ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ANALYSING LAYOUT ELEMENTS OF HISTORICAL MAPS

BACHELOR SEMESTER PROJECT REPORT

Jiaxun Liu

Supervisors:
Raimund Schnürer
Isabella di Lenardo

8th January 2025

ABSTRACT

This study focuses on analyzing layout elements in historical maps using advanced machine learning techniques. We propose a pipeline combining SAM2 for segmentation, BLIP-2 for image caption generation, and ImageBind for uniform embedding. SAM2 automatically segments key map elements, including cartouches, compass roses, and decorative patterns. BLIP-2 generates textual descriptions for these segments, enabling semantic analysis. ImageBind unifies image and text embeddings, enabling seamless cross-modal querying and retrieval. The pipeline also integrates dimensionality reduction methods, such as UMAP and T-SNE, to analyze embedding spaces and features an interactive user interface for intuitive exploration of map datasets. Additionally, we introduce a map-matching method that focuses on element-level comparisons rather than relying on a single global map embedding, which offers a more fine-grained and precise matching, capturing subtle similarities and unique details across maps. Together, these advancements bridge the gap between traditional historical cartography analysis and modern AI-driven techniques, equipping historians and researchers with powerful tools for map interpretation and exploration.

Contents

CHAPTER 1

INTRODUCTION

1.1 CONTEXT

Recent advancements in machine learning have enabled the extraction of various features from historical maps, significantly enhancing our understanding of their content. However, the peripheral elements, also known as layout elements, surrounding these maps remain unexplored. These elements, including decorative features (e.g., ornamentation, cartouches), orientation aids (e.g., scale bars, wind roses, north arrows), illustrative components (e.g., heraldic symbols, figures, landscape scenes), and descriptive annotations (e.g., titles, legends, explanatory text), offer valuable insights into a cartographer's background. Analyzing their style, arrangement, and purpose can reveal essential information about the historical and cultural context in which the maps were created.

1.2 DATASET

TABLE 1.1
Dataset Collection Data

Library Name	Number of Items
Beinecke Library	785
Biblioteca Digital Hispánica	90
Bibliothèque Nationale de France	2,586
Boston Public Library	518
Bodleian Library	476
David Rumsey	2,755
John Carter Brown Library	481
Library of Congress	413
New York Public Library	48
Royal Museums Greenwich	968
Ryhiner-Sammlung	14,176

The dataset consists of historic maps sourced from various renowned map libraries and collections, as summarized in Table 1.1. In total, the dataset includes maps from eleven distinct collections, each contributing a varying number of items.

The largest contribution comes from the Ryhiner-Sammlung with 14,176 maps, followed by significant

contributions from David Rumsey (2,755 maps) and the Bibliothèque Nationale de France (2,586 maps). Smaller yet valuable contributions include collections such as the Biblioteca Digital Hispánica (90 maps) and the New York Public Library (48 maps).

These maps span different historical periods, regions, and styles, providing a diverse representation of cartographic heritage. The processed dataset offers a rich resource for historical, geographical, and computational analysis.

1.3 GOALS

The goal of this project is to develop a robust and efficient tool for analyzing historical maps using state-of-the-art machine learning techniques. Specifically, our objectives are:

1. Automated Map Element Extraction: Utilize advanced segmentation models, such as SAM2, to automatically identify and extract key map elements, including cartouches, compass roses, legends, and decorative patterns.
2. Semantic Interpretation through Image Captioning: Employ multimodal models like BLIP-2 to generate meaningful textual descriptions of segmented map elements, facilitating deeper semantic analysis and interpretation.
3. Cross-Modal Querying and Retrieval: Leverage uniform embedding models, such as ImageBind, to enable cross-modal searches, allowing users to query map elements using both image and text inputs.
4. An Efficient Search Mechanisms: Implement optimized indexing techniques, such as IVF index, for efficient searching and retrieval of map features across large datasets, ensuring scalability and responsiveness.
5. Interactive Visualization Interface: Develop an intuitive user interface using frameworks like Vue.js and WizMap for embedding visualization, enabling users to explore, search, and analyze map elements interactively.
6. Dataset Analysis and Insights Generation: Perform high-density and outlier analysis on historical map datasets to uncover recurring patterns, artistic styles, and unique cartographic features. For example, the analysis can identify common elements that appear frequently across maps, reflecting widespread cartographic conventions, or highlight rare and unique elements that distinguish specific maps with exceptional artistic or thematic styles.

