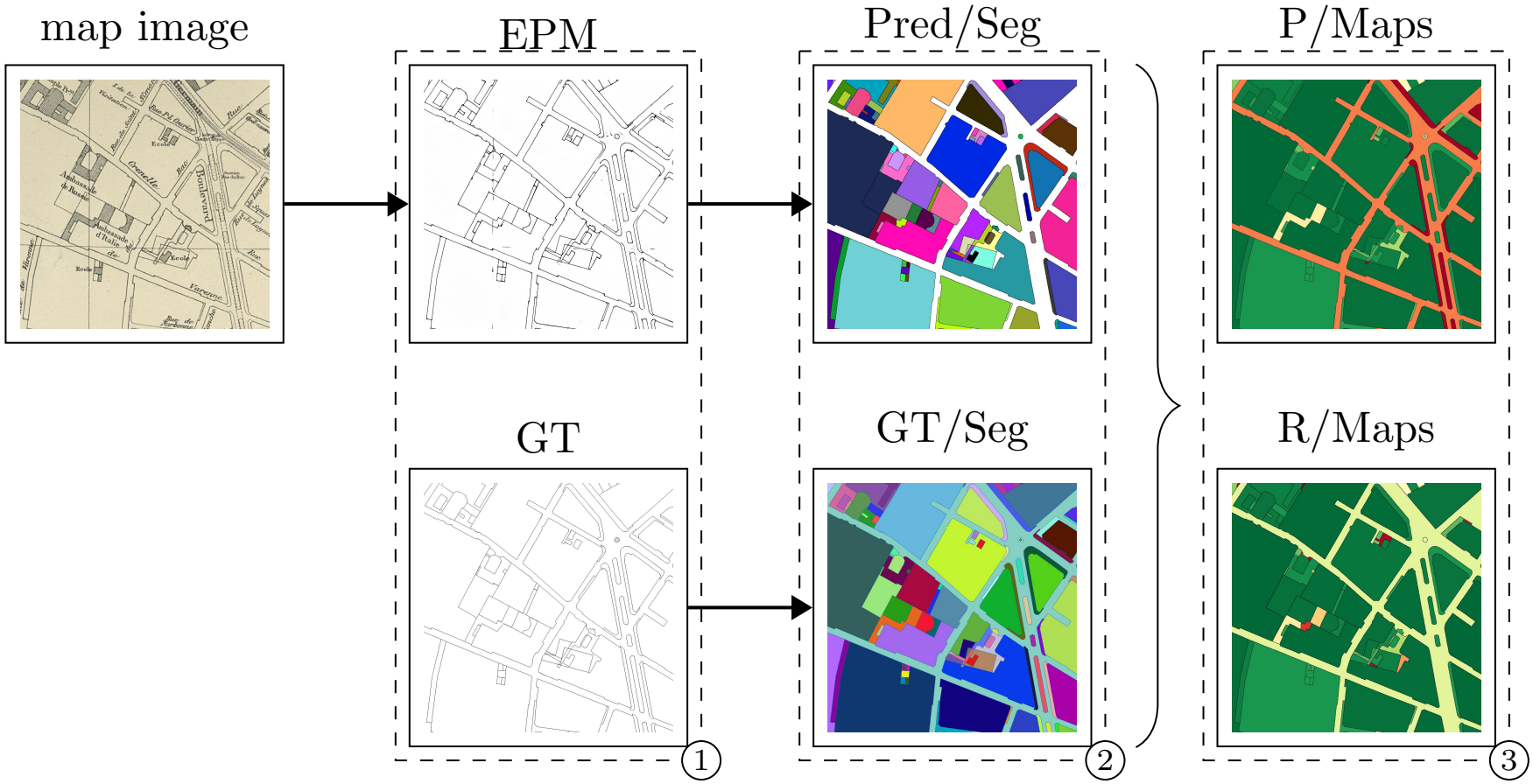



FIGURE 2.5: Our evaluation pipeline. The edge samples are shown in block 1. The prediction and segmentation samples are shown in the block of 2; The *COCO-PQ* evaluation measures the intersection over union (IoU) between prediction and ground truth segmentation. The Precision (P) and Recall (R) maps, which are visual, qualitative indicators, are shown in block 3.

As outlined in the introductory chapter of this thesis, historical maps may present many challenges due to their limited color and texture, content overlap as well as paper damage and artifacts. To overcome these challenges for extracting closed shapes for the purpose of vectorizing historical maps, we propose a two-stage pipeline which combines deep edge filter with watershed segmentation. Even though this pipeline has been put forward in previous literature, there remains a need for a comprehensive study and improvement within the different stages of this pipeline. The following chapters investigate the existing possibilities for this pipeline, with **better deep edge filters, preserve better topology in edge image, with better model robustness and by the redundacies can be leveraged through alignment techniques for the purpose of historical map vectorization.**

# Chapter 3

# Learning edges through deep neural architectures

To answer our second research question of **how to better filter the historical map images**, we compare multiple deep neural architectures for extracting semantically meaningful edges from historical maps.
In Section 3.1, we exhibit the advantage of applying a multiscale architecture in the edge detection task, followed by a solid benchmark of two state-of-the-art methods, HED and BDCN, with and without using pre-trained weights. In Section 3.2, we benchmark the results of two transformer architectures which enable to consider longer pixel dependencies with the hope of improving vectorization performance over CNN-based architectures. In Section 3.3, we detail our reimplementation of an end-to-end, deep watershed transform as a replacement of both deep edge filtering and closed shape extraction stages, and report its performance against our optimized pipeline. In Section 3.4, we carry out the study of using data augmentation techniques to improve the performance of historical map vectorization task through a joint optimization strategy. This chapter is an extended and adapted version of the contents of our publication submitted to IJGIS [3].

## 3.1 Multi-scale deep neural network architecture

### 3.1.1 Motivation

Detecting semantic meaningful edges from historical maps is a challenging task. It requires to separate the meaningful semantic edges (e.g, boundaries of objects) from other information (e.g, texts and textures), despite their similarity in their low-level. Early ways of extracting semantic meaningful edges are based on color gradient (such as Canny [102] and Local binary pattern (LBP) [103]) and feature learning based methods (such as Probabilistic boundary (Pb) [104], multiscale probabilistic boundary (mPb) [75] and structured edge (SE) [81]). Due to the limited colors and texture information especially in the *Paris Atlas Municipal*, such approaches are prone to fail at extracting semantic meaningful edges using only low-level features. Recently, deep learning based methods were developed to extract high-level semantic meaningful edges by learning to combine low-level and high-level features representatives. Among those deep learning approaches, multiscale deep neural network architectures (such as HED [56], RCF [105] and BDCN [84]) have been proven to be successful for merging low and high-level features, and

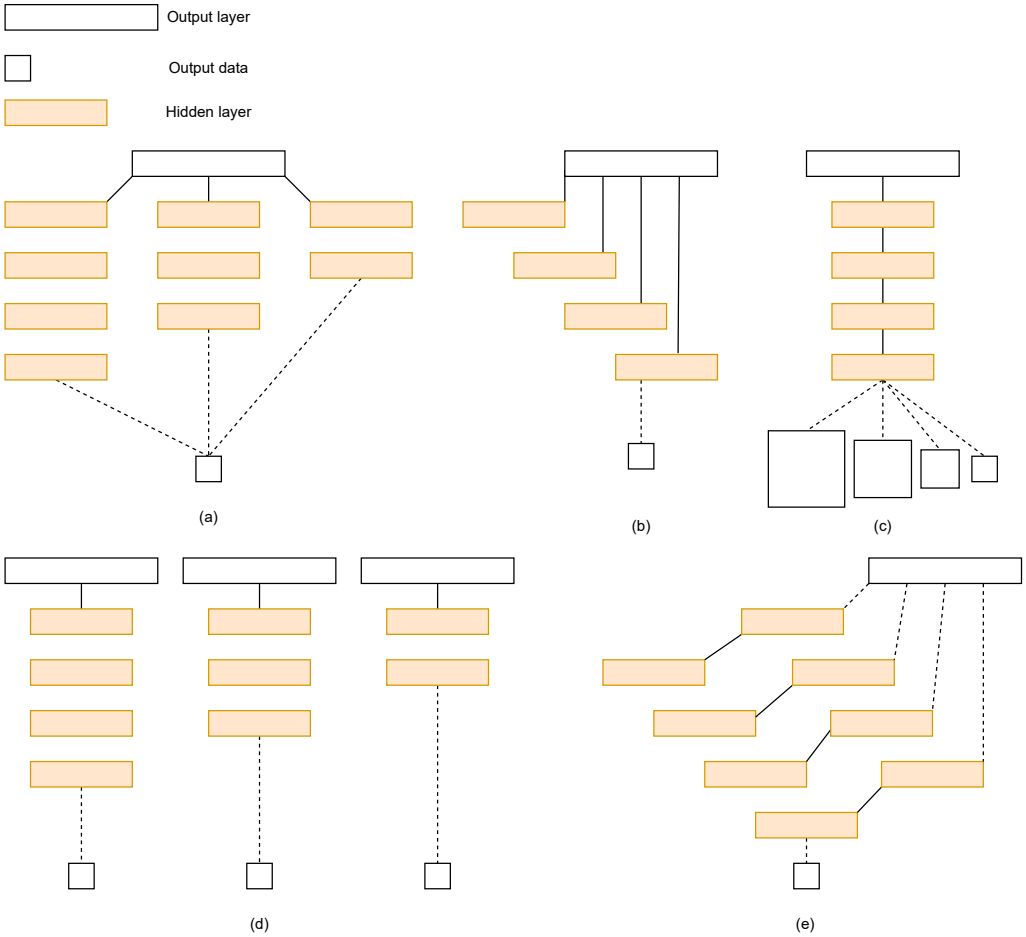



FIGURE 3.1: Five different multiscale architectures for semantic edge detection task recreated from paper [56]. Solid lines: connected layers in the intermediate layer of neural networks. Dash lines: connected layers in the output layer of neural networks. (a): multi-stream; (b): skip-layer; (c): single model in multi-scale inputs; (d): separate training; (e): holistically nested architecture.

achieve state-of-the-art results in various edge detection applications and we applied them to the task of historical map vectorization task.

### 3.1.2 Related work

Feature-based methods for learning the semantic edges from image through CNN can be classified into five types according to [56]: multi-streams, skip layer, single model on multiscale inputs, separate training and holistically-nested which are shown in Figure 3.1.

**Multi-stream (a) [106, 107]**: The multi-stream architecture uses several parallel network streams (related to multiple scales). The input is separated into several streams and fed into a global output layer.

**Skip-layer (b) [108–110]**: Different from the **(a)** architecture, skip layer added features from different levels of streams and added it by a shared output layer. These two architectures only requires one loss function to train the neural network architecture.

**Single model on multiscale inputs (c) [111, 112]**: This architecture *ensembles*

different input images with several scales into only one network.
**Separate training (d)**: The separate training strategy is to separate different training streams into three different networks with different output data and layer per scale compared to the **(a)** architecture. This strategy enables to train the network through networks with different settings (different level of depth, receptive fields) with different losses.
**Holistically-nested (e)**: The holistically-nested architecture is constructed by creating predictions from multiscale representations of features and eventually merging them into a single output layer.
In this thesis, we studied two representatives holistically-nested based architectures(e), which are **holistically edge detector (HED)**[1] and **Bi-directional cascade network (BDCN)**[2] for probabilistic boundary detection applied to historical map segmentation.

### 3.1.3 Multi-scale edge detection

**Some mathematical formulation of historical map segmentation (Vectorization) task**: We denote our map image as $I \in \mathbb{R}^{(H,W,3)}$ and its corresponding target edge map $y \in \mathbb{Z}^{H,W}, y \in \{0,1\}$. The deep edge detector transfers the image domain to a likelihood image $\hat{y} \in \mathbb{R}^{H,W}, \hat{y} \in [0,1]$ by learning the function $f : \mathbb{R}^{(H,W,3)} \rightarrow \mathbb{R}^{(H,W)}$. **Weighted loss function for imbalanced positive and negative edges samples**: Since the positive $y_-$ and negative $y_+$ edge samples are highly imbalanced (in training and validation set, edge pixels account for only 2.5% of image pixels), Xie et al. [56] proposed a rebalanced version of binary cross-entropy loss $\mathcal{L}_{BCE}$ by weighting positive and negative samples using a parameter $\beta$ and the output $X$ of the deep edge detector is parameterized by $w$:

$$\mathcal{L}_{wBCE} = -\beta \sum_{j \in y_+} \log \mathbb{P}(y = 1|X; w) - (1-\beta) \sum_{j \in y_-} \log \mathbb{P}(y = 0|X; w), \quad (3.1)$$

$$\beta = \frac{|y_-|}{|y|}; 1 - \beta = \frac{|y_+|}{|y|}. \quad (3.2)$$

**Holistically-edge detection architecture**: The multi-scale side output $y^m_{side}$ in the holistically-nested architecture is calculated by the sigmoid activation $\sigma$ of the intermediate feature output $f^m_{side}$ then fused by a $1\times1$ convolution layer $f_{comb}$ to single channel output $y$:

$$y^m_{side} = \sigma(y^m_{side}); \quad \sigma(x) = \frac{1}{1+e^{-x}}. \quad (3.3)$$

where $w_i$ is the weighted factor for $m$ number of intermediate feature output of $y^m_{side}$:

$$\hat{y}^{fuse} = \sigma(f_{comb}(y^m_{side})) = \sigma(\sum_{i=1}^{m} w_i * y^i_{side}). \quad (3.4)$$

The total loss of HED $\mathcal{L}_{total}$ is the sum of the side output of binary cross entropy and the fuse binary cross entropy loss, and $\hat{y}^{fuse}$ is the weighted sum

[1] BSDS500 state-of-the-art 2015
[2] BSDS500 state-of-the-art 2019

of the intermediate features:

$$\mathcal{L}_{total} = \mathcal{L}_{wBCE}(\hat{y}^{side}, y) + \mathcal{L}_{wBCE}(\hat{y}^{fuse}, y). \tag{3.5}$$

**Bidirectional cascade edge detection architecture**: To calculate accurate loss for the intermediate output for the edge detection task, He et al. [84] designed the BDCN network with shallow to deep (s2d) and deep to shallow (d2s) propagation of the network to better approximate edges at different scales:

$$y_{s2d} = y - \sum_{i<s} P_i^{s2d}; y_{d2s} = y - \sum_{i>s} P_i^{d2s}, \tag{3.6}$$

As explained by the BDCN authors [84], the better edge approximation $y_s$ for each scale $s$ can be approximated as:

$$P_s^{s2d} + P_s^{d2s} \sim 2y - \sum_{i<s} P_i^{s2d} - \sum_{i<s} P_i^{d2s}. \tag{3.7}$$

To make the edge detection easier to train at different scales $s$, He et al. [84] inserted **Scale Enhancement Modules (SEM)** in intermediate layers of their neural networks thanks to the use of dilated convolutions:

$$y_{ij} = \sum_{m,n}^{h,w} x_{[i+r\cdot m, j+r\cdot m]} \cdot w_{[m,n]}, \tag{3.8}$$

where $r$ is the dilation rate and $w$ is the parameter. The SEM module is used to to combine several outputs of dilated convolution into one output $y^{side}$. Similar to the total loss calculation in the HED network, where the $w_{side}$ and $w_{fuse}$ are the weights for the side and fuse outputs which can be rewritten as the total loss of BDCN can be formulated as:

$$\mathcal{L} = w_{side} \cdot \mathcal{L}(\hat{y}, y) + w_{fuse} \cdot \mathcal{L}(\hat{y}, y), \tag{3.9}$$

$$\mathcal{L}_{total} = \sum_{s=1}^{S} \mathcal{L}(p_s^{d2s}, y_s^{d2s}) + \mathcal{L}(p_s^{s2d}, y_s^{s2d}). \tag{3.10}$$

**U-Net (baseline approach)**: Inspired by FCN [39], this famous U-shaped architecture features a symmetrical structure that can preserve high performance prediction through accurate pixel spatial localization for semantic or instance segmentation tasks. It is essential for the geospatial applications (images of remote sensing or historical maps), where the image predictions require preserving the high accuracy of spatial information.

### 3.1.4 Experimental settings

To prove the effectiveness of our two-stage pipeline, we designed the following training protocols:

- **Data Preprocessing:** By dividing the original RGB values by 255, we normalize their value in the $[0, 1]$ range.
- **Weight Initialization:** We use a Kaiming initialization [113].
- **Batch Size:** We use a batch size of 4 (following preliminary experiments with sizes of 1, 2, 4 and 6).

- **Pre-trained weights:** We reuse the weights from the PyTorch Image Models ("timm") library [114] for the VGG backbone.
- **Loss formulation:** we use HED's re-weighting strategy.

We follow the training and selection procedure as follows:

- **Training:** For a given Deep Edge Filter $DEF$, using the train set, train the network for $M$ epochs. At each epoch $i$, we obtain $DEF_i$ which generates an Edge Probability Map (EPM).
- **DEF selection:** Using the validation set, compute the corresponding set of Edge Probability Maps $EPM_i$ using $DEF_i$ for $i \in \{0, \ldots, M\}$, then using a naive Closed Shape Extractor $CSE_{naive}$ (described hereafter), select the best Deep Edge Filter $DEF_{best}$ based the topological score (*COCO-PQ*) of the predicted shapes:

$$\begin{aligned} shapes_i &= CSE_{naive}(EPM_i) \\ DEF_{best} &= argmin_i(PQ(shapes_i)) \end{aligned}$$

- **CSE parameter tuning:** Then, using the best Deep Edge Filter $DEF_{best}$ as a base, restore or recompute $EPM_{best}$, the set of Edge Probability Maps for the validation set, and grid-search for the best $\theta$ parameters of the Meyer Watershed for Closed Shape Extraction ($CSE_{best}$), over the set of possible parameters $\Omega$:

$$\begin{aligned} shapes_\theta &= CSE_\theta(EPM_{best}) \\ CSE_{best} &= argmin_\theta(PQ(shapes_\theta)) \end{aligned}$$

- **Global evaluation:** The final evaluation on the test set is performed by combining the best Deep Edge Filter $DEF_{best}$ and the best Closed Shape Extractor $CSE_{best}$ to compute the shapes from test set samples.

We use a threshold of the EPM at 0.5 followed by a connected component labelling. However, for a fairer comparison with the watershed CSE, we add an edge-thinning step which allows obtaining thin, 1-pixel-large shape boundaries.

We use the same Meyer watershed CSE as the original authors, considering the following values for area filtering with value of 50, 100, 200, 300, 400, 500 number of pixels, and for dynamic value we use value from 1 to 10 with step of 1. The area and dynamic filters are the pre-filtering step for removing non-meaningful local minimum for Meyer watershed. The area is used to merge the regions with size lower than a specific area threshold, while the dynamic refers to the water elevation that is used to merge with other regions. This procedure gives us the opportunity to report the performance (on test set) of two different pipelines:

- **Best Meyer Watershed for the CSE stage:** This variant reports the performance of the full pipeline previously described, combining the best Deep Edge Filter $DEF_{best}$ and the best Closed Shape Extractor $CSE_{best}$ to compute the shapes from test set samples.
- **Naive Connected Component Labelling for the CSE stage:** This variant reports the performance $DEF_{best}$, combined with the naive Closed Shape Extractor $CSE_{naive}$. The purpose of reporting this simpler pipeline is to confirm the benefit of using an elaborated CSE stage based on some watershed.

TABLE 3.1: The following parameters are static and their respective columns are hidden: the CSE used is a naive connected component labelling ([2] used a grid search to find the best threshold $\theta$ for EPM binarization while we use a fixed value of 0.5), the loss function is the binary cross entropy, the best DEF is selected using the protocol of [2], no augmentation is performed. For the architectures, * indicate pre-trained variants.

| DEF Archi. | Training config. | CSE Param. | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|
| | | $\theta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | Proposed | 0.5 | 46.8 | 87.5 | 53.5 | 41.2 | 85.4 | 48.2 |
| HED | Proposed | 0.5 | 52.2 | 86.8 | 60.2 | 42.7 | 85.2 | 50.1 |
| HED* | Proposed | 0.5 | 32.4 | 87.0 | 37.3 | 44.5 | 85.2 | 52.3 |
| BDCN | Proposed | 0.5 | 51.4 | 86.5 | 59.5 | 43.4 | 85.2 | 50.9 |
| BDCN* | Proposed | 0.5 | 55.7 | 87.0 | 64.0 | 41.4 | 86.1 | 48.1 |

TABLE 3.2: COCO Panoptic scores on validation and test set for the training configuration study, using the Meyer Watershed (MWS) for CSE. The following parameters are static and their respective columns are hidden: the loss function is the binary cross entropy, the best DEF is selected using the protocol of [2], no augmentation is performed. For the architectures, * indicate pre-trained variants.

| DEF Archi. | Training config. | CSE Param. | | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | Proposed | 50.0 | 8.0 | 59.8 | 87.7 | 68.2 | 46.7 | 86.9 | 53.7 |
| HED | Proposed | 400.0 | 10.0 | 47.5 | 86.8 | 54.7 | 41.0 | 85.2 | 48.1 |
| HED* | Proposed | 400.0 | 10.0 | 51.5 | 87.5 | 58.9 | 43.9 | 86.2 | 50.9 |
| BDCN | Proposed | 400.0 | 10.0 | 48.9 | 86.9 | 56.3 | 41.3 | 85.8 | 48.1 |
| BDCN* | Proposed | 400.0 | 10.0 | 54.7 | 88.6 | 61.7 | 46.4 | 87.1 | 53.3 |

Based on this procedure, we report in Table 3.1 and Table 3.2 results for the following Deep Edge Filters: U-Net, HED and BDCN. For HED and BDCN, we report whether we used **pre-trained weights** as weight initialization before fine-tuning, or trained the network **from scratch**.

### 3.1.5 Numerical results and analysis

These results allow to draw the following conclusions.

- For both HED and BDCN architectures, we demonstrate the superiority of the pre-trained networks which always exhibit a better performance than when trained from scratch.
- The U-Net architecture exhibits much higher generalization performance, and achieves the best overall performance on the test set with a *COCO-PQ* score of 46.7%.

#### 3.1.5.1 Results of joint optimization

To enable the joint optimization of both stages, we propose to run a parameter selection (using a grid search) for the CSE stage, for each epoch of the training of each Deep Edge Filter. At the end of each epoch $i$ of the training

TABLE 3.3: COCO Panoptic scores on validation and test set for the joint optimization study, using the Meyer Watershed (MWS) for CSE and Joint Optimization (JO) for DEF selection. The following parameters are static and their respective columns are hidden: we use our proposed training configuration, the loss function is the binary cross entropy, no augmentation is performed. For the architectures, * indicate pre-trained variants.

| DEF Archi. | CSE Param. | | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|
| | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | 50.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| HED | 400.0 | 10.0 | 47.6 | 86.8 | 54.9 | 40.8 | 85.0 | 47.9 |
| HED* | 400.0 | 10.0 | 51.8 | 87.5 | 59.2 | 43.7 | 86.2 | 50.7 |
| BDCN | 400.0 | 10.0 | 49.1 | 86.9 | 56.5 | 41.1 | 86.0 | 47.8 |
| BDCN* | 400.0 | 9.0 | 55.0 | 88.5 | 62.1 | 47.0 | 87.3 | 53.8 |

of the DEF, generate the set of Edge Probability Maps $EPM_i$ for the validation set using $DEF_i$, then, using a grid-search, select the parameters for the CSE stage which leads to the best performance (based on the *COCO-PQ* score) on the validation set. The overall, final performance is reported on the test set (which was never used in any part of the joint optimization).

$$\begin{aligned} EPM_i &= DEF_i(\text{val. set}) \\ shapes_{i,\theta} &= CSE_\theta(EPM_i) \\ \{DEF, CSE\}_{best} &= argmin_{i,\theta}(PQ(shapes_{i,\theta})) \end{aligned}$$

We compare the following segmentation pipelines:

- **Finding best DEF parameters by using connected components labelling for Meyer watershed** The best DEF parameters (epoch) are selected based on the *COCO-PQ* score obtained on the validation set, using a simplified CSE stage: thresholding the EPM with a fixed value of 0.5, then extracting the shapes using CC labelling. The best CSE parameters (for the watershed extractor) are then computed for one DEF model only. This corresponds to the **Best Meyer Watershed ("best MWS")**.
- **Joint Optimization** This variant tests all possible combinations of DEF and CSE configurations for each epoch, reaching the best possible combination of DEF and CSE systems.

For the DEF stage, the set of possible parameters is defined as the different model trainings obtained at each epoch, and for the CSE stage with area filtering and dynamic filtering for the watershed stage.

#### 3.1.5.2 Results for the U-Net, HED and BDCN networks

Results are reported in Table 3.3, for the same U-Net, HED and BDCN networks as previously, and show that the systematic superiority of the joint optimization strategy over the baseline approach on the validation set (enforced by the selection protocol), **does not always guarantee to reach the best performance on the test set for the HED network and the BDCN network trained from scratch.** Figure 3.6 provides qualitative samples of the EPM generated by the predicted edges of HED and BDCN networks with

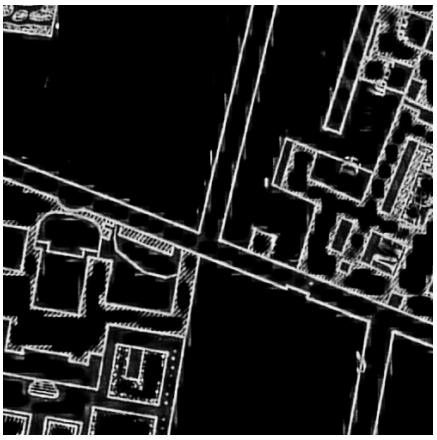


FIGURE 3.2: HED

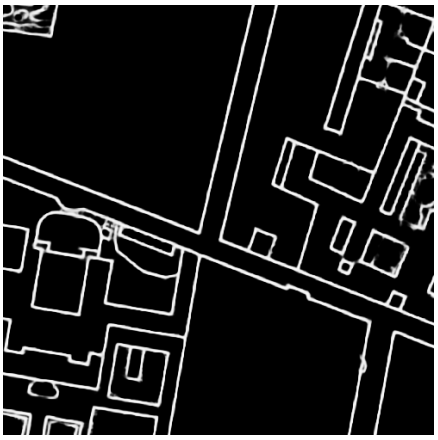


FIGURE 3.3: HED*

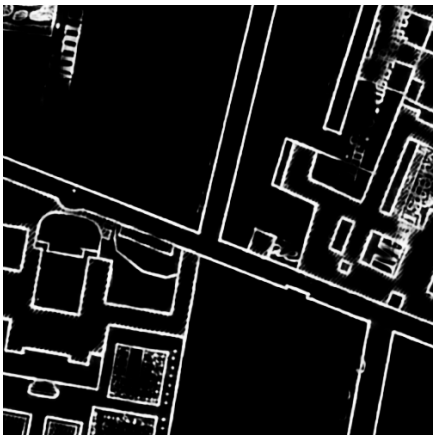


FIGURE 3.4: BDCN

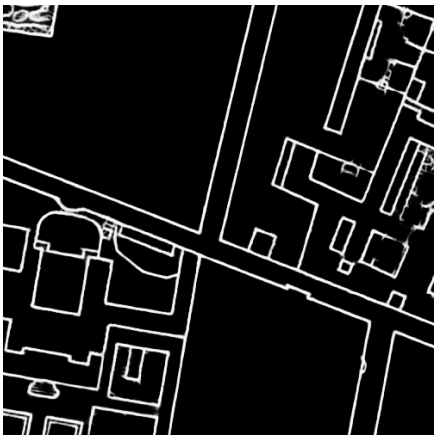


FIGURE 3.5: BDCN*

FIGURE 3.6: Predicted EPMs (Size: 500 px×500 px) with U-Net, HED and BDCN with (*) or without pre-trained weights. Pre-trained model produces less noise on the edges.

and without using pre-trained weights. **The networks of HED and BDCN with pretrained weights produce less noise on the edges images.**

## 3.2 Transformer architectures

### 3.2.1 Motivation

To detect the linear structure from images with topological properties, CNN architectures can achieve satisfactory results with different topological-based losses. However, they suffer from the limited range of their receptive field, as well as the discontinuity of their feature maps which may lead to topological inconsistencies in the predictions. Transformer architectures [115], applied to computer vision tasks, can address these issues thanks to their larger receptive field and longer pixel dependencies, and we propose to consider the two following architectures: Vision Image Transformer (ViT) [116] and Pyramid Vision Transformer (PVT) [117].

