School of Computer and Communication Sciences
Master Project

Vectorization of Historical Cadastral Maps by Artificial Intelligence

Student: Shanci LI

Supervised by: Dr.Mathieu Salzmann

September 25, 2023

Computer Vision Laboratory - CVLAB

EPFL

Abstract

The potential of historical cadastral maps for social science research and contemporary public administration is immense. However, these maps, which were created by expert geographers, have intricate topological and symbolic features that make their digitization a laborious task even for specialists. In this work, we present a comprehensive pipeline that automates the process by harnessing the power of artificial intelligence.

In order to obtain vectorized parcels with attributes such as semantic class and index, we employ deep-learning neural networks to eliminate the background noise and decipher complex symbols. Using Geographic Information System (GIS) software, we also develop an annotation workflow that can generate ground truth segmentation in both raster and vector formats. Deformable Convolutional Networks and Vision Transformers are applied to segment the borderline and other semantic elements of the cadastral map. Further enhancement of vectorization accuracy is studied with graph-based methods and active contour models. Moreover, a pre-trained Optical Character Recognition (OCR) model is tested to extract the parcel index. Finally, we aggregate all the extracted information in spatially referenced vector polygon format.

Contents

1 Introduction

Since the introduction of the Swiss Civil Code in 1912, cadastral plans like Fig.1 according to federal standards have been established to create the official cadastral survey. In order to ensure its update, numerous plans and sketches of mutations were drawn up, as soon as the boundary of a parcel changed (due to parcel reorganization, merging, splitting, milestone rectification, etc.). The cartographic writing is handwritten at an early age but later standardized and kept consistent. These paper plans and registers constitute the archives of the parcel development of a territory. They enable the historical reconstruction of the cadastral, as they contain not only parcel boundaries but also other features such as buildings, stairs, rivers, street names, etc.

Figure 1: Cadastral map of Geneva in 1850s: City Center (left) and Countryside (right)

Vectorization is the process of converting rasterized representations of geographic entities, such as maps, into instanced vector data. This data can be manipulated using Geographic Information Systems (GIS) for spatio-temporal analysis. However, vectorization is a challenging task due to the variations in historical maps, such as differences in legend, geographic features, and text fonts [1, 2]. Additionally, objects in historical maps have limited color and texture information, which sometimes creates ambiguities in interpretation. Currently, tracing the evolution of a parcel or a building is a manual task that requires extensive knowledge of the cadastral survey archives.

The conventional digitization process encompasses several steps, including segmenting the distinctive regions depicted in the plan, identifying potential line vertices for polyline creation, forming polygons, and consolidating associated attributes. As a result, a geographically referenced topology with shared edges is generated. Each polygon encapsulates attributes like semantic class, parcel index, and supplementary information. However, the vectorization of a single plan can extend over days [3]. With thousands of plans spanning the nation within a given timeframe, this labor-intensive vectorization approach becomes impractical and unfeasible.

In order to enhance the efficiency of professionals, it is imperative to create and prototype an automated technique for vectorizing cadastral plans. The realm of automated vectorization has been under exploration for over five decades, delving into the capabilities of computer vision and pattern recognition technologies. However, most of these approaches rely on pixel color statistics or gradient features, which confine their applicability to certain spatial and temporal scenarios. Additionally, the varying content of the map necessitates experts to fine-tune the algorithm's hyperparameters accordingly. While these methods do offer partial automation of the vectorization process, they still entail a notable degree of manual intervention.

The fast evolution of deep learning raises the hope for a better solution. Facilitated by convolutional neural networks (CNN), the model can sometimes perform human-level interpretation on the cadastral symbology by learning from human-annotated samples, which enables generalized models applicable to multiple scenarios.

The resource for this project is the cadastral map from the canton of Geneva in the 1850s. All objects on the map are handwritten as shown in Fig.1. The first priority of the project is to digitize: **Parcel**, **Building**, **Road**, **River**, and numerical **Index** if exists. Instances with complicated patterns like streams or walls are not strongly required.

The main contribution of this project is summarized as follows:

1) We first developed a comprehensive method to annotate the cadastral map with both raster and vector format ground truth, which can be generalized to any historical image dataset.

2) A prototype to vectorize cadastral maps automatically is built. Binary and multi-class semantic segmentation are performed for borderline extraction and parcel classification respectively. Besides, a graph-based approach is further adapted to improve the vectorization performance.

3) An open-source text recognition framework is employed and aggregated with the vectorized result.

In Section 3, the project's comprehensive workflow and the employed evaluation metric were introduced. Section 4 delves into the annotation strategy and provides intricate insights into the training datasets. Moving on to Section 5, we initiate the implementation of a baseline binary semantic segmentation model. Subsequently, we explore the cutting-edge segmentation framework and multi-class semantic segmentation. In Section 6, we present an elementary vectorization method grounded in segmentation predictions. Additionally, we introduce a sophisticated approach engineered to enhance performance. Section 7 involves the application of an open-source text recognition framework and the aggregation of preceding outcomes. Within Section 8, we discuss the potential of transfer learning and future work. The concluding remarks for the project are encapsulated in Section 9.

2 Related Work

The digital humanities field has shown increasing interest in the vectorization of historical maps [3, 4, 5, 6, 7]. While cadastral maps and other historical maps are not synonymous, they have many similarities as they are both handwritten historical documents with systematic symbology. Given the scarcity of literature solely focused on cadastral maps, we opted to broaden our scope to all types of historical maps. Furthermore, the vectorization workflow encompasses some intermediate stages. Therefore, this section is dedicated to the exploration of three key subjects: Vectorization of historical maps, Semantic Segmentation, and Closed shape extraction.

2.1 Vectorization of historical maps

The traditional method of vectorizing historical maps is a complex process that requires manual labeling by experienced professionals. However, this approach is too expensive to apply to large-scale databases. To address this issue, automatic solutions have been developed based on traditional methods such as color segmentation, template matching, shape descriptors, or mathematical morphological operators [6, 8, 9], which can efficiently extract high-level features and generate acceptable initial results. Nevertheless, these methodologies typically exhibit tailored designs aimed at specific objectives, such as roads, buildings, or wetlands, rendering them inadequate for effectively addressing other objects depicted on the map. Furthermore, the cartographic representation does not consistently retain robustness, given that maps are occasionally generated without strict adherence to the designed ontology. Potential defects can arise due to pollution or cartographic errors introduced during the manufacturing process. In certain instances, the methodologies lack the necessary intelligence to proficiently analyze complex scenarios wherein symbology intersects with each other.

Figure 2: Automatic vectorized historical cadastral map of Lausanne (DHLAB, EPFL)

Various attempts to automate the vectorization workflow using open-source GIS software have validated the viability of this approach, however, its overall efficiency remains limited. Ionut et al. [5] examined Siegfried's map in Zurich, Switzerland. Using the Geospatial Data Abstraction Library (GDAL/OGR) and the ImageMagick library, they

devised a technique to convert the raster elements such as buildings, rivers, and contours to vector format with morphological operation and outlier elimination. However, manual cleaning and configuration are still mandatory for the method. Drolias et al. [8] implemented a pipeline with QGIS (an open-source GIS platform) to extract the building block from a scanned map. This semi-automated workflow relies on the color features to filter out desired objects. The pipeline has nine steps and the parameters for each step require optimization for desired results. Therefore, despite the support of automatic vectorization workflow, professionals still have considerable work to do.

Automatic vectorization of the cadastral map has not been widely explored by researchers yet. While there are some resemblances between cadastral maps and general historical maps, there remains a substantial gap within the domains. Historical maps have the ability to portray objects using texture and pixel-based features, whereas cadastral maps predominantly rely on boundary and textual information to define object existence. Additionally, the availability of public datasets for historical maps is scarce [3], and it's even scarcer for cadastral maps. The only example of a related achievement is the vectorization of the historical cadastral map in Lausanne [1], as demonstrated in Fig.2, carried out by the Digital Humanities Laboratory at EPFL.

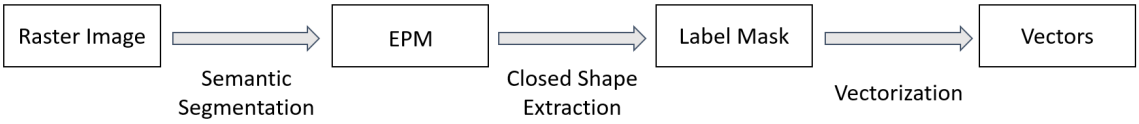

Figure 3: Generic workflow for vectorization of the historical cadastral map

A novel approach to the problem shown in Fig.3 first employs Convolutional Neural Networks (CNN) and deep learning technology [10, 11, 12]. The convolution operation empowers the model to capture the image's low-level representation, mitigating the risk of crafting features that fail to capture the primary target. Facilitated by training data and the backpropagation mechanism, deep learning models with proper loss function autonomously fine-tune both the encoder, responsible for feature engineering and the decoder, responsible for predictions and image reconstruction. Moreover, the application of the residual network structure, as introduced by ResNet [13], serves to minimize information loss during network propagation. By conducting the semantic segmentation task of the historical map, CNNs effectively eliminate noise and generate preliminary outputs, such as an edge probability map or contour mask, at an early stage. This initial output is markedly cleaner and holds more significance.

Subsequently, various closed-shape extraction methods are developed to derive polygon from the interpretation of CNNs. The most common one is the watershed algorithm [14], which treats edge probability values as a local topography (elevation) and floods water from local minima until they meet on the boundary representing segmentation. Traditional computer vision algorithms such as connected component labeling can also be utilized for this purpose [15]. By clustering connected foreground pixels, the contour of the objects lies on the boundary of the clusters. Finally, with a one-pixel-wide raster line mask, vector polygons can be easily acquired.

This pipeline empowers the neural network as an intermediate stage that decouples the vectorization from the original raster cadastral maps, transforming the challenge to finding a generic model that can produce accurate edge probability maps.

However, apart from the above solution, some other strategies are explored as well. Weiwei et al. [16] attempted to align the modern vector map with the georeferenced his-

torical map with reinforcement learning. Zuoyue et al. [7] introduced a novel PolyMapper model with CNN-RNN architecture that directly generates vector lines or polygon results. The model takes aerial images as input and demonstrates promising performance in delineating building blocks and road segments. This method is of great potential as it produces end-to-end vector predictions, which would enable intelligent closed-shape extraction.

2.2 Semantic Segmentation

Leveraging the enhanced computation of the Graphics Processing Unit (GPU), CNN with deep learning has dominated the semantic segmentation task for a long time. With the availability of high-quality large-scale image datasets, data scarcity is no longer a major challenge for general segmentation tasks. Later, the transformer architecture, which emerged from the attention mechanism, revolutionized the field of natural language processing and was soon applied to the domain of computer vision [17]. By enabling global attention, these Vision Transformers (ViT) can capture long-range dependencies and perform adaptive spatial aggregation on the image frame, surpassing the traditional CNNs with restricted kernel size and receptive field. However, as the ViTs increased the model size to billions of parameters, they became extremely computationally demanding. Furthermore, with higher model complexity, ViTs require more data to converge than conventional CNNs. For semantic segmentation, Enze et al. [18] proposed the Segformer model with a simple and efficient design, which achieved state-of-the-art performance on the ADE20K dataset with half the parameters. In addition to the ViTs, Yao et al. [19] attempted to combine the attention mechanism with CNN and developed a deep object attention network for building block extraction. This network shows significant improvement over the winner of the ICDAR 2021 Competition [3] on historical map segmentation who used DenseNet-121 [20]. Besides, Wenhai et al. [21] attempted to gain a dynamic spatial kernel by deformable convolution.

The process of semantically segmenting historical maps deviates significantly from that of natural images. In natural images, semantic details like the sky, roads, pedestrians, or buildings are intricately embedded into the pixel composition of objects. Conversely, in the context of historical cadastral maps, semantics are encoded within the topology structure, rather than pixel features. Simply identifying the pixels that belong to the background or borderline cannot reveal the semantics of a closed shape. Instead, this scenario necessitates a comprehensive interpretation encompassing contextual elements such as surrounding text and shape patterns.

Consequently, ensuring the preservation of pixel connectivity and topology throughout the segmentation process is critical to success. From the pixel perspective, one viable strategy involves the utilization of models like the Conditional Random Field or Markov Random Field [22, 23, 24, 25]. These models possess the capability to integrate pixel relationships with neighboring pixels into the training phase. Another approach is to learn and infer the topology distribution using CNNs like ConnNet [26, 27]. Moreover, [28] and IterNet [29] proposed a methodology that can progressively enhance pixel connectivity through multiple iterations. In terms of preserving topology, Boundary-Aware losses [2] based on persistent homology or minimum barrier distance are designed for the end-to-end deep neural network of segmentation. However, these methods do not guarantee a closed shape as the final outcome.

2.3 Closed shape extraction

Illustrated in Fig.3, the subsequent step following the semantic segmentation involves the extraction of closed shapes. This process is crucial in constructing the intended objects for the eventual vector outputs. Approaches to address this challenge vary considerably, falling into three main categories: image segmentation algorithms, corner point detection, and active contour models.

Image segmentation algorithms The simplest approach for identifying closed shapes involves segmenting the raster image. However, due to the fact that historical cadastral maps aren't segmented solely based on texture, directly applying an image segmentation algorithm on the map doesn't yield the intended topological outcome. Nevertheless, by utilizing prior semantic segmentation of the historical map, a potential solution emerges: conducting segmentation on the resultant edge probability map or the border mask.

Many researchers have opted for the watershed algorithm, a segmentation method originally designed for image processing. This technique, rooted in morphology, considers grayscale images as topographical landscapes wherein pixel values equate to elevations. By virtually pouring water onto this simulated terrain, water surfaces emerging from individual local minima ultimately converge. The boundary formed at their convergence represents the image's segmentation. When applied to historical maps, wherein the edge probability map mirrors the elevation, the algorithm can be customized to effectively differentiate and segment the enclosed shapes presented.

The Felzenszwalb algorithm is another prominent technique employed for image segmentation [30]. Named after its creator, this algorithm focuses on segmenting an image into regions based on variations in intensity and color. Unlike traditional edge-based methods, the Felzenszwalb algorithm considers the global context of the image, striving to group together pixels that exhibit similar properties. The core idea behind is to transform the image into a hierarchical graph structure, where pixels are represented as nodes, and edges are weighted based on color and intensity differences. The algorithm progressively merges smaller regions into larger ones while considering a predefined threshold for the maximum allowable difference in pixel feathers.

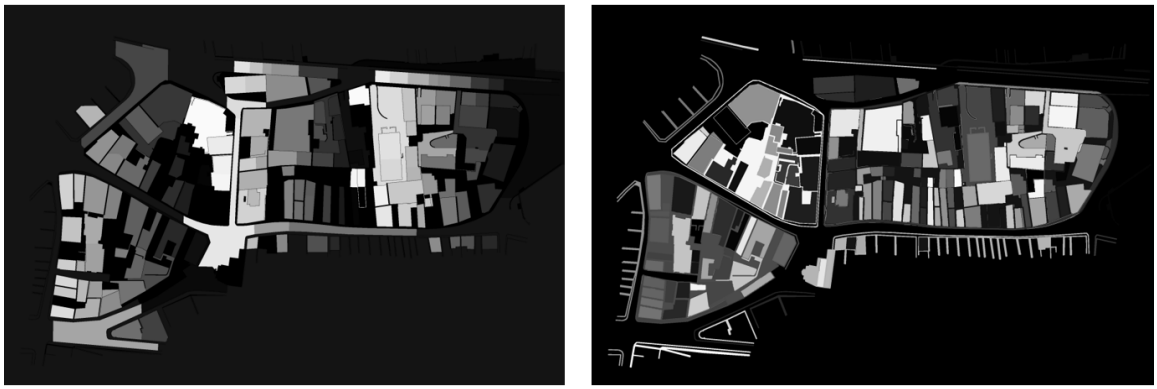

Figure 4: Extracted polygons with watershed (left) and Felzenszwalb (right) algorithms

The watershed and Felzenszwalb algorithms are employed as shown in Fig.4, with promising results. However, both of these algorithms necessitate predetermined parameters related to image features. For instance, the watershed algorithm requires a minimum basin distance, while Felzenszwalb's method involves an observation scale. These parameters highly depend on the user experience, and they can exhibit substantial variation

across different maps. This variability poses challenges when attempting to construct an automated pipeline for large-scale datasets.