Through these goals, the project aims to bridge the gap between traditional historical map analysis and modern machine learning techniques, offering historians, researchers, and geographers a powerful tool for exploring and interpreting historical cartographic data.

CHAPTER 2

RELATED WORKS

2.1 FEATURE DETECTION IN HISTORICAL MAPS BY DEEP LEARNING

Neural networks have significantly advanced the analysis of historical maps by automating the extraction of features such as roads, buildings, and text from scanned images. For instance, (Petitpierre 2021) uses deep convolutional neural networks (CNNs) for semantic segmentation of historical city maps, demonstrating superior flexibility and performance compared to traditional algorithms. Additionally, (Xia, Heitzler and Hurni 2022) uses CNN-based template matching has been utilized to detect wetlands in historical maps, enhancing the accuracy of information extraction. These advancements facilitate the digitization and interpretation of historical cartographic documents, providing robust tools for historians and geographers to automatically digitize large map series and collections with good precision.

Recent advancements have demonstrated (Sertel, Hucko and Kabadayı 2024; Wu et al. 2023) the efficacy of Transformer architectures in analyzing historical maps due to their ability to capture global relationships and long-range dependencies across map elements, outperforming CNNs which are limited by their local receptive fields. For instance, the SegFormer model (Sertel, Hucko and Kabadayı 2024) has been applied to automatically extract various road types from digitized historical maps, outperforming traditional CNN architectures in accuracy. These studies highlight the potential of Transformer-based models in processing and interpreting complex features present in historical cartographic documents.

2.2 MULTIMODAL MODELS FOR IMAGE CAPTIONING AND EMBEDDING GENERATION

In the field of image captioning, multimodal techniques have advanced with models like BLIP (J. Li, D. Li, Xiong et al. 2022) and BLIP-2 (J. Li, D. Li, Savarese et al. 2023). BLIP (Bootstrapping Language-Image Pre-training) integrates vision and language understanding, enabling the generation of coherent textual descriptions from images. Building upon this, BLIP-2 introduces a more efficient pre-training strategy by incorporating a lightweight Querying Transformer (Q-Former) between a frozen image encoder and a frozen large language model, significantly reducing computational costs while maintaining high performance.

In the domain of uniform embedding for multimodal data, several models have been developed to integrate diverse modalities into a cohesive representation space. ImageBind (**girdhar_imagebind_2023**) is notable for learning a joint embedding across six modalities—images, text, audio, depth, thermal, and IMU (Inertial Measurement Unit) data—by leveraging image-paired data to align these modalities without requiring all combinations of paired data. Building upon this, ImageBind-LLM(Han et al. 2023) extends

the approach by incorporating large language models (LLMs) through multi-modality instruction tuning, enabling the model to respond to various multimodal inputs, including audio and 3D point clouds, using only image-text alignment during training. Additionally, E5-V (Jiang et al. 2024) adapts multimodal large language models to generate universal embeddings, effectively bridging the modality gap between different input types and demonstrating strong performance in multimodal tasks without fine-tuning. These advancements facilitate applications such as cross-modal retrieval and generation, enhancing the integration and utilization of diverse data types.

CHAPTER 3

METHODS

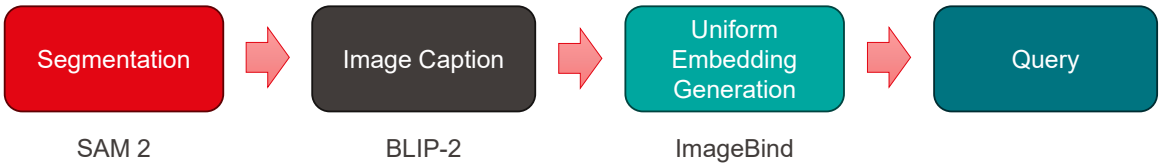


FIGURE 3.1
Pipeline Overview

3.1 OVERVIEW

In this task, we aim to extract and analyze map elements using machine learning methods. To achieve this, we have designed a pipeline (Figure 3.1) for map element extraction and querying, which enhances efficiency for further research.