### 3.2.2 Methods

**Basic building blocks of encoder in transformer architectures** The basic building blocks of encoder in transformer module which uses multi-head self-attention $SA$ can be calculated through Query (Q), Key (K) and Value (V) [115] and dot products of the query with all keys (will be scaled by $\sqrt{D_k}$, the dimension of keys):

$$SA(z) = softmax(Q \cdot K^T / \sqrt{D_k}) \cdot V. \tag{3.11}$$

**Multi-head attention** is used for merging information from different subspaces and positions:

$$MultiHead(Q, K, V) = Concat(head_1, ...head_h)W^O, \tag{3.12}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V). \tag{3.13}$$

**Positional encoding** The input of transformer architectures are flatten image vectors as well as the architectures have no recurrence and convolutional operation to keep the sequence order (in NLP tasks) or relatively spatial position (in image related task). So positional encodings are designed by Vaswani et al. [115] to tackle this issue. The positional encodings ($PE$) is added to the input image embeddings with *sine* and *cosine* functions, where $pos$ is the position and $i$ is its dimension, $d_{model}$ is the dimension of the model:

$$PE_{pos,2i} = sin(pos/10000^{2i/d_{model}}), \tag{3.14}$$

$$PE_{pos,2i+1} = cos(pos/10000^{2i/d_{model}}). \tag{3.15}$$

**Vision Image Transformer (ViT). ([116])** Inspired by the success of transformers architectures ([115]) in the field of natural language processing by Dosovitskiy et al. [116], proposed to adapt these architectures to computer vision, introduced the Vision Image Transformer (ViT). ViT architectures outperform CNN architectures in many computer vision task (image classifications and segmentations) due to its self-attention mechanism which integrate global context of images which is crucial for detecting long range pixel dependencies and consistent shape patterns in historical maps. Recently, Strudel et

TABLE 3.4: COCO Panoptic scores on validation and test set for transformer architectures. The following parameters are static and their respective columns are hidden: we use the Meyer Watershed (MWS) for CSE and Joint Optimization (JO) for DEF selection, we use our proposed training configuration, the loss function is the binary cross entropy, no augmentation is performed. For the architectures, * indicate pre-trained variants.

| DEF | CSE | | Evaluation | | | | | |
|---|---|---|---|---|---|---|---|---|
| Archi. | Param. | | Val. set | | | Test set | | |
| | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | 50.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| ViT* | 500.0 | 10.0 | 38.6 | 80.9 | 47.8 | 34.7 | 80.4 | 43.1 |
| PVT* | 400.0 | 9.0 | 45.7 | 85.4 | 53.5 | 36.6 | 83.0 | 44.2 |

al. [118] adapted ViT to semantic segmentation tasks by using two different decoder architectures. Although using the transformer architectures as network backbone achieved new state-of-art performance in many computer vision tasks, its features are extracted at single scale. Moreover, the computational and memory costs remains high for common input image sizes, and the output resolution depends on the size of the input *patches* (the visual equivalent of textual tokens in transformer), which can lead to blocky predictions.

**Pyramid Vision Transformer (PVT) ([117])** To tackle these two issues, Wang et al. [117] proposed another pure transformer-based backbone architecture named Pyramid Vision Transformer (PVT) that enables the network to learn different scales of features while significantly decreasing the number of parameters compared to traditional ViT architectures, leveraging the concept of feature pyramid proposed by Lin et al. [119]. According to recent publications from Guo et al. [120] and Tuli et al. [121], the combination of CNN and transformer architectures have become a promising trend in a wide range of computer vision tasks. These models reach the new state-of-the-art in a wide range of computer vision tasks.

### 3.2.3 Experimental settings

As training transformer architectures from scratch huge amount of training data (millions of images) which are not available in the case of historical map images, we use weights pre-trained on the Cityscape dataset [122] to initialize our network. We use the ADAMW optimizer for both transformers with a learning rate of $1 \cdot 10^{-5}$.

### 3.2.4 Experimental results and analysis

Results, summarized in Table 3.4, show that despite their larger receptive field, transformer architectures reach much lower validation and test scores in our experiments, compared to the traditional U-Net architecture. This low performance is caused by the fact that ViT has a low-resolution output, and


FIGURE 3.7: Input (MAP)

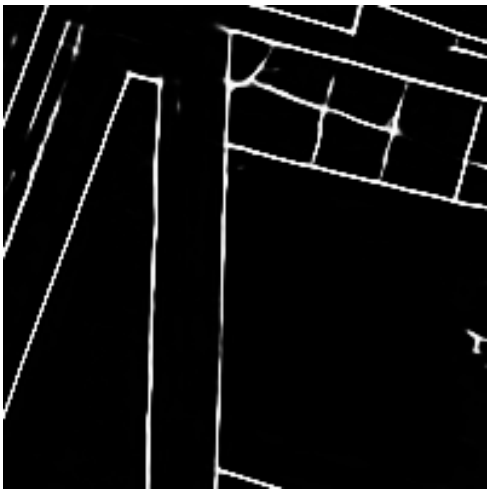


FIGURE 3.8: EPM (U-Net)

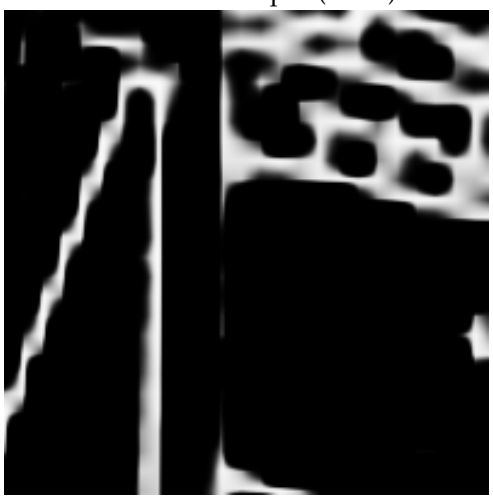


FIGURE 3.9: EPM (ViT)

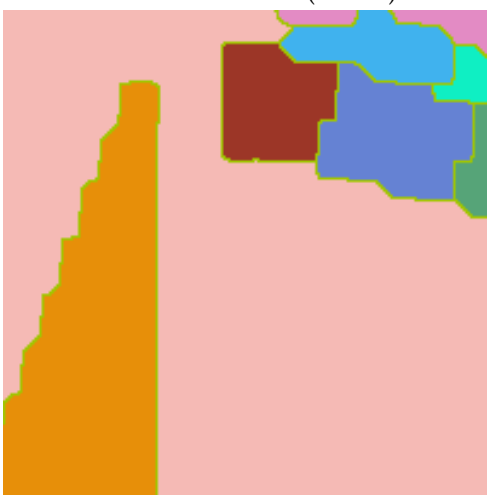


FIGURE 3.10: EPM (Pvt)

FIGURE 3.11: Comparison of the edge filtering produced by U-Net and ViT. ViT exhibits a *zigzag* effect, mainly because of the tokenization of the input image.

that ViT and PVT may require a much larger dataset for fine-tuning. However, the performance of these architectures is not always worse than the U-Net one, especially for larger shapes (log(area) > 15) where ViT outperforms U-Net as shown in Figure 3.12. It may indicate that transformer architectures can better preserve line consistency, compared to CNN architecture, thanks to their richer context. Some combination of these systems may be possible to obtain the best possible performance; keeping only smaller objects from U-Net and larger ones from ViT. Regarding PVT, its overall performance is not better compared to the conditions (U-Net) we tested it against. **Looking at the values of the best parameters for the area and dynamic filtering, we can see that both ViT and PVT models require stronger filtering compared to U-Net. This suggests that transformer-based models are suffering from false positive background noise in the predicted EPMs.**

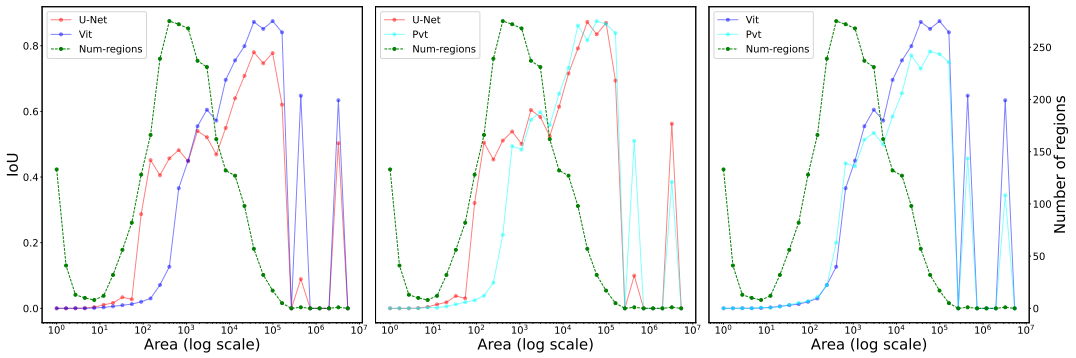



FIGURE 3.12: Shape statistics of convolutional-based (U-Net) and transformer-based (ViT and PVT) models. The figure represents the average IoU with steps of 0.5 in logarithmic scale. The green dash line corresponds to the distribution of number of regions with corresponding area values. ViT and PvT transformer-based architecture have better performance for detecting large objects, while U-Net better preserves the boundaries of small objects (leading to a superior IoU).

## 3.3 Deep watershed transform

### 3.3.1 Motivation

To tackle the key limitations of classical watershed techniques, i.e. noise sensitivity and difficulty to select filter parameters, Bai et al. [50] proposed a Deep Watershed Transform which learns the discrete watershed levels directly from multichannel images. However, learning watershed levels from original images is a difficulty task because of the limited receptive field of convolutional networks. To be able to learn the long-range dependencies between pixels and capture their level of inclusion (or distance to object boundaries), the authors introduced an intermediate step where a direction fields is learned, mimicking the *water flow* of the watershed segmentation algorithm. This integrated architecture is supposed to exhibit good images filtering properties, which can prevent the over-segmentation issues of traditional watershed approaches, while avoiding the need for extra prior knowledge of the filtering attributes and their optimal values.

### 3.3.2 Method

**Mathematical notations of deep distance transform** The traditional distance transformation function $D_t : \hat{I} \rightarrow T$ is to map each point of binary image $\hat{I} \in \mathbb{Z}^{(H,W)}$ to its closest boundary, resulting in a transformation image $\mathcal{T} \in \mathbb{R}^{(H,W,2)}$. Instead of computing the distance transformation over the binary image, the deep distance transform directly transforms the multichannel image (RGB) $I \in \mathbb{R}^{(H,W,3)}$ into the transformation image $\mathcal{T} \in \mathbb{Z}^{(H,W)}$ by using a neural network as a non-linear transformation function. However, as mentioned by Bai et al. [50], learning the transformation through a neural network directly is a complex task since distance transform requires information from a global context, while existing CNN architectures have a limited receptive field to capture global information.

**Learning the direction field from image** To tackle this problem, Bai et al. [50] add an intermediate step by firstly learning the direction of the distance transform which is also the gradient of the distance transformation image. The direction of the distance transform of the ground truth image $v_{\text{gt}} \in \mathbb{R}^{(H,W)}$ is calculated through normalized image gradient of distance transform as follows:

$$\vec{v_{gt}} = \frac{\nabla \mathcal{T}_{gt}}{|\mathcal{T}_{gt}|}. \tag{3.16}$$

The angular loss $\mathcal{L}_{ang}$ is used to learn the ground truth direction through measuring the cosine similarity between vector of prediction $\vec{v_{pred}}$ and ground truth $\vec{v_{gt}}$ with parameter $w_p$ which is the weighting factor based on the area of instances $A_{obj}$:

$$l_{ang} = w_p||cos^{-1} < \vec{v_{pred}}, \vec{v_{gt}} > ||^2; w_p = \frac{1}{A_{obj}}. \tag{3.17}$$

**Learning the distance transform from direction field** Once the direction field of distance transform is learned, the next step is to learn the function $f : v \rightarrow \mathcal{T}$ which integrates the direction field and turns it into a distance field. To better learn the levels of distance transform, Bai et al. [50] quantized the distance transform into 16 inhomogeneous level sets by using a scaling factor $c_k$ which is the pre-defined weight for each level. Since the levels of distance transform are sorted in descending order, the parameter $c_k$ is used to emphasize and give stronger weight close to the boundary of instances instead of the center of the instances. In the end, a modified cross entropy loss $\mathcal{L}_{dt}$ is used to train the distance transform between quantized prediction $P_{ls}$ and ground truth $G_{ls}$ level sets, $K$ is the number of discrete watershed levels with common value of 16 and $w_p$ is the weighting factor to adjust the importance of the objects based on the size of the objects:

$$\mathcal{L}_{dt} = \sum_{k=1}^{K} w_p c_k \mathcal{L}_{ce}(P_{ls}, G_{ls}). \tag{3.18}$$

**Joint learning of direction field and distance transform** The architecture proposed by Bai et al. [50] is replaced by two U-Net [47] networks for maintaining better spatial localization in the prediction during the training of the direction field (direction network) and distance transform (deep watershed). We modified the original training strategy to better fit the historical map data. Instead of training direction field and watershed level separately, we use weight trained from direction field and jointly trained for discrete watershed levels. The resulting training process is composed of the two following steps:

1. Train the direction field network/predictor in order to obtain pre-trained weights for the second step;
2. Jointly train the distance transform network, using the pre-trained weights of the direction network, with the following global loss.

$$\mathcal{L}_{total} = \mathcal{L}_{ang} + \mathcal{L}_{dt}. \tag{3.19}$$

TABLE 3.5: COCO Panoptic scores on validation and test set for U-Net+Meyer Watershed vs Deep Watershed. We use the MWS as a post-processing without filtering on Deep Watershed outputs to thin the prediction edges. The following parameters are static and their respective columns are hidden: no augmentation is performed, and DEF selection is performed with Joint Optimization (JO).

| DEF Archi. | CSE Method | Param. | | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | MWS | 50.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| DWS | MWS | 0.0 | 0.0 | 54.0 | 87.4 | 61.7 | 28.5 | 84.9 | 33.5 |

### 3.3.3 Experimental settings

The training strategy of deep watershed is to train direction field and discrete watershed level separately, then perform fine-tuning in an end-to-end style. We trained the direction network using an ADAM optimizer with the initial learning rate of $1 \cdot 10^{-5}$, a momentum of 0.9 and a weight decay of $1 \cdot 10^{-5}$. Then end-to-end fine-tuning used the same settings except for a smaller learning rate of $1 \cdot 10^{-6}$. In order to ensure a fair comparison with other approaches, we generate object boundaries with the following process: we first perform the equivalent of a "watershed cut" by selecting the highest value on the learned watershed levels (this creates thick boundaries), then we perform an edge-thinning to recover thin, 1-pixel large object boundaries.

### 3.3.4 Numerical experiments and conclude remark

We compare in Table 3.5 the results of the Deep Watershed approach and of the leading approach, composed of a U-Net Deep Edge Filter combined with a Meyer Watershed for Closed Shape Extraction, trained using the joint optimization procedure. Despite encouraging performance on our validation set, the Deep Watershed fails to generalize on our test set, reaching much lower performance than our leading approach. **It is due to the fact that deep watershed learns an approximated function which transforms images into watershed levels without providing any topological guarantee (about closed shapes) in the final prediction of watershed levels due to limited spatial context.**

## 3.4 Data augmentations

### 3.4.1 Motivation

Though many data augmentation techniques were proposed for computer vision (we refer the reader to the work of [123] for an overview), [124] appropriately pointed out that not all of these transformations can be safely applied to historical map images. Indeed, while color transformation, noise and geometric transformations, to some extent, can preserve the original signal, augmentation techniques like feature space transformation, mixups, as well as strong geometric transformation, would break object boundaries and may

prevent the network from capturing local edge consistencies. Furthermore, some text and symbols may not appear in all orientations, and their symmetric counterparts may not exist. The effects of such augmentation are not well studied. We prefer to avoid them by restricting our study to a safe subset of image augmentation techniques, in order to mimic the variations from different scanning conditions of historical maps: contrast and color changes, and paper rotation and bumps. We therefore propose to consider the following augmentation techniques: **contrast stretch** and **geometric transformations**.

### 3.4.2 Methods

**Image Contrast** : The contrast and brightness adjustment can be controlled through $\alpha$ and $\beta$ parameters, respectively:

$$f(I) = \alpha * I + \beta, \alpha \in [a, b], \beta = 0. \tag{3.20}$$

When $0 < \alpha < 1$ is to decrease the contrast of the image and $\alpha > 1$ is to increase the contrast. The contrast value is picked in the value range $[a, b]$ with uniform distribution. In this thesis, we only apply contrast change, and do no consider brightness. Hence, $\beta = 0$.

**Geometric transformations** : Online data augmentation at training time can improve the generalization performance of the Deep Edge Filters. We consider the following geometric augmentation techniques: affine transformation, homography transformation (full perspective), and thin-plate splines (TPS) transformation. These geometric transformations can maintain the topology properties in historical map images shown in Figure 3.13.

**Affine Transformation:** Affine transformation is a category of linear image transformations which include rotation, translation, non-isotropic scaling and shearing. This transformation has 6 degrees of freedom which can be encoded in a $2 \times 3$ matrix $A$ with offset $x$ and $y$:

$$A = \begin{bmatrix} a1 & a2 & x \\ a3 & a4 & y \end{bmatrix} \tag{3.21}$$

The affine transformation from point $p_a$ to point $p_b$ can be calculated through:

$$p_a = \begin{bmatrix} a1 & a2 \\ a3 & a4 \end{bmatrix} p_b + \begin{bmatrix} trans_x \\ trans_y \end{bmatrix} \tag{3.22}$$

The matrix $A$ is defined according to the Singular Value Decomposition (SVD) of the affine transformation through rotation angle $\theta$, shear angle $\sigma$, anisotropic scaling factor $\lambda$ and transformation vector $x$ and $y$ which equals to:

$$A = R(\theta)R(-\sigma)diag(\lambda_1, \lambda_2)R(\sigma). \tag{3.23}$$

**Homography transformation:** Homography transformation is another linear transformation which can be through the transformation applied to the four corners of images. The homography transformation of four coordinates can

be generated through a controllable offset $\delta_{xi}$ and $\delta_{yi}$, where the four point homography matrix can be calculated as:

$$H_{4p} = \begin{bmatrix} x1 + \delta_{x1} & y1 + \delta_{y1} \\ x2 + \delta_{x2} & y2 + \delta_{y2} \\ x3 + \delta_{x3} & y3 + \delta_{y3} \\ x4 + \delta_{x4} & y4 + \delta_{y4} \end{bmatrix}. \tag{3.24}$$

Then $4 \times 2$ homography transformation matrix can be transformed into $3 \times 3$ homography transformation matrix from every corresponding source points (x, y) into target points $(x^{'}, y^{'})$ through Linear direct transform (LDT) where $Ah_{3\times3} = 0, h_{3\times3} = \{h_1, h_2, ...h_9\}$:

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ h9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.25}$$

where this matrix can be formulated from source points to target points by elements in the homography matrix $H = \{h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}\}$:

$$x = \frac{h_{11}x' + h_{12}y' + h_{13}}{h_{31}x' + h_{32}y' + h_{33}}, y = \frac{h_{21}x' + h_{22}y' + h_{23}}{h_{31}x' + h_{32}y' + h_{33}}. \tag{3.26}$$

In the end, we can get the $3 \times 3$ homography transformation matrix ($h_{3\times3}$) through SVD.

**Thin-plate spline transformation (TPS):** TPS is a transformation which transform set of source points $p_s = \{p_{s1}, p_{s2}, ...p_{si}\}$ into target points $p_t = \{p_{t1}, p_{t2}, ...p_{ti}\}$. The source points are selected through a $k \times k$ uniform grid in the source image, while target points can be generated through random value of offset $\sigma$ with uniform distribution so that:

$$p_t = p_s + \sigma \tag{3.27}$$

The TPS transformation is estimated through:

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{N} w_i U(||(x_i, y_i), (x, y)||), \tag{3.28}$$

and should satisfy:

$$\sum_{i=1}^{N} w_i = 0, \tag{3.29}$$

and

$$\sum_{i=1}^{N} w_i x_i = \sum_{i=1}^{N} w_i y_i = 0. \tag{3.30}$$

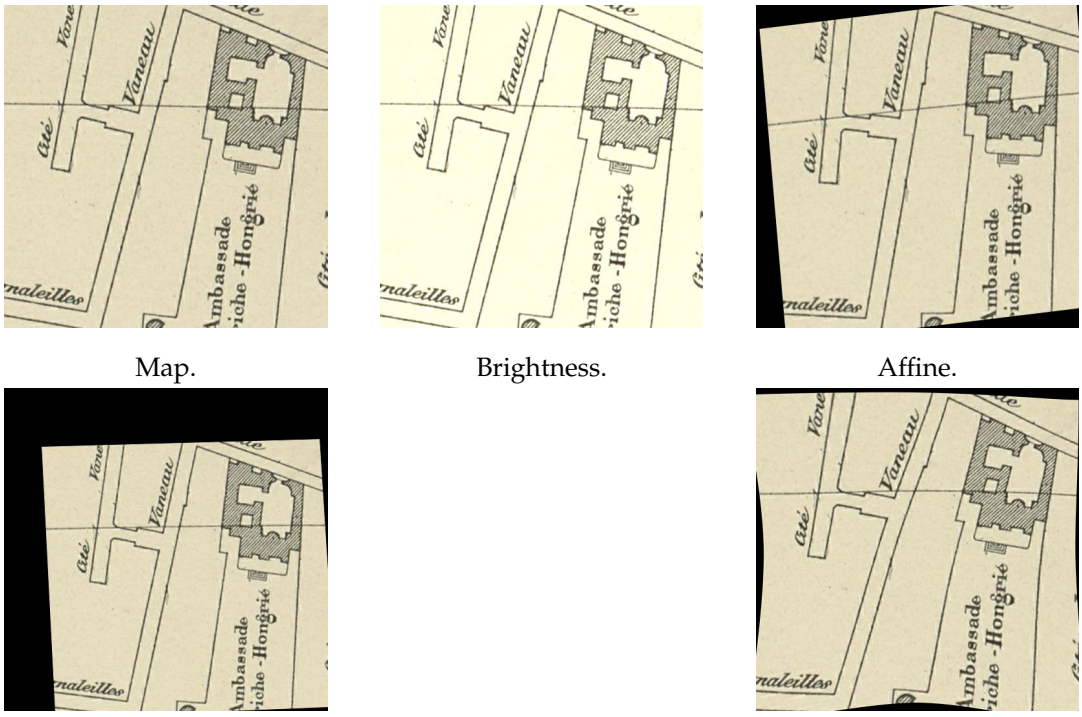



FIGURE 3.13: Image examples with four different augmentation methods.

The function $U(r) = r^2 log(r)$ is a TPS kernel, where points close to the center will return a higher value.

The TPS can be calculated through a linear system:

$$\begin{bmatrix} K & P \\ p^T & O \end{bmatrix} \times \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix} \tag{3.31}$$

where $k_{ij} = U(||(x_i^{'}, y_i^{'}), (x_j, y_j)||)$ is the distance between two points, P is the coordinates, O is $3 \times 3$ matrix of zeros; we can compute $a$ and $w$ by solving the linear equation.

### 3.4.3 Experimental settings

We study the effect of separate and combined geometric and contrast augmentation.

Furthermore, we choose contrast range of $\alpha$ from 0.8 to 1.2 (to prevent unrealistic dark or bright cases which will not occur in scanned maps) with uniform distribution. For the setting of geometric transformation, we change settings in the work of [125] to prevent large geometric transform. For affine transformation, we choose rotation angle $\theta \sim \mathcal{U}(-10°, 10°)$[3], translation $t_x, t_y \sim \mathcal{U}(-0.1, 0.1)$, anisotropic scaling factor $\lambda_1, \lambda_2 \sim \mathcal{U}(0.9, 1.1)$ and shear angle $\phi \sim \mathcal{U}(-10°, 10°)$. For homography transformation, we add a random translation $\sigma_x, \sigma_y \sim \mathcal{U}(-0.1, 0.1)$ to four control points in the corner. For TPS transformation, we use 9 points with random translation $\delta_x, \delta_y \sim \mathcal{U}(-0.1, 0.1)$ to prevent strong deformations.

[3] $\mathcal{U}$ is the uniform distribution.

TABLE 3.6: COCO Panoptic scores on validation and test set for the augmentation study. The following parameters are static and their respective columns are hidden: model architecture is U-Net (trained from scratch), the loss function is the binary cross entropy, the best DEF is selected using joint optimization, and Meyer Watershed (MWS) is used for CSE.

| DEF Augmentation | | CSE Param. | | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|---|
| Contrast streching | Geometric transform | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| no | none | 50.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| yes | none | 100.0 | 6.0 | 57.3 | 88.2 | 65.0 | 47.2 | 86.7 | 54.4 |
| no | Aff. | 100.0 | 9.0 | 61.0 | 87.9 | 69.4 | 47.7 | 86.5 | 55.1 |
| yes | Aff. | 100.0 | 10.0 | 61.1 | 88.1 | 69.4 | 50.7 | 86.8 | 58.5 |
| no | Hom. | 200.0 | 10.0 | 58.4 | 87.9 | 66.5 | 49.6 | 86.9 | 57.1 |
| yes | Hom. | 200.0 | 10.0 | 59.5 | 88.2 | 67.4 | 50.4 | 86.7 | 58.2 |
| no | TPS | 100.0 | 10.0 | 59.8 | 88.3 | 67.8 | 47.9 | 86.9 | 55.1 |
| yes | TPS | 100.0 | 7.0 | 59.6 | 88.2 | 67.5 | 51.1 | 86.8 | 58.8 |

### 3.4.4 Numerical experiments and analysis

Table 3.6 reports the results for the various combinations of contrast and geometric augmentations. The combination of the contrast and affine transformation have the highest improvement leading to a *COCO-PQ* score of 0.79 on the val. set compare to U-Net baseline. All methods lead to improved performance on the test set. The combined use of contrast and TPS augmentations leads to the best *COCO-PQ* score, reaching 51.1%, which represents between 0.7 and 4.6 points of improvement over the other combinations and all variants lead to similar results, TPS being slightly superior to the others in our experiments. As the results, the contrast+TPS achieves the highest *COCO-PQ* score 51.08 in our dataset.