Corner point detection In their study of the historical Swiss Siegfried map, Magnus and Lorenz developed an approach that generates simplified yet faithful building shapes [6]. To achieve this, they outline a series of essential steps and techniques based on corner point detection.

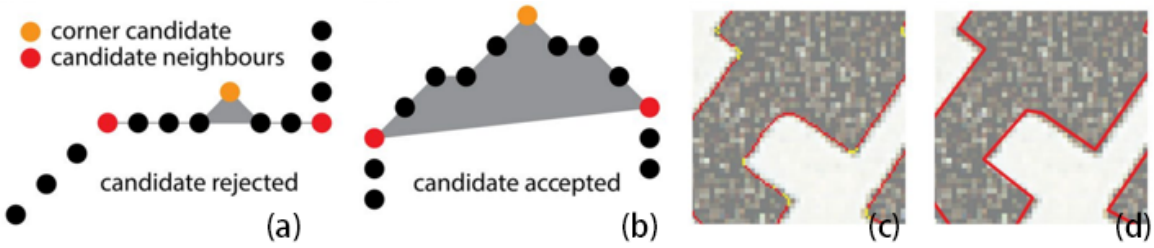

Figure 5: **(a)**Rejected corner point; **(b)**Accepted corner point; **(c)**Before; **(d)**After

The corner detection strategy, illustrated in Fig.5, commences with the creation of a contour for each identified building. This contour, derived through the Canny algorithm, effectively traces the building's perimeter. To address the intricacies of corner identification, a dual-pronged methodology is implemented. First, an angle-smoothing process is employed to detect evident corners. Subsequently, scrutiny is directed toward the sequence of pixels situated between neighboring corners, aimed at identifying any potential corners or inaccuracies. For a more precise corner detection, polygons are formed by utilizing adjacent pixels flanked between each corner pair, depicted in Fig. 5 (a, b). The evaluation hinges on the ratio of the polygon area to its circumference. Pixels surpassing a predetermined threshold are recognized as new corners.

This corner detection technique integrates angle computations, smoothing, and geometric scrutiny, resulting in improved accuracy when identifying corner points. By following this approach, the process contributes to the creation of meticulously defined and representative polygons, capturing the essence of historical building structures.

Active contour models The Graph-Based Growing Contours method is an active contour model designed to extract object boundaries from images with accuracy and efficiency [31, 32, 31]. This technique combines principles from graph theory and image analysis to iteratively refine a contour around the object of interest.

In this method, the image is treated as a graph, with nodes representing pixels or regions, and edges depicting relationships between neighboring pixels as shown in Fig.6. The contour evolves through successive steps, adding new nodes to form a connected boundary that closely follows the object's shape.

The process starts with an initialization step where a seed point or initial contour is placed close to the object's boundary. Then, through contour expansion, neighboring pixels or regions are considered based on edge information. Nodes are strategically added to the contour based on gradients or other image features, facilitating the growth of the contour.

This method's strength lies in its adaptability to irregular object boundaries. It excels where traditional edge-based techniques might falter due to image noise or intensity variations. The Graph-Based Growing Contours method achieves accuracy by continually

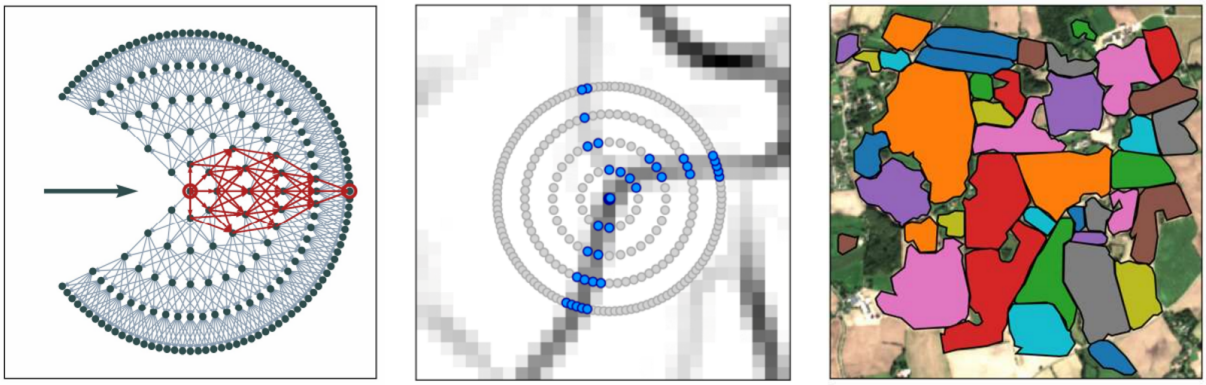

Figure 6: **Left**: contour evolution on the graph; **Middle**: potential split of a contour; **Right**: example of extracted polygons on agricultural field

adjusting the contour's shape based on local image features. However, it's worth noting that the method might encounter challenges in managing potential contour splits. Besides, the extraction always ends with separate polygons, even when there's topological continuity indicated by shared edges. An instance of this behavior is visually depicted in Fig.6, illustrating detected polygons in an agricultural field application [33].

3 Methodology

In this section, we will outline the evaluation metrics to be employed and have an overview of the vectorization workflow, stretching from the initial dataset generation to the eventual achievement of geo-referenced vectorized outcomes.

3.1 General Workflow

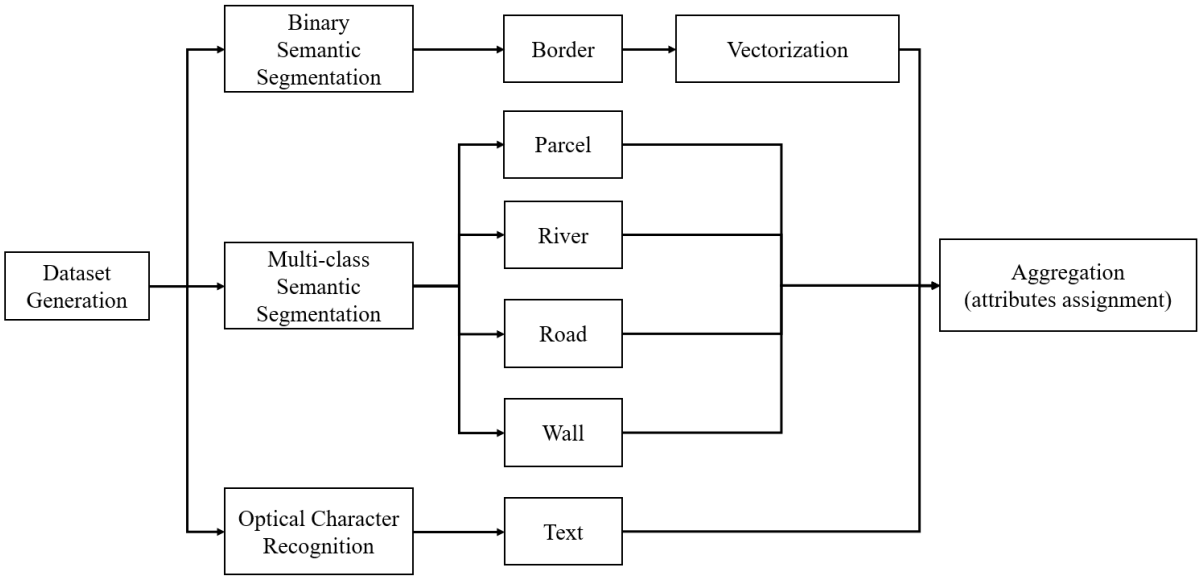

Figure 7: General pipeline of proposed vectorization process

As discussed in Section 2, our innovative approach to cadastral map vectorization leverages artificial intelligence through deep-learning neural networks in the preprocessing. Subsequently, intermediate outcomes are subjected to closed shape extraction and vectorization to yield the desired vector results.

Illustrated in Fig.7, the proposed methodology unfolds with a workflow that initiates with dataset generation and subsequently branches into three distinct modules: Binary Semantic Segmentation, Multi-class Semantic Segmentation, and Optical Character Recognition. Each of these modules is anchored by a dedicated neural network, serving a distinct purpose. The Binary Semantic Segmentation module is geared towards extracting the borderline mask, pivotal for optimal vectorization. The Multi-class Semantic Segmentation module aims to categorize the object semantics of the derived closed shapes. The Optical Character Recognition module centers on the recognition of parcel and building indexes. The information is then integrated as attributes into the vectorized cadastral polygons, which are verified manually by domain experts.

The segmentation task is bifurcated into two separate networks, a strategic move inspired by prior research indicating a substantial decline in segmentation performance with an increasing number of classes. The design of an individual module for borderline detection specifically aims to optimize topology identification and restoration.

Our methodology adheres to semantic segmentation instead of advanced architectures capable of generating vector format outputs. This choice is guided by the project's long-term objective, which aims at not only a method applicable to cadastral maps from the canton of Geneva in the 1850s but also a more universal approach that fits diverse cantons

and time periods. A model relying solely on ontology and symbology to produce vector outputs may not effectively generalize its knowledge from one case to another. Conversely, generic features like borderline and object edges, exhibit a degree of similarity across all maps due to their conceptual and cartographic significance. In this context, a strategy characterized by robustness and compatibility is favored.

3.2 Metrics: IoU and Hausdorff Distance

To assess performance, we introduce IoU (Intersection over Union) and Hausdorff Distance as metrics for comparing various models and experiments, as shown in Fig.8.

IoU In semantic segmentation evaluation, IoU is a crucial metric used to assess the accuracy of a model's segmentation results. It quantifies the overlap between the predicted segmentation mask and the ground truth mask for each class of interest. It's a widely adopted metric in the field of computer vision for evaluating the quality of object localization and segmentation. IoU is computed as the ratio of the intersection of prediction and ground truth to the union of prediction or ground truth. Mathematically, it's expressed as:

$$IoU = \frac{\text{Intersection area}}{\text{Union area}}$$

This value lies between 0 and 1, where higher values indicate better segmentation accuracy. A score of 1 represents perfect overlap between the prediction and the ground truth, while a score of 0 implies no overlap at all. IoU is particularly useful because it considers both false positives and false negatives, providing a balanced assessment of segmentation quality. It's robust even when dealing with imbalanced class distributions.

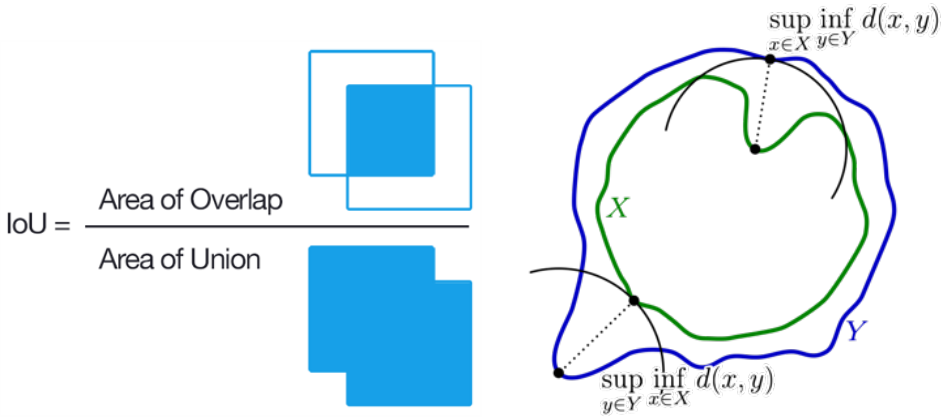

Figure 8: Left: the formula of IoU; Right: the illumination of Hausdorff distance definition

Hausdorff Distance Hausdorff distance is a metric applied in the assessment of vector polygons, designed to measure the dissimilarity between two sets of geometric shapes. In the context of vector polygons, often encountered in tasks like semantic segmentation, Hausdorff distance is used to quantify the degree of variation between a predicted polygon and a ground truth polygon.

Mathematically, the Hausdorff distance (d_H) between two sets of points X and Y can be defined as follows:

$$d_H(X, Y) = \max\left(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\right)$$

Here's a more descriptive explanation of how Hausdorff distance operates in evaluating vector polygons:

Given a predicted polygon P and its corresponding ground truth polygon G, the Hausdorff distance calculates the greatest distance between any point on polygon P and the nearest point on polygon G, and vice versa. It quantifies how much the two polygons differ from each other in terms of shape.

The metric takes into account discrepancies in both directions: from predicted to ground truth and from ground truth to predicted. The larger of these two distances becomes the Hausdorff distance between the polygons. Hausdorff distance is particularly useful in scenarios where accurate delineation of object boundaries is crucial. It's effective in capturing shape variations, distortions, and local deformations between the predicted and ground truth. It's important to note that Hausdorff distance is sensitive to outliers or isolated points that significantly deviate from the overall shape. A single distant point can have a notable impact on the final result.

While Hausdorff distance provides a comprehensive assessment of shape dissimilarity, its computation can be computationally intensive for complex or high-resolution polygons. Nevertheless, its ability to offer detailed insights into shape similarity facilitates its application in our evaluation where precise boundary representation is most significant.

4 Dataset generation: annotation pipeline and datasets

4.1 Previous studies on annotation strategy

4.1.1 Historical Maps

The examination of cadastral map vectorization encounters obstacles due to the insufficiency of datasets. In the context of historical maps, the largest dataset available is the 'Historical City Maps Semantic Segmentation Dataset'. This compilation contains 635 annotated maps, which are featured into 5 distinct semantic classes, namely buildings, non-built regions, water, road networks, and background areas. Notably, around 50% of the dataset comes from the urban area of Paris, while the remaining is sourced from diverse global municipalities.

An additional dataset is the MapSeg dataset, which was published by the ICDAR21 Competition on Historical Map Segmentation. Domain experts expended approximately 400 hours to meticulously annotate 5 map sheets to facilitate the detection of building blocks. This substantiates the considerable workload requisite for manual annotation.

These datasets offer a comprehensive representation of the domain. The former dataset is characterized by pixel-level annotations, thereby furnishing semantic ground truth information. In contrast, the latter dataset is presented in vector, engendering a more application-friendly representation.

4.1.2 Cadastral Maps

In contrast to conventional historical maps, cadastral maps lack distinctive textures attributed to individual categorical entities. Consequently, the interpretation of closed shapes necessitates auxiliary information from a global perspective like neighboring boundaries. Evidently, in the Geneva cadastral map, it is not possible to discriminate the semantics of roads, unbuilt areas, and the center part of the river given only texture, as illustrated in Fig.9.

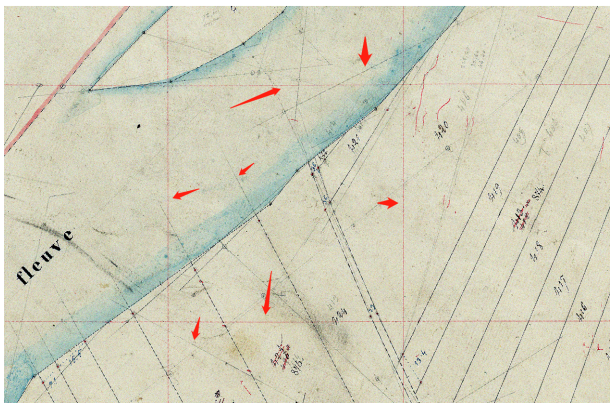

Figure 9: Cadastral map with graticule lines: the texture of road, unbuilt area and center of the river are the same

Nevertheless, there is a scarcity of publicly accessible datasets of the historical cadastral map. Supported by the European Time Machine Project, the Digital Humanities Laboratory at EPFL has conducted several projects with cadastral maps. These projects have been oriented towards devising an automated methodology capable of digitizing cadastral maps from the years 1721 to 1895 in Lausanne (Melotte-Perey; Berney), Venice,

Neuchâtel, and Paris. Their overall objective is to facilitate the reconstitution of a four-dimensional cartographic model of these urban centers.

Debating about the annotation strategy had happened in previous studies. Remi et al. [1] summarized their experience from previous studies on five cadastral map projects and gave the following guidelines:

1. **Annotated objects must be comprehensible and classifiable solely based on visual attributes, disregarding latent semantics.** This guideline underscores the importance of confining annotations to visual properties, such as color, texture, and morphology. This strategy is necessitated by the neural network's lack of interpretation regarding implicit contextual latent. While human annotators can understand implicit meanings, neural networks lack this semantic comprehension. Training the network to rely solely on visible features aligns with its eventual mode of object identification and categorization.