First, we employ SAM2 (Segment Anything Model 2) (Ravi et al. 2024) to extract elements from the maps. Next, BLIP-2 (J. Li, D. Li, Savarese et al. 2023) is used to generate image captions, providing semantic interpretations of the patterns. These captions offer an additional dimension for querying patterns based on their textual descriptions.

To further unify these representations, we leverage ImageBind (Han et al. 2023) to create uniform embeddings for both images and their captions. This enables querying using either text or images, allowing retrieval of not only similar images but also their corresponding captions.

For query optimization, we experimented with indexing techniques (Douze et al. 2024) to accelerate query performance. However, on our small-scale dataset, indexing led to a decrease in accuracy. As a result, we opted for GPU-based matrix multiplication to directly perform queries, ensuring both efficiency and accuracy in the search process.

In addition to these methods, to identify related maps based on segmented elements, we calculate similarity by comparing individual elements rather than using a single embedding for the entire map.

To better interpret and explore the results, we also developed a visualization module to illustrate the uniform embeddings. For this purpose, we employed both UMAP (McInnes, Healy and Melville 2018)

and T-SNE (Van der Maaten and Hinton 2008) techniques to reduce the dimensionality of the embeddings, enabling clearer and more intuitive representation of the data.

3.2 SEGMENTATION

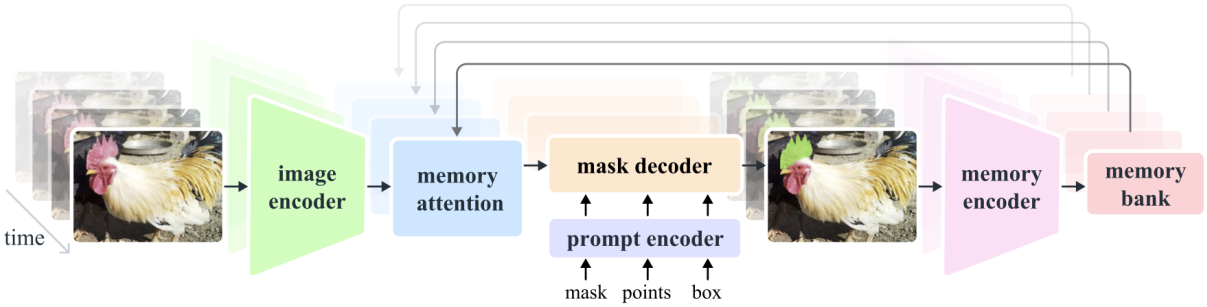


FIGURE 3.2
Segment Anything 2 model illustration (Ravi et al. 2024)

SAM2 (Ravi et al. 2024) is utilized in our pipeline for image segmentation tasks. The illustration of the model is shown in Figure 3.2. SAM2 processes an input image by first extracting multi-scale features using a simplified Transformer architecture. User-provided prompts, such as points, boxes, or masks, are encoded into prompt vectors that align with these image features. For those task without user prompt, the model employs a regular grid of points over the input image to generate candidate masklets—small, preliminary segmentation masks. These masklets are then subjected to a verification step to assess their quality. A streaming memory module then integrates the current image features with the prompt vectors and any prior information, enhancing the model's ability to accurately identify target regions. Based on this combined information, SAM2 generates precise segmentation masks that delineate the desired objects within the image. The final output is a segmentation mask matching the input image's dimensions, ready for further analysis or application.

Compared to other models, SAM2 introduces several innovations. It unifies image and video segmentation within a single model, streamlining processing across different visual data types. These advancements enable SAM2 to achieve superior performance in segmentation tasks, requiring fewer user interactions compared to prior approaches.