**To conclude, data augmentations are proved to an effective training tool which improves the generalization performance of a network trained for historical map vectorization.**

This chapter describe and compare different techniques for filtering edges from historical maps and subsequently extracting closed shapes. Firstly, we observed that multiscale neural network architectures have worse performance compare to baseline U-Net due to overfitting because we have little training data in *Paris Atlas Municipal*. Moreover, multiscale neural network architectures using pre-trained weights exhibit less noise on the predicted edges images compared to the ones not using pre-trained weights. Secondly, transformer architectures achieve good performances on recovering objects with larger sizes compare to our baseline CNN-based U-Net architecture while the latter exhibits a better performance over a large scale of shapes, globally resulting in superior performance. Thirdly, we reimplemented a deep watershed transform to extract watershed lines that are similar to an edge extraction process. Yet, the deep watershed transform has worse generalizability to test data, and we also show that this architecture cannot offer any topological guarantee about the closed shapes formed by its watershed lines. Lastly, we proved that data augmentations techniques based on contrast stretch and geometric transformations significantly improve the generalizability of our proposed pipeline when facing unseen historical map images.

# Chapter 4

# Topology-aware loss functions

To answer the third research question of **how to guarantee the topological properties in the prediction**, we overcome this challenge by introducing new loss functions which aim to preserve the topological properties of the predicted edge images. In spite of the fact that deep edge filtering with joint optimization strategy achieves satisfactory results for extracting closed shapes from historical maps, the problem of missed detection of critical pixels in the object boundaries can lead to the failure of extracting closed shapes from edge images. This is also called a **topological failure** because the edges predicted by the CNN do not maintain topology properties of the original shapes.
We divided this chapter into two sections. Section 4.1 details the motivations for topological loss functions. Afterwards, the mechanism of topological loss functions is introduced and four topology-oriented loss function (two existing SOTA and our two novel topology-oriented loss functions) are tested to maintain the topological correctness in the predicted edge output. Section 4.2 explores another direction to solve topological correctness by using information on local pixel connectivity to enhance topological correctness in the final predicted edges.
This chapter is an extended and adapted version of the contents of our previously published [52] and submitted [3, 126] publications. Our two novel topology-preserving loss functions are initially tested in the neuron segmentation and then adapted into historical map segmentation.

## 4.1 Introduction to topology-awareness loss functions

### 4.1.1 Motivation

When evaluating the topological quality of the extracted map objects, the pixel-level performance of boundary detection does not always correlate well with shape-level performance. Indeed, a missed detection for a single critical pixel on the boundary of an object may create some leakage, and lead to a topological error at the shape level, while the error at pixel level remains negligible shown in Figure 4.1. Although CNN perform well at filtering images, such networks are trained to minimize pixel-based losses: the limited range of pixel dependencies and restricted spatial context in CNNs do not permit to guarantee the expected topological properties in the predictions (which can leads to a significant drop in topology-level performance over the course of training). Recent approaches proposed to improve the topology performance

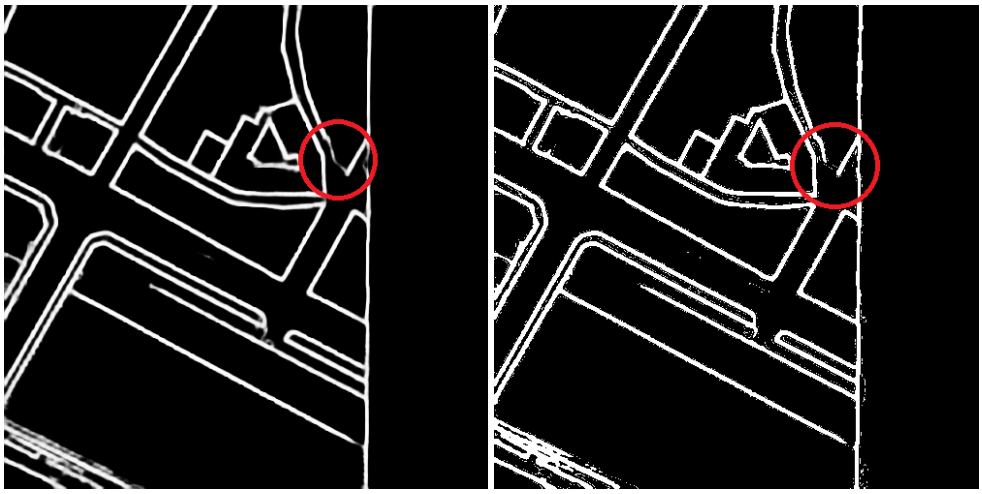


FIGURE 4.1: Connected component extraction (**CC labelling**): we threshold the edge probability map and run connected component analysis to get instances from the images. The red circle shows a weak boundary that disappears after threshold.

by adding a topology-oriented loss to approximate the correct topology in the predictions. We propose to consider the following three categories of mainstream methods to preserve topology in edge prediction: architecture-based, persistent-homology-based and boundary-based loss functions.

As previously mentioned, several loss functions were designed in the literature [127, 128] to better comply with the topological requirements of several tasks. **We also contributed new loss functions BALoss [52] and Pathloss [126] to preserve topology properties in image segmentation tasks.**

### 4.1.2 Related work

Topology-preserving image segmentation methods mainly consist of indirect and direct ways. The **indirect** way uses an iteration framework to gradually refine the elongated structure in the predicted output, while latter one directly uses strong topology priors and constraints to improve topological results in the predictions. The worth noted indirect way stems from Mosinska et al. [129], where a topology-awareness loss is proposed by measuring the similarities of features in VGG-19 [87] between ground truths and predictions. However, this work assumed that the line continuity of trained filters in the VGG-19 can maintain the topology properties in the predicted probability map, albeit it does not have any guarantee of high dimensional topology such as loops (or closed boundaries). Therefore, the detected objects might not be topologically correct. Different from the Mosinka et al. [129] where the predicted likelihood is joined to the image input for refinement purposes, the work in Iternet [130], Multi-stage multi-recursive-input networks [131] and Flood-Filling Networks [132] stack several networks to gradually refine the likelihood in each stage of the networks. Still, those methods suffer the similar issue.

The **direct** topology-preserving methods focus on using persistent homology [133] to measure the topology features of image predictions [133]. Chen

| Model | M | TA | Bo-Pi | Cri-P | Cri-P-F |
|---|---|---|---|---|---|
| Iternet[130] | IN | ✗ | ✗ | ✗ | – |
| Mosin[127] | IN+Loss | ✗ | ✗ | ✗ | – |
| BL[139] | Loss | ✗ | ✓ | ✗ | – |
| clDice[138] | Loss | ✓ | ✓ | ✗ | – |
| BALoss[52] | Loss | ✓ | ✓ | ✗ | – |
| Topoloss[128] | Loss | ✓ | ✓ | ✓ | ✗ |
| Pathloss[126] | Loss | ✓ | ✓ | ✓ | ✓ |

TABLE 4.1: A comparison between our method and state-of-the-art methods: **M**: methods use IN (iteration based network) or Loss (topology-preserving loss); **TA**: The topology aware in the training process of the method; **Cri-P**: methods use information of critical points or not; **Bo-Pi**: methods focus on boundary pixels; **Cri-P-F**: methods do not require filtering the critical points. In this thesis, we aim to close more shapes based on the condition of the first dimensional topology and focused on fixing the critical points in the boundary pixels.

et al. [134] introduced the topology priors and integrate them into Conditional Random Fields (CRF) image models to improve the image segmentation task. Persistent homology has been widely used as a topology feature. Recently, persistent homology has been re-designed and has proved its differentiable properties. Then it can be used as a topology-preserving loss function and can be applied to any end-to-end deep neural networks [128, 135–137]. However, the limitations of using persistent homology in deep image segmentation tasks are its memory consumption, convergence speed, and sometimes the training difficulty when combined with other loss functions. Shit et al. [138] proposed a new topology-aware metrics to measure the similarity between images based on morphological skeletons so called centerline dice (clDice). However, clDice still yield missing centerlines on the leakage locations in the prediction image.

### 4.1.3 Types of topology-aware loss functions

**VGG architecture based loss function (MosinLoss)** This early design of a topology oriented loss leverages elongation properties of the features of the VGG-19 architecture. To preserve line consistency in the trained features, the differences between the VGG-19 features of the predicted and ground truth images are calculated for each layer to form a global loss. Although the **MOSIN** loss function can improve the pixel consistency in the output (e.g. road detection), it does not directly improve the performance of detecting the closed instances.

Denote input historical map image as $I \in \mathbb{R}^{(H,W,3)}$, where its correspond binary ground truth labelling $y \in 0,1^{H,W}$ represents the boundary of the objects. The MOSIN architecture features two stages. The first one is a U-Net which assigns to each pixel an edge probability $\hat{y}$ as $f : I \to \hat{y}; \hat{y} \in [0,1]^{(H,W)}$. Then, the BCE loss is used to measure the difference between prediction $\hat{y}$ and the ground truth $y$ as:

$$\mathcal{L}_{bce}(x, y, w) = -\sum_{i}^{N}(1 - y_i) \cdot \log(1 - f_i(x, w)) + y_i \cdot \log f_i(x, w). \quad (4.1)$$

The second stage is to measure the topology difference between prediction $\hat{y}$ and the ground truth $y$ through measuring the intermediate feature differences as:

$$\mathcal{L}_{vgg}(x, y, w) = -\sum_{n=i}^{N} \frac{1}{M_n \cdot W_n \cdot H_n} \sum_{m=1}^{M_n} ||l_n^m - l_n^m(f(x, w))||_2^2, \tag{4.2}$$

where $l_n^m$ is the m feature map in the $n^{th}$ layer, $M_n$ is the number of channels and $W_n, H_n$ are the width and height in $n^{th}$ layer, respectively. The total **MOSIN** loss is the sum of binary cross entropy loss $\mathcal{L}_{bce}(x, y, w)$ with the topology-oriented loss $\mathcal{L}_{vgg}(x, y, w)$ with weighted factor $\lambda$:

$$\mathcal{L}_{total} = \mathcal{L}_{bce}(x, y, w) + \lambda \mathcal{L}_{vgg}(x, y, w). \tag{4.3}$$

### 4.1.4 Persistent-Homology-Based loss function (TopoLoss)

This loss is based on the theory of persistent homology which enables to identify the critical failure points of the predicted objects boundaries. It takes into account the width and depth of the gaps in a differentiable loss function which will encourage the network to recover lost boundary components. However, this loss function is highly sensitive to noisy images. Denote a historical map image $I \in \mathbb{R}^{(H,W,3)}$ and a binary ground truth labelling $y \in \{0, 1\}^{(H,W)}$, the likelihood image $\hat{y}$ is predicted by a deep neural network. The **Topoloss** the topological difference between the likelihood $\hat{y}$ and ground truth image $y$ through **persistence diagrams** $Dgm$ which capture all possible topological structures both of likelihood $Dgm(\hat{y})$ and ground truth $Dgm(y)$. The goal is to minimize the *Wasserstein distance* between $i^{th}$ persistent dots $p_i$ of likelihood and ground truth. Each persistent dot encodes the birth and death information of a topology structure.

$$\min Dgm(\hat{y}, y) = \min \sum_{i=1}^{N} ||p_i(\hat{y}) - p_i(y)||^2. \tag{4.4}$$

Every persistent dot $p$ corresponds to a topological structure which borns at a specific threshold $a$ (y-axis in persistent diagram as birth time) and dead at threshold $b$ (x-axis in persistent diagram as death time).

$$\min Dgm(\hat{y}, y) = \sum [birth(\hat{y}) - birth(y)]^2 + [death(\hat{y}) - death(y)]^2. \tag{4.5}$$

The Topoloss $\mathcal{L}_{topo}$ is trained combined with the binary cross entropy loss $\mathcal{L}_{bce}$ to preserve topology structure in the likelihood image with balanced weight $\lambda$:

$$\mathcal{L}_{total} = \mathcal{L}_{bce} + \lambda \cdot \mathcal{L}_{topo}. \tag{4.6}$$

#### Boundary-awareness loss function (BALoss)

**Overview of the method** Compared to Mosin [127] and Topoloss [128], we proposed BALoss [52] to preserve topology in deep image segmentation task. Our method is a seeded two-step approach (Figure 4.2), in which the object

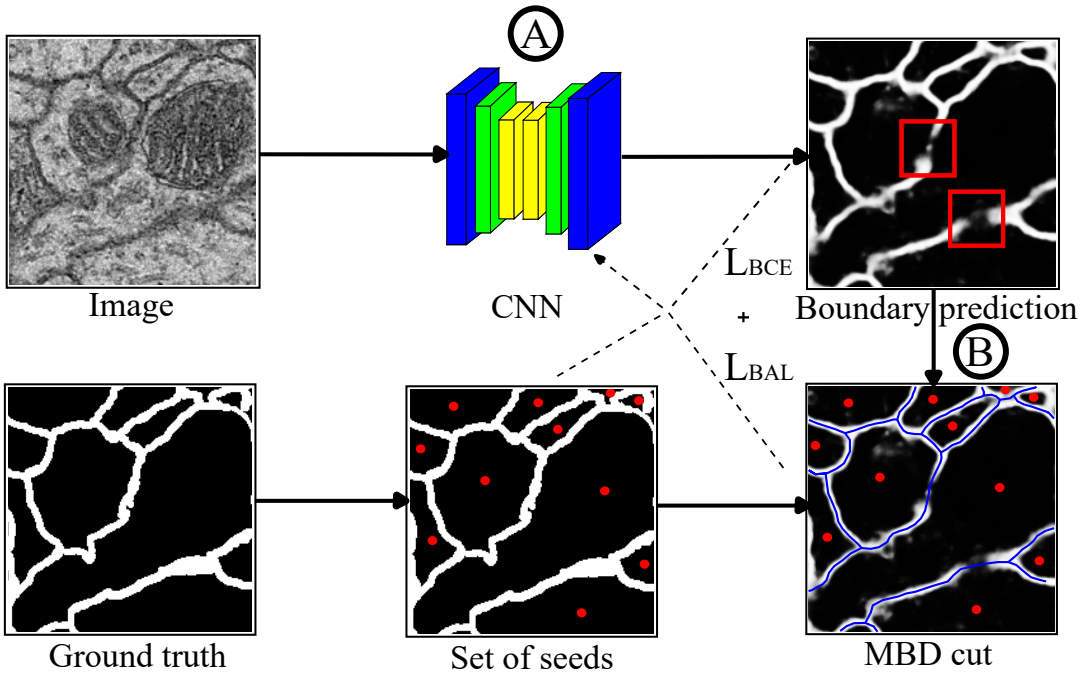
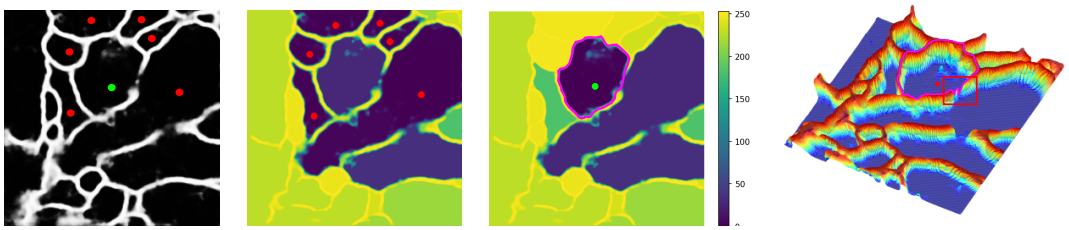



FIGURE 4.2: Overview of our approach. A set of seed points (red points) are placed from the ground truth. A boundary prediction image is the output of the CNN network (step A). Leakage positions are highlighted inside the red squares. We locate the boundaries of regions by using the MBD cut algorithm (blue lines), then compute the *BALoss* function (step B).

boundaries can be extracted during training and eventually encourage the network to focus on boundary structures. To train the neural network, we are using the combination of the binary cross-entropy (BCE) and the *BALoss*. First, a convolutional neural network predicts a boundary likelihood map from the original image. Secondly, the *BALoss* is computed from the extracted boundaries of regions by using with a seed node provided inside each connected component of the training and validation ground truth label images. These seeds are the points that have the maximum Euclidean distance value w.r.t the boundary of each connected component. We respectively consider the inside/outside seeds of the region as foreground/background seeds, then compute a foreground/background distance map on the prediction image thanks to the MBD distance. Through comparing the values on these two distance maps, the boundaries of the regions are identified and a newly designed *BALoss* is derived by computing the pixel-wise error between the extracted boundaries and the ground truth image. The advantage of our *BALoss* function is that it helps the network to focus on important missing pixels on each region (the ones which cause breaks in the boundaries), thus preserving the topological structure of the image, without any class information.

**MBD-cut** The boundaries of the regions are extracted using a method so-called *MBD-cut*. This preserves the topology of the image and measures the quality of the segmentation. The MBD distance has a low sensitivity to noise, blur, and seed positions [51, 140]. In our approach, we push the




(a) Likelihood image from the CNN network (b) Background MBD distance map (c) Foreground MBD distance map (d) The density map

FIGURE 4.3: (a) Prediction image from CNN network (green/red points are foreground/background seeds). (b) Background MBD distance map from the seed of neighbor regions. (c) Foreground MBD distance map and the MBD cut (pink contour). (d) The density map represents the values of the prediction image. Z-axis represents the value of pixels the prediction image. Leakage position is shown inside the highlight square.

idea of seeded graph-cut based segmentation further by using the high-level features computed from a CNN.

We denote the likelihood prediction map as $u$, and the ground truth label image as $S$. We respectively consider the seed point $x_i$ inside the region $S_i$ as the *foreground* seed and all the seeds $x_j$ of the neighbor connected components $S_j$ as the *background* seeds. We respectively compute the MBD distance map from the background/foreground seeds by using the front propagation approach [141]. The idea behind is that we consider the seed pixels as sources of water, the water can flow from source pixels to other pixels with a different priority which is determined by the MBD cost. We use the priority queue to keep track of the order of pixels to propagate the distance value to every pixel in the image (lower cost means earlier flow). The algorithm stops when all pixels in the image were scrutinized.

The complexity of our front propagation algorithm is $\mathcal{O}(n \log n)$, where $n$ is the number of pixels in the image. Our method is efficiently computed, so that we can get the MBD distance map immediately from the set of the foreground and background seed points. The background/foreground MBD distance maps are illustrated in Figure 4.3(b) and Figure 4.3(c). After computing these maps, we are able to label the pixels as *background* or *foreground* based on their distances to the seed set. We also recover the boundary of the region $C_i$ (pink contour in Figure 4.3(d)). The segmented boundary is pivotal in computing the Boundary-Aware loss function.

**Training using the Boundary-Aware loss** Most CNN-based segmentation networks use the binary cross-entropy (BCE) as a loss function. It is defined as a measure of the difference between two probability distributions for a given random variable or set of events [142]. BCE is known to be adapted to measure boundary shifts [143, 144]. Here, we present a new *BAL* function to enhance segmentation results and detail how to implement it. The *BAL* function is computed from the values of the binary extracted contour $C_i$ of the region $S_i$ using the *MBD-cut*. The total loss is the sum of the *BALoss* for

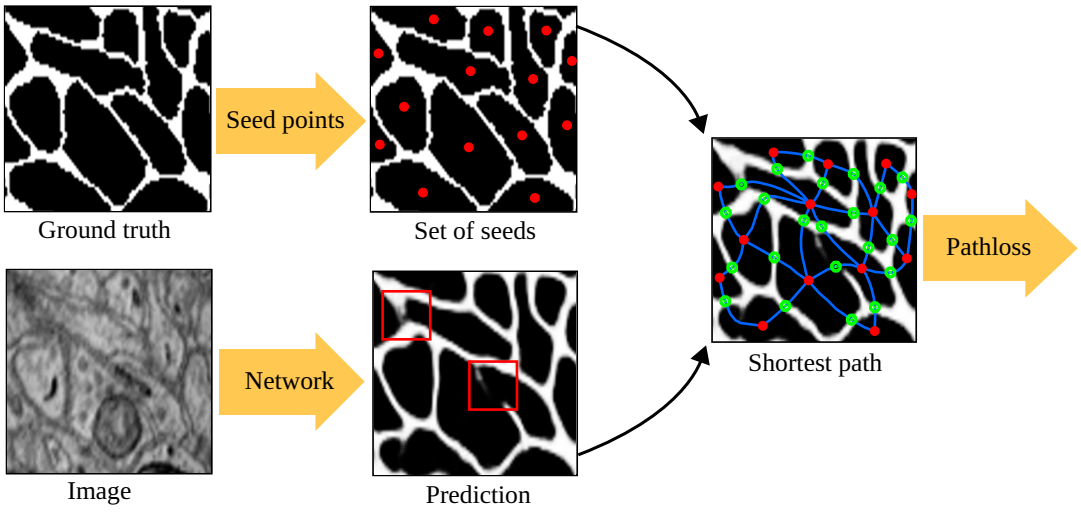



FIGURE 4.4: The pipeline of our method. We first generate the set of seeds (red dots) which correspond to each region in the ground truth image. We then combine the set of seeds and the boundary prediction (the output of the network), to correctly identify the topological errors on the boundaries (red rectangle). To do so, we search for the intersections (green dots) of the boundaries in the ground truth and the shortest paths (blue lines) between red dots in the prediction. These green dots are the critical points, with possible leakages. Then we compute our Pathloss function using these critical points.

every region:

$$\mathcal{L}_{BAL}(u, GT) = \sum_{i \in N} BCE(u \odot C_i, GT \odot C_i), \tag{4.7}$$

where $u$ represents the likelihood prediction map, $GT$ is the boundary ground truth image, and $\odot$ is the Hadamard product.

Our loss function measures the segmentation quality for each region. We check if there are leakage positions on the boundaries, thereby ensuring the topological structure in the image. A high value of the Boundary-Aware loss corresponds to many broken connections. When the loss function $L_{BAL}$ is zero, the prediction image is exactly the same as the ground truth image. The pixel-wise binary cross-entropy remains crucial to maintain the global information of every pixel in the image.

$$\mathcal{L}_{total} = \mathcal{L}_{BCE}(u, GT) + \lambda \, \mathcal{L}_{BAL}(u, GT), \tag{4.8}$$

where $\lambda$ tunes the trade-off between both losses.

**Activating critical points through path-based loss (Pathloss)** We design a path-based loss function based on **geodesic distance** to activate critical points in the boundary which improves the topology correctness of detected shapes. The Pathloss aims to enforce the activation in the broken connections on the boundaries of the regions.

**Geodesic distance at a glance** We recall the definition of the geodesic distance, a simple but effective method to find the shortest path between two points in the image. Formally, an image is modeled as a 2D function $u : \Omega \rightarrow \mathbb{R}$, where $\Omega$ is the image domain. The geodesic strength $\tau$ of a smooth curve

$\gamma$ between two pixels $s, s'$ in the given image is defined as:

$$\tau(\gamma) = \int \|\dot{\gamma}(t)\|_{\gamma(t)} dt, \tag{4.9}$$

in which $\dot{\gamma}$ is the *velocity vector* of $\gamma$ and $\|.\|_{\gamma(t)}$ is a norm (see Sommer et al. [145] for more details). Note that the geodesic strength is computed by splitting the integration into pieces where the curve is smooth [146]. From that definition, the geodesic distance $d(s, s')$ between two points $s, s'$ is deduced as the minimum of the geodesic strengths of all the curves between two given points:

$$d(s, s') = \min_{\gamma \in \Pi(s,s')} \tau(\gamma), \tag{4.10}$$

where $\Pi(s, s')$ is the set of paths $\pi$ going from $s$ to $s'$. In the topographical view, the geodesic distance is computed by considering an image as a landscape. Distances between pixels on the flat terrain are shorter than pixels that have hills and valleys in the heightmap [147].

In discrete form, the image can be modeled as a graph, in which $W_{s_i,s_j}$ represents the weight along the edge $[s_i, s_j]$ on the graph. The geodesic distance on the graph is:

$$d(s, s') = \min_{\pi \in \Pi(s,s')} \sum_{s_i \in \pi, i=0}^{N-1} W_{s_i,s_{i+1}}, \tag{4.11}$$

where $s_0 = s$ and $s_N = s'$. The integration becomes the sum of edge weights along the path connecting $s$ and $s'$. Then, the geodesic distance is the distance along a path where the accumulation of image gradient reaches the minimum. In a flat zone, the shortest path is similar to the Euclidean distance. We propose to use this distance to recover the shortest path between points in the image, as a basis to compute our Pathloss function.

**Overview of Pathloss** The overview of our method is exposed in Figure 4.2. The total loss of the network is defined as $\mathcal{L}_{total}(u, g)$ (where $u$ is the predictions and $g$ is the ground truth), which equals the sum of the weighted pixel-level binary cross entropy $\mathcal{L}_{BCE}$ and Pathloss $\mathcal{L}_{PL}$:

$$\mathcal{L}_{total}(u, g) = \mathcal{L}_{BCE}(u, g) + \lambda \cdot \mathcal{L}_{PL}(u, g), \tag{4.12}$$

where the selected hyperparameter $\alpha$ is used to control the trade-off between the two losses. The usage of the $\mathcal{L}_{BCE}$ is already efficient to train a deep neural network to segment an image. However, it may lead to leakages on the boundaries. The idea with the Pathloss is to detect these leakages on the boundaries and to penalize them. We detail in the following subsections how to detect and use these leakages to improve the segmentation.

**Critical points detection** In order to treat the leakage problem during the learning process of the neural network, we have to correctly locate these points. Our method initiates with considering the likelihood map (the output of the network) as a landscape with mountains and valleys. In this map, a broken connection on the boundary of the region corresponds to a dip on

**Algorithm 1:** Shortest path computation between two pixels $s, s'$ in the image.

**Data:** Image $U$, Point $s, s'$, Geomap $D$, Parent image $par$
**Result:** $\pi(s, s')$

Initiate $Q = \emptyset$; $D(s) = 0$ ;
$Q.push(s, D(s))$ ;
$par(s) = s$ ;
**while** $!Q.empty$ **do**
  $p = Q.pop()$ ;
  **if** $p! = s'$ **then**
    **for** $n \in N_8(p)$ **do**
      $d = Update\ distance(n)$ (Eq. 4.11);
      **if** $d < D(n)$ **then**
        $D(n) = d$ ;
        $Q.push(n, D(n))$ ;
        $par(n) = p$ ;
  **else**
    $\pi(s, s')$ = Trace back the $par$ relationship;
    **return**$(\pi(s, s'))$

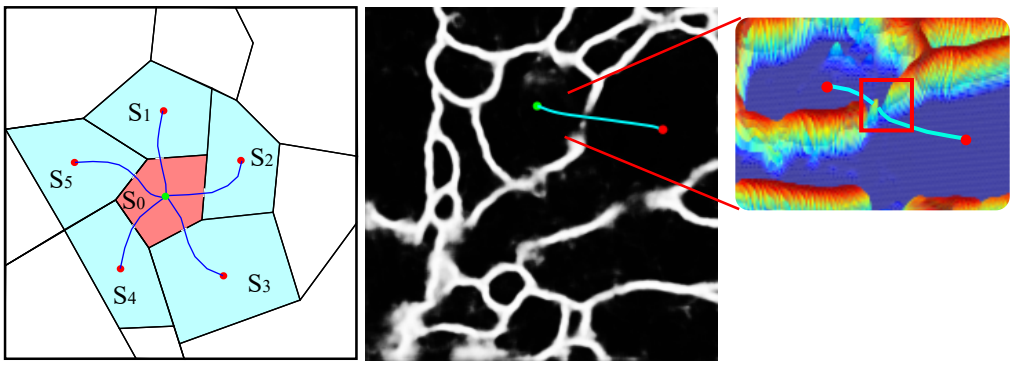



FIGURE 4.5: Leakage (critical points) detection at the intersection of the shortest path and the boundaries of regions. The blue lines represents the shortest path between two seeds.

the mountain ridge. The idea of the Pathloss function is inspired by Liebig's law [148] (law of the minimum). This law was first developed in agricultural science. It states that a growth is dictated not by total resources available, but by the scarcest resource (limiting factor). The broken connection can also be called a critical point, which is a pixel that has the weakest value compared to other pixels located in the boundary of the region. If we drop water to the basin, the critical point is the position where water first leaks from one region to its neighbor. Therefore, the values of the critical points relate to the correct topology in the image. By finding these critical points, we are able to capture missing pixels, thereby aiding the network to penalize pixels near these structures using the Pathloss function.