2. **Objects within a semantic category should exhibit distinctive visual characteristics within their class and share reasonable visual features for grouping.** The principle of visually distinct and coherent object representation within a semantic class is advanced to ensure a homogenized visual language, while also demarcating objects from other classes visually. This strategy to retain visual distinctiveness and consistency within each class facilitates the neural network's capacity to classify objects through visual patterns.

3. **Decrease the number of semantic categories to a minimum.** The rationale behind limiting the number of categories resides in mitigating unwarranted complexity. Fragmented categories introduce a high potential for disorientation and misclassification, particularly when addressing intricate or ambiguous entities. A restricted category pool enhances the neural network's efficiency by allowing focused learning of distinct visual attributes associated with a concise collection of classes.

4. **Keep annotation uniformity across instances.** Consistency in annotations serves to ensure equitable categorization of analogous objects across diverse instances. Neural networks thrive on pattern recognition, and incongruities in annotations introduce ambiguity that destroys precise learning. With uniform annotations, the neural network interfaces with explicit and clear training data.

5. **Embrace an iterative approach; validate preliminary outcomes and add more data as needed.** Starting with a limited number of annotations serves as a preliminary stage for model training. As initial outcomes are reviewed, the neural network's proficiencies and limitations are inspected. Iteratively enrich the training data to refine performance.

When considering how to annotate data effectively, it's crucial to utilize raster masks for training semantic segmentation networks. This means that each pixel in the map is aligned with its corresponding label. A common practice involves generating raster masks through image editing tools like Photoshop. However, relying solely on raster masks presents limitations, particularly when evaluating vectorization performance. The evaluation process is confined to the IoU metric, lacking vector ground truth information.

Another challenge associated with raster masks is the complexity of making adjustments compared to utilizing vector data. With vector annotations, tasks such as modifying

boundary widths or rectifying prior annotations are easily achievable. Conversely, these adjustments prove more challenging with raster annotations.

To tackle these issues, we propose an alternative approach: initiate the annotation with vector format and subsequently convert these vector annotations into raster masks. This method allows us to address both network training and vectorization assessment while avoiding the complications linked to manipulating raster annotations.

4.2 Workflow with GIS software

Figure 10: Example for polygon layer with classified objects, which is the desired vectorization result as well

In this project, we opted for ArcGIS Pro as the designated annotation tool, which is a Geographic Information System (GIS) software developed by ESRI, a leading company in the GIS industry. Renowned for its advanced functionalities in spatial analysis, data visualization, and 3D mapping, ArcGIS Pro provides a user-friendly and efficient solution for annotation.

Illustrated in Fig.11 and Fig.10, the annotation process follows a specific workflow, which is further elucidated below:

1. The initiation of the process involves importing a geo-referenced TIFF file into a new *map* project. Navigating through the Content panel and subsequently selecting the *Raster layer* tab, the choice of *None* under *Stretch Type* and setting the *Layer Gamma* to 1.0 ensures that ArcGIS Pro's color management remains consistent with conventional image viewer software. If the initial raster cadastral map lacks geo-referencing, consequently, vector annotations are generated within the raster image framework, utilizing (row, column) as the coordinate system.

2. In the *Catalog* panel, locate the *Databases* folder and the *ProjectName.gdb* file.

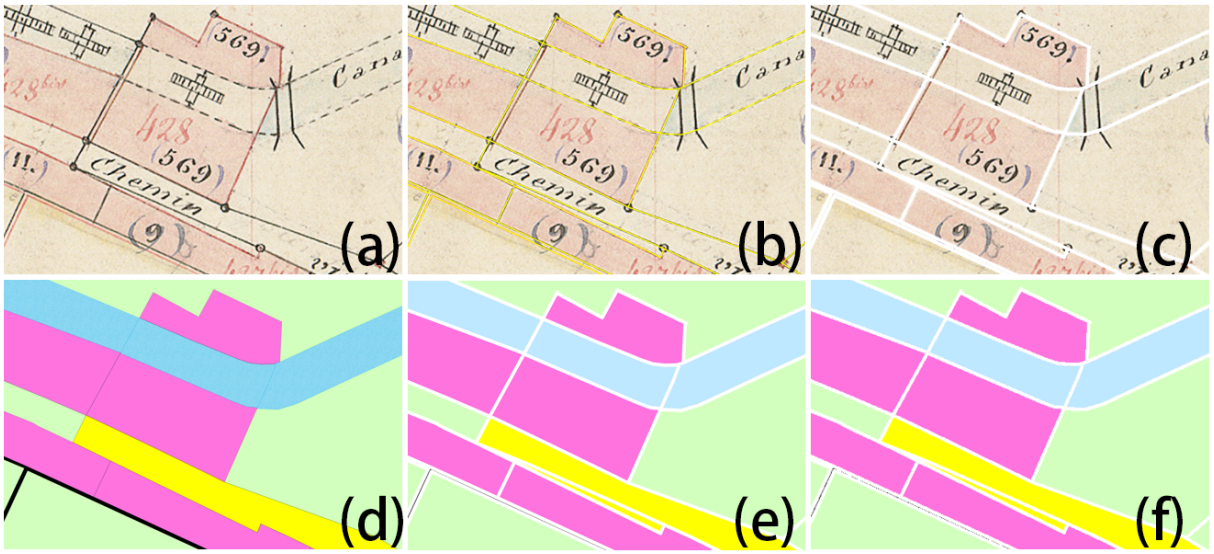

Figure 11: Annotation workflow: (a)initial cadastral map; (b)polyline annotation; (c)buffered borderline polygon layer; (d)constructed object polygon layer; (e)merged vector mask layer; (f)rasterized mask layer

Establish a line feature class within this geodatabase file, creating an additional field to record line width.

3. Transition to the *Edit* tab and activate *Snapping*. Initiate the labeling of borderlines on the cadastral map by selecting the *Create* button. During annotation, adhere strictly to pixel-level ground truth, refraining from correction or human interpretation. A strategic approach emerges here: annotating absent lines or non-existent boundaries that are a part of desired objects and employing a width of 0 for the vector line so that they can be erased later. This trick facilitates the creation of closed shapes within the annotation area, beneficial when topological completeness is desired.

4. Create a polygon feature class in the geodatabase. Create fields related to the desired cadastral map information, such as class, parcel number, building number, text, and time. Subsequently, select all previously created polylines and execute the *construct polygon* function within the *Edit* panel and *Modify* button. Employ the empty polygon feature class as a template and initiate the algorithm. For each polygon, input the corresponding information within its attribute table. Configure the layer's symbology and designate unique colors for distinct semantic classes.

5. Eliminate polylines labeled with the width of 0 from the attribute table. Repeat the previous step, substituting the *Construct Polygon* function with the *Buffer* function. Set an appropriate buffer distance in consideration of the raster borderline's width, enabling *Dissolve*. This results in the creation of the vector polygon layer which represents the borderline.

6. Utilize the *Merge (Data Management Tools)* function within the *Geoprocessing* panel to merge the two polygon feature classes encompassing objects and borderlines. This integration yields a new vector polygon layer, denoted as the 'Mask' layer.

7. Apply the *Polygon to Raster (Conversion Tools)* function within the *Geoprocessing* panel to convert the vector 'Mask' layer into the raster format.

8. Display the cadastral map within the active map window. Employ the *Export map* function under the *Share* tab to independently export the PNG files of the cadastral map layer and the rasterized mask layer with the original resolution. For the latter, generate two PNG files featuring 24-bit and 32-bit color depths respectively.

9. Utilize Photoshop to eliminate the unlabeled region of the cadastral map, leveraging the 32-bit PNG file of the rasterized mask for precise refinement.

Through this workflow, a precise alignment at the pixel level is established between the cadastral map and the raster mask. In addition to acquiring the image in raster format and its corresponding semantic mask, the ground truth data in vector format is also obtained. This vector-based ground truth proves to be particularly advantageous for accommodating potential modifications. For instance, if an alternative annotation strategy were to be explored, like thinning or widening the borderline semantics, the vector ground truth offers greater convenience for implementing such changes, while it is almost not feasible for raster annotation.

4.3 Datasets

In order to assess the inherent generalization capabilities of the semantic segmentation applied to cadastral maps, an exchange of datasets took place between us and DHLAB(EPFL). As a result, two additional datasets from Lausanne and Neuchâtel were obtained.

4.3.1 Pretrain Dataset from Lausanne and Neuchatel

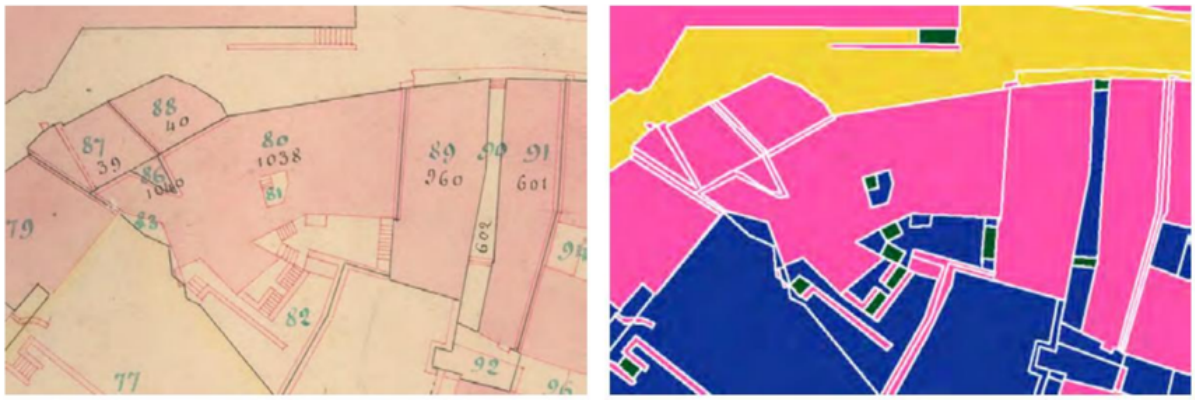

Figure 12: Semantic segmentation dataset example from Neuchatel: borderline - white, road - yellow, building - red, unbuilt - blue, stairs - green

A total of 20 cadastral maps, comprising 10 maps from Lausanne and Neuchâtel respectively, have been employed in this context. The distinctive closed shapes present across these maps have been systematically categorized into six distinct classes, namely background, building, unbuilt, road, stair/brick, and borderline, as visually depicted in Fig.12.

The two datasets originate from the year 1827 and 1869, aligning closely with the time that Geneva cadastral maps were created. As an illustration, consider Fig.12 and Fig.13, which represent cadastral maps from Neuchâtel and Geneva. While distinctions exist in symbology and ontology, there is a significant degree of similarity. The observed similarity raises the possibility of exploring whether the common features present within the cadastral maps could be transferred across different datasets using transfer learning. Should this idea yield encouraging outcomes, there exists a strong likelihood that we could capitalize on pre-existing labeled datasets when confronted with entirely new data. By leveraging the acquired knowledge and features, we could potentially generate preliminary annotations that significantly facilitate the annotation process for the upcoming dataset.

4.3.2 Annotated Dataset: Geneva Dufour Plans

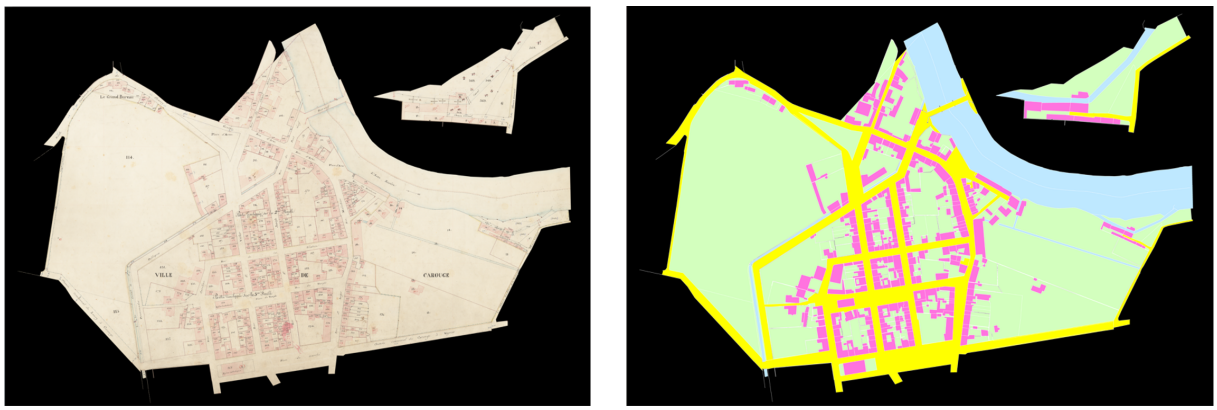

Figure 13: Cadastral map (left) and semantic mask (right) of Geneva dataset

As depicted in Fig.13, the Geneva cadastral map dataset has been meticulously annotated using the workflow described in the previous section, eliminating the area without annotation. Essentially, the procedure followed strictly the guidelines outlined in Section 3.1, despite slight differences. In this dataset, it was decided to classify certain dashed lines as continuous because they form the constituent elements of closed structures such as buildings or streams, and these dashed lines help delineate these topologies. Conversely, dashed lines indicating the distance between objects were omitted from labeling. Besides, pixels are classified across six distinct classes: background, borderline, building, unbuilt areas, roads, and rivers. Furthermore, the vector polygons were labeled with parcel numbers and building numbers.

A total of eight plans from the Geneva dataset were subjected to annotation, as the Geneva maps are significantly larger in comparison to others. While cadastral maps from Lausanne or Neuchâtel usually have dimensions around 3500x4500 or 4500x5500 pixels, Geneva maps consistently exceed 12000x8000 pixels, rendering them four to six times more expansive. The selected annotated maps come from the Carouge and Cartigny communes, representing typical urban and countryside terrain, as evident in Fig.1. The speed of annotation exhibited a correlation with the complexity of map content. For rural regions characterized by large parcels and simple topology, the annotation process demanded approximately 4-6 hours per map. Conversely, urban areas with intricate layouts, such as the commercial center or irregularly dashed stream, necessitated about 2-3 times more effort.

Addressing a prior research concern, potential overlap or omissions arising from the merging of distinct cadastral maps to construct a city scope posed challenges, as highlighted by the yellow arrow in Fig.2. However, this issue did not occur in the Geneva dataset. The geographer highlighted the boundary of the unique area on a cadastral map with thick yellow or red lines. Only the region inside held validity. This insightful indication offered a substantial mitigation for the issue.

5 Semantic Segmentation

Within this section, we reproduced the dhSegment method, initially proposed by DHLAB at EPFL, as our baseline for comparison. Its favorable performance on the Neuchâtel dataset makes it a suitable starting point. However, it's noteworthy that dhSegment relies on the Unet architecture, which is now considered outdated. Thus, following a comprehensive literature review, we implement the InternImage framework, the state-of-the-art semantic segmentation solutions across several large-scale public datasets.

The InternImage framework introduces a novel approach by replacing the conventional convolution kernel with deformable convolution. This alteration enhances the model's receptive field without incurring additional computational costs. Importantly, this framework is compatible with conventional CNNs. We opted to employ UperNet [34] as the backbone model within this framework. Besides, Segformer, a transformer-based segmentation network, is implemented as well. It utilizes the Multi-Head Self-Attention (MHSA) mechanism and can effectively capture global information.

Through rigorous experimentation, we evaluate the performance of both UperNet and Segformer. Comparing the several scenarios, the potential explanation for the observed results is discussed. This exploration aims to reveal the capabilities of these two networks in the context of the semantic segmentation module.