SAM2 (Segment Anything Model 2) offers several segmentation modes to accommodate various user needs:

- Interactive Prompt-Based Segmentation: Users can provide prompts such as points, bounding boxes, or masks to guide the model in segmenting specific objects within an image. This mode allows for precise control over the segmentation process, enabling users to iteratively refine the results based on their inputs.
- Automatic Mask Generation: SAM2 can automatically generate segmentation masks without user prompts. This mode is particularly useful for quickly obtaining segmentation results across an entire image, facilitating tasks that require comprehensive object delineation.

In our pipeline, given the need to process a large volume of maps, we employ automatic mask generation for segmentation. However, this automated approach can sometimes produce irrelevant patterns, such as low-resolution segments or patterns lacking meaningful information, like blank regions. To address this, we implemented filtering mechanisms to eliminate low-resolution patterns and other non-informative segments, ensuring cleaner and more valuable segmentation results.

3.3 IMAGE CAPTION GENERATION

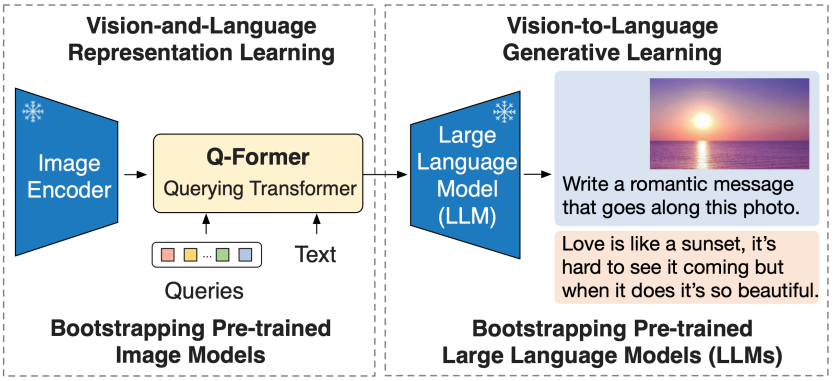


FIGURE 3.3
Image Search Interface (J. Li, D. Li, Savarese et al. 2023)

We use BLIP-2 (Bootstrapping Language-Image Pre-training 2) in this part. The illustration of the model is shown in Figure 3.3. BLIP-2 enhances image captioning by integrating a frozen image encoder with a frozen large language model (LLM) through a lightweight Querying Transformer (Q-Former). This architecture allows BLIP-2 to generate coherent and contextually relevant textual descriptions for images. The Q-Former serves as an intermediary, extracting pertinent visual features from the image encoder and conveying them to the LLM, which then produces the corresponding textual output. This design enables BLIP-2 to perform zero-shot image-to-text generation, effectively producing captions without requiring task-specific fine-tuning.

Compared to other models, BLIP-2 introduces several innovations. By utilizing frozen pre-trained image encoders and LLMs, it significantly reduces the computational costs associated with end-to-end training of large-scale vision-language models. Notably, BLIP-2 achieves state-of-the-art performance on multiple vision-language benchmarks while maintaining a lower parameter count compared to existing methods, exemplifying its efficiency and effectiveness in the field.

Compared to BLIP, BLIP-2 offers improved performance and supports conditional image caption generation. This capability allows us to provide domain-specific background information to BLIP-2, guiding it to generate captions more closely aligned with the historical map domain.

3.4 UNIFORM EMBEDDING

In this part of our pipeline, we utilize ImageBind, a multimodal representation model designed to establish a shared embedding space for a wide range of data modalities. The illustration of the model is shown in Figure 3.4. The core principle of ImageBind lies in its ability to align these heterogeneous data types into a single, unified embedding space without requiring pairwise training data for every modality combination. Instead, ImageBind leverages pre-trained unimodal encoders (e.g., image encoders like CLIP (Radford et al. 2021)) and aligns them via a common reference modality — typically vision data (images) — to create a bridge across modalities.