We formulate the segmentation problem as follows: given the input image $I$, we aim to provide a prediction $u$ (output of the network) that is topology equivalent to the ground truth image $S$. To do that, a set of seed nodes

$M = m_1, m_2, ..., m_N$ with $m_i \in S_i$ is provided inside each region $S_i$ in the ground truth image. The inter-segment topology $A$ is defined by the pairwise adjacency relations among all the segments, i.e., $S_i, S_j \in A$ *iff* $S_i$ and $S_j$ are adjacent in segmentation ground truth.

To preserve the topology of the boundary and measure the quality of segmentation, the shortest path $\pi_{ij}$ between the pair composed of the seed of $m_i$ and each of its neighbors $m_j$ is computed using the geodesic distance. The intersection point $\chi_{ij}$ between the shortest path $\pi_{ij}$ and the boundary of the connected component $S_i$ provides the lowest value on the boundary or the weakest edge (Figure 4.5). These weak points must be enhanced during the learning process to improve the quality of the segmentation.

This is the reason why we must find the shortest path between two regions using the geodesic distance. Such an algorithm is explained in Algorithm 1. It is a shortest path algorithm (also can be called as Dijkstra algorithm). We use the 8 adjacency $N_8$ to define the relationship between neighboring pixel. The propagation procedure is employed by using a priority queue $Q$. Firstly, the geodesic distance map $D(s)$ and the parent relation $par(s)$ are initiated at the seed pixels. The starting point $s$ is then put into the queue $Q$. In the next step, we pop out pixels $p$ from the queue $Q$ for the propagation process. Next, we need to update the geodesic distance value $D$ (in Algorithm 1) at every neighboring pixel $n$ of $p$ along the path using Equation (4.11). For pixels that can be reached from different paths, we select the path that minimizes the distance between the starting point and the current point. We then sort pixels in the queue according to the geodesic distance $D$. If the updated distance $d$ is lower than its previous value, we update the parent relation $par$ and its new distance value $D(n)$. The process is repeated until the destination point $s'$ is found. The shortest path between two points $s, s'$ is easily traced back using the parent relation $par$ that we updated in the propagation step.

We present here our method to compute the $\mathcal{L}_{PL}$ function for granting the closed shape properties in image segmentation task. From the shortest path, we deduce critical pixels as the intersection points $\chi_{ij}$ of the shortest paths $\pi_{ij}$ and the object boundaries $C_i$ to find the weakest edge on the boundary. Example of the shortest path is illustrated in Figure 4.5. The higher value of the intersection point $\chi_{ij}$, the better the segmentation result. The Pathloss $\mathcal{L}_{PL}$ is the sum of the Pathloss for every connected component, which is defined as:

$$\mathcal{L}_{PL}(u, g) = \sum_{i \in N} \sum_{j \in A(i)} MSE(u(\chi_{ij}), g(\chi_{ij})), \tag{4.13}$$

where $u$ and $S$ are respectively the likelihood and ground truth images, $N$ is the number of regions in the ground truth label image.

Our loss function is used to quantize the value of the critical points in the image, thereby evaluating the segmentation quality. A high value of the Pathloss corresponds to many broken connections on the boundary of the connected component. When the loss function $\mathcal{L}_{PL}$ is zero, the likelihood of the critical points is 1, i.e., the prediction image is exactly the same as the ground truth image. The advantage of our $\mathcal{L}_{PL}$ loss function is that it helps the network to focus on important broken missing pixels on each region, thus

preserving the topological structure of the image.

**Discussion on the Pathloss and training details** We choose to predict edge probability maps instead of labeling the regions to allow our method to handle the relation between neighboring regions in the image. Let call $M$ the binary mask that represents a set of all detected critical pixels, which are intersections between the shortest paths between seed pixels in the image and the region boundaries. It guarantees that all the critical points belong to the ground truth boundary of the image. The set of critical pixels is used to check if there are leakage positions or broken connections on the boundaries. If so, the $\mathcal{L}_{PL}$ will force the network to improve the likelihood values on these structures.

We also notice that the edge probability map $u$ will be updated at every epoch. That leads to a re-computation of the shortest path between the neighbor seeds and the critical points. Our method locates the intersection pixels which only depends on $u$, and the change on the mask $M$ at each epoch is not continuous. This set of critical points is not directly predicted by the network. Therefore, the gradient of $\mathcal{L}_{PL}$ exists and can be computed naturally.

Our $PL$ function is architecture-agnostic: it can be integrated into any kind of CNN. In practice, we first pre-train the neural network with only the $\mathcal{L}_{BCE}$ to get the global prediction of the edge probability map, and then train the network with the combined loss (binary cross entropy loss + Pathloss). This way provides us more precise intersection pixels of the region boundary, that lead to a better computation of the $\mathcal{L}_{PL}$.

### 4.1.5 Experimental settings

We report here the performance of a U-Net network trained alternatively with each of the following loss functions: **Mosin**: We set the $\lambda = 0.001$. **BALoss**: We set the $\lambda = 1000$. **Topoloss**: We set the $\lambda = 0.01$. **Pathloss**: We set the $\lambda = 0.01$. **Binary cross-entropy**: As a baseline.

All variants are trained using ADAM optimizer except for the Topoloss one, which is trained using SGD according to authors' recommendation. We set the initial learning rate to $1 \cdot 10^{-4}$, a momentum of 0.9 and a weight decay of $1 \cdot 10^{-5}$.

We report results based on the joint optimization detailed in Section 2.2.6 of the Deep Edge Filter and Closed Shape Extraction stages, and joint optimization leads to the best performance.

### 4.1.6 Numerical experiments and analysis

Table 4.2 summarizes the results for the different variants, and shows that the Pathloss variant is able to achieve a better performance on the validation set, and it provides a slightly better performance than the baseline version (U-Net) on the test set. This proves that the Pathloss leads to the best generalization among all the topology-oriented loss functions. **Hence, using the Pathloss to preserve topology in the predicted edges seems the most promising topology-oriented loss function for the validation and testing datasets.**

TABLE 4.2: COCO Panoptic scores on validation and test set for study on topological on BALoss and Pathloss. The following parameters are static and their respective columns are hidden: we use the Meyer Watershed (MWS) for CSE and Joint Optimization (JO) for Deep Edge Filtering (DEF) selection, no augmentation is performed. For the architectures, * indicate pre-trained variants: the network is trained by using binary cross entropy for 50 epochs then trained with BALoss and Pathloss.

| DEF Archi. | Loss function | CSE Param. | | Evaluation Val. set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma$ | $\delta$ | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | BCE | 50.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| U-Net* | BALoss | 50.0 | 1.0 | 63.1 | 87.6 | 72.0 | 45.6 | 86.3 | 52.9 |
| U-Net* | Topoloss | 100.0 | 6.0 | 59.9 | 88.1 | 68.0 | 36.9 | 84.2 | 43.8 |
| U-Net* | Mosin | 50.0 | 1.0 | 57.7 | 88.3 | 65.3 | 36.0 | 87.4 | 41.2 |
| U-Net* | Pathloss | 100.0 | 3.0 | 62.4 | 88.4 | 70.6 | 47.3 | 86.4 | 54.7 |

The qualitative results of three different topology-preserving techniques are shown in Figure 4.6. The first/second row in each example, respectively, shows the edge prediction map (EPM) and the recall map for each method. At the first glance, we can see that even a small discontinuity on the EPM can significantly damage the structure of the image, as depicted by the red regions on the recall map. The Pathloss function is able to identify and penalize topological errors on the prediction image to maintain the closeness property. **As a consequence, our proposed method outperforms the Topoloss and Mosin methods to recover more correct objects on the historical map.**[1]

## 4.2 Enhancing pixel connectivity as a loss function

### 4.2.1 Motivation

The pixel connectivity is one of the important elements for maintaining the correct topology in EPM images. However, the conventional neural network learns single-pixel probability instead of connectivity probability between two pixels. To leverage the pixel relationships while in training, we learns whether two pixels should be connected by using so-called pixel connectivity loss mentioned in the paper *ConnNet* [149]. Unlike BCE loss which is considered independent pixel values, pixel connectivity loss (PCL) predicts the probability of connectivity for every single pixel instead of predicting the pixel-wise probability of edges. For example, the center pixel is considered positive if the probability of at least (exactly one / at most one) neighbor pixel for laying on an edge is larger than a pre-defined threshold. In this section, we describe PCL method and report its performance in *COCO-PQ*.

### 4.2.2 Related work

By considering the image as a graph, maximin affinity learning of image segmentation (MALIS) [150] classifies the connectivity between pixel pairs (also

[1] We don't put the BALoss qualitative results here since Pathloss is an upgraded version of BALoss.

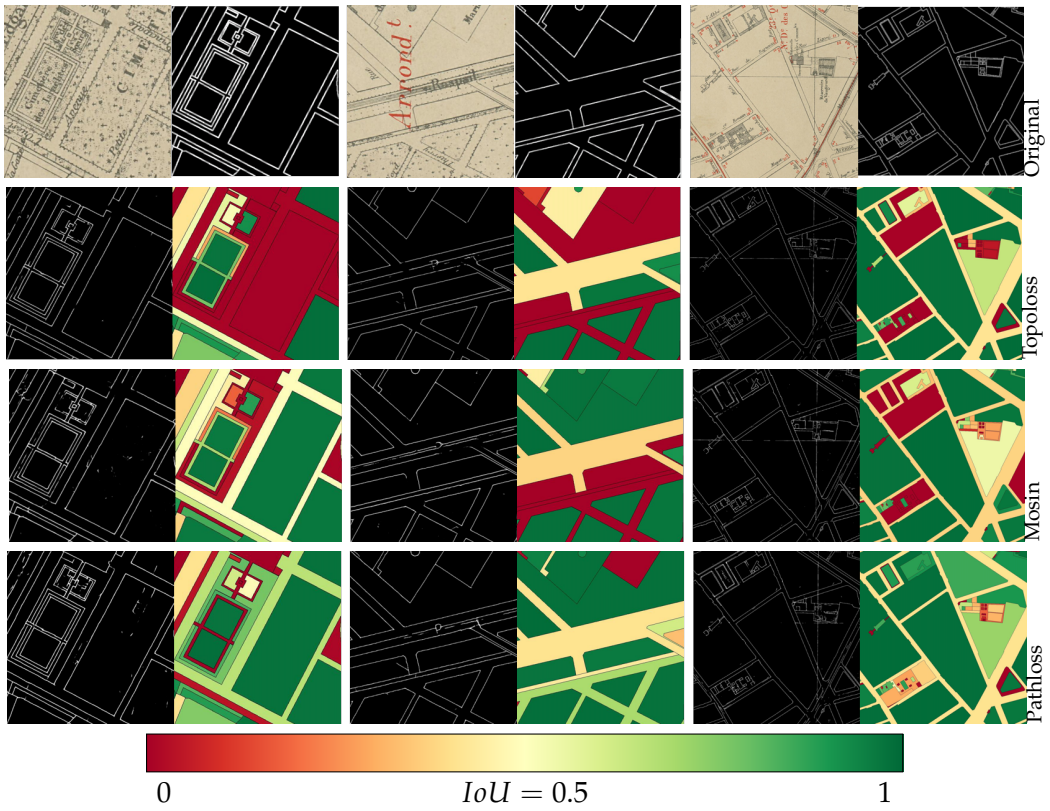



FIGURE 4.6: Qualitative results of our proposed method compared to state-of-the-art methods. The first/second of each example respectively corresponds to the edge probability map (EPM) and the recall map of each method. As indicated in the color scale, the quality of the segmentation ranges from bad (red), with low $IoU$ between predicted and target shapes, to good (green) with high $IoU$.

called affinity graph), the threshold affinity graph can be used to form connected components for instance segmentation purposes. However, using a simple thresholding to partition the grap can lead to misclassify of one or a few edges of the affinity graph. Inspired by ideas from MALIS [150], Oner et al. [151] first predicted the Euclidean distance map and used a hybrid loss by combining the correctness of the distance map and the pairwise shortest cost between two connected components to improve the connectivity in road segmentation task.

One way to maintain topology properties in the object detection task is to make sure information of inter-pixel correlations (also called pixel connectivity) is correct. Nowozin et al. [152] proposed a method which consists in learning the parameters of a Markov Random Field (MRF) by incorporating global connectivity information to connect pixels of every object. Kampffmeyer et al. [149] proposed a network *ConnNet* for saliency object detection which first encodes and predicts the inter-pixel relationships as well as the number of foreground neighboring pixels by using neural networks. Since pixel connectivity has symmetry properties, Yang et al. proposed Biconnect [153], which improves the connectivity network by using shared weights between pairwise neighboring pixels (so-called bilateral voting), instead of using independent weights for pair-wise connectivity. Leveraging previous works [149, 153], we use the connectivity network to detect more

consistency edges in historical map segmentation task. Alternatively, Jie et al. [154] proposed a spatial attention network that jointly learns the segmentation and connectivity information for road detection to focus on the pixels that are connected to their neighbors. In [155], Yan et al. used a graph neural network (GNN) to propagate and aggregate the features of the vertices for regularized road extraction. To conclude, the aforementioned approaches apply information of pixel connectivity (zero-dimensional topology properties) that can boost the performance of object segmentation results.

### 4.2.3 Encoding pixel connectivity

We encode and predict pixel connectivity information for every pixel in the output. It is a learnable way to binarize EPM, making use of pixel-based spatial information at training time. The original *ConnNet* exhibits a significant amount of parameters and may not be efficient on our data. We use a U-Net to encode information of pixel connectivity (any deep edge detector can work). We denoted input image as $I \in \mathbb{R}^{(H,W,3)}$, the connectivity output $\mathcal{C} \in \mathbb{R}^{(H,W,N)}$, where N equals 4 or 8, number of connectivity $\mathcal{C}$ with each of its neighbors. To optimize the PCL for historical vectorization, the connectivity loss $\mathcal{L}_{Conn}$ is calculated through measuring the binary cross entropy loss between prediction connectivity $\hat{\mathcal{C}}$ and ground truth of connectivity $\mathcal{C}_i$:

$$\mathcal{L}_{conn} = \frac{1}{N}\sum_{c=1}^{C}[\hat{\mathcal{C}}_i \log y_i + (1-\hat{\mathcal{C}}_i)\log \mathcal{C}_i]; C \in \{4,8\}. \tag{4.14}$$

At inference time, the pairwise connectivity probability should satisfy the bilateral condition with the threshold $\gamma$:

$$\mathcal{C}_{(i,j)} = \begin{cases} 1, & \text{if } \mathcal{C}_{(i,j),(i+a,j+b)} = \mathcal{C}_{(i+a,j+b),(i,j)} \\ 0, & \text{otherwise,} \end{cases} \tag{4.15}$$

and the final output $\hat{y} \in \mathbb{R}^{(H,W,1)}$ is the **argmax** value of encoded pixel connectivity:

$$\hat{y} = argmax(\mathcal{C}). \tag{4.16}$$

### 4.2.4 Experimental settings

We report here the performance of the PCL trained by using the ADAM optimizer. We set the initial learning rate to $5 \cdot 10^{-5}$, a momentum of 0.9, and a weight decay of 0.02. Furthermore, we use an early-stopping scheme that limits the number of total epochs to consider, setting an upper limit to 60 epochs for each network.

### 4.2.5 Numerical experiments and analysis

We test PCL in *Paris Atlas Municipal* and reported the benchmark results against our U-Net pipeline (with BCE loss without using relations between pixels) in Table 4.3. It should be noted that PCL we introduced here for deep edge detection has a lower performance compared to our baseline, where it

achieves a 29.9 *COCO-PQ* score, compared to the score of 47.1 of the U-Net baseline on the testing set. Still many object missing shown in Figure 4.7. **To conclude, PCL does not directly use and guarantee any topological property compared to watershed segmentation and other topology-awareness loss functions by using much stronger topology guarantees.**

TABLE 4.3: Global *COCO-PQ* results (in %) of our evaluation, for each combination of deep edge detector. Best results on validation and test sets are indicated **in bold**.

| | Connected Component Labeling | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Parameters | | Validation | | | Test | | |
| Stage 1 | $\theta$ | | PQ | SQ | RQ | PQ | SQ | RQ |
| U-Net | 5.0 | 10.0 | 60.4 | 88.2 | 68.5 | 47.1 | 86.8 | 54.3 |
| PCL | 0 | 0 | 51.6 | 86.6 | 59.5 | 29.9 | 85.3 | 35.0 |

We used topological-oriented or pixel connectivity loss functions to maintain the topological properties of edges predicted by a traditional neural network. According to the evaluation results with these topology-oriented loss functions, we found that boundary- and path-based topological preserving loss functions can maintain a better topology structure in the edge predictions compared to other topology-oriented loss functions. It is due to the fact that the pixel connectivity loss function capture only local information while topological correctness requires the predicted edge pixels to maintain global maintain a global context to ensure consistency.

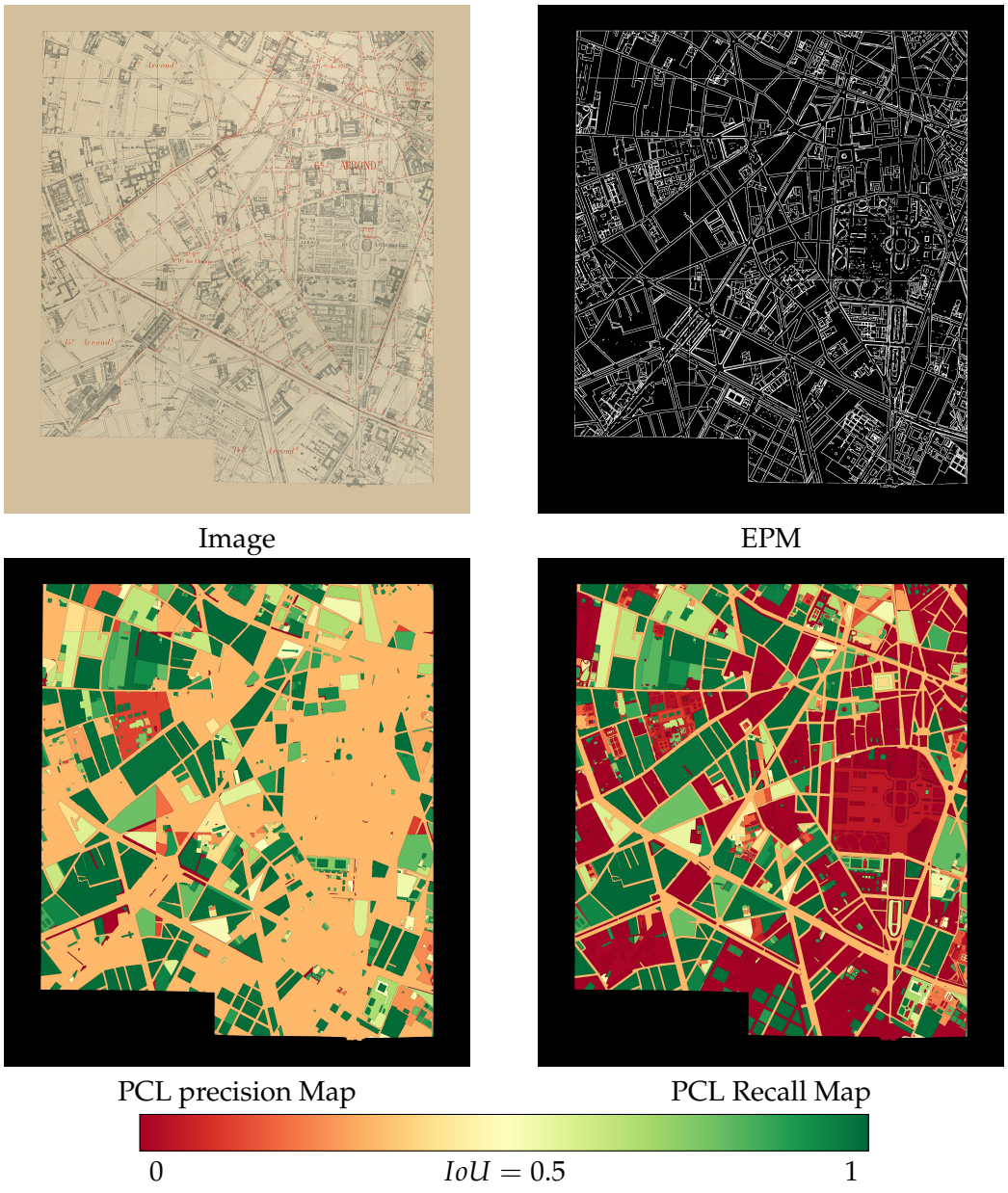



FIGURE 4.7: Image, EPM, Precision, and Recall maps of pixel connectivity loss (PCL). Many objects are miss detected shown in precision and recall maps due to miss detected pixels in the boundaries.

# Chapter 5

# Improving model robustness of deep edge detectors

To answer the fourth question of how to improve the model robustness of deep edge detectors, we propose a novel module to improve the quality of curvlinear detection task as well as the model robustness in different contrast conditions. As discussed in the previous chapters, improving deep neural network architectures and topological-oriented loss functions with joint optimization can both achieve satisfactory results in extracting closed shapes from historical map images. However, these methods may capture irrelevant texture information instead of learning the specific properties of the curvilinear signal used to draw objects' contours. This could be explained by the thin boundaries, biasing the training in favor of texture data. Our designed module encodes the pixel gradient into a traditional convolution kernel, so-called contrast convolution **CConv**. We reveal the potential of both segmentation quality and robustness to different image contrast conditions by applying this module to historical map vectorization tasks.
The content of this chapter is organized as follows. In Section 5.1, we demonstrate the challenges in the curvilinear structure segmentation task. In Section 5.2, we illustrate the existing literature for curvilinear structure segmentation, learning inter-pixel relationships, and residual and attention mechanisms for improving network performance. In Section 5.3, we are going into detail about how these modules are designed. Lastly in Section 5.4 and Section 5.5, we detailed the experimental settings and report the segmentation and contrast robustness with our designed modules on our historical map dataset.
This chapter is an adapted version of the contents of the submitted paper [156].

## 5.1 Motivation

The segmentation of the curvilinear structures is a key challenge in many domains [157–160] ranging from medical images (retinal vessel and neuron segmentation) to geographical data and mapping (object detection in historical maps and road segmentation). In these applications, curvilinear structure enhancement is pivotal e.g., for flow computation or change detection, and calls for designing detection methods that enhance the curvilinear feature by using some traditional method such as local binary patterns (LBP) [161], Sobel operator [162], or Roberts cross [163]. The rapid

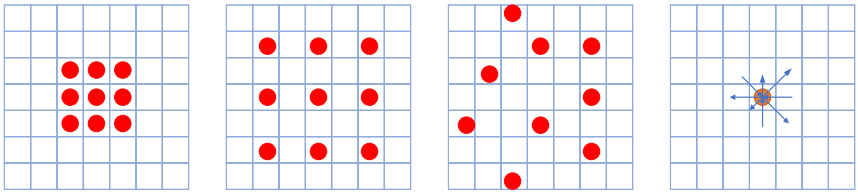
development of convolutional neural networks has significantly improved the performance in detecting curvilinear structures. These methods rely on the increasing complexity of CNN architectures, with advanced concepts such as attention module [164], residual corrections [165], adversarial training [166] or pre-processing and post-processing methods [167]: e.g., a predefined number of objects or spatial scales [168], integration of human-annotated data. While per-pixel-level performance has been boosted, they often fail to maintain the correct structures in the predictions. For a successful detection of curvilinear patterns, not only the pixels should be correctly individually detected, but also keep spatial coherence among inter-pixels [131, 169–171]. These coherences can also be considered as contrast relationships between pixels.

Three main issues lead to such failure cases. First, the inherent downsampling process of CNNs makes it difficult to maintain the fine structure of the learned features in the original image when the networks go deeper, even if Fully Convolutional Networks [39]. For detecting curvilinear structures, Mosinka et al. [127] found that more pixels are connected at the early layers. It was because the segmentation model began to learn progressively the low, middle and high-level features, from the bottom, intermediate to top convolutional layers [172, 173]. It caused that the curvilinear structure is more prominent in low-level features, so we added our propose method in the bottom of segmentation networks. Secondly, rigid convolutional kernels are not always adapted for detecting fine multiscale non-linear patterns. Different receptive fields allow to handle multiple spatial scales and image resolutions, e.g., Atrous convolution [174, 175] or deformable convolution [176] (Figure 5.1), yet assuming such scales are known. These convolution operations calculate the linear sum of learned kernels and ignore the correlation between neighboring pixels (e.g, the difference between two pixels). Thirdly, a key issue remains poor cross-domain and unseen dataset generalization. Alleviated by data augmentation strategies [177, 178] and transfer learning methods [179, 180], these methods are all from the perspective of increasing data or data characteristics, and do not reduce the differences between the datasets from the model. This explains why current state-of-the-art methods (are tailored to) perform well only on specific datasets.

To tackle these issues, inspired from the traditional edge detection operators, we propose **contrast convolution** (CConv, Figure 5.2) which integrates the pixel gradient into the contrast convolution, thereby combining the efficiency of the low-level method with the strength of the deep neural network. The **CConv** is a mask-guided learnable operation, which can improve the data generalizability of the model. In addition, we design module so called **contrast blocks** based on our proposed contrast convolution: concatenation, addition (residual block), or multiplication (soft attention) [181].

## 5.2 Related work

**Curvilinear structure segmentation networks** For the historical map segmentation, our publications and this thesis benchmarked U-Net [47], HED [56], BDCN [57] and ConnNet [149] to estimate the probability of building boundaries in order to extract instances from the map images.