5.1 Baseline: dhSegment

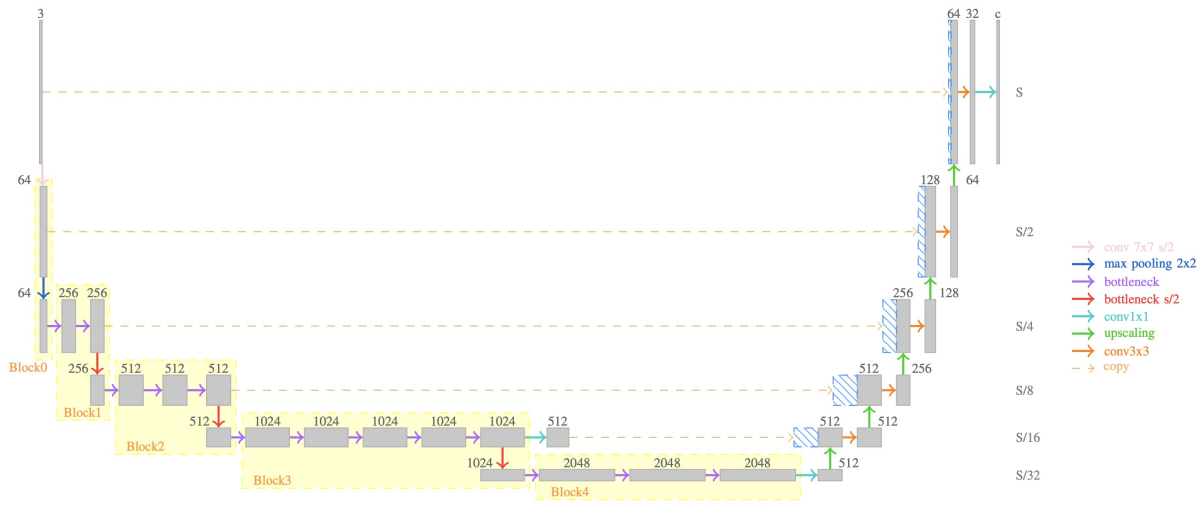

Figure 14: dhSegment network structure

As shown in Fig.14, the dhSegment network architecture consists of two main parts: an encoder and a decoder. The encoder is based on the ResNet-50 architecture and is used for feature extraction. It has five steps, each reducing the feature map size by half. Pretrained models from ImageNet are employed to initiate the weights and enhance the robustness.

The decoder focuses on mapping low-resolution encoder features to full-resolution input features. It includes five blocks and a final convolutional layer for assigning class labels to pixels. Each deconvolutional step involves upscaling the previous feature map, concatenating it with a corresponding residual block, and applying a convolutional layer

followed by ReLU activation. To reduce parameters and memory usage, the feature channels are decreased using 1x1 convolutions before concatenation. Upsampling is done through bilinear interpolation.

5.2 InternImage: Deformable Convolutional Network

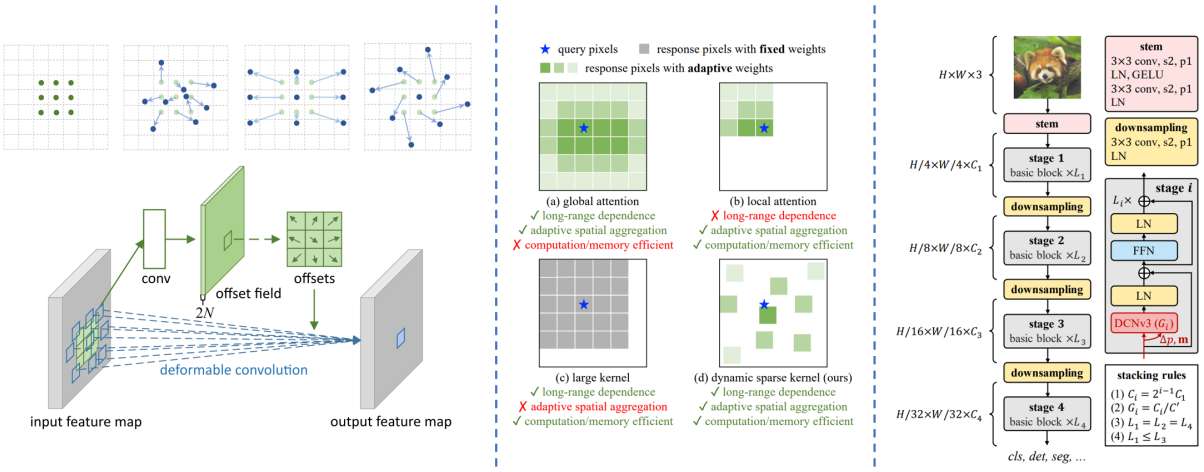

Figure 15: **Left**: illustration of how deformable convolution expands the receptive field with an additional offset kernel compared with normal convolution operator; **Middle**: the features of different kernel operators. (a)multi-head self-attention (MHSA) (b)limited MHSA (c)normal convolution with large kernel size (d)deformable convolution; **Right**: the architecture of the InternImage encoder

The InternImage method introduces a comprehensive approach to constructing large-scale CNN-based foundation models for vision tasks as shown in Fig.15. Central to this approach is the development of a novel convolution variant called deformable convolution v3 (DCNv3) as the core operator. This operator addresses the limitations inherent in traditional convolutions, while also incorporating some of the advantageous characteristics found in attention-based mechanisms like MHSA (Multi-Head Self Attention).

By extending the capabilities of DCNv2 [35], the approach seeks to build a model that is not only efficient but also effective in large-scale vision tasks. This is achieved through a set of key modifications:

- Weight sharing among convolutional neurons is employed to optimize the model's efficiency and performance.
- The introduction of a multi-group mechanism enhances spatial aggregation, enabling the model to learn richer information from different representation subspaces.
- Normalized modulation scalars, implemented along sampling points, contribute to stability during the training process.

The resulting InternImage model takes advantage of the enriched DCNv3 operator and incorporates other advanced components inspired by Vision Transformers (ViTs). These components include Layer Normalization [36], Feed-Forward Networks [37], and GELU activation [38]. The architecture is further augmented by stem and downsampling layers, which contribute to the formation of hierarchical feature maps.

In essence, InternImage presents a powerful framework for developing powerful vision models that can handle large-scale tasks. It capitalizes on the novel DCNv3 operator, integrates it with advanced block designs, and employs scaling strategies to deliver models capable of robustly learning from extensive datasets.

5.2.1 CNN - Upernet v.s. transformer - Segformer

Segformer and UperNet are two distinct semantic segmentation architectures, each designed to excel in capturing intricate image details and providing accurate segmentations, although with different approaches.

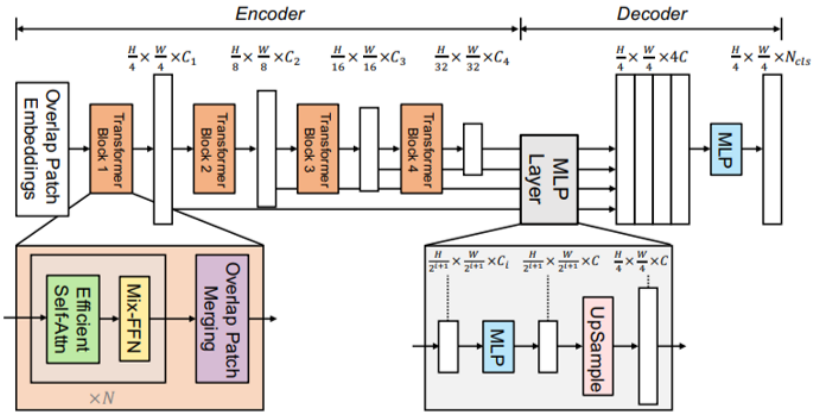

Figure 16: Model architecture of Segformer

Segformer takes an innovative approach by bridging the gap between transformers, originally designed for sequential tasks in natural language processing, and the domain of image segmentation. It introduces a transformer-based framework shown in Fig.16, adapting it to efficiently handle image data. The core of Segformer lies in an axial-attention mechanism, which allows the model to perform self-attention across both rows and columns in an image. This unique arrangement is particularly effective in capturing complex visual patterns. By blending an encoder-decoder architecture, Segformer harnesses the power of transformers to grasp the global context while meticulously refining segmentations in the decoder. This integration of components addresses challenges posed by conventional convolutional methods, mitigating issues such as information loss in deeper layers and the sensitivity of receptive field size.

In contrast, UperNet shown in Fig.17 operates within the domain of traditional convolutional networks, but it introduces an ingenious strategy to enhance semantic segmentation. This approach revolves around multi-scale feature fusion. UperNet excels at capturing both fine-grained image details and broader contextual understanding. The architecture achieves this by integrating features from various levels of a convolutional network. By incorporating lateral connections, UperNet establishes links between low-level and high-level feature maps, facilitating the holistic integration of information across diverse scales. This design proves particularly advantageous for scenarios where objects exhibit a wide range of scales, demanding accurate boundary delineation and simultaneous object category classification.

In summary, Segformer introduces a transformer-driven example to address semantic segmentation challenges, efficiently incorporating global context and localized intricacies. UperNet, on the other hand, relies on the strengths of traditional convolutional networks

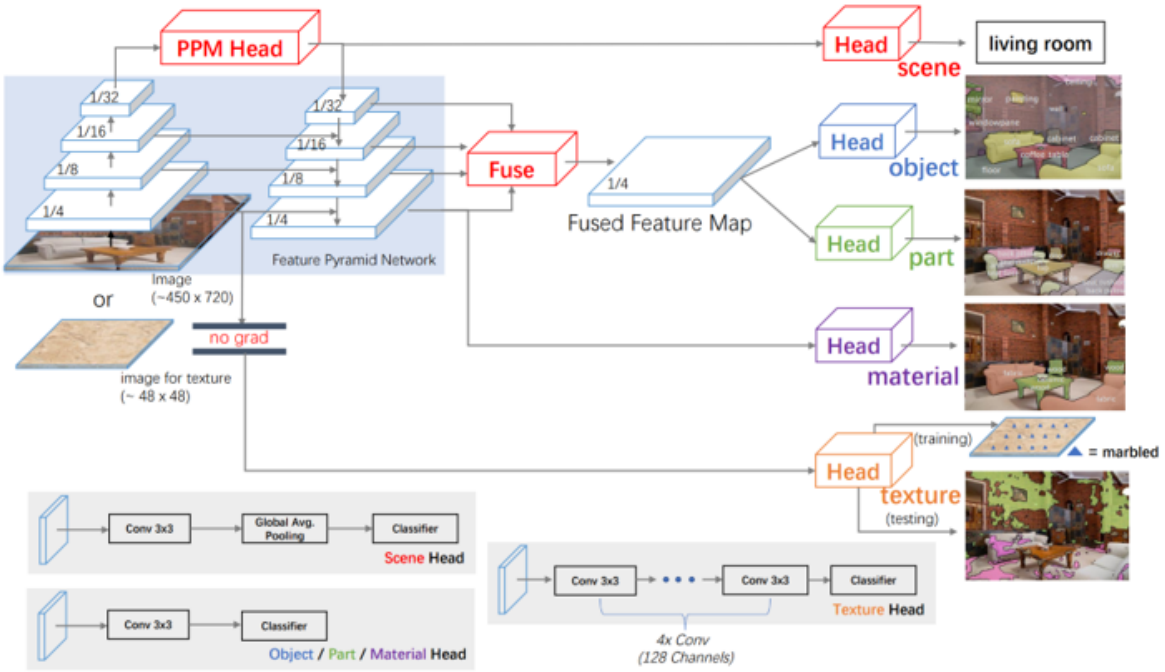

Figure 17: Model architecture of UperNet

while ingeniously fusing features across scales for a comprehensive understanding of complex scenes. Notably, UperNet is further enhanced within the InternImage framework, where it is combined with deformable convolution to harness global context. Consequently, both models demonstrate competitiveness at a comparable level.

5.2.2 Binary semantic segmentation: borderline

As discussed in Section 3, we introduced a dedicated module for binary semantic segmentation. The purpose of this module is to extract the object borders, effectively mitigating the influence from other categories. To evaluate its effectiveness, we conducted experiments involving three distinct networks. These networks are based on dhSegment or InternImage with Segformer or UperNet architectures. We trained these networks on the Geneva dataset.

Moreover, we assessed the potential of transfer learning. For the zero-shot scenario, we trained the network on the Lausanne-Neuchâtel dataset and then directly applied inference to the Geneva cadastral map. This test aimed to test the generalization capability of the borderline segmentation model across different map datasets.

In the case of the finetuned scenario, we followed a two-step process. Initially, the network underwent pre-training on the Lausanne-Neuchâtel dataset. Subsequently, it was finetuned using the Geneva dataset. Our objective here was to leverage the enriched dataset to enhance the network's ability to capture features and latent information from the cadastral map, consequently leading to an improved segmentation performance.

Implementation details Regarding the implementation specifics, the original cadastral map's high resolution poses a challenge due to GPU memory limitations, necessitating a resizing step. The 8 annotated cadastral maps are split into training (6), validation (1) and test (1) sets. Then, all maps are cropped to 1024x1024 resolution for network train-

ing. Furthermore, the InternImage framework demands even smaller input (512x1024) to accommodate GPU memory constraints. Given the use of NVIDIA V100 32GB GPUs, the batch size for Segformer and UperNet models is capped at 2 per GPU. During inference on the complete cadastral map, predictions are obtained patch by patch through sliding windows, with concatenation afterward.

To enhance performance, we incorporated random data augmentation techniques such as rotation, flipping, resizing, and photometric distortion. These techniques bolster the model's adaptability to diverse lighting, contrast, and color scenarios, enhancing its suitability for varying environments.

As for optimization, we opted for AdamW due to its capacity to augment training stability and convergence in neural networks. AdamW directly incorporates weight decay into optimization, restrains weight growth, and sustains adaptive learning rate features. This choice helps in preventing overfitting and improving overall training efficiency.

Quantitative analysis Table.1 presents a comprehensive evaluation of the performance of three models, namely Segformer, UperNet, and dhSegment, across various scenarios. The primary evaluation metric used is the IoU for the "Borderline" class in the Geneva dataset.

Scenery	Model	Borderline - IoU
In domain (Geneva)	dhSegment	0.645
	Segformer	0.711
	UperNet	0.732
Transfer Learning (Zeroshot)	Segformer	0.622
	UperNet	0.649
Transfer Learning (Finetuned)	Segformer	0.721
	UperNet	0.734

Table 1: Comparison of borderline IoU on different models and scenarios

For the in-domain (Geneva) scenario, dhSegment achieves an IoU of 0.645. Segformer achieves a higher IoU of 0.711, indicating improved performance. UperNet outperforms both other models with an IoU of 0.732, showcasing its superiority in the Geneva cadastral map domain. For the Transfer Learning (Zeroshot) scenario, both Segformer and UperNet are evaluated without domain-specific fine-tuning. The IoU of UperNet is about 2% higher than Segformer. The Transfer Learning (Finetuned) scenario involves fine-tuning the models after pre-training, potentially adapting them to the target domain. Segformer achieves an improved IoU of 0.721 while UperNet further elevates its performance to an IoU of 0.734.

Shown in Fig.18, dhSegment provides a baseline performance. However, its performance is surpassed by both transformer and CNN with deformable operators in various scenarios. Segformer performs well across all scenarios, showcasing its versatility and adaptability. It exhibits competitive performance both in-domain and after fine-tuning. UperNet consistently exhibits the best performance across all scenarios. It achieves the highest IoU but only has negligible improvement after adaptation through fine-tuning.

In conclusion, UperNet stands out as the top-performing model and will be chosen for the binary semantic segmentation module.

Qualitative analysis

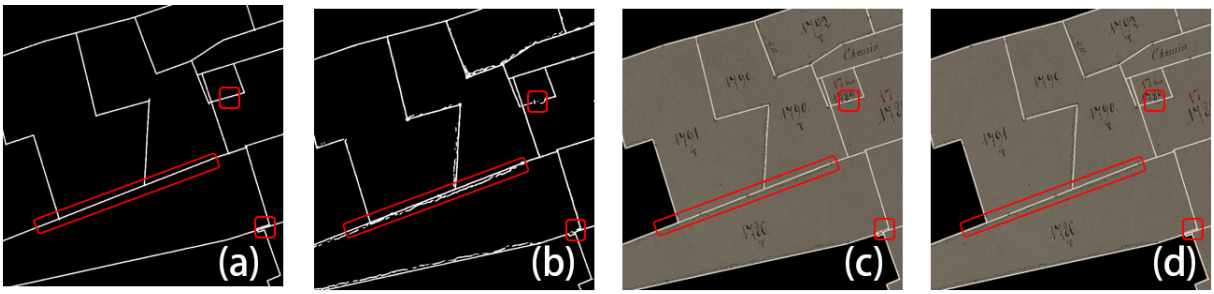

Figure 18: Comparison of predictions on Geneva dataset. **(a)**Ground Truth; **(b)**dhSegment; **(c)**Segformer; **(d)**Upernet

dhSegment Fig.19 depicts the predictions generated by dhSegment on the Neuchâtel dataset. The Neuchâtel dataset showcases the exceptional performance of dhSegment, as highlighted by the red box. The majority of borderline pixels receive accurate positive predictions. However, a minor portion of corner cases exhibits errors due to the intersection of multiple lines or other symbols. The raster mask effectively captures the overall topology, which holds considerable promise for subsequent vectorization processes. Remarkably, in this specific case, dhSegment achieves an impressive IoU of 0.808.