The innovative architecture allows cross-modal retrieval and synthesis tasks by embedding all modalities into the same latent space, where semantic similarities between different types of data can be directly measured using simple distance metrics, such as vector dot product. And ImageBind can retrieve relevant images, text descriptions corresponding to a train scenario, as shown in the visualization.

Compared to previous multimodal models, ImageBind's key innovation is its ability to scale seamlessly

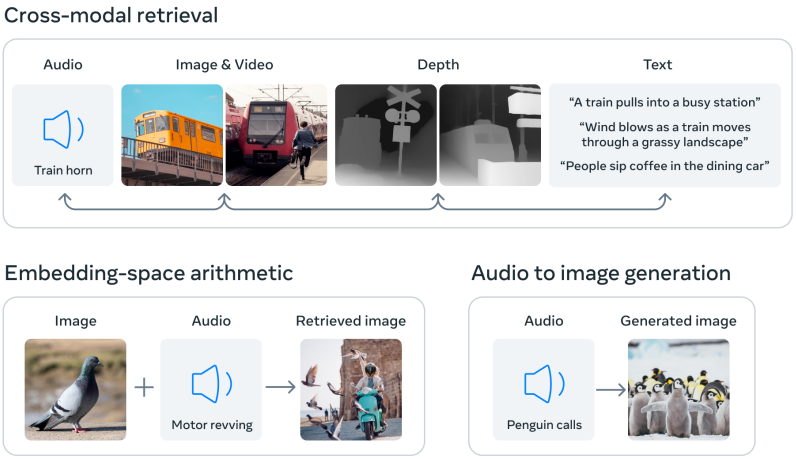


FIGURE 3.4
Image Search Interface (Han et al. 2023)

across modalities without requiring explicit paired training data between every modality pair. Traditional multimodal systems often rely on curated datasets explicitly aligning modalities (e.g., text-image pairs), which limits scalability. ImageBind breaks this limitation by aligning all modalities to a shared image-centric latent space, making it far more flexible and extensible. This approach enables tasks like cross-modal retrieval, zero-shot learning, multimodal content generation, and feature manipulation across previously unconnected data types.

To calculate the similarity between each pair of elements, given two embeddings e_i and e_j generated by ImageBind, we use the vector dot product:

$$\sigma(e_i, e_j) = e_i \cdot e_j$$

where:

- σ denotes the similarity function, representing the dot product between the vectors e_i and e_j.

3.5 EMBEDDING SEARCH INDEX

The Inverted File (IVF) Index is a technique, among others, designed to enhance search efficiency by narrowing the search area through the use of neighbor partitions or clusters. The illustration is shown in Figure 3.5.

In the training phase, the dataset undergoes clustering, typically using algorithms like k-means, to partition the vector space into multiple clusters. Each cluster is represented by a centroid, serving as a reference point for the vectors within that cluster.

During index construction, each vector in the dataset is assigned to its nearest centroid. This assignment creates an inverted file list, where each list corresponds to a centroid and contains the vectors closest to it.

When a query vector is introduced, the system calculates its distance to all centroids and selects the top nearest clusters. The search is then confined to these clusters, significantly narrowing the search space. Within the selected clusters, the system computes the distances between the query vector and the vectors in the inverted lists to identify the most similar vectors.