(a) Std. Conv. (b) Atrous (c) Deformable (d) CConv(ours)

FIGURE 5.1: Sampling positions in 3×3 convolution kernels. Red points denote the pixels of interest. (a) Standard convolution[182]. (b) Atrous convolution [174, 175]. (c) Deformable convolution [176]. (d) Contrast convolution (our method): the blue arrows give the relative directions between the center point and the eight neighbors, and its length represents the degree of influence on the center point. CConv takes such pixel contrast into consideration more finely.

**Learning inter-pixel relationships** Inter-pixel relationships in the images are used in training procedures standard with MRF/CRF [152, 183–185]. Recently, inter-pixel relationships are used for salient object detection tasks. Kampffemeyer et al. [149] proposed ConnNet for salient object segmentation, where the ground truth of binary salient objects was encoded into eight directions of pixel connectivity, which can be predicted through designed neural networks. Similar to ConnNet, Yang et al. [153] proposed BiconnNet, which added a bilateral voting module to pay extra attention to the probability misclassified direction. Both approaches are designed to fully leverage the inter-pixel relationships as loss function in the salient object detection tasks compared to traditional loss functions based on intersection of unions (IoU).

**Residual and attention units** The residual learning is first proposed by He et al. [88] to fasten network convergence and solve the vanishing/exploding gradient problems when we stack more layers in the networks. Shortcut connections are also used in residual blocks [88] to learn the residual functions and make all information easily passed through the whole network. The gating function can be added in shortcut connections [186], transferring the block between residual and non-residual functions to control the information passing.

Soft attention has been deeply studied in [187, 188]. Chen et al. [187] used attentions to soften the weight of unimportant features in different scales. Compared with traditional pooling methods (average pooling or max pooling), attention can help networks better select the important features for each region [188, 181]. Jaderberg et al. [188] proposed a differential module called spatial transformer that learns invariance to translation, scale, rotation, and more generic warping of images. The module contains a localization network that progressively learns the transformation parameters. Then a soft-attention mechanism merges information from the original images according to the transformation parameters, to achieve spatial invariance. Therefore, in this thesis, we use the advantage of these units to prevent the loss of detailed information about curvilinear structure along the network.

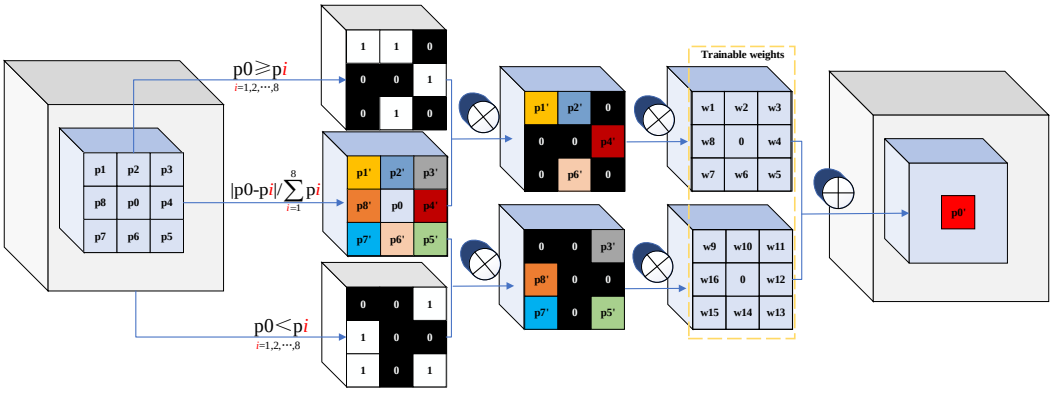



FIGURE 5.2: Illustration of 3×3 contrast convolution. The existence of the binary masks shown with value of 0 and 1 which makes the connection method between neighboring pixels constantly change during the training process, because the binary masks are constantly changed with the relationship between the center pixel and the surrounding pixels.

## 5.3 Method

**Contrast convolution** The standard 2D convolution operation is mainly composed of two parts. First, the input feature map is sampled by k×k convolution kernels, and then the sampled values are weighted and eventually summed and fused. Let us take k=3 as an example, and the standard 3×3 convolution operation is defined as follows:

$$\mathbf{Conv}\left(x,y\right)=\sum_{dx=-1}^{1}\sum_{dy=-1}^{1}\omega\left(dx,dy\right)\mathbf{I}\left(x+dx,y+dy\right). \tag{5.1}$$

where $\mathbf{Conv}\left(\cdot\right)$ is the feature map after convolution operation. $\mathbf{I}\left(\cdot\right)$ denotes the original feature map. $x$ and $y$ represent the location of the pixel in the image coordinate system. $\omega\left(dx,dy\right)$ denotes the weight of convolution kernel. Each position of the convolution kernel is designed by $-1\leq dx\leq 1$ and $-1\leq dy\leq 1$.

According to Equation (5.1), the standard convolution operation does not specifically emphasize the spatial coherence between pixels whether there is a connectivity between $\mathbf{I}\left(x+dx,y+dy\right)$ and $\mathbf{I}\left(x,y\right)$. The following contexts describe the details of how contrast convolution is calculated. Firstly, the 3×3 convolution kernel is designed to measure the gradient between the center pixel $\mathbf{I}(x,y)$ with the eight neighboring pixels $\mathbf{I}(x\pm 1,y\pm 1)$.

Inspired by the boolean pruning mask [189] that categorizes filters into important ones and unimportant ones, considering the fact that large differences often happen on the boundary, we use two criteria to define the relationship between pairs of pixels.

The first criterion is: If the center pixel is greater than the neighboring

pixel, it is considered as *positive contrast* as shown in Equation (5.2), and vice-versa in Equation (5.3).

$$\begin{aligned}\mathbf{mask}_{+}(x+dx,y+dy) &= 1 \text{ if } \mathbf{I}(x,y) \geq \mathbf{I}(x+dx,y+dy), \\ &\quad 0 \text{ otherwise,} \end{aligned} \tag{5.2}$$

$$\begin{aligned}\mathbf{mask}_{-}(x+dx,y+dy) &= 1 \text{ if } \mathbf{I}(x,y) < \mathbf{I}(x+dx,y+dy), \\ &\quad 0 \text{ otherwise.} \end{aligned} \tag{5.3}$$

where $dx$ and $dy \in \{+1, 0, -1\}$, except center pixels which have $dx = dy \neq 0$.

Here is the reason why we need to define two masks at Equation (5.2) and Equation (5.3), for example, when $\mathbf{I}(x,y) > \mathbf{I}(x+1,y+1)$ and $\mathbf{I}(x,y) < \mathbf{I}(x-1,y+1)$, $|\mathbf{I}(x,y) - \mathbf{I}(x+1,y+1)| = |\mathbf{I}(x,y) - \mathbf{I}(x-1,y+1)|$, but in terms of $\mathbf{I}(x,y)$, $\mathbf{I}(x+1,y+1)$ and $\mathbf{I}(x-1,y+1)$ have completely different meanings. The positive and negative contrast cases are separated by the function of guided masks to separate the weights sharing. It is a heuristic approach that these two cases should be learned separately through channel aggregation process. As shown in Figure 5.5, if the proposed contrast block is used without the binary mask, the curvilinear features are not enhanced.

The second criterion is based on the normalization of pixel difference $\sigma(dx,dy)$:

$$\mathbf{dd}(x+dx,y+dy) = \frac{|\sigma(x+dx,y+dy)|}{\sum_{dx=-1}^{1}\sum_{dy=-1}^{1}|\mathbf{I}(x,y) - \mathbf{I}(x+dx,y+dy) + \delta|}. \tag{5.4}$$

where $\delta$ is a tolerance value equals to 1e-6 to prevent that the denominator from being null. Finally, the contrast convolution is defined by combining Equation (5.2), Equation (5.3) and Equation (5.4):

$$\begin{aligned}\mathbf{CConv}(x,y) \;=\; &\textstyle\sum_{dx=-1}^{1}\sum_{dy=-1}^{1}\omega_{+}\;(dx,dy) \\ &(\mathbf{dd}(x+dx,y+dy) \times \mathbf{mask}_{+}(x+dx,y+dy)) \\ &+\omega_{-}\;(dx,dy)\,(\mathbf{dd}(x+dx,y+dy) \times \mathbf{mask}_{-}(x+dx,y+dy)).\end{aligned} \tag{5.5}$$

As mentioned above, the contrast convolution is very different from the standard 2D convolution. For the features on feature maps and convolutional kernels, the standard 2D convolution directly uses the calculation method of mult. and sum ops and is trained to maximize the correctness of learned convolutional kernels. However, the contrast convolution uses the absolute difference between neighboring pixels to separate neighboring pixels, which makes the convolutional learn for specific directions for pairwise contrast difference between pixels.

**Contrast blocks** Residual learning [88] and attention mechanism [190] have achieved a wide range of applications in the field of deep learning. We design three different blocks which is the most common way to fuse data (concatenate, attention and residual) shown in Figure 5.3. The output of **Concatenate block (CB)** $O_{CB}$ is defined by

$$O_{CB} = Concatenate(\mathbf{I}(x,y), \mathbf{CConv}(x,y)) \tag{5.6}$$

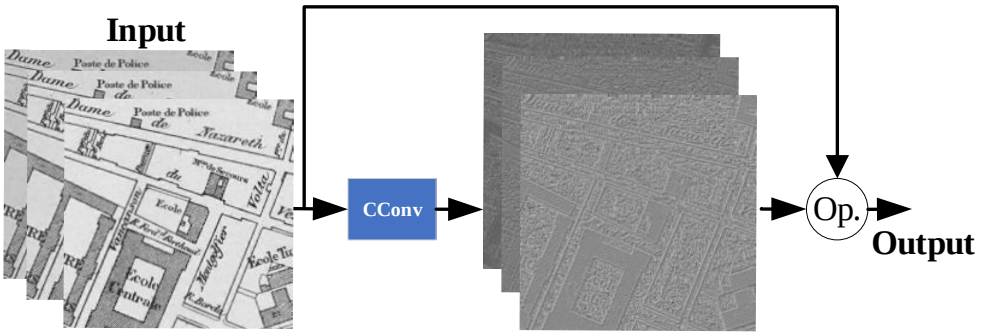



FIGURE 5.3: Different contrast convolution (CConv) blocks. Input: RGB historical maps with three channels. *Op*. includes three optional operations (concatenate, add and multiply operations).

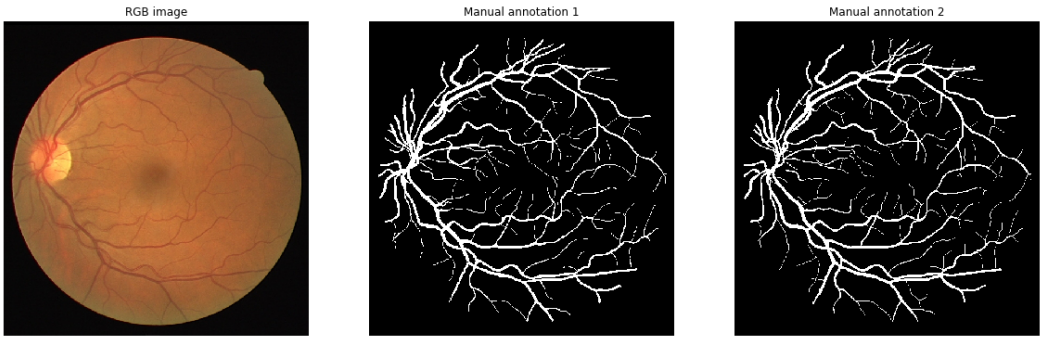



FIGURE 5.4: Example of Drive datasets [191].

The outputs of **Attention block (AB)** $O_{AB}$ and **Residual block (RB)** $O_{RB}$ are defined by:

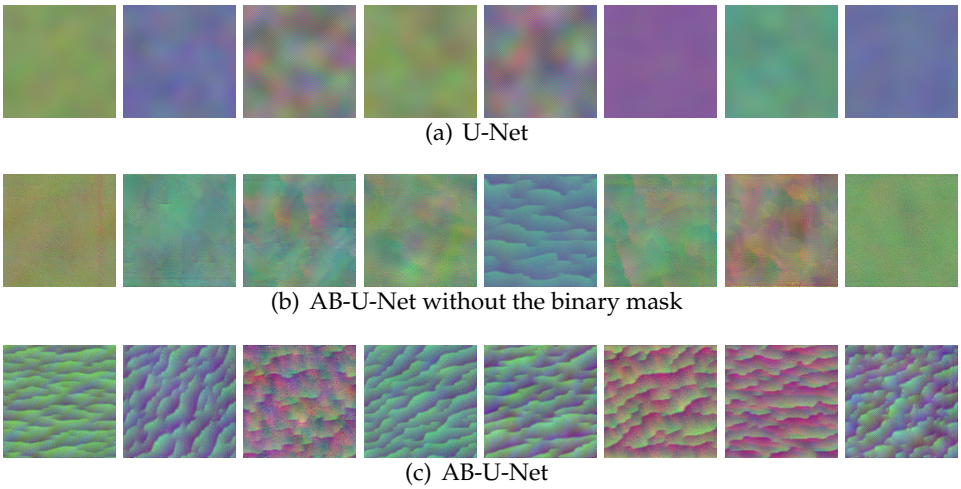
$$O_{AB} = \mathbf{I}(x,y) \otimes \mathbf{CConv}(x,y), \tag{5.7}$$

$$O_{RB} = \mathbf{I}(x,y) \oplus \mathbf{CConv}(x,y). \tag{5.8}$$

The **CB** is different from **RB** and **AB** because it increases the number of feature maps, while **RB** and **AB** update the features on each channel and do not change the channel count. As shown in Figure 5.5, we verify that our proposed modules learn the linear structures that are important for detecting curvilinear structures. With **CConv**, the network is fed with the knowledge of curvilinear structures in Figure 5.5(c).

## 5.4 Experimental settings for segmenting historical maps

To evaluate the relevance of our **contrast blocks**, we compare these blocks with our baseline methods (U-Net), and simply add after the input layer of these methods. We add RB/AB/CB layers with kernel size of 3x3 (common choice in most of deep neural networks) in the early layer of the neural networks respectively. Our experiments on historical maps were conducted using Nvidia Quadro P6000 and RTX 8000 GPUs servers. Since the image sizes



(a) U-Net

(b) AB-U-Net without the binary mask

(c) AB-U-Net

FIGURE 5.5: Filter visualization of the convolution layers of U-Net after CConv layer on DRIVE [191] dataset which use Figure 5.4 as input image. Here is an example with our attention block (AB). We verify that our proposed modules learn the linear structures that are important for detecting curvilinear structures. Encoder-decoder segmentation methods propagate the linear structure information through the layers (skip/residual connections), assuming such information exists. Otherwise, (Figure 5.5(a)), the curvilinear structure feature cannot be guaranteed. With CConv, the network is fed with the knowledge of curvilinear structures (Figure 5.5(c)).

of our map images are too large ($4500 \times 9000$ pixels for training), the image is cropped into patches of $500 \times 500$ pixels. The ADAM [192] is used with a learning rate of $1 \cdot 10^{-4}$ as an optimizer in our experiments. The BCE loss is used to train dataset of historical maps.

## 5.5 Experimental results and analysis

### 5.5.1 CConv improves segmentation performance

We first evaluate the performance of segmentation models for adding **contrast layer** for segmenting historical maps for the year 1926 where the training and validation datasets are from the same map. The results are listed in Table 5.1. Compared to the baseline results without applying contrast blocks, the contrast blocks maximally improve the *COCO-PQ* score by 4.5 points (of percentage) (2.0 points with pre-training), and BDCN with no pre-trained weight with 5.02% and pre-trained 4.71% in the validation dataset. It implies that contrast blocks consistently improve the segmentation quality.

### 5.5.2 CConv improves contrast robustness

By applying the contrast blocks to the traditional architectures, contrast blocks lead to a significant improvement of the robustness to contrast performance with the neural networks in test datasets both in the instance and pixel evaluations. As shown in Figure 5.7, using contrast blocks improves detection quality for the boundaries of map objects. The contrast robustness the segmentation model in historical maps can help creating accurate EPM for historical maps with style shift (see Figure 5.7).

TABLE 5.1: The segmentation results on *Atlas Municipal*. The baseline U-Net results are copied and tested with the same setting according to our publication [2].

| Method | Validation | | | Test | | | |
|---|---|---|---|---|---|---|---|
| | PQ ↑ | SQ ↑ | RQ↑ | PQ ↑ | SQ ↑ | RQ ↑ | Dice ↑ |
| UNet [47] | **34.80** | 80.50 | 43.30 | 8.10 | 78.20 | 10.40 | 68.12 |
| RB-UNet (ours) | 31.86 | 79.13 | 40.26 | **25.52** | 78.00 | 32.72 | **74.49** |
| AB-UNet (ours) | 32.20 | 79.52 | 40.50 | 24.06 | 78.20 | 30.77 | 74.42 |
| CB-UNet (ours) | 32.63 | 79.07 | 41.28 | 24.50 | 78.36 | 31.27 | 74.25 |
| HED-Scratch [56] | 23.20 | 76.50 | 30.30 | 14.00 | 74.80 | 18.80 | 24.03 |
| RB-HED-Scratch (ours) | **28.85** | 77.65 | 37.15 | 18.94 | 77.56 | 24.41 | **71.67** |
| AB-HED-Scratch (ours) | 28.19 | 77.72 | 36.27 | **21.73** | 77.03 | 28.21 | 69.50 |
| HED-Pretrain [56] | 27.60 | 76.50 | 30.30 | 16.20 | 76.10 | 21.30 | 53.98 |
| RB-HED-Pretrain (ours) | 28.69 | 78.07 | 36.75 | 17.91 | 76.91 | 23.29 | 69.68 |
| AB-HED-Pretrain (ours) | **29.57** | 77.98 | 37.92 | **23.98** | 76.88 | 31.19 | **71.74** |
| BDCN-Scratch [57] | 27.70 | 80.60 | 34.30 | 14.00 | 74.80 | 18.80 | 61.21 |
| RB-BDCN-Scratch (ours) | 29.28 | 78.88 | 37.12 | 19.68 | 77.40 | 25.43 | **74.27** |
| AB-BDCN-Scratch (ours) | **32.72** | 79.19 | 41.31 | **23.63** | 78.62 | 30.06 | 72.38 |
| BDCN-Pretrain [57] | 27.60 | 82.10 | 33.70 | 8.90 | 82.80 | 10.70 | 61.90 |
| RB-BDCN-Pretrain (ours) | **32.31** | 78.77 | 41.01 | 18.03 | 78.94 | 22.84 | 72.09 |
| AB-BDCN-Pretrain (ours) | 32.00 | 79.49 | 40.26 | **24.01** | 79.41 | 30.23 | **74.10** |
| ConnNet [149] | – | – | – | 14.2 | 73.6 | 19.3 | – |

TABLE 5.2: Average of the *COCO-PQ* score after contrast perturbations *Atlas Municipal*

| Method | PQ↑ | SQ↑ | RQ↑ | Parameters ↓ |
|---|---|---|---|---|
| U-Net | 6.19 | 85.34 | 7.92 | 31,032,837 (∼31M) |
| RB-U-Net (Ours) | 11.93 | 78.87 | 15.36 | 31,032,984 (∼31M) |
| AB-U-Net (Ours) | **14.59** | 78.38 | 18.77 | 31,032,984 (∼31M) |
| CB-U-Net (Ours) | 10.29 | 82.08 | 13.03 | 31,034,712 (∼31M) |

To assess the contrast robustness capability of the proposed contrast block, we evaluated the performance of the proposed architecture. First we tested on the same test set but with different image contrast to mimic different scanning conditions. We also test the testing set of historical maps with different image contrast to mimic different scanning conditions of the historical maps. Then the *COCO-PQ* score is measured for contrast value range from 0.05 to 0.8 with the step of 0.05. As it is shown in Table 5.2 and Figure 5.6, the AB-U-Net, RB-U-Net, and CB-U-Net have better average *COCO-PQ* scores and area under curves (AUC) compared to original U-Net architecture.

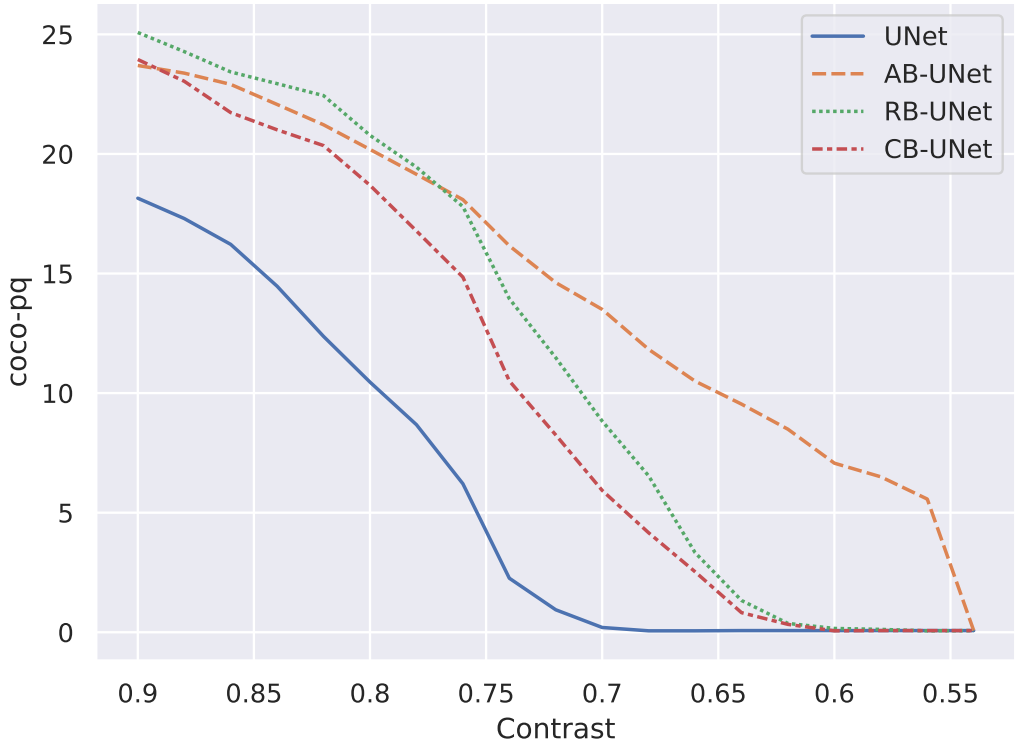



FIGURE 5.6: The *COCO-PQ* score of AB-, RB-, and CB-U-Net in different contrast conditions ranging from 0.05 to 0.8. This figure shows that our methods have better contrast tolerance compared to the original U-Net architecture.

In this chapter, we evaluated the performance of curvilinear structure segmentation with our designed **CConv** module. These modules encode and learn the information of pixel gradients inside the neural networks. Our contrast blocks are built from the **contrast convolution** and residual, attention, and concatenate operations to merge original inputs with outputs from contrast convolution operations. Moreover, we proved that simply stacking the contrast blocks to existing neural networks can significantly improve the performance in predicting curvilinear structure in historical map segmentation tasks. Furthermore, we found that networks with added contrast blocks are more robust to contrast changes in the images compared to the baseline architectures by only adding a maximum of thousands of parameters to the models. Finally, the invention of contrast blocks opens a new and promising research direction on directly bringing structural properties inside deep neural architectures.

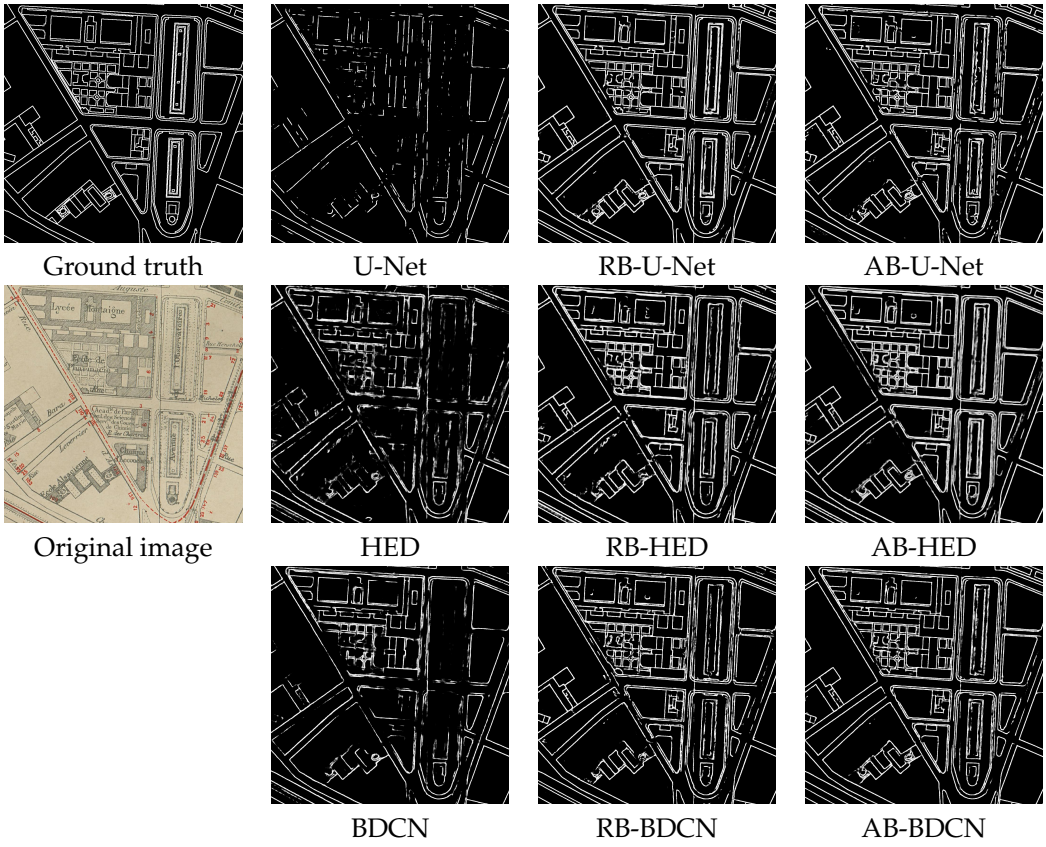



FIGURE 5.7: Visualization of the segmentation results on *Atlas Municipal* using different state-of-the-art segmentation architectures as base network.

# Chapter 6

# Leveraging redundancies of historical maps

To answer the fifth question of **how to leverage the redundancies of historical maps**, we align historical map sheets representing the same area for different years in the *Atlas Municipal* map series. These alignment results can be used to build **edge consensus** which can be used to refine predicted EPMs and subsequently improve per-date segmentation and feed change detection pipelines.
The content of this chapter is organized as follows. In Section 6.1, we demonstrate our incentives and challenges in aligning historical maps. In Section 6.2, we explain the related work for the non-parametric and parametric geometric alignment techniques. In Section 6.3, we demonstrate the unsupervised alignment techniques for aligning a pair of historical maps without requiring any annotation key points. In the end, Section 6.4 and Section 6.5, we explain our experiment setups with some findings of our alignment results and future works.

## 6.1 Motivation

One of the most important properties of the *Paris Atlas Municipal* is the redundancies that are visible over different periods, as shown in Figure 6.1. These redundancies can be useful in three aspects. Each useful aspect depends on the previous ones: geo-referencing (thanks to proper alignment), refine segmentation and self-supervision for training some segmentation networks (automatic ground-truth generation from reliable results predicted using the first network).