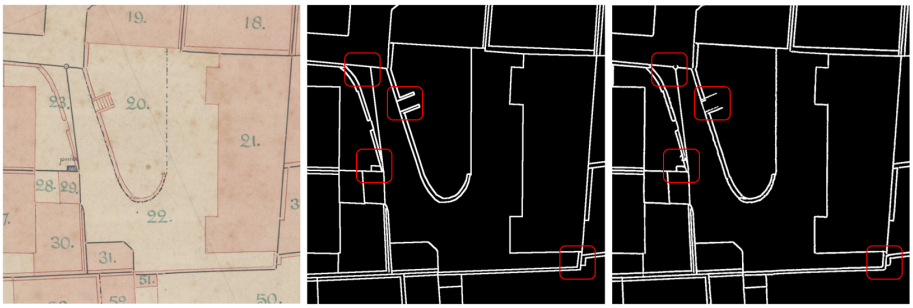

Figure 19: dhSegment - Neuchâtel: Input (left); Ground truth (middle); prediction (right)

Figure 20: dhSegment - Geneva: Input (left); Ground truth (middle); prediction (right)

However, this level of performance is not replicated in the Geneva dataset. The Geneva cadastral maps present intricate patterns and closely adjacent lines, leading to a substantial reduction in dhSegment performance, as evident in Fig.20. For instance, the middle red box illustrates vegetation near the borderline, which becomes indistinguishable from the borderline itself, resulting in numerous false positive predictions. Challenges also arise from cases such as text overlapping with the borderline or the presence of closely adjacent lines, both of which can disrupt the detection of accurate topology. Consequently, the effectiveness of dhSegment performance at this level falls significantly short of expectations.

Segformer and UperNet Despite the minor discrepancy in the IoU metric between these two networks, their overall performance from a qualitative perspective remains remarkably similar. Consequently, we opt to discuss these two networks collectively. As illustrated in Fig.21, both networks surpass dhSegment performance on the Geneva dataset, displaying an improved capability to comprehend intricate background object patterns.

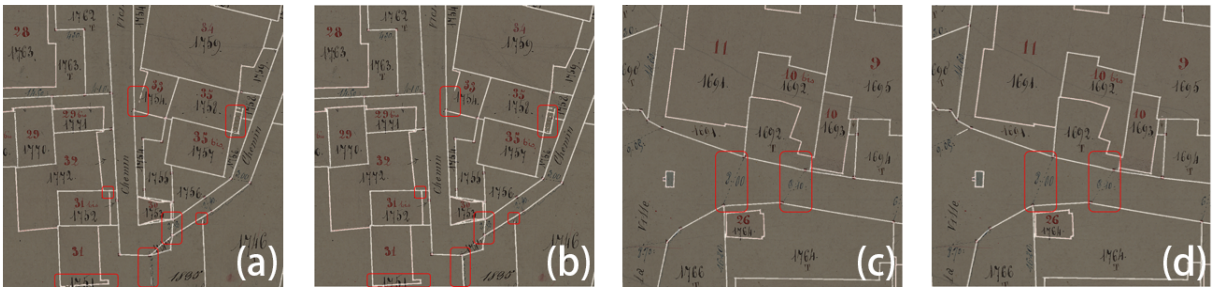

Figure 21: Comparison of predictions with Segformer **(a,c)** and UperNet **(b,d)**.

In the majority of cases, these two models generate accurate predictions for borderlines, effectively mitigating the impact of noise stemming from text and other patterns. The limitations of the dhSegment model are successfully addressed through the integration of more advanced network architecture. Moreover, these networks demonstrate the ability to differentiate between desired dashed lines and undesired ones by learning the insights encoded within human annotations. This level of sophistication surpasses initial expectations.

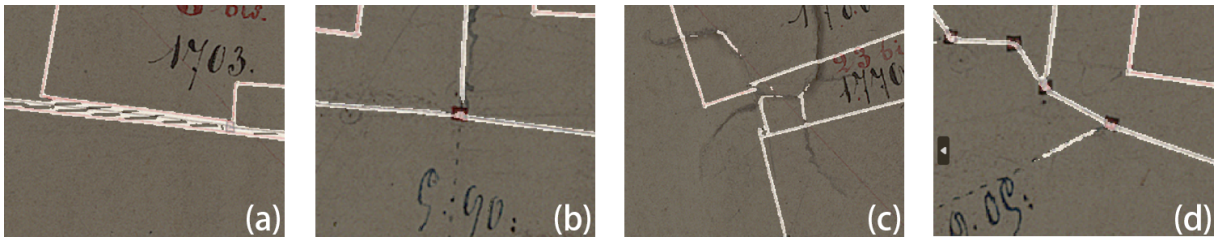

Figure 22: Defects regarding borderline prediction. **(a)**several adjacent lines; **(b)**false negatives - topology broken; **(c)**polluted document; **(d)**false positives for graticule line

However, certain defects surface when the input cadastral map's ground truth falls short of ideal conditions. Shown in Fig.22, false predictions arise with instances including red borderlines with insufficient color depth, intersections accompanied by red-black rectangular symbols, and borderlines that terminate prematurely near intersections. Despite these drawbacks, the overall outcome is promising and sufficiently satisfactory to proceed with the vectorization module.

Discussion and improvement on connectivity We proposed a possible explanation for the minor distinction between Segformer and UperNet. While the transformer architecture of Segformer possesses a more intricate structure and its Multi-Head Self-Attention mechanism captures a broader range of information compared to deformable convolution, theoretically suggesting Segformer's potential for superior performance, it's noteworthy that UperNet consistently outperforms Segformer. It's possible that the design and performance optimization of UperNet, especially with the integration of deformable convolution, contribute to its competitive edge. Another significant factor could be that the

performance of Segformer is possibly constrained by the dataset size. Transformer-based networks typically thrive on large-scale datasets; however, our training stage leveraged only 6 cadastral maps, corresponding to approximately 400 patches with 1024x1024 resolution. This scarcity of data could potentially lead to Segformer underfitting. Benefiting from its more streamlined architecture, UperNet might achieve superior metrics, as visually depicted in Fig.23 (Left).

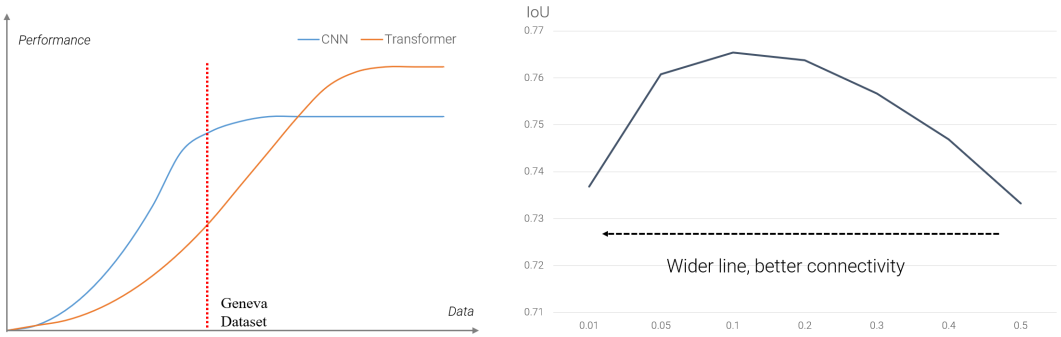

Figure 23: **Left**: possible explanation regarding model performance; **Right**: IoU performance with different prediction threshold

Through experiments, we came to recognize that IoU (Intersection over Union) serves as a comprehensive metric for line segmentation, but its maximization doesn't equate to optimal results for historical map vectorization. The central objective of the segmentation module lies in the detection of the existence of lines, irrespective of their width, with a strong emphasis on capturing the connected topology during the segmentation process. Although IoU is tightly intertwined with this objective, its relationship is not strictly linear or deterministic.

Figure 24: Comparison of prediction with different thresholds. **Left**:0.5; **Middle**: 0.1; **Right**: 0.01

As a consequence, the default decision threshold for translating network output logits into the final binary outcome, conventionally set at 0.5 due to default designations and backpropagation algorithms, may not be optimal. Our perspective shifts from merely minimizing false positives to prioritizing enhanced connectivity and topology, even if it leads to a broader scope of detected lines or an increased count of false positives. The right side of Fig. 23 showcases our exploration of how predictions change when adjusting the decision threshold, and examples of these alterations can be observed in Fig. 24. Fundamentally, lowering the threshold results in a slightly expanded prediction area, yielding thicker lines and improved connectivity. The overlap between ground truth and prediction increases initially, followed by an increase in the union area, causing the IoU metric to ascend and subsequently descend as the decision threshold decreases. Consequently, a lower threshold aligns better with our goal by retaining more topology information within

predictions. The trade-off is that adjacent line predictions might merge, but the challenge of adjacent lines remains significant regardless of this adjustment.

5.2.3 Multi-class semantic segmentation

Multi-class semantic segmentation serves the purpose of categorizing vectorized polygons based on their semantic classes. The determining factor for this module's effectiveness lies in the overall accuracy of predictions at the pixel level. As such, the most relevant metric in this context is **Accuracy**.

In adapting the networks, we opted for a straightforward modification of models designed for binary semantic segmentation. The non-uniformity of semantic classes across different datasets and variations in ontology deter the application of transfer learning for multi-class semantic segmentation in this context.

Table 2: Comparison of Segformer and UperNet for multi-class semantic segmentation

Segformer - iteration 60,000			UperNet - iteration 108,000		
Class	IoU	Acc	Class	IoU	Acc
Background	99.65	99.88	Background	99.66	99.85
Borderline	69.43	72.35	Borderline	71.05	74.58
Building	87.86	88.74	Building	89.19	90.09
Unbuilt	93.73	99.67	Unbuilt	93.87	99.74
Wall	44.86	60.19	Wall	50.57	64.69
Road	75.22	77.2	Road	74.32	75.7
River	71.32	97.86	River	66.31	98.58
aAcc	mIoU	mAcc	aAcc	mIoU	mAcc
95.3	77.44	85.13	95.41	77.85	86.17

The comprehensive comparison of multi-class semantic segmentation performance between Segformer and UperNet is presented in Table.2. Segformer reached its optimal performance at iteration 60,000, whereas UperNet achieved its peak at iteration 108,000, implying that Segformer might converge faster and potentially face underfitting issues as we proposed in the previous section.

Evidently, both models excel in accuracy for the Background and Unbuilt classes, both surpassing 99% accuracy. Across all classes except for Road, UperNet consistently outperforms Segformer in terms of pixel accuracy, consequently being chosen as the preferred network for the multi-class semantic segmentation module. A sample result within the city center area is depicted in Fig. 25.

The Wall class poses a particularly challenging scenario in multi-class segmentation due to the adjacency of lines representing walls, leading to border prediction invading wall pixels. This can lead to a lot of false negatives for walls. Additionally, the road, unbuilt areas, and parts of the river share similar textures, further complicating classification. As the network's input size limits the map scope that can be processed at once, achieving entirely accurate predictions without boundary information is implausible even for human judgment.

However, the present results effectively support the classification of polygons. The UperNet model makes just one single error in both scenarios involving city centers (223 polygons) or rural areas (122 polygons).

Figure 25: Prediction of multi-class semantic segmentation

6 Vectorization

In the context of utilizing binary semantic segmentation masks to extract boundaries, three distinct vectorization strategies are discussed in section 2.3. These strategies revolve around addressing the challenges of accurately converting these masks into vector representations.

Image segmentation algorithms for vectorization The initial approach involves the application of image segmentation algorithms. However, the effectiveness of this method hinges on the careful adjustment of algorithmic parameters for each individual cadastral map. An illustration of the vectorized outcomes of the watershed and Felzenszwalb algorithms in urban and rural locales can be observed in Fig.26. Evidently, the watershed algorithm encounters difficulties when extracting objects with intricate topologies, such as roads. Both algorithms are notably impacted by noise from the segmentation mask, leading to issues like false positives for graticule lines, thereby generating undesirable polygons.

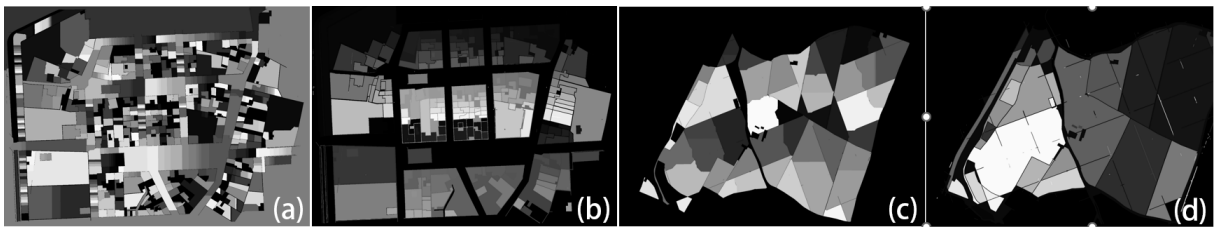

Figure 26: Test results with watershed (a,c) and felzenszwalb (b,d) algorithm

Alternative approaches The other two alternative methodologies share a common prerequisite, necessitating the conversion of the raster mask into one-pixel-wide raster lines through skeletonization. Subsequently, the first approach involves identifying corner

points or desired vertices to generate the vector output. In this scenario, the vectorized representation primarily focuses on retaining crucial points instead of meticulously capturing every detail of a line. To illustrate, consider a manually drawn lengthy straight line; minor shifts between the start and endpoint are plausible. Consequently, the ideal vectorized line might not perfectly align with the ground truth line. This method leans towards a more conceptual interpretation of vectorization.

The second approach entails the comprehensive vectorization of all intricacies present on the raster line. Here, the vectorized topology aligns more closely with the ground truth depicted in the image, eliminating the need for conceptual interpretation. After the vectorization process, various topology simplification algorithms can be employed. These algorithms offer the flexibility for users to decide whether to retain critical points, bends, or effective areas.

When considering the demands of the canton of Geneva, they have shown a preference for the final strategy, opting to preserve a substantial amount of information while affording them the flexibility to fine-tune the resulting vectorized output. Drawing from this preference, we have devised two distinct methods: an elementary method and a more sophisticated one, both aimed at addressing the issue at hand.

Main concerns As depicted in Fig.22 and as discussed in the qualitative analysis of binary semantic segmentation, there are four distinct types of defects related to the extracted borderline prediction topology. Pollution occurrences are infrequent within the Geneva dataset, and while false positives for graticule lines emerge, they are excluded from the resulting vector output if they don't constitute a closed shape. Hence, these two defects aren't the primary concerns. On the contrary, adjacent lines and false negatives yield substantial repercussions, significantly affecting the vectorized topology. False negatives result in the omission of vector lines and the merging of two neighboring polygons. Adjacent lines give rise to numerous minor closed shapes and fragment wall semantics into multiple, disorderly polygons. The mitigation of these concerns involves reducing the decision threshold. Nevertheless, this adjustment doesn't guarantee the complete elimination of these issues from the process. Consequently, we have put forth corresponding remedies. For tackling adjacent lines, we've introduced connected component analysis to identify and eliminate these small enclosed shapes. Regarding false negatives, manual correction comes into play before polygon construction.