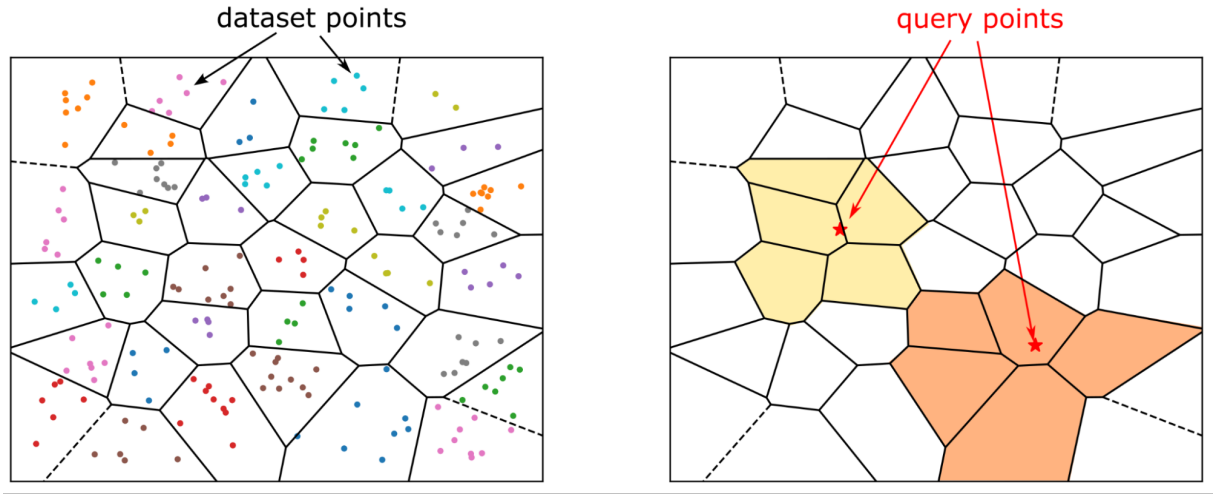


FIGURE 3.5
IVF index illustration (Douze et al. 2024)

By focusing the search on a subset of clusters, the IVF Index reduces computational load and accelerates query response times, making it particularly effective for large-scale, high-dimensional vector data.

Since index-based search may yield incomplete results due to clustering limitations, we also employ matrix multiplication to perform a comprehensive search. This approach directly utilizes the GPU to boost matrix multiplication and calculate the embedding similarity across the entire dataset, ensuring that no potential matches are missed. However, this method may introduce more false positives due to the lack of pre-filtering, highlighting a trade-off between completeness and precision in the search results.

3.6 EMBEDDING VISUALIZATION

Wizmap (Wang, Hohman and Chau 2023) is utilized to visualize the density distribution of embeddings across different regions by contours. The entire embedding space is divided into grids, with summaries provided for each grid cell. When hovering over a specific point, the corresponding pattern becomes visible, offering an intuitive way to explore and interpret the spatial distribution of patterns.

3.7 ENTIRE MAP MATCHING

Since we want to find related maps based on the elements. we calculate the the number of similar elements between two maps to evaluate their similarity rather than directly use the embedding of the whole map.

The similarity calculation between maps is performed in two stages to ensure accuracy and clarity:

1. **Element-wise Similarity Calculation:**
    - For each segmented element e_i from the query map M_q, calculate its similarity with all elements from each candidate map M_c.
    - The similarity score for each element is calculated as the average of the similarity values between e_i and all elements e_j in the candidate map:

$$S(e_i, M_c) = \frac{\sum_j \sigma(e_i, e_j)}{n_c}, \quad e_j \in M_c, \quad n_c = |M_c|$$

2. **Map-level Similarity Aggregation:**
    - After computing the similarity for all segmented elements from the query map, the similarity scores are aggregated across the map level by calculating the average of the similarity contributions of all elements:

$$S(M_q, M_c) = \frac{\sum_i S(e_i, M_c)}{n_q}, \quad e_i \in M_q, \quad n_q = |M_q|$$

3. **Ranking Similarity Scores:**
    - Once the map-level similarity scores are calculated, rank all candidate maps based on their aggregated similarity scores to identify the most similar maps.

This two-step approach ensures both fine-grained element-level comparison and global map-level similarity assessment, producing a reliable ranking of similarity scores among maps.

3.8 INTERFACE

In the interface design (Figure 3.6), we adopt a frontend-backend separation architecture. For the frontend, we use Vue.js (You 2014), a progressive JavaScript framework known for its flexibility, reactivity, and seamless integration with existing projects, to build the primary user interface. Additionally, we embed WizMap, developed with Svelte (Harris 2016), via an iframe. Svelte, a modern JavaScript framework, performs most of its processing at compile-time, enabling the creation of highly efficient and lightweight applications with minimal runtime overhead.