1. **Alignment helps geo-referencing:** by only georeferencing one map, the georeferencing information can be propagated through other maps.
2. **Refine segmentation:** the redundancy information existing in the same and different time series of atlases can be used to refine the quality of segmentation.
3. **Self-supervision for training some segmentation network:** map redundancies can be used to create pre-generated weak ground truth which can be used in training a self-supervision network.

We keep the last point as future work and it will not be discussed further. In this thesis, we are particularly interested in using aligned historical maps to refine EPMs for improving the quality of closed shapes extraction process.

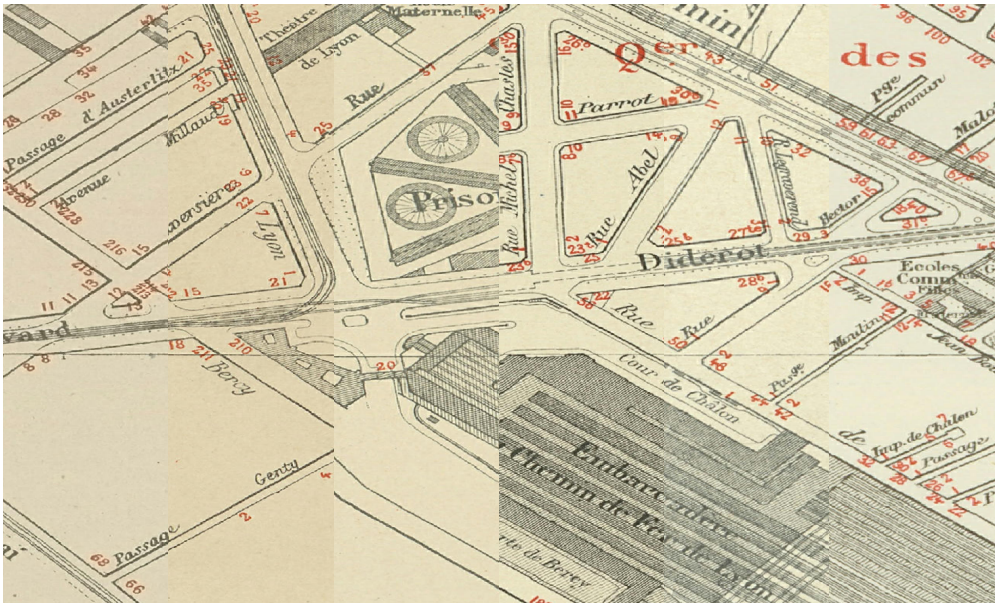



FIGURE 6.1: How maps change over their successive editions. Most of the objects of the map are consistent between the 1882 and 1900 editions, as well as between the 1926 and 1937 editions, where these redundancies can be used. Significant change is visible between the 1900 and 1912 editions and little redundant information can be found.

To leverage the redundancies in the historical maps, aligning the maps is a prerequisite. However, aligning historical maps is a challenging task. Firstly, the topological properties should remain after the map alignment process. Secondly, finding the correspondences of manually annotated key points of two historical maps is not yet practical, since the quality varies for different people who annotate the maps, and it is a very time-consuming process to annotate the key points for large-scale historical maps. Thirdly, historical map images contain many repeated patterns and noises (such as texts and textures) that will create many false alarms when using classical registration frameworks like SIFT [193] (local detector/descriptor) + RANSAC [194].

To make sure the aligned image can maintain its topological properties, we choose to use geometric alignment instead of optical flow. We test an off-the-shelf unsupervised image alignment technique based on deep optical flow, **RANSAC-flow** [195], and report the qualitative results in Figure 6.2. Despite the fact that the two historical map sheets have a high quality alignment, as shown in Figure 6.2, the texts and texture can not be recognized for the aligned source image and topological properties such as line consistency are changed due to the lack of smoothness of optical flow. To address the challenge of generating expensive annotated key-points for the fine alignment of two images, we use a weakly supervised instead of a fully supervised or self-supervised techniques.

For the purpose of eliminating the effects of unrelated information in the

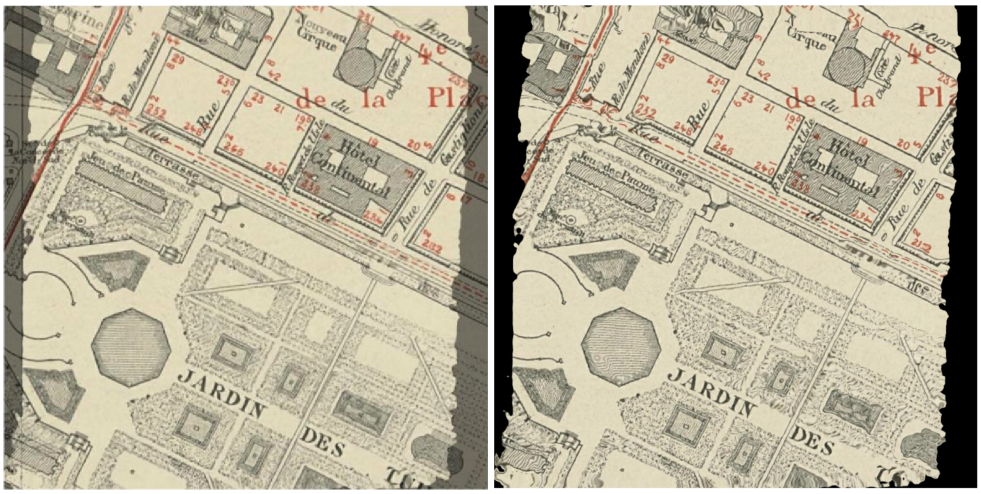



FIGURE 6.2: Results of aligning historical maps by using unsupervised state-of-the-art image alignment with geometric transformation + optical flow (RANSAC-flow) [195]. Left: two image aligned by optical flow. Right: Source image after fine alignment.

maps (textures and texts), which can significantly downgrade the performance of the fine image alignment process, we adopt the EPM for fine alignment instead of the original RGB map images which is also called sparse registration.

To refine missing edge pixels, we use a voting strategy as a baseline to find a consensus between predicted edges in different map sheets. We aim at keeping reliably aligned edges while filtering out other edges caused by bad alignment quality as well as changes over different map sheets. This voting strategy can be used to refine predicted edges based on the consensus in historical map alignment results. At last, the aligned historical map time series with redundancies information can be potentially used to **refine the segmentation**.

## 6.2 Related work

Early techniques for geometric alignment consist in optimizing a matching energy and are based on the combination of traditional image descriptors (such as SIFT [196] or HOG [197]) with hand-crafted alignment models [198–200]. More recently, techniques [201–204] have been proposed to improve the performance of image alignment by replacing traditional image descriptors by CNN descriptors with pre-trained networks. Other techniques [205–207] improve the geometric alignment models with trainable image descriptors. To make image alignment end-to-end trainable, Rocco et al. [125] proposed a parametric geometric pipeline where image descriptors can be trained, and the pixel correspondence is differentiable. Nonetheless, these methods are trained in a self-supervised fashion with synthetically warped image pairs as ground truth, which is difficult to generalize to unseen data with significantly variations and changes. To leverage the limitation of strong supervision in the existing geometric alignment techniques, Rocco et al. [208] developed a weakly-supervised end-to-end trainable technique for dense alignment without requiring any ground truth of pixel correspondence. This technique fits

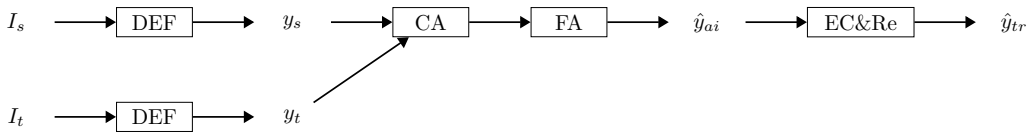



FIGURE 6.3: Pipeline for refining edges by leveraging redundancies. DEF:Deep edge filter; CA: Coarse alignment (graticule alignment); FA: fine alignment (weakly supervised image alignment); EC&Re: Edge consensus and refinement.

our alignment task where the ground truth of pixel correspondence is expensive to produce and impractical to generate in large-scale historical maps.

## 6.3 Map image alignment method

### 6.3.1 Overview of our pipeline

Our pipeline for aligning historical maps consists in three stages : the coarse alignment, the fine alignment and the edge consensus stages. The coarse alignment stage is used to prevent large displacement of two aligned maps by finding global geometric transformation, while the fine alignment stage is used to match the boundary of objects at pixel level by finding local geometric transformations between two local image patches.

### 6.3.2 Coarse alignment

We use graticule point which is intersection of gratiucule, each represent a constant coordinate as key points for aligning two map images coarsely, as illustrated in Figure 6.4. Graticule points for our dataset have been manually annotated and can be obtained from the database of the ICDAR 2021 Competition on Historical Map Segmentation [99]. Given N graticules points $p_g = \{p_1, p_2, ...p_i\}; i \in [1, N]$ in source map images $I_s$ and N other points $p'_g = \{p_1, p_2, ...p_j\}; j \in [1, N]$ in target map images $I_t$, performing the coarse alignment of two map images consists in estimating the geometry transformation between $p_g$ and $p'_g$. In this thesis, we use a homography for our coarse alignment stage, but another geometric transformations such as affine or TPS could also be used.

### 6.3.3 Refined alignment

We can see that global coarse alignment by using graticules is not sufficient enough to align the boundaries of two images shown in Figure 6.5. The misalignment of object boundaries can lead to failure of finding good edge consensus between different map series. Thus, we adapt fine alignment between two maps, which can achieve better alignment quality of boundaries of objects shown in Figure 6.6.

This stage is based on the weakly-supervised image alignment techniques proposed by Rocco et al. [208], whose original publication contains the details we summarize here. Once the source $I_s$ and target image $I_t$ are transferred into EPMs $y_s$ and $y_t$, both $y_s$ and $y_t$ are used as input for the weakly supervised geometric alignment module. The optimization objective function

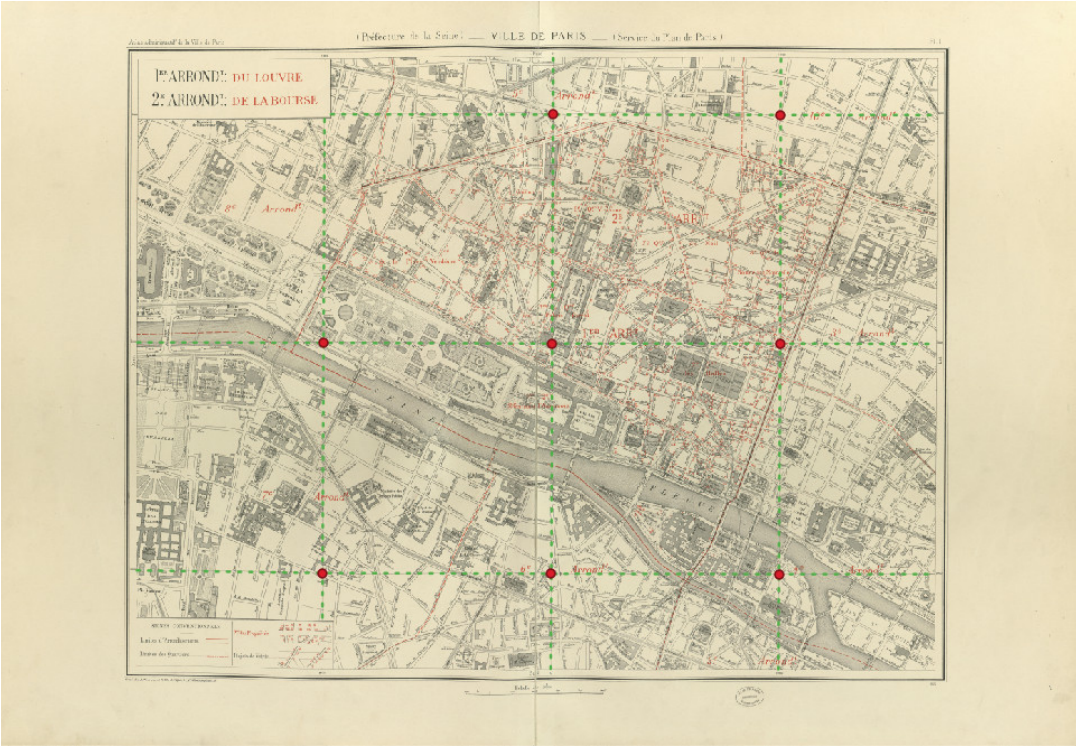



FIGURE 6.4: Graticule points of historical maps (red dots). In this example, 8 graticules points are visible and usable for coarse alignment.

$S(f_s, f_t)$ of the weakly-supervised geometric alignment module is the pairwise *cosine* similarity score between image of source and target feature maps $f_s$ and $f_t$. The feature maps $f_s, f_t : \mathbb{R}^{H,W,D}$ are the mapping of source and target images $I_s, I_t : \mathbb{R}^{H,W,D}$ by using a fully-convolutional neural network with pre-trained weights. Then the feature maps of $f_s, f_t$ are used to calculate pairwise similarity matrices as:

$$s_{ijkl} = S(f^s, f^t)_{ijkl} : \mathbb{R}^{H,W,D} \times \mathbb{R}^{H,W,D} \rightarrow \mathbb{R}^{H,W,H,W}, \tag{6.1}$$

with the pairwise score $S(f_s, f_t)$ defined by Rocco et al. [208]:

$$s_{ijkl} = S(f^s, f^t)_{ijkl} = \frac{< f^s_{ij}, f^t_{kl} >}{\sqrt{\sum_{i,j} < f^s_{ij}, f^t_{kl} >^2}}. \tag{6.2}$$

According to the mathematical definition in Equation (6.2), we want to maximize the pairwise similarity score $S(y_s, y_t)$ through $S(f_s, f_t)$. To filter elements which are not boundaries in the content (like text which can be printed at different location over time) to avoid perturbing the alignment module:

$$\max S(y_s, y_t) \approx \max S(f_s, f_t). \tag{6.3}$$

The final step of the alignment pipeline is to predict a geometric transformation function $G : \mathbb{R}^{H,W,H,W} \rightarrow \mathbb{R}^p$ which is the mapping between the pairwise similarity $S$ and the geometric transformation $\theta$, where $p$ is the number of parameters in $\theta$: $p = 6$ for the affine transformation, $p = 9$ for the homography transformation, and $p = 18$ for the thin-plate-spline transformation with 8 control points.

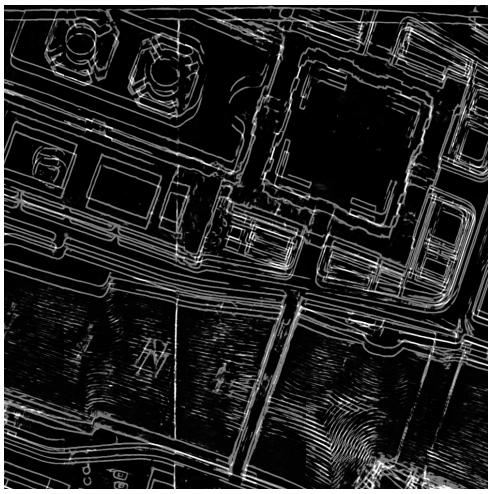


FIGURE 6.5: Global coarse alignment by using graticules.

FIGURE 6.6: Local fine alignment with geometric alignment.

FIGURE 6.7: Results of two aligned EPMs predicted from the map in the years 1898 and 1909.

Each transformation parameter $\theta$ can create geometric 2D warping-grid $\mathbb{G}_{geo} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ from source image to the target image. We apply the idea of *soft-inlier count* proposed by Rocco et al. [208] and inspired by the RANSAC method [194]. It is differentiable and can be used with any neural networks. The idea is to calculate a *soft-inlier count* $c \in \mathbb{R}$ by multiplying pairwise similarity $s_{ijkl}$ and the discrete inlier mask $m_{ijkl}$:

$$c = \sum_{i,j,k,l} s_{ijkl} \cdot m_{ijkl}. \tag{6.4}$$

This mask $m_{ijkl}$ is generated using a spatial transformer [188] layer by wrapping the source image into target image with spatial attention. Finally, the best alignment is selected based on the highest value of *soft-inlier count* $c$.

### 6.3.4 Refining edges with consensus

The purpose of this stage is to produce a new, more reliable EPM $y_{tr}$ for a map image $I_t$ by considering its original predicted EPM $y_t$ as well as the predicted EPMs $y_{ai}$ for other similar map sheet images $I_i$, transformed using the previous stages of our pipeline, so they are aligned with $y_t$. Each edge image aligned with $y_t$ (corresponding to different editions of the map sheets) are represented as $y_{ai} \in \mathbb{Z}^{H,W,N}, i \in [1, N]$ and its binarized version is calculated in Equation (6.5)[1].

$$\hat{y}_{ai} = 1 \text{ where } y_{ai} > 0.5, 0 \text{ elsewhere.} \tag{6.5}$$

Finally, the consolidated EPM $\hat{y}_{tr}$ is calculated as follows:

$$\hat{y}_{tr} = 1 \text{ where } \hat{y}_t + \sum_{i \in [1,\ldots,N]} \hat{y}_{ai} > \tau, 0 \text{ elsewhere.} \tag{6.6}$$

[1] We choose value of threshold as 0.5 for binarization of image.

## 6.4 Experimental settings and results

To be able to evaluate the alignment quality, we selected the map sheet number [2] from the 1898 edition as the target image whose segmentation need to be improved. We chose this image because some ground truth segmentation was manually created for it. Then, we sequentially align the map sheets from the 1888, 1895, 1909, and 1912 editions against this target image. All the maps are aligned to the map in year 1898 which is the middle of time sequence of historical map atlases which can minimize the effect of object changes.

In our experiments to train the map alignment, an ADAM optimizer [192] is used with a learning rate of $1 \cdot 10^{-6}$ and a batch size of 1. We select the best alignment results of two images with the highest number of inlier scores in the feature space generated by the pre-trained network of the refined alignment module.

**Evaluation matrices** We use pixel and COCO panoptic evaluation protocols to evaluate our edge consensus strategy in terms of improving both correctly detected pixels and segmentation quality in historical map segmentation. The COCO panoptic evaluation is explained in Section 2.5. The pixel evaluation protocol includes *Precision*, *Recall* and *Fscore* measures which are calculated based on $TP$ true positive; $TN$: true negative, $FP$: false positive and $FN$: false negative:

$$Precision = \frac{TP}{TP + FP}, \tag{6.7}$$

$$Recall = \frac{TP}{TP + FN}, \tag{6.8}$$

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \tag{6.9}$$

### 6.4.1 Pixel-based analysis

Based on the pixel evaluation results in Table 6.1, we note two findings:

1. We monitor that **the fine alignment consistently improves the number of corrected detected pixels in different value of** $\tau$, shown in recall value.
2. However, fine alignment has a lower average F1 score compared without fine alignment, due to the fact that **the refined process added more background noise which lowers the precision value**. More advanced techniques are required to filter the background noises to improve pixel-level accuracy in future studies.

### 6.4.2 Segmentation-based analysis

Based on the *COCO-PQ* results in Table 6.1, we note two findings:

[2] Atlas municipal des vingt arrondissements de Paris. 1898. Bibliothèque de l'Hôtel de Ville. Ville de Paris. http://bibliotheques-specialisees.paris.fr/ark:/73873/pf0000935524

1. **Higher $\tau$ leads to lower *COCO-PQ* values**. It is due to the fact that a higher value of the consensus will increase the noise level in the background.
2. **Unfortunately, fine alignment does not produce better *COCO-PQ* values compared to the one without fine alignment**. There are two reasons. Firstly, there exist object changes in different map series which increases the difficulties for unsupervised fine alignment of two historical maps. Secondly, fine alignment creates poor alignment results which leads to over-segmentations.

To sum up, our proposed edge consensus strategy of improving segmentation results fails due to poor alignment results, unexpected background noise level and object changes between different maps. More advanced strategies for weakly unsupervised image alignment and edge voting are required as future work.

| Fine align. | Extra sheets (ed. year) | | | | Con. | Pixel-based | | | Shape-based | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1888 | 1895 | 1909 | 1912 | $\tau$ | P | R | F | PQ | RQ | SQ |
| ✗ | ✗ | ✗ | ✗ | ✗ | — | **22.81** | 83.47 | **35.83** | **43.52** | **85.27** | **51.03** |
| ✗ | ✓ | ✗ | ✗ | ✗ | 1 | 13.48 | 85.80 | 23.30 | 24.78 | 77.34 | 32.04 |
| ✗ | ✗ | ✓ | ✗ | ✗ | 1 | 13.45 | 85.76 | 23.25 | 22.45 | 76.74 | 29.26 |
| ✗ | ✗ | ✗ | ✓ | ✗ | 1 | 12.96 | 86.64 | 22.55 | 25.49 | 75.70 | 33.68 |
| ✗ | ✗ | ✗ | ✗ | ✓ | 1 | 13.70 | 87.43 | 23.68 | 29.74 | 78.05 | 38.11 |
| ✗ | ✓ | ✓ | ✗ | ✗ | 2 | 10.86 | 87.17 | 19.31 | 22.49 | 74.72 | 30.10 |
| ✗ | ✓ | ✗ | ✓ | ✗ | 2 | 10.00 | 88.29 | 17.97 | 20.38 | 73.79 | 27.62 |
| ✗ | ✓ | ✗ | ✗ | ✓ | 2 | 10.39 | 88.95 | 18.60 | 21.74 | 74.49 | 29.19 |
| ✗ | ✗ | ✓ | ✓ | ✗ | 2 | 10.11 | 88.21 | 18.13 | 20.40 | 73.42 | 27.78 |
| ✗ | ✗ | ✓ | ✗ | ✓ | 2 | 10.38 | 88.94 | 18.59 | 21.66 | 74.10 | 29.23 |
| ✗ | ✗ | ✗ | ✓ | ✓ | 2 | 10.23 | 89.39 | 18.36 | 22.24 | 74.42 | 29.89 |
| ✗ | ✓ | ✓ | ✓ | ✗ | 3 | 8.85 | 89.28 | 16.11 | 19.93 | 72.77 | 27.38 |
| ✗ | ✓ | ✓ | ✗ | ✓ | 3 | 9.06 | 89.92 | 16.46 | 20.52 | 73.02 | 28.10 |
| ✗ | ✓ | ✗ | ✓ | ✓ | 3 | 8.61 | 90.55 | 15.73 | 18.72 | 73.10 | 25.61 |
| ✗ | ✓ | ✓ | ✓ | ✓ | 4 | 7.89 | **91.28** | 14.53 | 18.63 | 72.44 | 25.72 |
| ✓ | ✓ | ✗ | ✗ | ✗ | 1 | 14.05 | 86.02 | 24.16 | 28.78 | 78.66 | 36.59 |
| ✓ | ✗ | ✓ | ✗ | ✗ | 1 | 13.96 | 85.94 | 24.02 | 27.12 | 77.98 | 34.77 |
| ✓ | ✗ | ✗ | ✓ | ✗ | 1 | 13.16 | 86.67 | 22.85 | 27.07 | 76.64 | 35.32 |
| ✓ | ✗ | ✗ | ✗ | ✓ | 1 | 13.08 | 86.51 | 22.72 | 25.85 | 76.72 | 33.70 |
| ✓ | ✓ | ✓ | ✗ | ✗ | 2 | 11.21 | 87.61 | 19.87 | 26.99 | 76.28 | 35.39 |
| ✓ | ✓ | ✗ | ✓ | ✗ | 2 | 10.57 | 88.32 | 18.88 | 24.61 | 74.63 | 32.98 |
| ✓ | ✓ | ✗ | ✗ | ✓ | 2 | 10.55 | 88.19 | 18.85 | 23.94 | 74.45 | 32.15 |
| ✓ | ✗ | ✓ | ✓ | ✗ | 2 | 10.59 | 88.33 | 18.91 | 25.22 | 74.74 | 33.74 |
| ✓ | ✗ | ✓ | ✗ | ✓ | 2 | 10.50 | 88.20 | 18.76 | 23.62 | 74.47 | 31.72 |
| ✓ | ✗ | ✗ | ✓ | ✓ | 2 | 10.34 | 88.46 | 18.52 | 23.13 | 74.56 | 31.02 |
| ✓ | ✓ | ✓ | ✓ | ✗ | 3 | 9.32 | 89.49 | 16.89 | 24.55 | 74.09 | 33.13 |
| ✓ | ✓ | ✓ | ✗ | ✓ | 3 | 9.06 | 89.71 | 16.47 | 22.33 | 73.36 | 30.43 |
| ✓ | ✓ | ✗ | ✓ | ✓ | 3 | 9.06 | 89.74 | 16.46 | 22.64 | 73.71 | 30.71 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 4 | 8.33 | 90.63 | 15.26 | 22.17 | 73.51 | 30.16 |

TABLE 6.1: Pixel and *COCO-PQ* results (in %) for our preliminary edge refinement process from the sheets 1888, 1895, 1909, 1912 to target map sheet edited in 1898 in *Paris Atlas Municipal*. Fine align.: whether fine alignment is used; Con.: value of edge consensus $\tau \in [1, 4]$; P, R, F: precision, recall and F1 score for pixel-based evaluation; PQ, RQ, SQ: panoptic, recognition and segmentation quality for shape-based evaluation.

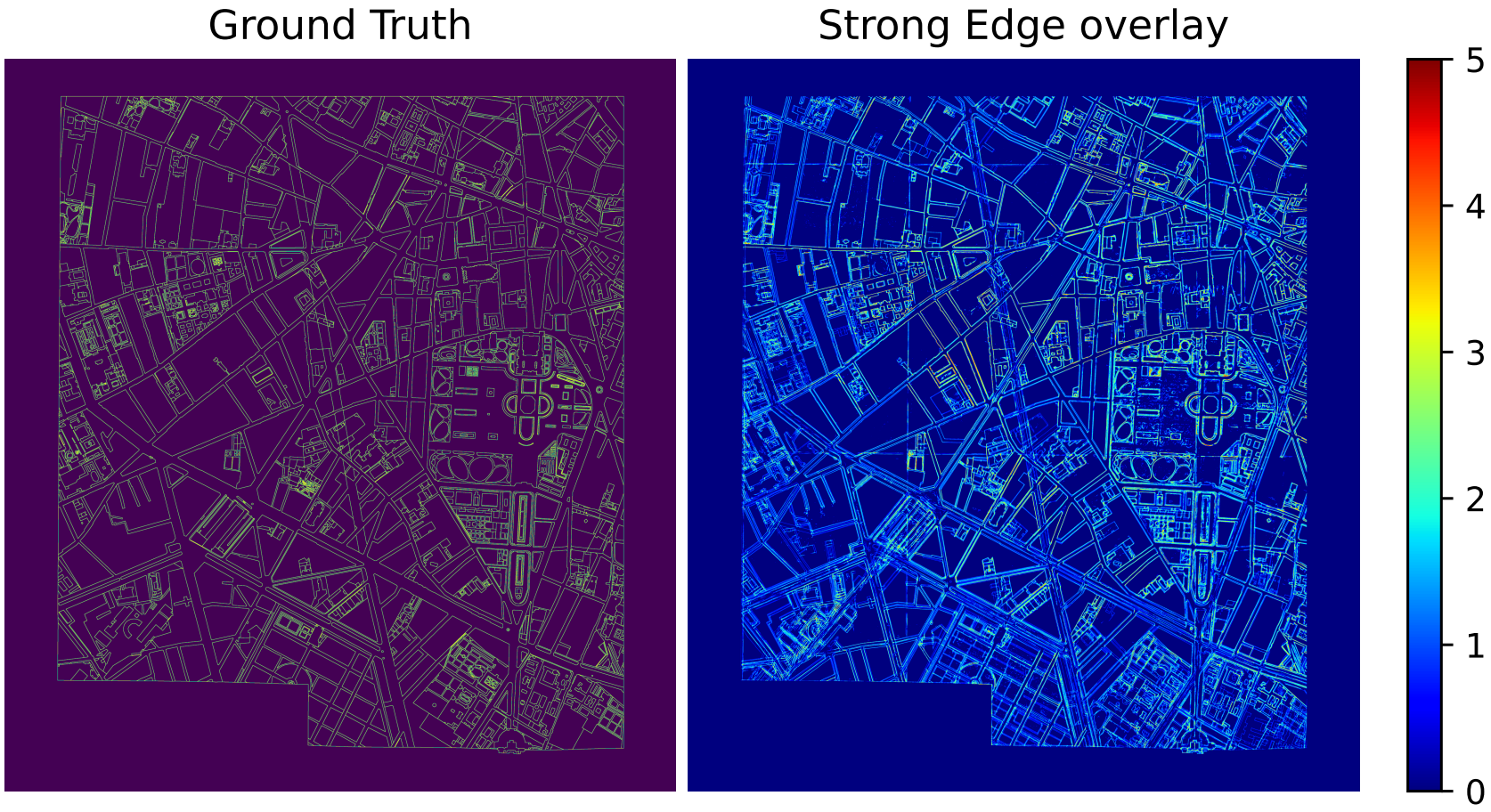



FIGURE 6.8: Maps alignment results for our test image. The right image shows the edge consensus value for each pixel of the aligned image stack. Strong consensus is indicated in red, while dark blue indicates background for all map sheets.