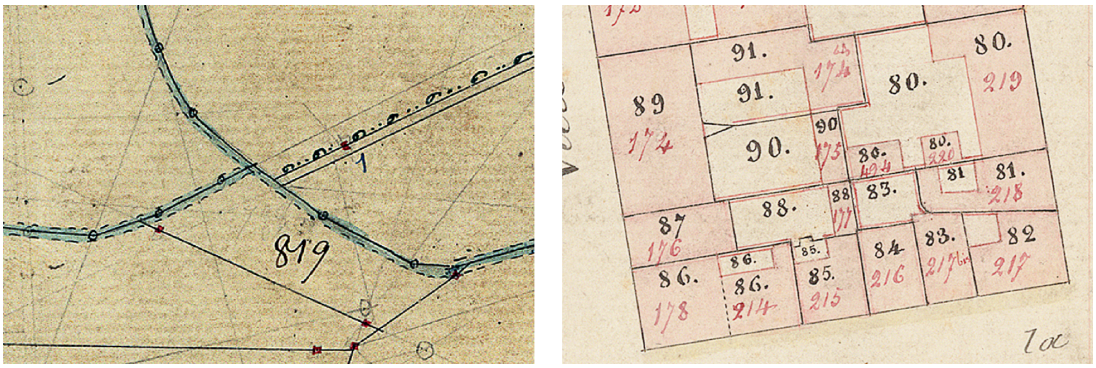

Figure 27: Second priority objects. **Left**: steams with blue texture; **Right**: walls represented by black parcel lines and red building lines.

Vectorization objectives Based on our prior knowledge and segmentation outcomes, we are aware that the current approach faces significant challenges when it comes to extracting small or irregular objects. Following a conversation with the Canton of Geneva, they communicated that there's no imperative need for an automated extraction of these intricate objects. In any case, specialists in the field will manually review the final vectorized outcomes, allowing them to rectify such exceptional scenarios throughout the process. The first priority of this project revolves around the automated extraction of parcel and building topology, unbuilt areas, roads, and major rivers. Objects of smaller size and irregular shape, such as walls and streams depicted in Fig.27, are not of primary concern at this juncture.

6.1 Mask Completion: connected component analysis

Connected component analysis Excluding the identification of walls and streams, the application of connected component analysis (CCA) is employed to improve the topology of the borderline mask. CCA serves as a fundamental technique in image processing, used for the recognition and categorization of connected areas (components) within a binary image. In the specific context of the borderline mask, where pixels are designated as either the borderline (foreground) or the background, connected components represent regions comprised of foreground pixels that share connectivity through neighboring pixels.

Conventionally, two pixels are defined as connected if they possess a shared edge or corner, termed as 4-connectivity or 8-connectivity, respectively. The primary objective of CCA is to allocate a distinct label to each connected component, thus facilitating the sharing of a common label among pixels belonging to the same component.

Figure 28: Mask completion with connected component analysis.

However, in this scenario, the connected components solely pertain to the foreground. The algorithm treats all background pixels as a single label, irrespective of their interconnectedness. In the desired processing of the borderline mask, illustrated in Fig.28, the intended connected region actually is the background pixels. Consequently, to rectify this issue along with removing minor regions of false positives, two iterations of CCA are implemented on the borderline mask, which involves an inversion of foreground and background in between.

It is important to note that for the foreground designated as the borderline, an 8-connectivity criterion is employed, while in the reversed situation, a 4-connectivity criterion must be applied.

Adaptive thresholding After obtaining information about the positions and statistics of the connected components, it becomes necessary to set a threshold based on the total

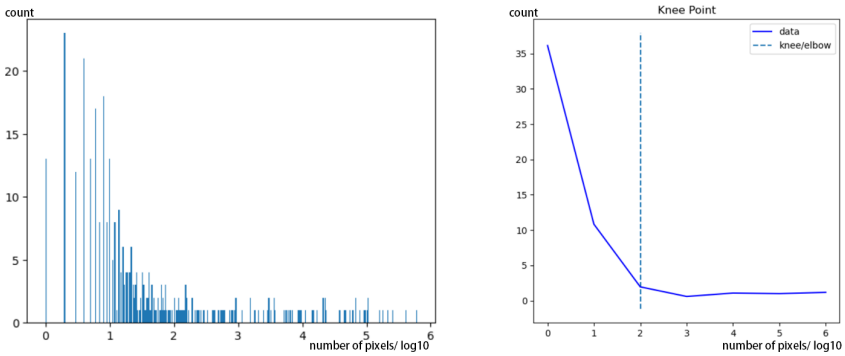

Figure 29: Distribution of connected component on the borderline mask. **Left**: Original distribution in log scale; **Right**: Smoothed distribution and knee point detection

number of pixels within each region. This threshold assists in determining which tiny regions should be eliminated. Notably, the threshold value for this purpose is intricately linked with the geographical location and scale of the cadastral map being analyzed. Maps situated in urban centers, which tend to exhibit numerous tiny objects, should employ a lower threshold than those in rural areas characterized by larger, more scattered objects. To streamline this process and minimize the need for manual intervention, an adaptive thresholding technique has been developed to automatically determine the minimal pixel number within a region.

Illustrated in Fig.29, the statistical distribution of connected component statistics is assumed to be stable. The x-axis logarithmically represents the region's pixel count, while the y-axis represents the count of these regions. Evidently, the distribution adheres to a power law, and the regions of interest are typically situated at the tail of the distribution. As these regions constitute the predominant part of the borderline mask, they tend to be connected and have a large number of pixels. Consequently, a knee point calculated using curvature or the second derivative emerges as an effective way to determine the optimal threshold.

6.2 Elementary method: skeletonize segmentation mask

Upon its completion, the border mask now possesses an optimal topology suitable for vectorization. Powered by the intelligence of deep learning neural networks, this mask is capable of capturing the desired target topology regardless of whether it intersects multiple symbols or consists of dashed lines. The exceptional segmentation performance positions it as a prime candidate for vectorization, if the tiny object is ignored. Thus, we propose an elementary vectorization approach.

The workflow is depicted in Fig.30. The cadastral map is inferred through a binary semantic segmentation network. Subsequently, connected component analysis with adaptive thresholding is employed to obtain the finalized mask. Leveraging the skeletonization algorithm provided by the Python skimage library, the border mask is thinned into one-pixel-wide raster lines. These lines are then subjected to vectorization using GIS software. This process is followed by the application of a topology simplification algorithm, which is designed to reduce the complexity of geometric shapes while preserving the essential characteristics and geometry of the vector data. Finally, polygons are generated from the vectorized polylines. Samples of results are illustrated in Fig.30 (g,h), showcasing the accurate capture of most target objects and the concise boundary facilitated by the

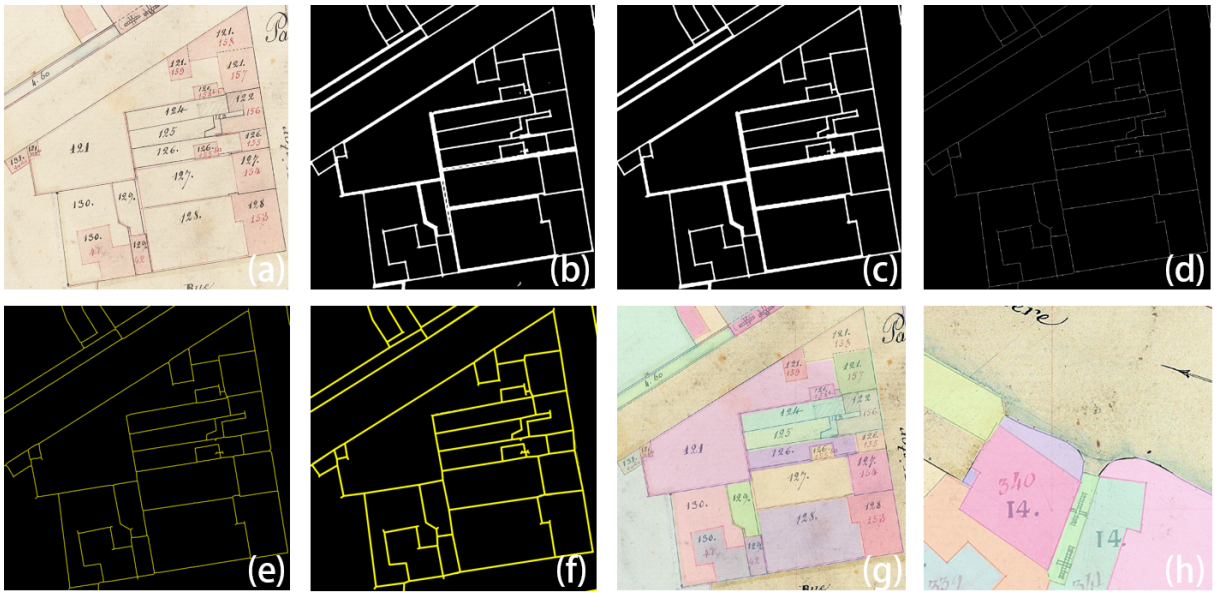

Figure 30: Elementary method pipeline. **(a)**cadastral map; **(b)**binary semantic segmentation result; **(c)**completed mask; **(d)**skeletonized mask; **(e)**vectorized topology; **(f)**topology simplification; **(g,h)**vectorized samples

simplification algorithm.

In the realm of topology simplification algorithms, ArcGIS Pro presents four options: Retain critical points (Douglas-Peucker), Retain critical bends (Wang-Müller), Retain weighted effective areas (Zhou-Jones), and Retain effective areas (Visvalingam-Whyatt). In practical application, the final choice stands out as the best choice, aligning notably well with the ground truth of the borderline. The Visvalingam-Whyatt algorithm functions based on the iterative removal of the least significant vertices from polylines. The significance of a vertex is measured by the area of the triangle formed by the vertex and its neighboring vertices. Smaller area values imply a higher likelihood of vertex removal, indicating its negligible importance in the polyline geometry. The corner point detection approach discussed in section 2.3 mirrors this algorithm, where the acceptance or rejection of a new corner point is determined by the calculation of a similar effective area.

However, besides small objects undetected, the elementary method does come with another drawback concerning multiple adjacent borderlines. The skeletonized line typically lies at the center of the previously completed line area, leading to inaccuracies in cases where a black parcel line and a red building line are adjacent. Although this is not unacceptable for the beneficiary, achieving a better performance in accurately placing the vector line on the black parcel line would undoubtedly enhance their satisfaction.

6.3 Sophisticated method: graph-based approach

To tackle the intricate task of identifying small objects and accurately delineating the black parcel line, a sophisticated approach is explored. These challenges essentially arise from ambiguous predictions concerning adjacent lines. A potential solution emerges if the segmented mask of these neighboring lines can be divided into well-defined topological regions. However, although technologies like super-resolution can extend the scope of the region with adjacent lines, relying solely on resolution enhancement holds limited promise, as information augmented remains relatively rare. Consequently, our strategy

involves revisiting image pixels and exploring algorithms within the realm of traditional computer vision, which are more robust compared to neural networks.

Delineation and Masking Implementing delineation algorithms directly on the original cadastral map can lead to unfavorable outcomes due to factors such as background noise, textual elements, and intersections of text with borders. As illustrated in Fig.31 (b), this often results in the generation of undesirable closed shapes. However, when utilizing the completed borderline mask, it precisely identifies the area of interest for extracting topology. The masked area contains all the essential information necessary for boundary topology extraction. This effectiveness is demonstrated in Fig.31 (c), where applying the completed mask to the delineated cadastral map successfully filters out noise.

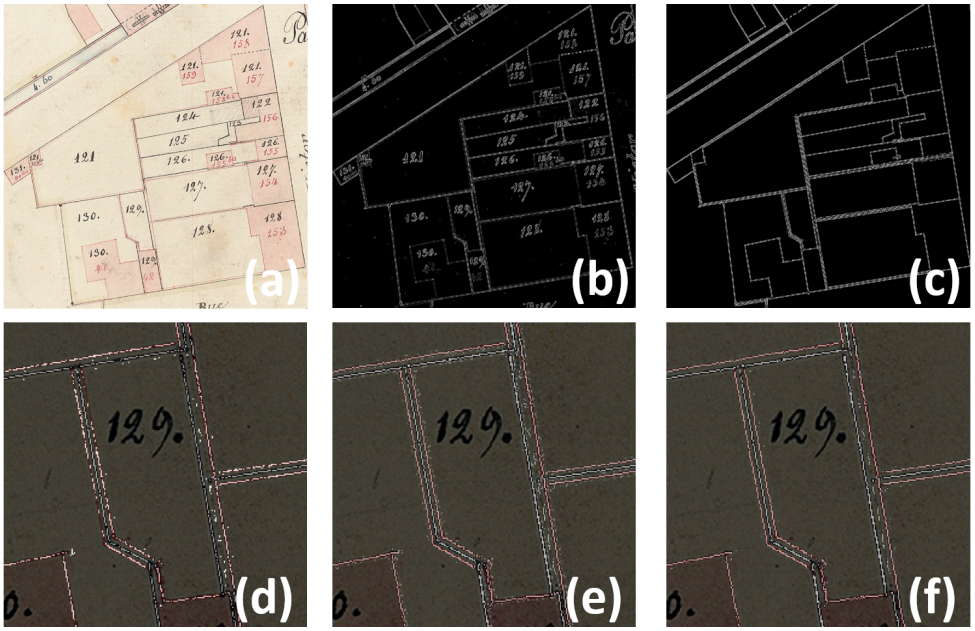

Figure 31: Delineation and masking of the sophisticated method. **(a)**Cadastral map; **(b)**Edge detection; **(c)**Filter noise with completed borderline mask; **(d)**Canny detector with intensity; **(e)**Sobel operator; **(f)**Laplacian operator.

In the context of delineation algorithms, the objective is to achieve one-pixel-wide raster lines. The Canny edge detector, which employs non-maximum suppression, seems like the ideal choice. However, the traditional Canny algorithm identifies two edge pixels for a single borderline, as shown in Fig.31 (b, c), rather than the desired delineation result. To address this, a modification to the traditional Canny edge detector is devised. Notably, the center pixels of the borderline tend to be deep black or red, exhibiting distinct statistical features. Analyzing these pixels reveals that they represent local minima in terms of intensity (grayscale value) within the gradient direction. Armed with this unique statistical characteristic, we managed to adapt the Canny detector for accurate borderline delineation.

Fig.31 (d) showcases the performance of the modified Canny detector. While it effectively identifies the central pixels of the borderline, it severely disrupts the connectivity of the raster lines, rendering topology generation infeasible. Additionally, a significant number of false positives emerge in the background area between adjacent lines. Consequently, the outcome of the delineation is unsatisfactory.

To address this, the Sobel and Laplacian operators are introduced for delineation, both employed with non-maximum suppression from canny, as shown in Fig.31 (e, f). Following fine-tuning and evaluation, the most promising outcome is achieved using the Laplacian operator after manual assessment.

Graph-based endpoints connection To restore the continuity of the intermediate Laplacian delineation output, a novel graph-based endpoint connection algorithm has been developed, as depicted in Fig.32 (left). The algorithm initiates with detecting the endpoints of the existing line mask utilizing a customized convolution kernel. This convolution kernel confirms that an endpoint possesses only one neighboring line pixel. Subsequently, the algorithm treats each pixel as a node and establishes a directed graph, assigning weights to each pixel.

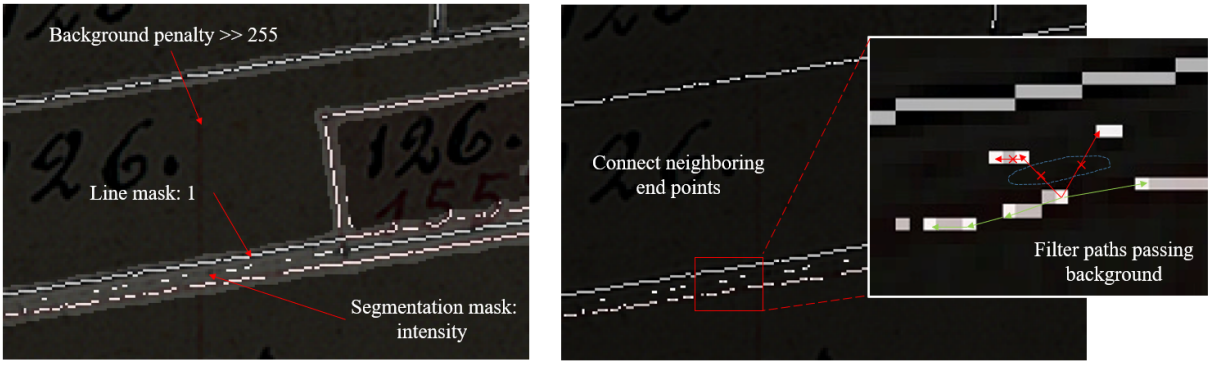

Figure 32: Illustration for sophisticated method. **Left**: graph weights assignment; **Right**: visualization of connecting endpoints

For pixels that register as negative in the completed borderline segmentation mask, a high background penalty, surpassing 255, is designated. Pixels located on the present delineated line mask are assigned a small constant weight of 1. As for other pixels situated within the segmentation mask, their weights align with their intensity values. This distinctive weighting scheme is thoughtfully designed. Consequently, when connecting adjacent endpoints, the shortest path between two endpoints inherently corresponds to the absent portion of the line mask.

However, the presence of false positives within the line mask introduces undesired connections when linking neighboring endpoints. A solution is presented in Fig.32 (right). The concept revolves around the fact that the shortest path between endpoints first passes through positive pixels in the line mask, followed by pixels between endpoints with minimal intensity. The intended connections consistently remain within the borderline pixels. Conversely, undesirable connections navigate through background pixels possessing a bright color and high intensity. To mitigate this, a thresholding mechanism is applied to the highest intensity along the shortest path. This action effectively filters out the undesirable connections, as indicated by the red arrow in the illustration.

The creation of connections between endpoints has greatly enhanced the line mask's continuity. However, certain exceptional scenarios still remain unresolved. The progress achieved is depicted in Fig.33 (a, b, c), where a notable improvement can be observed. Yet, despite these advancements, instances of line mask discontinuity persist, as evidenced in (c). The issue arises particularly at endpoints resembling T-junction patterns, where a suitable neighboring endpoint for connection cannot be identified, resulting in incomplete

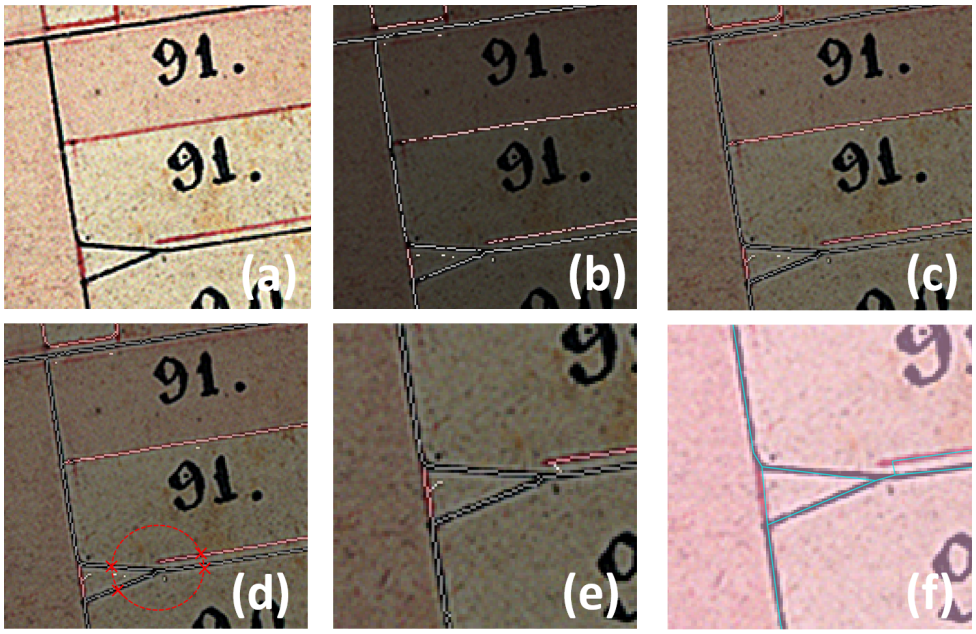

Figure 33: Sophisticated method pipeline. **(a)**Cadastral map; **(b)**Laplacian operator; **(c)**Connect endpoints; **(d)**Defects with ground truth and T-junction; **(e)**Connect with potential links; **(f)**Vectorized polygons.

topology recovery. Furthermore, within this context, certain defects caused by ground truth are left unaddressed. For instance, the parcel boundary represented by the black line is disconnected from the red line denoting a wall.

Inspired by the thinking of the growing contour model, a solution is devised involving the use of a circular pattern to detect candidate pixels for endpoint connection. Illustrated in Fig.33 (d), this method employs a circle to identify potential connecting pixels for the endpoints. By removing the thresholding limit and iteratively increasing the radius of the circle, this approach mitigates the existing defects despite some side effects. Fig.33 (e,f) demonstrate the improved performance.

6.4 Evaluation of vectorized results

Qulitative analysis When comparing the outcomes of the vectorization processes, both methods exhibit strengths and shortcomings. The elementary method excels in identifying the intended polygons; however, its boundaries do not align seamlessly with the target borderlines. Conversely, the sophisticated method demonstrates improved boundary alignment, but it falls short of recognizing dashed lines, resulting in the creation of meaningless polygons. The primary defects are illustrated in Fig.34.

Given the scope for limited human intervention to rectify automatically generated results, a more efficient approach involves addressing incorrect topology after raster line vectorization but before polygon construction. In an effort to determine the most optimal performance using this hybrid workflow, we examined the time and effort required for manual correction of cadastral maps vectorized by both methods.

For a countryside Geneva cadastral map, manual digitization of the topology takes approximately 6 hours. When employing the elementary method, rectification demands around 20 minutes, while the sophisticated method demands doubled effort. Notably, for countryside maps without many intricate topologies, ideal performance can be achieved

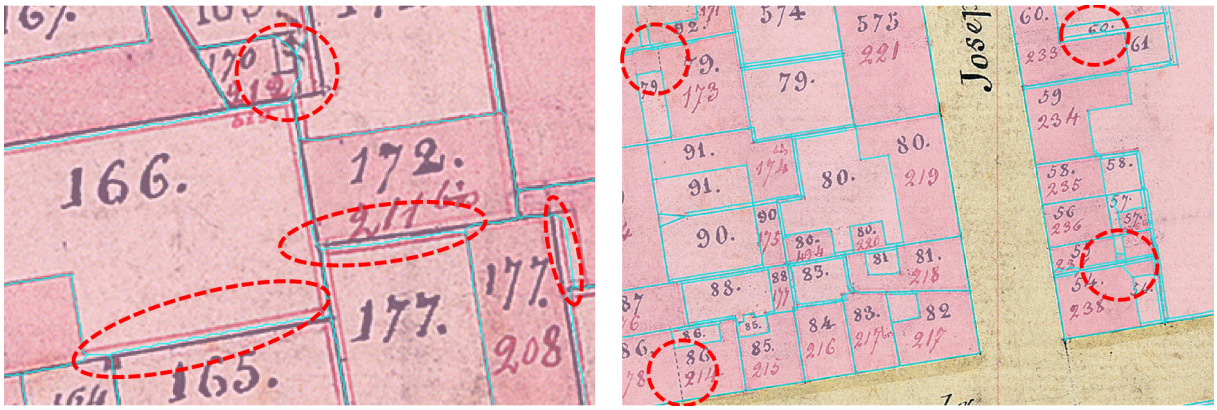

Figure 34: Defects with elementary and sophisticated method. **Left**: Vectorized boundary does not align with the black parcel line; **Right**: Dashed line lost and undesired polygon generated

with both methods. For the city center, refined vectorized outputs, which effectively mitigate the alignment defects, can only be achieved through the sophisticated method.

Consequently, it can be concluded that a collaborative approach involving the sophisticated method and manual rectification can result in saving more than 80% of the manual effort.

Method	Detected Polygons	Mean IoU	Median Hausdorff Distance
Elementary	199	0.979	0.117
Sophisticated	202	0.971	0.133

Table 3: Quantitative performance of elementary and sophisticated methods

Quantitative analysis For vectorization results of the methods, a qualitative assessment is conducted after necessary manual rectification. This involves identifying matched polygons where the IoU metric exceeds 0.7 between the automatically generated results and the ground truth vector data. Subsequently, statistical metrics related to IoU and the Hausdorff distance are computed. It's worth noting that due to occasional ambiguities inherent in annotating the ground truth for cadastral maps, the matched polygons could include outliers generating high Hausdorff distances, while they are not errors. Therefore, to mitigate the sensitivity of the Hausdorff distance, we rely on the mean IoU and the median Hausdorff distance to evaluate the performance.

Across a total of 223 polygons within the test cadastral map, both methods exhibit similar performance by detecting 199 and 202 matched polygons, respectively. In terms of mean IoU, both methods achieve values surpassing 0.97. As for the median Hausdorff distance, the sophisticated method records a slightly higher value (0.133 meters) compared to the elementary method (0.117 meters). However, when projected back to the raster cadastral map, these values translate to a mere 3 pixels for both methods. Consequently, while there exists a marginal disparity in the metric of the two methods, the proximity of their performances makes the quantitative evaluation less substantively meaningful.

7 Aggregation with Optical Character Recognition

In preceding sections, we successfully converted the cadastral map topology into vectors and categorized polygon semantics. In this section, we will employ an open-source text recognition model to detect the index of the extracted parcels and buildings. Subsequently, with all components finalized, the outcomes will be consolidated into vector data and transformed from image frames to spatial reference coordinates.

7.1 EasyOCR: A one-line solution for text recognition

EasyOCR is an open-source Python library designed for optical character recognition (OCR). This functionality empowers the automated identification and extraction of text from images and scanned documents. By providing user-friendly interfaces and pre-trained models for diverse languages, the library can significantly streamline the integration of text recognition into the digitization of cadastral maps.

Supported by the open-source community, EasyOCR is equipped with pre-trained models encompassing over 70 languages. However, it's important to acknowledge that the precision of the extracted text remains influenced by factors like image quality, text font, and language complexity. Hence, in line with common practices for any OCR solution, it is advisable to review and verify the extracted text, particularly in our contexts involving handwritten text.

The process employed by EasyOCR, depicted in Fig.35, is composed of two principal parts: text detection and recognition. These sequential phases collaborate to first pinpoint regions containing text within an image and subsequently extract the actual textual content from these regions.

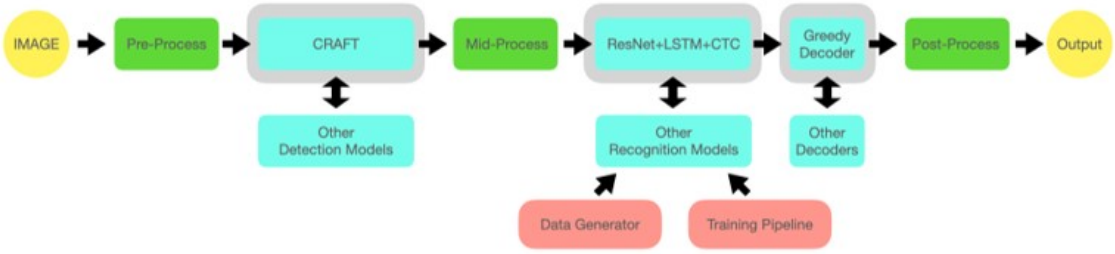

Figure 35: EasyOCR framework

Text Detection Text detection is to identify specific areas within an image that contain textual content. EasyOCR harnesses the power of deep learning methodologies for executing this text detection process. The framework employs either the EfficientDet [39] or CRAFT (Character Region Awareness for Text Detection) models [40], both adept at locating regions that hold text.

The framework is initiated by the preprocessing of the input image, ensuring its compatibility with the subsequent text detection model. This preparatory stage might involve operations such as resizing, normalization, and other transformations. Then, the preprocessed image becomes the input for the text detection model, which has been fine-tuned to recognize and delineate sections containing textual information. As an output, this model generates bounding boxes that encompass the identified text regions, along with corresponding confidence scores indicating its certainty.

Following this, the bounding boxes undergo mid-processing steps to be refined. These refinements include the removal of regions associated with low confidence scores and the overlapping boxes. Non-maximum suppression (NMS) is commonly employed to achieve this, efficiently eliminating redundant or intersecting boxes and thus retaining the most probable and distinct text regions.

Text Recognition After the identification of text regions through the detection process, each detected region signified by a bounding box was cropped from the input image. Subsequently, the cropped region undergoes processing via a text recognition model, which is structured on the foundation of recurrent neural networks (LSTM) and convolutional neural networks (ResNet). This specific model has been trained to recognize individual characters and to predict the sequence of characters comprising the given text. As an outcome, the recognition model generates a sequence of character predictions specific to the text region.

To transition from characters to coherent words or sentences, a greedy decoder mechanism is employed. This entails a transformation from individual character predictions into meaningful textual content by language modeling and the management of character-level probabilities.

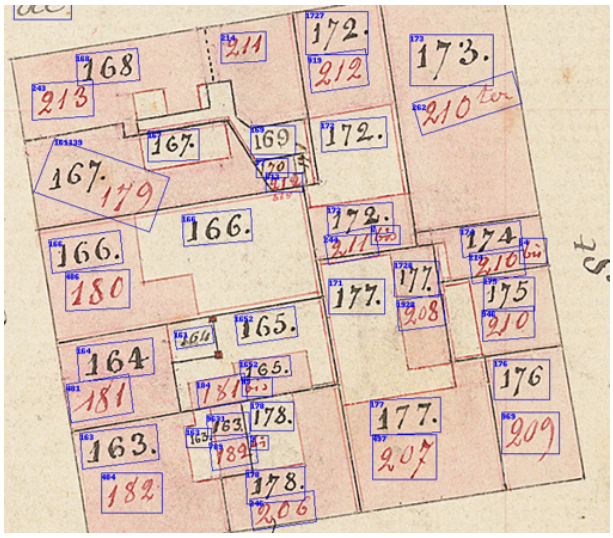

Figure 36: EasyOCR detection and recognition performance

Experiment on cadastral maps Fig.36 shows a sample of the detection and recognition result by the EasyOCR pre-trained model on the Geneva cadastral map. Overall, the text region detection is comprehensive and precise; however, a minor anomaly is observed where a few adjacent regions have been merged. In terms of recognition performance, a consistent pattern of errors has been witnessed across several maps, notably involving the misclassification of digits 1, 4, and 7. It is noteworthy that the recognition performance experiences significant fluctuations when dealing with handwritten digits, as the accuracy depends on the particular handwriting style of the map creator.

The recognition accuracy varies considerably, with some maps boasting an accuracy of around 50%, while others exhibit performance that falls below acceptable thresholds. This variance can be attributed to the fact that the pre-trained model was developed using datasets composed of printed fonts. Due to time constraints, a dedicated dataset

specifically focusing on handwritten digits wasn't feasible during this study. However, it's important to highlight that EasyOCR supports fine-tuning with customized datasets, and it is confidently anticipated that recognition accuracy would witness substantial enhancement upon undertaking such a process.

7.2 Aggregation and geo-referenced results

Aggregation The standard geospatial coordinate reference system is conventionally established on a Cartesian plane, where the origin (0,0) is positioned at the bottom left corner. As one moves upwards and towards the right, both X and Y coordinates increase. However, raster data, originating from image processing, employs an alternative referencing system for pixel access, referring image frame or pixel coordinates system. This approach employs row and column designations, with the origin (0,0) situated at the upper left corner. Rows increment as one moves downwards, while columns increase towards the right.

The process of aggregating vector polygons with semantic classifications and OCR results happens within this image frame. The vectorization conducted using ArcGIS Pro yields vector data based on pixel coordinates. However, a particular difference arises regarding the definition of the positive direction for columns within ArcGIS Pro. Consequently, a preprocessing step is essential to invert the y coordinates of all vertices within the vector polygons. Following this, the multi-class semantic segmentation mask, the bounding boxes of the OCR-detected digits, and the vector polygons are aligned within the image frame. By employing a majority voting approach, the classification of the polygons is deduced from the segmentation mask. Simultaneously, the text within the polygons is captured by the intersections between the bounding boxes and the polygon geometries.