FIGURE 6.9: Edge refinement for an excerpt of our test image. From left to right: ground truth edges; edges predicted using our best deep edge filter; refined edge predictions produced by our consensus refinement considering 4 images for all values of $\tau$. We use coarse with refined alignment in this example.

## 6.5 Perspectives

To unlock the automation of temporal analysis and change detection for large-scale historical maps, we propose a new pipeline, by aligning historical maps to leverage the redundancy information through multiple map EPMs from time to time, and building up the edge consensus to refine the target EPM. This eventually can improve the pixel and segmentation quality in the target EPM. The pipeline consists in three parts, including coarse alignment, fine alignment and edge consensus. Whereas the quality of EPMs, alignment and edge refinement influence the final pixel and segmentation quality of the target EPM that have not been fully evaluated yet due to the missing ground truths and changes of object boundaries (for EPM evaluation), key points (for alignment evaluation) and object changes (for consensus evaluation) in historical map atlases. It is still an open question of which stages are the bottleneck for improving the quality of the target EPM and worth studying in our future work.

In this chapter, we leverage the redundancy information contained in the *Atlas Municipal* by aligning historical map sheets. Although the coarse alignment stage requires very limited graticules ($\leq 9$ points) to align two historical map images, the fine alignment stage does not require any annotated keypoint (which would be difficult to get given the number of map sheets we are dealing with). The alignment maps images can be to compute a pixel-wise **edge consensus** among the predictions. This consensus information is used to determine detect missed edge pixels and bring the back in the predicted EPM, in the hope of repairing broken edges and improving the final resulting segmentation. However, our first baseline approach adds more noise pixels than it recovers missing ones, lowering the global *COCO-PQ* in our quantitative assessment. Based on these findings, more advanced strategies for edge refining would be required.

# Chapter 7

# Conclusions and perspectives

## 7.1 Conclusion

In this thesis, we focused on vectorizing closed shape in late $19^{th}$ and early $20^{th}$ century historical map atlases of Paris. The vectorization step is considered as the pre-requisite for understanding the morphogenesis of the city. Such vectorization is usually manually performed in GIS software which is expensive and have heterogeneous data quality. To tackle this issue, we proposed five research questions to leverage the human annotation effort as well as providing reliable and consistent historical map vectorization outputs for building a geo-historical database which can eventually benefit for multiple research areas. These research questions were separated into **finding a universal solution for historical map vectorization (Chapter 2)** as well as **better filtering unrelated map contents (Chapter 3)** while maintaining **topological properties (Chapter 4)** and **model robustness (Chapter 5)** and **leverage redundancies by using temporal properties in our historical map dataset (Chapter 6)**.

To address our first research question of ***how to design a pipeline that can extract reliably closed shapes from map images***, we proposed a two-stage pipeline (explained in Chapter 2) and explored numerous possibilities on both sides of the framework for automatic and efficient vectorization of historical maps. **Joint optimization** was proved to be an effective tool which improves the closed shape extraction of Edge Probability Maps by using watershed segmentation. We also found that the joint optimization did not always improve the vectorization performance due to the fact that watershed segmentation suffers from the noisy Edge Probability Maps.

We developed our second research question of ***how to find better edge filtering techniques of the historical map images*** in Chapter 3. Multiscale neural networks using pre-train weights generate less background noise in the edge predictions compare to the model without using pre-train weights. Transformer-based architectures with self-attention mechanism and longer pixel context compared to traditional convolutional architecture, and they are proven to be effective in detecting large instances compared to convolutional neural networks. However, they lack the ability in detecting fine-grained local cues which leads to the failure of detecting objects with small areas. The end-to-end learnable watershed level (**Deep watershed**) for adapting the historical map vectorization tasks where it is proved to have worse performance compared to our joint optimization process. The reason is that, **deep watershed** does not guarantee the topology properties in the predicted

likelihood image. Concerning **data augmentation** strategies, contrast stretch combined with thin-plate-spline has the most generalizability.

To answer our third research question about ***how to guarantee the topological properties in the prediction***, several signature loss functions were tested and analysed in Chapter 4 and our newly designed **Pathloss** appeared to perform best loss among all the topology-related variants. Similar to the cell counting problem in biomedical applications, historical maps also have very little texture and object's edges are very thin which will lead to topology failure. We design **BALoss** and **Pathloss** which can activate the boundary and critical pixels in the predicted EPM, eventually enhance the topology properties in EPMs. Moreover, the **Pathloss** successfully extended the **BALoss** [52] for recovering the closeness property of the image segmentation and it was able to penalize the leakages between the neighboring objects to preserve the first dimensional topological structure in the deep image segmentation task. These two loss functions can be applied to improve the perceptual edge detection in the natural, satellite and biomedical images.

The fourth research question ***how to improve the model robustness*** was addressed in Chapter 5. Our contrast modules were built upon the contrast convolution **CConv** with residual, attention, and concatenate operations to merge original inputs and outputs from contrast convolution operations. We proved that simply stacking the contrast modules to existing neural networks can improve the segmentation performance as well as model robustness to the variation of contrast in historical map vectorization task by only adding thousands of parameters to the models.

First attempts to tackle the fifth research question on ***how to leverage the redundancies of historical maps*** were illustrated in Chapter 6. The redundancies of historical maps was leveraged through aligning the historical maps in a unsupervised manner without requiring any ground truth of pixel-correspondence, which is impractical to retrieve in large-scale historical maps. We monitored that the **edge consensus** created by our current voting strategy refined missing edges. However, the voting strategy also created noise which downgraded the performance in extracting polygons as shown by the PQ score. A better voting strategy is required to eliminate the unexpected noise.

To conclude, we wish this thesis work can help researchers to extract high quality vectorizations from historical maps as well as leverage the human annotation effort. Our research is open-source and all the code, dataset, and results are freely available.

## 7.2 Perspectives

This section demonstrates some perspectives and future work of this thesis.

### 7.2.1 Explore all the experiment variants

There are many other variants we did not test in Chapter 3, such as the combination of data augmentation techniques with multiscale and transformer architectures (which requires heavy computation resource and a large datasets) with different topology-preserving loss functions in Chapter 4 or even **CConv**

in Chapter 5. These variants can potentially improve the segmentation quality or model robustness of the historical map vectorization process.

### 7.2.2 Run experiments in larger dataset

Other historical maps such as the Verniquet Atlas, old IGN maps and cadasters also contain rich information for studying urban morphogenesis. Extending our methods in terms of robustness and generalizability for vectorizing those maps is a very challenging perspective.

### 7.2.3 Try to train some CNNs with CConv modules only

In Chapter 5, despite **CConv** improve the contrast robustness by only adding limited parameters, these modules have been only added in an early stage of the neural network. It is an interesting research direction to see whether modules can replace all the convolutional operations in the whole network. This would boost performance while being memory efficient at the same time.

### 7.2.4 Assess how map alignment can facilitate geo-referencing

In Chapter 6, the aligned historical maps are used to refine the quality of map segmentation. Moreover, map redundancies (or aligned maps) can be used to provide fast and better map geo-referencing. The geo-referencing aspects of the map (or map sheet) alignment problem have not been thoroughly discussed in this thesis. Providing better approaches to the automatic geo-referencing of historical maps is a very promising perspective of our work.

### 7.2.5 Improve the redundancy model

In Chapter 6, we presented the voting mechanism to create **edge consensus** for refining the segmentations. However, this voting mechanism brings more noise to the background which decreases the performance of closed shape extraction. One way of solving the background noise is to model the distributions of stable edge probabilities by using some generative model such as probabilistic U-Net [209]. This would enable to learn the voting strategy automatically instead of setting it manually.

Furthermore, we can leverage the temporal model of object changes proposed by Costes et al. [43] to better filter the background noises. The temporal model requires a list of ordered binary values which are easily retrieved through aligned historical maps. Combining the temporal model with our map alignment results is a very promising perspective of our work to better detect object changes in historical maps.

### 7.2.6 Fully explore map redundancies

Redundancies not only appear between different editions of the map atlas, but also exist within each atlas between maps sheets as shown in Figure 7.1. These redundancies can be useful for stitching separate map sheets together.

Moreover, these redundancies can be used as data augmentation to create more diverse training data for improving historical map vectorization.

### 7.2.7 Predict polygons directly

Mentioned in Section 2.2, polygons can be learned and extracted from images by using Polygon-RNN [9], Polygon-RNN++ [68] or Poly-CNN [69]. Moreover, if we add a decoder module to Vision Image Transformer (VIT) or any other transformer-based encoder, we can try to predict a sequence of polygon nodes directly instead of using an RNN structure. Such sequences are considered learnable by using curriculum learning as recently demonstrated by Coquenet et al. [210]. Predicting polygons by using these end-to-end polygon prediction architectures is end-to-end learnable compared to our existing two-stage pipeline. However, these models might generate invalid polygons since it does not have any topological guarantees in training. Tackling this issue is a challenge for extracting closed shapes from historical maps which can be considered as future work of this thesis.

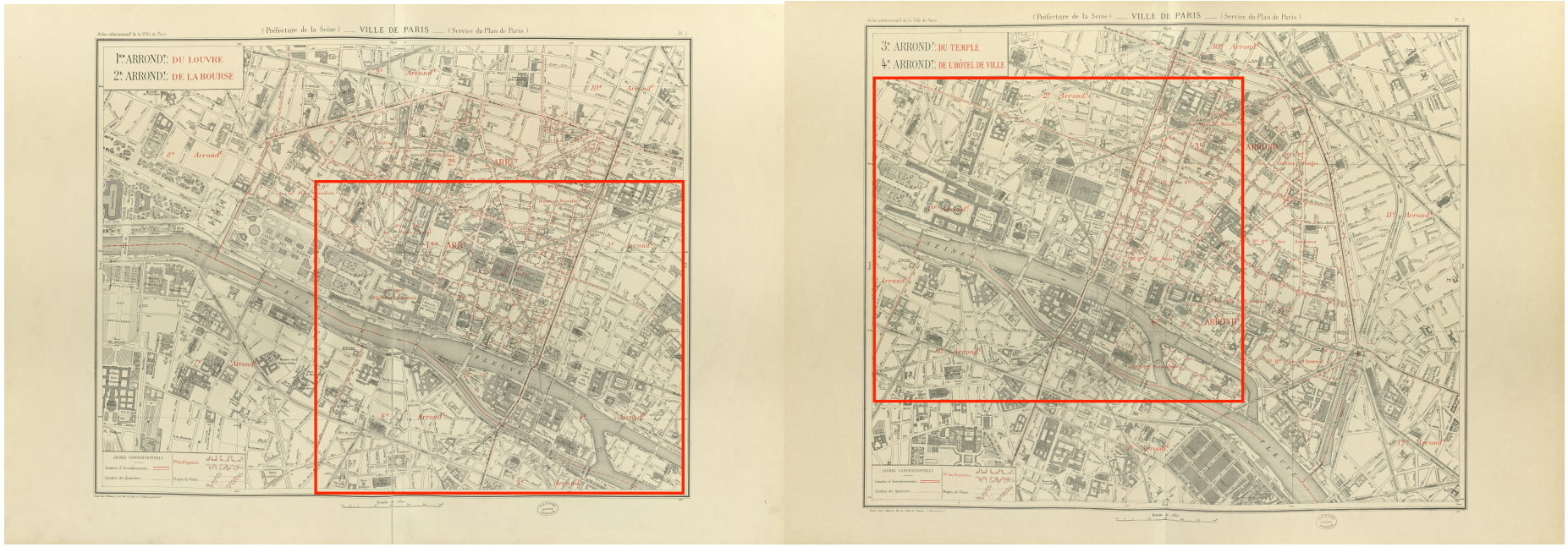



FIGURE 7.1: The red rectangle shows the redundancies information between two maps.

# Appendices

# Appendix A

# Résumé substantiel

## A.1 Introduction

Cette thèse discute de l'importance des cartes historiques pour comprendre les changements dans l'espace géographique au fil du temps et de leur valeur dans les sciences historiques et sociales, l'architecture et l'urbanisme. La numérisation des collections d'archives a augmenté la quantité d'informations géospatiales disponibles. La vectorisation, le processus de transformation de représentations graphiques en données géographiques instanciées, est essentielle pour mieux préserver, analyser et diffuser le contenu à des fins d'analyse spatiale et spatio-temporelle. La thèse expose les défis de la vectorisation et les derniers développements en traitement d'image qui permettent la construction automatique de bases de données géo-historiques, en mettant l'accent sur la détection de formes fermées dans les atlas de cartes historiques de Paris du XIXe et du début du XXe siècle.

Les cartes historiques posent des défis liés à la fois à leur caractère cartographique et à leur caractère historique, du point de vue de la communauté d'analyse et de reconnaissance de documents (DAR). Les défis liés au caractère cartographiques comprennent les variations de sémiotique, le manque d'informations de texture, la palette de couleurs limitée et les objets superposés. Les défis liés au caractère historique incluent le papier endommagé, les lignes non droites, l'incohérence de l'image, les informations manquantes, le changement des propriétés topologiques et les représentations incohérentes des textes manuscrits. Cette thèse vise à accélérer la détection des structures urbaines essentielles et leur processus de géoréférencement tout en assurant une grande précision.

Le principal défi scientifique à relever consiste à extraire automatiquement des formes fermées de haute qualité à grande échelle à partir de cartes historiques. Pour y répondre, le problème est décomposé en cinq questions auxiliaires, qui seront abordées dans des chapitres dédiés. Les questions comprennent la conception d'une chaîne de traitement pour extraire de manière fiable des formes fermées, un meilleur filtrage d'images de cartes historiques, la garantie des propriétés topologiques dans les prédictions, l'amélioration de la robustesse du modèle dans différentes conditions de numérisation et l'exploitation des redondances des cartes historiques.

Les contributions de cette thèse sont énumérées comme suit. Pour vectoriser les cartes historiques, cette thèse propose une chaîne de traitement universelle, axée sur l'extraction de formes closes qui représentent des objets cartographiques variés tels que les bâtiments, les îlots, les jardins ou les cours d'eau pour l'étude de la morphogenèse urbaine. L'approche proposée

combine les forces des réseaux de neurones convolutifs et de la morphologie mathématique pour l'extraction de formes fermées. Un benchmark est introduit pour la tâche de vectorisation de cartes historiques, comprenant une comparaison approfondie des détecteurs de contours profonds avec des fonctions de perte préservant la topologie, une optimisation conjointe des étapes de détection de contours et d'extraction de forme, et une étude des effets des techniques d'augmentation. De plus, deux nouvelles fonctions de perte orientées topologie sont proposées pour préserver les propriétés topologiques de la segmentation des cartes historiques : *BALoss* et *Pathloss*. Enfin, un nouveau bloc de convolution de contraste (*CConv*) est proposé pour améliorer la robustesse du modèle dans la tâche de vectorisation, en se concentrant sur la détection de structures curvilignes.

## A.2 Conception d'une chaîne de traitement pour la vectorisation de cartes historiques

Les chaînes de traitement à une seule étape existantes présentent plusieurs limitations. L'approche de détection de contour implique de combiner l'extraction de contour avec la simplification de polygones, mais elle présente des inconvénients comme une qualité limitée des polygones provoquée par une instabilité des cartes de classification et la nécessité d'un processus de raffinement manuel. Bien que l'approche de détection de polygones de bout en bout comprenne des réseaux tels que Polygon-RNN, Polygon-RNN++ et PolyMapper, ces réseaux peuvent générer des polygones invalides en raison de l'auto-intersection des sommets et des chevauchements de lignes. Cela pose un défi pour l'ajustement fin de ces méthodes pour les cartes historiques, que nous avons laissé pour des travaux futurs. L'approche de partition de polygones consiste à utiliser des outils tels que LSD et à construire un diagramme de Voronoï ou à utiliser KIPPI, mais ces techniques dépendent fortement de la qualité des segments de ligne et ne sont pas suffisantes pour vectoriser des objets avec des frontières courbes. Ces limites rendent ces approches insuffisantes pour la vectorisation de cartes historiques.

La conception d'une chaîne de traitement à deux étapes est proposée pour extraire les structures géométriques à partir de scans de cartes historiques. Les approches de segmentation sémantique traditionnelles sont insuffisantes en raison du contenu limité en texture et en couleur des données. Au lieu de cela, le problème est formulé comme une tâche de segmentation d'instance qui peut être transformé en une tâche d'extraction de contour et de détection de formes closes. La chaîne de traitements proposée combine une étape de détection de contours basée sur un réseau de neurones convolutifs (CNN) avec une étape d'extraction de formes fermées basée sur la morphologie mathématique pour combiner des forces des deux stratégies. Elle est supervisée et tire profit à la fois des cartes vectorisées de référence et des architectures CNN pré-entraînées qui sont disponibles. L'avantage principal de l'approche proposée est sa faible sensibilité au bruit dans les cartes et la fourniture de primitives plus saillantes et robustes pour l'étape d'extraction d'objet subséquente. L'image d'entrée de l'étape de *Watershed* (issu de la morphologie mathématique) est une image à canal unique d'activations de

frontière. L'efficacité de cette approche a été démontrée dans divers travaux. La méthode proposée est évaluée quantitativement en utilisant la métrique *COCO Panoptic* (COCO PQ), et montre des résultats supérieurs à ceux des chaînes de traitement à une seule étape.

## A.3 Apprentissage des contours à travers des architectures de réseaux neuronaux profonds

Différentes techniques sont comparées, et il est constaté que les architectures de réseaux neuronaux multiéchelle sont moins performantes que U-Net, architecture de référence, à cause d'un phénomène de sur-apprentissage. Les architectures de type *transformer* fonctionnent mieux pour les objets de plus grande taille, tandis que U-Net est plus performant pour une large gamme de formes. Les *deep transformers* ont en effet une faible capacité de généralisation et ne garantissent pas la production de formes fermées. Les techniques d'augmentation de données améliorent la capacité de généralisation de la chaîne de traitement proposée lors du traitement de nouvelles images de cartes historiques.

La tâche de détection des contours sémantiques des cartes historiques est difficile, et les méthodes traditionnelles utilisant le gradient de couleur et l'apprentissage de caractéristiques échouent souvent. Les méthodes basées sur l'apprentissage profond ont été développées pour extraire des contours sémantiques de haut niveau en combinant des caractéristiques de bas et haut niveau. Les architectures de réseaux neuronaux profonds multiéchelle tels que HED, RCF et BDCN ont réussi à fusionner les caractéristiques de bas et haut niveau et à obtenir des résultats de pointe dans les applications de détection de contours. Ces architectures ont également été appliquées à la tâche de vectorisation de cartes historiques.

Les architectures de réseaux neuronaux convolutifs sont limitées par deux éléments dans la détection de structures linéaires à partir d'images possédant des propriétés topologiques : leur champ réceptif est limité spatialement, et les cartes de caractéristiques qu'ils construisent présentent des discontinuités. Pour résoudre ces problèmes, nous proposons l'utilisation d'architectures de transformateurs telles que Vision Image Transformer (ViT) et Pyramid Vision Transformer (PVT) qui ont des champs réceptifs plus larges et permettent, théoriquement, de capturer des dépendances spatiales plus longues.

Nous discutons des défis liés à l'application des techniques d'augmentation de données aux images de cartes historiques, car certaines transformations peuvent rompre les limites d'objets et causer d'autres problèmes. Au lieu de cela, nous proposons d'utiliser un sous-ensemble sûr de techniques incluant l'étirement de contraste et les transformations géométriques, pour imiter les variations des différentes conditions de numérisation des cartes historiques.

La technique de *Deep Watershed* surmonte les limites des techniques classiques du *Watershed* en apprenant directement les iso-niveaux discrets à partir d'images multicanaux. Pour capturer les dépendances à longue distance entre les pixels et la distance aux contours des objets, une étape intermédiaire a été introduite dans laquelle un champ de direction est appris pour imiter

la propagation de front à la base de l'algorithme du *Watershed*. Cette architecture intégrée évite le besoin de connaissances préalables sur les attributs de filtrage et leurs valeurs optimales, mais ne présente qu'une performance limitée en pratique.

En résumé, nous comparons différentes techniques pour filtrer les contours et extraire des formes fermées à partir de cartes historiques. Les architectures multiéchelle donnent de moins bons résultats que U-Net à cause de problèmes de sur-apprentissage, bien que l'utilisation de réseaux pré-entraînés réduise le bruit sur les contours prédits. Les architectures de transformateurs fonctionnent bien pour les objets de grande taille, mais U-Net est meilleur pour une plus large gamme de formes, ce qui se traduit par des performances globales supérieures. Le *Deep Watersed* présente une faible capacité de généralisation et ne peut pas garantir des formes fermées. Les augmentations de données en utilisant l'étirement de contraste et les transformations géométriques améliorent considérablement la capacité de généralisation de la chaîne de traitement proposée pour les images de cartes historiques inconnues.

## A.4 Fonctions de perte sensibles à la topologie

Nous décrivons le développement de nouvelles fonctions de perte pour préserver les propriétés topologiques des images de contours prédites, afin de résoudre le problème de détection manquée de pixels critiques dans les contours des objets conduisant à une défaillance topologique. Deux sections sont présentées. La première détaille les motivations et les mécanismes des fonctions de perte topologiques, et la seconde explore l'utilisation de la connectivité locale des pixels pour améliorer la précision topologique dans les contours prédits finaux. Quatre fonctions de perte orientées topologie (deux existantes et deux nouvelles) sont testées pour maintenir la précision topologique dans la carte de contours prédite.

Tout d'abord, nous discutons de la connectivité des pixels dans les images prédites, qui est importante pour maintenir la topologie correcte. Les réseaux de neurones classiques apprennent les probabilités de pixels individuels plutôt que les probabilités de connectivité entre deux pixels. Pour surmonter ce problème, nous proposons l'utilisation de la *Pixel Connectivity Loss* (PCL) qui prédit la probabilité de connectivité pour chaque pixel individuel, plutôt que de prédire la probabilité de contour pixel par pixel. Le pixel central est considéré comme positif si la probabilité qu'au moins un (exactement un/au plus un) pixel voisin se trouve sur un contour est supérieure à un seuil prédéfini. La méthode PCL est décrite en détail, et ses performances au regard de la métrique *COCO Panoptic* sont rapportées.

Ensuite, nous abordons le défi de maintenir la précision topologique dans les tâches de segmentation d'images, qui ne peut être assurée par l'op-ti-mi-sa-tion de pertes basées sur les pixels seuls. Nous étudions trois méthodes préservant la topologie pour relever ce défi : *TopoLoss* (méthode déjà existante), ainsi que *BALoss* et *Pathloss* (deux méthodes que nous avons proposées). Nous présentons ensuite les contributions de ces deux nouvelles fonctions de perte préservant la topologie, BALoss et Pathloss, qui sont testées dans des tâches de segmentation de neurones et de cartes historiques.

En résumé, des fonctions de perte orientées vers la topologie ou la connectivité des pixels sont évaluées afin d'estimer leur capacité à préserver les propriétés topologiques des contours prédits par un réseau neuronal traditionnel. Sur la base des résultats d'évaluation, nous avons constaté que les fonctions de perte proposées *BALoss* et *Pathloss* peuvent maintenir une meilleure structure topologique dans les prédictions de contours par rapport à d'autres fonctions de perte orientées vers la topologie. Nous avons également noté que bien que la correction de la connectivité des pixels soit fortement corrélée à la précision topologique, les fonctions de perte de connectivité des pixels ont tendance à ne capturer que des informations locales, tandis que la précision topologique exige que les pixels de contours prédits maintiennent un contexte global pour assurer la cohérence.

## A.5 Améliorer la robustesse du modèle des détecteurs de contours profonds

Pour améliorer la qualité des tâches de détection de concours courbes et la robustesse du modèle dans différentes conditions de contraste, nous introduisons un nouveau module appelé convolution de contraste (CConv). La performance de la segmentation des structures curvilignes est évaluée en utilisant le module CConv conçu, qui code et apprend l'information des gradients de pixels à l'intérieur des réseaux neuronaux. Les blocs de contraste sont construits à partir des opérations de convolution, de résidus, d'attention et de concaténation de contraste, qui peuvent améliorer significativement la performance de la prédiction des structures curvilignes dans les tâches de segmentation de cartes historiques. L'inclusion de blocs de contraste dans les réseaux a entraîné une augmentation de la robustesse aux changements de contraste d'image, par rapport aux architectures de base. Cela a été réalisé en ajoutant un nombre marginal de paramètres supplémentaires aux modèles existants. L'invention de blocs de contraste offre une nouvelle direction de recherche prometteuse pour intégrer les propriétés structurales à l'intérieur des architectures de réseaux neuronaux profonds.

## A.6 Exploitation des redondances des cartes historiques

La redondance présente au sein de l'Atlas Municipal de Paris est une propriété utile qui peut être exploitée pour le géoréférencement, l'affinage de la segmentation et l'auto-supervision pour l'entraînement des réseaux de segmentation. Dans cette thèse, les cartes historiques sont alignées pour améliorer la qualité de l'extraction de formes fermées. L'alignement des cartes historiques est difficile en raison de la préservation des propriétés topologiques, de la recherche de correspondances entre des points clés annotés manuellement, ainsi que des motifs répétitifs et du bruit dans les images. Pour relever ces défis, l'alignement géométrique est utilisé au lieu du flux optique, des techniques faiblement supervisées sont utilisées sur la base des cartes de contours prédites par les filtres profonds proposés, au lieu des images de carte RVB, afin de limiter l'impact du bruit dans le procédé. Nous

utilisons une stratégie de vote pour restaurer les pixels de contours manqués en trouvant un consensus entre les contours prédits dans différentes planches. Le but est de maintenir des contours alignés de manière fiable tout en filtrant les autres contours causés par une mauvaise qualité d'alignement ou des changements sur différentes planches. Cette stratégie peut être utilisée pour affiner les contours prédits en fonction du consensus dans les résultats d'alignement de la carte historique. Enfin, la série chronologique de cartes historiques alignées à l'aide de leurs redondances peut potentiellement être utilisée pour affiner la segmentation.