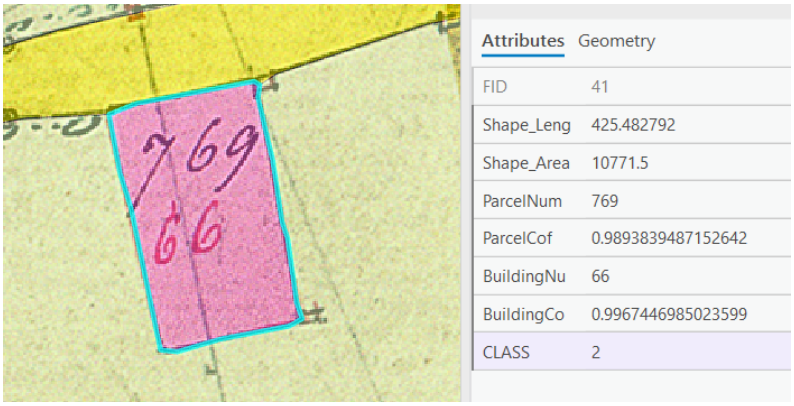

Figure 37: Aggregated vector results. Semantic classification and text recognition results are aggregated within vector polygon as attributes.

A single polygon might overlap with multiple text regions. In such instances, an analysis of color statistics is initially performed on the red channel within the text region to differentiate between the red building index and the black parcel index. Furthermore, when multiple occurrences of the same type of index are encountered, the one exhibiting a higher confidence coefficient is retained. The finalized vector results are depicted in Fig.37.

Geo-referencing The process of georeferencing historical cadastral maps necessitates the expertise of domain specialists. This undertaking involves manual manipulation of the maps within GIS software, all conducted under the spatial coordinate system. The alignment procedure involves the identification of control points, each characterized by a pair of corresponding pixel coordinates (row, column) and spatial reference coordinates (x, y). Following this meticulous alignment process, the projected cadastral map is enriched with geographic information and subsequently saved as a GeoTIFF file.

The GeoTIFF file functions as a repository for both the raster information of the cadastral maps and the requisite parameters for executing an affine transformation. This transformation operates to project the raster image from its native pixel coordinate system to the target spatial coordinate reference system. The general formulation for an affine transformation can be succinctly expressed as follows:

$$x' = ax + by + c$$
$$y' = dx + ey + f$$

Where:

- (x, y) are the original pixel coordinates.
- (x', y') are the transformed coordinates in the spatial reference system.
- a, b, d, and e are scaling, rotation, and skewing factors.
- c and f are translation offsets in the x and y directions.

The values of a, b, d, e, c, and f are determined based on at least three pairs of control points, using linear algebra methods, such as matrix inversion or least squares.

Figure 38: Geo-referencing: transform vectorized result from image frame to spatial-referenced coordinate system.

Fig.38 demonstrates geo-referencing vector results. For practical implementation, a strong recommendation is to employ the "affine" Python library. This advice stems from the fact that the ordering of the six parameters has not been universally standardized between ESRI World File and GDAL (Geospatial Data Abstraction Library) – two of the most widely used GIS tools. Furthermore, in instances where a non-zero rotation parameter exists, the affine transformation implementation of alternative libraries may encounter failures while the "affine" library manages such scenarios.

8 Discussion and Future Work

8.1 Discussion: Potential benefit from transfer learning

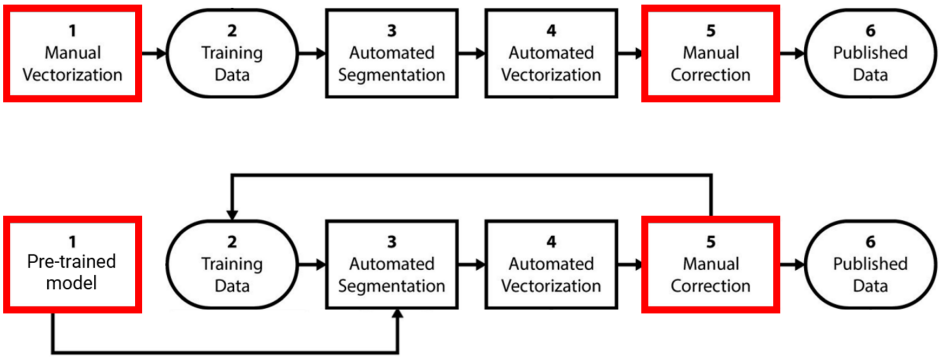

Figure 39: **Top**: Current workflow; **Bottom**: Workflow updated with transfer learning

The completion of the entire vectorization pipeline has been achieved after aggregation and geo-referencing modules. The upper part of Fig.39 provides an overview of the current workflow. The process initiates with manual vectorization for selected maps, generating both raster and vector ground truth as training data. Then, a segmentation neural network is trained, utilizing the in-domain dataset. Automated segmentation is subsequently applied to all the cadastral maps. Followed by automated vectorization, manual correlation is needed to rectify and verify the generated topology. Ultimately, this announces the completion of vectorization, rendering the data ready for publication.

The feasibility and performance of this workflow have been validated with both quantitative and qualitative assessments. While manual correction remains essential for refining results, a remarkable workload reduction of over 80% is achieved during vectorization. However, a notable prerequisite is the availability of an in-domain training dataset. This empowers the segmentation model to capture image features and extract cadastral topology. In the Geneva case, more than 2 weeks were invested in annotating 8 cadastral maps, apart from the time spent developing the annotation workflow. Although this cost is affordable given the broad applicability of the segmentation model across over 200 cadastral maps for the entire Geneva canton, it might not be the same case for maps from other cantons or temporal periods.

To address this challenge, we introduce a pre-trained model with transfer learning into the workflow. Leveraging existing datasets such as the Lausanne and Neuchâtel cases, training can be initiated without the in-domain dataset (Geneva) ground truth. The IoU metric of such zero-shot transfer learning reached around 0.65 with UperNet architecture and borderline prediction is shown in Table.1 and Fig.40. Although this accuracy is not as competitive as the model trained with in-domain data, and certain patterns such as green vegetation lines next to the borderline remain unrecognized, the overall topology is reasonably extracted and acceptable for the vectorization stage.

In light of these considerations, a more practical and efficient workflow is proposed, depicted in the bottom section of Fig.39. The pre-trained model stemming from zero-shot transfer learning initializes the automated segmentation and vectorization for the first iteration. While zero-shot performance is limited, it could extract the majority of the topology, allowing the time-consuming process of manual vectorization to be replaced

Figure 40: Transfer learning (Zeroshot): Borderline segmentation performance with pre-trained model

by manual correction during the first iteration. This leads to a significant reduction in annotation workload. Subsequently, with corrected results at hand, a second iteration fine-tunes the model using in-domain training data, ultimately resulting in optimal performance.

8.2 Future work: Network SNAKEs

The current optimal performance is generated with the sophisticated approach we proposed, which can locate the boundary at the center of borderlines. However, it needs at least two times the effort to rectify manually as it can not recognize dashed lines and create undesirable polygons. Besides, the pattern of walls and steams is so complicated that both methods do not manage to extract. In other words, if we can refine the elementary method so that the vectorized boundary can center on borderline pixels, it would outperform the sophisticated approach.

Active Contour Models This concept is effectively applicable through the utilization of active contour models, often referred to as SNAKEs or deformable contours. These models represent mathematical frameworks employed for detecting object boundaries and segmenting them within images. Their utility becomes particularly prominent where conventional edge detection methods face challenges due to noise, interruptions, or variations in intensity.

The active contour model is initialized close to the object of interest and then evolves based on the forces acting on it. It progressively refines an initial approximation to align more accurately with the desired boundaries. This process involves energy minimization, where the contour seeks to find the configuration that best fits the image features and the user-defined constraints. The contour evolves by balancing internal energy, which enforces smoothness and contour curvature, and external energy, which pulls the contour towards image features like edges or gradients.

However, it's important to note that the conventional active contour model is ideally suited for a singular closed shape. It can effectively contract to conform concave regions or expand to encapsulate convex areas. While this adaptability renders active contour models highly suitable for segmenting objects characterized by irregular shapes and intricate

boundaries, their application is limited in cases involving topology with multiple polygons that share edges.

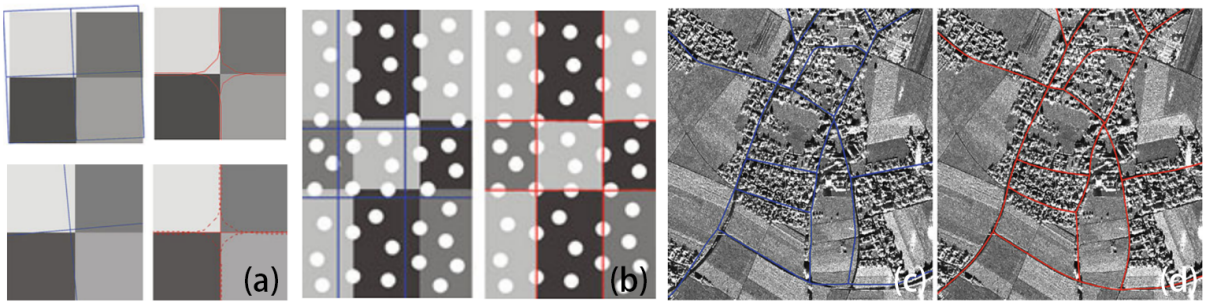

Figure 41: Samples to illustrate Network SNAKEs. **(a)**left: initial contour; right: results of Network SNAKEs (solid line) and traditional SNAKEs (dashed line). **(b)**Network SNAKEs in an intricate pattern that can simulate dashed lines in cadastral maps. **(c,d)**Network SNAKEs performance on the road network topology.

Network SNAKEs Network SNAKEs, an extension of the traditional active contour models, are innovative mathematical frameworks designed to address the segmentation challenges posed by complex networks, such as those encountered in images with intricate interconnected structures. These models, often referred to as "snakes on a graph," combine the versatility of active contour methods with the ability to capture and delineate complex topologies formed by networks of interconnected components.

Network SNAKEs offer a powerful solution for scenarios where traditional active contour models fall short due to the presence of multiple polygons sharing edges or intricate interconnections. By leveraging graph-based representations, these models enable the incorporation of contextual information, such as spatial relationships and connectivity, into the segmentation process. Fig.41 illustrates with three demo cases.

Regrettably, owing to time constraints, we were unable to build this algorithm from scratch. Despite attempts to reach out to the algorithm's author, their implementation has been lost over the past decade. However, it is believed that with the preliminary outcomes produced by the elementary method, network SNAKEs would facilitate the refinement of topology and align with the borderline precisely.

9 Conclusion

The task of vectorizing historical cadastral maps remains challenging with the help of artificial intelligence techniques. In this project, our journey begins with a manual annotation pipeline, where we create raster ground truth masks based on vector data. To extract borderlines and classify detected objects, we employ semantic segmentation networks. Throughout this process, we experiment with innovative architectures such as transformers and deformable convolutions. After that, an elementary vectorization method and a more sophisticated graph-based approach are developed. The latter contributes to satisfactory results with necessary manual refinement.

To conclude, fully automatic vectorization of cadastral maps with ideal accuracy is still not feasible yet. The hybrid mode is the optimal solution so far and we achieved more than 80% reduction in workloads.

This exploratory project will be the starting point to automate the historical reconstitution of a parcel over time. By harnessing computer vision and machine learning techniques for the automatic vectorization of plans and registers, we aspire to streamline archive retrieval, reconstruct the historical evolution of cadastral objects, and populate a temporal database within the cantonal geographic information systems.

References

[1] Remi Petitpierre and Paul Guhennec. Effective annotation for the automatic vectorization of cadastral maps. *Digital Scholarship in the Humanities*, page fqad006, 2023.

[2] Minh Ôn Vû Ngoc, Yizi Chen, Nicolas Boutry, Joseph Chazalon, Edwin Carlinet, Jonathan Fabrizio, Clément Mallet, and Thierry Géraud. Introducing the boundary-aware loss for deep image segmentation. In *British Machine Vision Conference (BMVC) 2021*, 2021.

[3] Joseph Chazalon, Edwin Carlinet, Yizi Chen, Julien Perret, Bertrand Duménieu, Clément Mallet, Thierry Géraud, Vincent Nguyen, Nam Nguyen, Josef Baloun, et al. Icdar 2021 competition on historical map segmentation. In *International Conference on Document Analysis and Recognition*, pages 693–707. Springer, 2021.

[4] Yizi Chen, Edwin Carlinet, Joseph Chazalon, Clément Mallet, Bertrand Dumenieu, and Julien Perret. Vectorization of historical maps using deep edge filtering and closed shape extraction. In *International conference on document analysis and recognition*, pages 510–525. Springer, 2021.

[5] Ionut Iosifescu, Angeliki Tsorlini, and Lorenz Hurni. Towards a comprehensive methodology for automatic vectorization of raster historical maps. *e-Perimetron*, 11(2):57–76, 2016.

[6] Magnus Heitzler and Lorenz Hurni. Cartographic reconstruction of building footprints from historical maps: A study on the swiss siegfried map. *Transactions in GIS*, 24(2):442–461, 2020.

[7] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. Topological map extraction from overhead images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1715–1724, 2019.

[8] Drolias Garyfallos Chrysovalantis and Tziokas Nikolaos. Building footprint extraction from historic maps utilizing automatic vectorisation methods in open source gis software. *Automatic vectorisation of historical maps. Department of Cartography and Geoinformatics, ELTE Eötvös Loránd University, Budapest*, pages 9–17, 2020.

[9] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. Deep hough-transform line priors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 323–340. Springer, 2020.

[10] Rémi Guillaume Petitpierre, Frédéric Kaplan, and Isabella Di Lenardo. Generic semantic segmentation of historical maps. In *CEUR Workshop Proceedings*, volume 2989, pages 228–248, 2021.

[11] Rémi 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.

[12] Chenjing Jiao, Magnus Heitzler, and Lorenz Hurni. Extracting wetlands from swiss historical maps with convolutionalneural networks. In *Automatic Vectorisation of*

Historical Maps. International workshop organized by the ICA Commission on Cartographic Heritage into the Digital 13 March, 2020 Budapest. Proceedings, pages 33–38. Department of Cartography and Geoinformatics, ELTE Eötvös Loránd University, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1-2):187–228, 2000.

[15] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25–43, 2017.

[16] Weiwei Duan, Yao-Yi Chiang, Stefan Leyk, Johannes H Uhl, and Craig A Knoblock. Automatic alignment of contemporary vector data and georeferenced historical maps using reinforcement learning. *International Journal of Geographical Information Science*, 34(4):824–849, 2020.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[19] Yao Zhao, Guangxia Wang, Jian Yang, Lantian Zhang, and Xiaofei Qi. Building block extraction from historical maps using deep object attention networks. *ISPRS International Journal of Geo-Information*, 11(11):572, 2022.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[21] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022.

[22] Sebastian Nowozin and Christoph H Lampert. Global connectivity potentials for random field models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 818–825. IEEE, 2009.

[23] Martin Ralf Oswald, Jan Stühmer, and Daniel Cremers. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 32–46. Springer, 2014.

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

[25] Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n–d images. *Computer vision and image understanding*, 112(1):81–90, 2008.

[26] Ziyun Yang, Somayyeh Soltanian-Zadeh, and Sina Farsiu. Biconnet: An edge-preserved connectivity-based approach for salient object detection. *Pattern recognition*, 121:108231, 2022.

[27] Michael Kampffmeyer, Nanqing Dong, Xiaodan Liang, Yujia Zhang, and Eric P Xing. Connnet: A long-range relation-aware pixel-connectivity network for salient segmentation. *IEEE Transactions on Image Processing*, 28(5):2518–2529, 2018.

[28] Agata Mosinska, Pablo Marquez-Neila, Mateusz Koziński, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3136–3145, 2018.

[29] Liangzhi Li, Manisha Verma, Yuta Nakashima, Hajime Nagahara, and Ryo Kawasaki. Iternet: Retinal image segmentation utilizing structural redundancy in vessel networks. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3656–3665, 2020.

[30] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004.

[31] Fernando A Velasco and Jose L Marroquin. Growing snakes: active contours for complex topologies. *Pattern Recognition*, 36(2):475–482, 2003.

[32] Matthias P Wagner and Natascha Oppelt. Extracting agricultural fields from remote sensing imagery using graph-based growing contours. *Remote sensing*, 12(7):1205, 2020.

[33] Matthias P Wagner and Natascha Oppelt. Deep learning and adaptive graph-based growing contours for agricultural field extraction. *Remote sensing*, 12(12):1990, 2020.

[34] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[35] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797, 2021.

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[37] George Bebis and Michael Georgiopoulos. Feed-forward neural networks. *Ieee Potentials*, 13(4):27–31, 1994.

[38] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[39] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[40] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019.