Nous alignons les planches de cartes historiques en exploitant les informations de redondance contenues dans l'Atlas Municipal de Paris. La méthode d'alignement est basée sur l'apprentissage non supervisé qui nécessite très peu de points de contrôle pour un alignement grossier et aucun point de contrôle annoté pour un alignement précis. Les images alignées sont ensuite utilisées pour calculer un consensus de bord pixel par pixel pour réparer les contours interrompus et améliorer la segmentation. Cependant, l'approche de base ajoute plus de pixels de bruit qu'elle ne récupère de pixels manquants, il est donc nécessaire d'utiliser des stratégies plus avancées pour l'affinage des contours.

## A.7 Perspectives

Nous suggérons plusieurs orientations pour les travaux futurs. Premièrement, explorer d'autres variantes d'expériences telles que la combinaison de techniques d'augmentation de données avec des architectures multiéchelles et des *transformers*, ainsi que différentes fonctions de perte préservant la topologie. Deuxièmement, réaliser des expériences sur un ensemble de données plus large, comprenant d'autres cartes historiques telles que l'Atlas Verniquet, les anciennes cartes IGN et les cadastres. Troisièmement, essayer d'entraîner certains CNN avec des modules CConv exclusivement afin de remplacer toutes les opérations de convolution dans l'ensemble du réseau. Quatrièmement, évaluer comment l'alignement des cartes peut faciliter la géoréférencement des cartes historiques. Cinquièmement, améliorer le modèle de redondance en modélisant les distributions des probabilités de contours stables à l'aide d'un modèle génératif et en exploitant le modèle temporel des changements d'objet. Sixièmement, explorer pleinement les redondances de cartes pour assembler des planches de cartes séparées et créer des données d'entraînement plus diverses pour améliorer la vectorisation de cartes historiques. Enfin, prédire directement des polygones à l'aide d'architectures de prédiction de polygones de bout en bout pour générer des polygones valides et extraire des formes closes des cartes historiques.

# Bibliography


[1] Y. Chen, E. Carlinet, J. Chazalon, C. Mallet, B. Duménieu, and J. Perret, "Combining deep learning and mathematical morphology for historical map segmentation," in *International Conference on Discrete Geometry and Mathematical Morphology*. Springer, 2021, pp. 79–92.

[2] Y. Chen, E. Carlinet, J. Chazalon, C. Mallet, B. Dumenieu, and J. Perret, "Vectorization of historical maps using deep edge filtering and closed shape extraction," in *International conference on document analysis and recognition*. Springer, 2021, pp. 510–525.

[3] Y. Chen, J. Chazalon, E. Carlinet, M. Ô. V. Ngoc, C. Mallet, and J. Perret, "Automatic vectorization of historical maps: a benchmark," *Submitted.*, 2022.

[4] S. Leyk, R. Boesch, and R. Weibel, "Saliency and semantic processing: Extracting forest cover from historical topographic maps," *Pattern recognition*, vol. 39, no. 5, pp. 953–968, 2006.

[5] Y.-Y. Chiang, S. Leyk, and C. A. Knoblock, "Efficient and robust graphics recognition from historical maps," in *Intl. Workshop on Graphics Recognition*. Springer, 2011, pp. 25–35.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[7] T. Wade, S. Sommer *et al.*, *A to Z GIS, An illustrated dictionary of geographic information systems*. Esri Press, 2006.

[8] K. Ostafin, D. Kaim, T. Siwek, and A. Miklar, "Historical dataset of administrative units with social-economic attributes for austrian silesia 1837–1910," *Scientific data*, vol. 7, no. 1, pp. 1–14, 2020.

[9] B. Budig, T. C. van Dijk, F. Feitsch, and M. G. Arteaga, "Polygon consensus: smart crowdsourcing for extracting building footprints from historical maps," in *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2016, pp. 1–4.

[10] H. Southall, P. Aucott, C. Fleet, T. Pert, and M. Stoner, "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*, vol. 13, no. 1, pp. 7–28, 2017.

[11] S. Muhs, "Computational delineation of built-up area at urban block level from topographic maps," 2019.

[12] R. Raveaux, J.-C. Burie, and J.-M. Ogier, "Object extraction from colour cadastral maps," in *2008 The Eighth IAPR International Workshop on Document Analysis Systems*. IEEE, 2008, pp. 506–514.

[13] A. Cordeiro and P. Pina, "Colour map object separation," in *Proceedings of the ISPRS Mid-Term Symposium*, 2006, pp. 243–247.

[14] D. B. Dhar and B. Chanda, "Extraction and recognition of geographical features from paper maps," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 4, pp. 232–245, 2006.

[15] Y.-Y. Chiang and C. A. Knoblock, "A general approach for extracting road vector data from raster maps," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 55–81, 2013.

[16] S. Leyk, "Segmentation of colour layers in historical maps based on hierarchical colour sampling," in *International Workshop on Graphics Recognition*. Springer, 2009, pp. 231–241.

[17] J. H. Uhl, S. Leyk, Y.-Y. Chiang, W. Duan, and C. A. Knoblock, "Spatialising uncertainty in image segmentation using weakly supervised convolutional neural networks: a case study from historical map processing," *IET Image Processing*, vol. 12, no. 11, pp. 2084–2091, 2018.

[18] A. Khotanzad and E. Zink, "Contour line and geographic feature extraction from usgs color topographical paper maps," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 1, pp. 18–31, 2003.

[19] Q. Miao, P. Xu, T. Liu, Y. Yang, J. Zhang, and W. Li, "Linear feature separation from topographic maps using energy density and the shear transform," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1548–1558, 2012.

[20] D. Schemala, "Semantische segmentierung historischer topographischer karten," *Technische Universität Dresden, Dresden, Germany*, 2016.

[21] J.-M. Viglino and M. Pierrot-Deseilligny, "A vector approach for automatic interpretation of the french cadatral map." in *ICDAR*, 2003, pp. 304–308.

[22] J. Wu, P. Wei, X. Yuan, Z. Shu, Y.-Y. Chiang, Z. Fu, and M. Deng, "A new gabor filter-based method for automatic recognition of hatched residential areas," *IEEE Access*, vol. 7, pp. 40 649–40 662, 2019.

[23] R. Brügelmann, "Recognition of hatched cartographic patterns," *International Archives of Photogrammetry and Remote Sensing*, vol. 31, pp. 82–87, 1996.

[24] J.-M. Ogier, R. Mullot, J. Labiche, and Y. Lecourtier, "Technical map interpretation: A distributed approach," *Pattern Analysis & Applications*, vol. 3, no. 2, pp. 88–103, 2000.

[25] T. Miyoshi, W. Li, K. Kaneda, H. Yamashita, and E. Nakamae, "Automatic extraction of buildings utilizing geometric features of a scanned

topographic map," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3. IEEE, 2004, pp. 626–629.

[26] C. A. Mello, D. C. Costa, and T. Dos Santos, "Automatic image segmentation of old topographic maps and floor plans," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012, pp. 132–137.

[27] S. Banda, A. Agarwal, C. R. Rao, and R. Wankar, "Contour layer extraction from colour topographic map by feature selection approach," in *2011 IEEE Symposium on Computers & Informatics*. IEEE, 2011, pp. 425–430.

[28] E. Katona and G. Hudra, "An interpretation system for cadastral maps," in *Proceedings 10th International Conference on Image Analysis and Processing*. IEEE, 1999, pp. 792–797.

[29] S. Salvatore and P. Guitton, "Contour line recognition from scanned topographic maps," 2004.

[30] N. W. Kim, J. Lee, H. Lee, and J. Seo, "Accurate segmentation of land regions in historical cadastral maps," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1262–1274, 2014.

[31] M. G. Arteaga, "Historical map polygon and feature extractor," in *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*, 2013, pp. 66–71.

[32] H. Yamada, K. Yamamoto, and K. Hosokawa, "Directional mathematical morphology and reformalized hough transformation for the analysis of topographic maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 380–387, 1993.

[33] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, "Raster-to-vector: Revisiting floorplan transformation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2195–2203.

[34] S. A. Oliveira, B. Seguin, and F. Kaplan, "dhsegment: A generic deep-learning approach for document segmentation," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 7–12.

[35] S. Ares Oliveira, I. di Lenardo, B. Tourenc, and F. Kaplan, "A deep learning approach to cadastral computing," in *Digital Humanities Conference*, no. CONF, 2019.

[36] R. Petitpierre, "Neural networks for semantic segmentation of historical city maps: Cross-cultural performance and the impact of figurative diversity," *arXiv preprint arXiv:2101.12478*, 2021.

[37] M. Heitzler and L. Hurni, "Cartographic reconstruction of building footprints from historical maps: A study on the swiss siegfried map," *Transactions in GIS*, vol. 24, no. 2, pp. 442–461, 2020.

[38] S. Wu, M. Heitzler, and L. Hurni, "Leveraging uncertainty estimation and spatial pyramid pooling for extracting hydrological features from scanned historical topographic maps," *GIScience & Remote Sensing*, vol. 59, no. 1, pp. 200–214, 2022.

[39] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[40] SoDUCo, "Compte-rendu intermédiaire du projet anr-18-ce38-0013," ANR, Tech. Rep., 2021.

[41] H. Noizet and E. Grosso, "The alpage project: Paris and its suburban area at the intersection of history and geography (9th-19th century)," in *25th International Cartographic Conference (ICC'11)*, 2011, pp. 1–10.

[42] B. Dumenieu, "Un système d'information géographique pour le suivi d'objets historiques urbains à travers l'espace et le temps," Theses, École des Hautes Études en Sciences Sociales, Dec. 2015. [Online]. Available: https://theses.hal.science/tel-03262391

[43] B. Costes, "Vers la construction d'un référentiel géographique ancien : un modèle de graphe agrégé pour intégrer, qualifier et analyser des réseaux géohistoriques," Theses, Université Paris-Est, Nov. 2016. [Online]. Available: https://theses.hal.science/tel-01565850

[44] B. Dumenieu, N. Abadie, and J. Perret, "Assessing the planimetric accuracy of Paris atlases from the late 18th and 19th centuries," in *Symposium on Applied Computing (SAC 2018)*, ser. Knowledge Extraction from Geographical Data (KEGeoD), vol. 8. Pau, France: ACM Press, Apr. 2018, pp. 876–883. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02388077

[45] R. Cura, B. Dumenieu, N. Abadie, B. Costes, J. Perret, and M. Gribaudi, "Historical collaborative geocoding," *ISPRS International Journal of Geo-Information*, vol. 7, no. 7, p. 262, Jul. 2018, wORKING PAPER. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02388035

[46] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," vol. 39, no. 12, pp. 2481–2495, 2017.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[48] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[49] B. Romera-Paredes and P. H. S. Torr, "Recurrent instance segmentation." Springer, 2016, pp. 312–329.

[50] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5221–5229.

[51] R. Strand, K. C. Ciesielski, F. Malmberg, and P. K. Saha, "The minimum barrier distance," *Computer Vision and Image Understanding*, vol. 117, no. 4, pp. 429–437, 2013.

[52] M. Ô. V. Ngoc, Y. Chen, N. Boutry, J. Chazalon, E. Carlinet, J. Fabrizio, C. Mallet, and T. Géraud, "Introducing the boundary-aware loss for deep image segmentation," in *British Machine Vision Conference (BMVC) 2021*, 2021.

[53] Z. Zhang, S. Fidler, J. Waggoner, Y. Cao, S. Dickinson, J. Siskind, and S. Wang, "Superedge grouping for object localization by combining appearance and shape informations," in *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 3266–3273.

[54] J. Favreau, F. Lafarge, A. Bousseau, and A. Auvolat, "Extracting geometric structures in images with delaunay point processes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 837–850, 2020.

[55] A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin, "Diffusion curves: A vector representation for smooth-shaded images," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–8, 2008.

[56] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[57] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bdcn: Bi-directional cascade network for perceptual edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020.

[58] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5221–5229.

[59] L. Xie, J. Qi, L. Pan, and S. Wali, "Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images," *Neurocomputing*, vol. 376, pp. 166–179, 2020.

[60] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[61] C. Rother, V. Kolmogorov, and A. Blake, ""' grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[62] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.

[63] F. De Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun, "An optimal transport approach to robust reconstruction and simplification of 2d shapes," in *Computer Graphics Forum*, vol. 30, no. 5. Wiley Online Library, 2011, pp. 1593–1602.

[64] C. Dyken, M. Dæhlen, and T. Sevaldrud, "Simultaneous curve simplification," *Journal of geographical systems*, vol. 11, no. 3, pp. 273–289, 2009.

[65] M. Li, F. Lafarge, and R. Marlet, "Approximating shapes in images with low-complexity polygons," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8633–8641.

[66] S. Zorzi and F. Fraundorfer, "Regularization of building boundaries in satellite images using adversarial and regularized losses," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 5140–5143.

[67] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.

[68] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[69] N. Girard and Y. Tarabalka, "End-to-end learning of polygons for remote sensing image classification," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 2083–2086.

[70] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1715–1724.

[71] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[72] L. Duan and F. Lafarge, "Image partitioning into convex polygons," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3119–3127.

[73] J.-P. Bauchet and F. Lafarge, "Kippi: Kinetic polygonal partitioning of images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3146–3154.

[74] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[75] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[76] A. Hanbury and B. Marcotegui, "Morphological segmentation on learned boundaries," *Image and Vision Computing*, vol. 27, no. 4, pp. 480–488, 2009.

[77] S. Ares Oliveira, I. di Lenardo, and F. Kaplan, "Machine vision algorithms on cadaster plans," in *Premiere Annual Conference of the International Alliance of Digital Humanities Organizations (DH 2017)*, no. CONF, 2017.

[78] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1964–1971.

[79] Z. Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1589–1596.

[80] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3158–3165.

[81] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1841–1848.

[82] Y. Liu and M. S. Lew, "Learning relaxed deep supervision for better edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 231–240.

[83] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe, "Learning deep structured multi-scale features using attention-gated crfs for contour prediction," *Advances in neural information processing systems*, vol. 30, 2017.

[84] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3828–3837.

[85] Z. Su, W. Liu, Z. Yu, D. Hu, Q. Liao, Q. Tian, M. Pietikäinen, and L. Liu, "Pixel difference networks for efficient edge detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5117–5127.

[86] M. Pu, Y. Huang, Y. Liu, Q. Guan, and H. Ling, "Edter: Edge detection with transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1402–1412.

[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[89] F. Meyer, "Topographic distance and watershed lines," *Signal processing*, vol. 38, no. 1, pp. 113–125, 1994.

[90] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: Thinnings, shortest path forests, and topological watersheds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 925–939, 2009.

[91] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.

[92] M. Couprie, L. Najman, and G. Bertrand, "Quasi-linear algorithms for the topological watershed," *Journal of Mathematical Imaging and Vision*, vol. 22, no. 2, pp. 231–249, 2005.

[93] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[94] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Transactions on image processing*, vol. 4, no. 8, pp. 1153–1160, 1995.

[95] S. Beucher, "Watershed, hierarchical segmentation and waterfall algorithm," 1994, pp. 69–76.

[96] B. Perret, J. Cousty, S. J. F. Guimarães, Y. Kenmochi, and L. Najman, "Removing non-significant regions in hierarchical clustering and segmentation," *Pattern Recognition Letters*, vol. 128, pp. 433–439, 2019.

[97] I. B. Barcelos, G. B. da Fonseca, L. Najman, Y. Kenmochi, B. Perret, J. Cousty, Z. K. do Patrocínio, and S. J. F. Guimarães, "Exploring hierarchy simplification for non-significant region removal," in *SIBGRAPI Conference on Graphics, Patterns and Images*, 2019, pp. 100–107.

[98] B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia, "Evaluation of hierarchical watersheds," *IEEE Trans. on Image Processing*, vol. 27, no. 4, pp. 1676–1688, 2017.

[99] J. Chazalon, E. Carlinet, Y. Chen, J. Perret, B. Duménieu, C. Mallet, T. Géraud, V. Nguyen, N. Nguyen, J. Baloun, L. Lenc, , and P. Král, "Icdar 2021 competition on historical map segmentation," in *Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR'21)*, Lausanne, Switzerland, 2021.

[100] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.

[101] J. Chazalon and E. Carlinet, "Revisiting the coco panoptic metric to enable visual and qualitative analysis of historical map instance segmentation," in *International Conference on Document Analysis and Recognition*. Springer, 2021, pp. 367–382.

[102] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[103] M. Pietikäinen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[104] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE, 2006, pp. 182–182.

[105] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3000–3009.

[106] P. Buyssens, A. Elmoataz, and O. Lézoray, "Multiscale convolutional neural networks for vision–based classification of cells," in *Asian Conference on Computer Vision*. Springer, 2013, pp. 342–352.

[107] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *European conference on computer vision*. Springer, 2015, pp. 474–490.

[108] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4380–4389.

[109] Y. Ganin and V. Lempitsky, "$n^4$-fields: Neural network nearest neighbor fields for image transforms," in *Asian conference on computer vision*. Springer, 2015, pp. 536–551.

[110] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[111] J. H. Elder and R. M. Goldberg, "Ecological statistics of gestalt laws for the perceptual organization of contours," *Journal of Vision*, vol. 2, no. 4, pp. 5–5, 2002.

[112] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[113] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[114] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[115] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[116] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[117] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 568–578.

[118] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7262–7272.

[119] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[120] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 175–12 185.

[121] S. Tuli, I. Dasgupta, E. Grant, and T. L. Griffiths, "Are convolutional neural networks or transformers more like human vision?" *arXiv preprint arXiv:2105.07197*, 2021.

[122] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[123] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[124] C. Jiao, M. Heitzler, and L. Hurni, "A novel data augmentation method to enhance the training dataset for road extraction from swiss historical maps," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 423–429, 2022.

[125] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6148–6157.

[126] M. O. V. Ngoc, Y. Chen, N. Boutry, J. Fabrizio, and C. Mallet, "Buythedips: Pathloss for improved topology-preserving deep

learning-based image segmentation," *arXiv preprint arXiv:2207.11446*, 2022.

[127] A. Mosinska, P. Marquez-Neila, M. Koziński, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3136–3145.

[128] X. Hu, L. Fuxin, D. Samaras, and C. Chen, "Topology-preserving deep image segmentation," *arXiv preprint arXiv:1906.05404*, 2019.

[129] A. Mosinska, P. Marquez-Neila, M. Koziński, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3136–3145.

[130] L. Li, M. Verma, Y. Nakashima, H. Nagahara, and R. Kawasaki, "Iternet: Retinal image segmentation utilizing structural redundancy in vessel networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3656–3665.

[131] W. Shen, B. Wang, Y. Jiang, Y. Wang, and A. Yuille, "Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2017, pp. 2391–2400.

[132] M. Januszewski, J. Maitin-Shepard, P. Li, J. Kornfeld, W. Denk, and V. Jain, "Flood-filling networks," *arXiv preprint arXiv:1611.00421*, 2016.

[133] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[134] C. Chen, D. Freedman, and C. H. Lampert, "Enforcing topological constraints in random field image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 2089–2096.

[135] J. R. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. P. King, "A topological loss function for deep-learning based image segmentation using persistent homology," *arXiv preprint arXiv:1910.01877*, 2019.

[136] F. Wang, H. Liu, D. Samaras, and C. Chen, "Topogan: A topology-aware generative adversarial network." Springer, 2020, pp. 118–136.

[137] X. Hu, Y. Wang, L. Fuxin, D. Samaras, and C. Chen, "Topology-aware segmentation using discrete morse theory," *arXiv preprint arXiv:2103.09992*, 2021.

[138] S. Shit, J. C. Paetzold, A. Sekuboyina, I. Ezhov, A. Unger, A. Zhylka, J. P. Pluim, U. Bauer, and B. H. Menze, "cldice-a novel topology-preserving loss function for tubular structure segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 560–16 569.

[139] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2019, pp. 285–296.

[140] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1404–1412.

[141] X. Huang and Y. Zhang, "Water flow driven salient object detection at 180 fps," *Pattern Recognition*, vol. 76, pp. 95–107, 2018.

[142] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, 2020, pp. 1–7.

[143] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," in *Proceedings of the International Conference on Learning Representations*, 2016.

[144] S. Borse, Y. Wang, Y. Zhang, and F. Porikli, "Inverseform: A loss function for structured boundary-aware segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5901–5911.

[145] S. Sommer, T. Fletcher, and X. Pennec, "Introduction to differential and Riemannian geometry," in *Riemannian Geometric Statistics in Medical Image Analysis*. Elsevier, 2020, pp. 3–37.

[146] G. Peyré, M. Péchaud, R. Keriven, and L. D. Cohen, *Geodesic methods in computer vision and graphics*. Now publishers Inc, 2010.

[147] P. J. Toivanen, "New geodosic distance transforms for gray-scale images," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 437–450, 1996.

[148] H. De Baar, "von liebig's law of the minimum and plankton ecology (1899–1991)," *Progress in Oceanography*, vol. 33, no. 4, pp. 347–386, 1994.

[149] M. Kampffmeyer, N. Dong, X. Liang, Y. Zhang, and E. P. Xing, "Connnet: A long-range relation-aware pixel-connectivity network for salient segmentation," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2518–2529, 2018.

[150] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung, "Maximin affinity learning of image segmentation," *arXiv preprint arXiv:0911.5372*, 2009.

[151] D. Oner, M. Koziński, L. Citraro, N. C. Dadap, A. G. Konings, and P. Fua, "Promoting connectivity of network-like structures by enforcing region separation," *arXiv preprint arXiv:2009.07011*, 2020.

[152] S. Nowozin and C. H. Lampert, "Global connectivity potentials for random field models," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 818–825.

[153] Z. Yang, S. Soltanian-Zadeh, and S. Farsiu, "Biconnet: An edge-preserved connectivity-based approach for salient object detection," *arXiv preprint arXiv:2103.00334*, 2021.

[154] J. Mei, R.-J. Li, W. Gao, and M.-M. Cheng, "Coanet: Connectivity attention network for road extraction from satellite imagery," *IEEE Transactions on Image Processing*, vol. 30, pp. 8540–8552, 2021.

[155] J. Yan, S. Ji, and Y. Wei, "A combination of convolutional and graph neural networks for regularized road surface extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[156] Y. C. et al., "Rethinking the pixel contrast in curvilinear structure segmentation," *Submitted.*, 2023.

[157] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua, "Supervised feature learning for curvilinear structure segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2013, pp. 526–533.

[158] P. Bibiloni, M. González-Hidalgo, and S. Massanet, "A survey on curvilinear object segmentation in multiple applications," *Pattern Recognition*, vol. 60, pp. 949–970, 2016.

[159] S. Leninisha and K. Vani, "Water flow based geometric active deformable model for road network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 102, pp. 140–147, 2015.

[160] L. Mou, Y. Zhao, H. Fu, Y. Liu, J. Cheng, Y. Zheng, P. Su, J. Yang, L. Chen, A. F. Frangi, M. Akiba, and J. Liu, "Cs2-net: Deep learning segmentation of curvilinear structures in medical imaging," *Medical Image Analysis*, vol. 67, p. 101874, 2021.

[161] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European conference on computer vision*. Springer, 2004, pp. 469–481.

[162] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[163] L. Roberts, "Machine perception of 3-dimensional solids, optical and electro-optical information processing," *J. Tippett. Cambridge, MA, MIT Press*, 1965.

[164] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[165] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[166] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[167] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[168] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[169] X. Hu, F. Li, D. Samaras, and C. Chen, in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[170] F. Wang, Y. Gu, W. Liu, Y. Yu, S. He, and J. Pan, "Context-aware spatio-recurrent curvilinear structure segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[171] M. Cheng, K. Zhao, X. Guo, Y. Xu, and J. Guo, "Joint topology-preserving and feature-refinement network for curvilinear structure segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7147–7156.

[172] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[173] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2070–2078.

[174] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[175] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[176] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 764–773.

[177] C. Chen, W. Bai, R. H. Davies, A. N. Bhuva, C. H. Manisty, J. B. Augusto, J. C. Moon, N. Aung, A. M. Lee, M. M. Sanghvi *et al.*, "Improving the generalizability of convolutional neural network-based segmentation on cmr images," *Frontiers in cardiovascular medicine*, vol. 7, p. 105, 2020.

[178] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *arXiv preprint arXiv:1805.12018*, 2018.

[179] É. Puybareau, Z. Zhao, Y. Khoudli, E. Carlinet, Y. Xu, J. Lacotte, and T. Géraud, "Left atrial segmentation in a few seconds using fully convolutional network and transfer learning," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2018, pp. 339–347.

[180] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Deep retinal image understanding," in *International conference on medical image*

*computing and computer-assisted intervention*. Springer, 2016, pp. 140–148.

[181] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[182] N. Ketkar, "Convolutional neural networks," in *Deep Learning with Python*. Springer, 2017, pp. 63–78.

[183] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler, "A higher-order crf model for road network extraction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1698–1705.

[184] Y. Zeng, D. Samaras, W. Chen, and Q. Peng, "Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in N-D images," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 81–90, 2008.

[185] M. R. Oswald, J. Stühmer, and D. Cremers, "Generalized connectivity constraints for spatio-temporal 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2014, pp. 32–46.

[186] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[187] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[188] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[189] L. Cai, Z. An, C. Yang, Y. Yan, and Y. Xu, "Prior gradient mask guided pruning-aware fine-tuning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 1, 2022.

[190] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[191] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE transactions on medical imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[192] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[193] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[194] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[195] X. Shen, F. Darmon, A. A. Efros, and M. Aubry, "Ransac-flow: generic two-stage image alignment," in *European Conference on Computer Vision*. Springer, 2020, pp. 618–637.

[196] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[197] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[198] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 26–33.

[199] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2307–2314.

[200] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 978–994, 2010.

[201] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 7, pp. 1711–1725, 2017.

[202] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4246–4255.

[203] N. Ufer and B. Ommer, "Deep semantic feature matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6914–6923.

[204] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen, "Object-aware dense semantic correspondence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2777–2785.

[205] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn, "Fcss: Fully convolutional self-similarity for dense semantic correspondence," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6560–6569.

[206] S. Kim, D. Min, S. Lin, and K. Sohn, "Dctm: Discrete-continuous transformation matching for semantic flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4529–4538.

[207] D. Novotny, D. Larlus, and A. Vedaldi, "Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5277–5286.

[208] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6917–6925.

[209] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. Eslami, D. Jimenez Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," *Advances in neural information processing systems*, vol. 31, 2018.

[210] D. Coquenet, C. Chatelain, and T. Paquet, "Dan: a segmentation-free document attention network for handwritten document recognition," *arXiv preprint arXiv:2203.12273*, 2022.