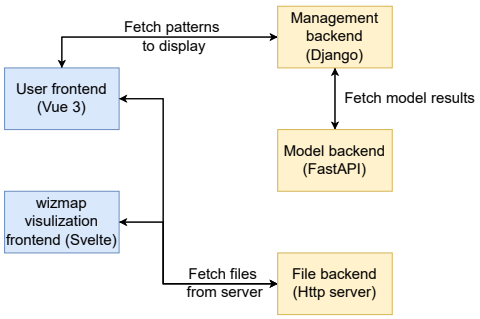


FIGURE 3.6
Interface framework

On the backend, we utilize Django (Foundation 2005), a high-level Python web framework known for its robustness and scalability, to handle frontend requests, and FastAPI (Ramírez 2018), a modern, fast, and highly efficient web framework for building APIs, to manage model responses, enabling more granular processing of data and computational tasks. Additionally, a file server is provided to facilitate efficient access to relevant file data by the frontend.

CHAPTER 4

RESULTS

4.1 SEGMENTATION

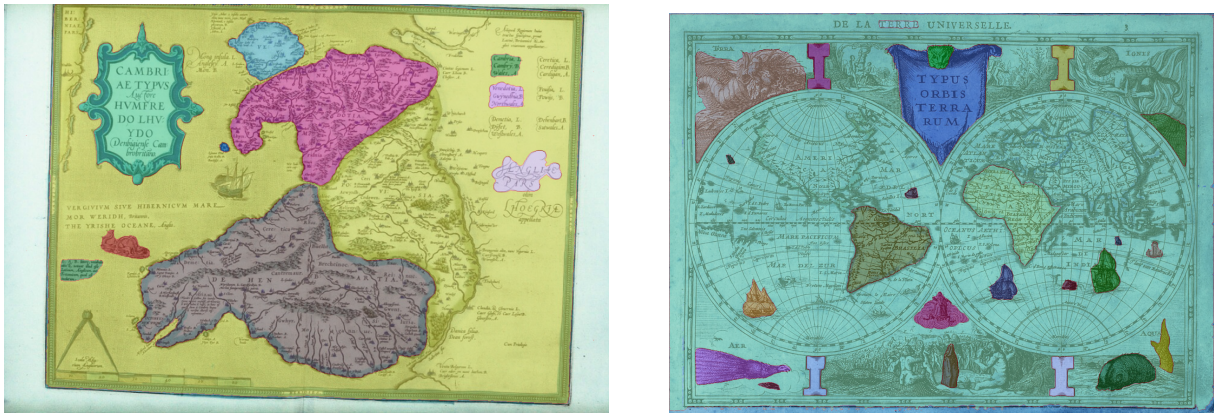


FIGURE 4.1
Map segmentation examples

We provide two examples of segmented maps shown in Figure 4.1. Additionally, to assess the quality of our segmentation work, we also rely on Predicted IoU (Intersection over Union) and Stability Score, as our dataset lacks ground truth annotations. These metrics provide an indirect evaluation of segmentation quality. The definitions of IoU, Predicted IoU, and Stability Score are outlined below.

IoU is defined as:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

- A: Predicted segmentation region by the model.
- B: Ground truth segmentation region.
- $|A \cap B|$: Number of pixels in the intersection of the predicted and ground truth regions.
- $|A \cup B|$: Number of pixels in the union of the predicted and ground truth regions.

Predicted IoU represents an estimated IoU predicted by a model, serving as a confidence score for the quality and accuracy of the segmented result. typically via an auxiliary branch:

$$\hat{IoU} = f(\mathbf{X})$$

- $\hat{IoU}$: Predicted IoU value.
- f: IoU prediction head (e.g., an auxiliary network branch).
- $\mathbf{X}$: Model features or predicted mask representation.

In our method, the prediction head is a component of the SAM2 network, which takes the image and mask as inputs and outputs the Predicted IoU value.

Stability measures the consistency of segmentation results under different thresholds. It is defined as:

$$S = \frac{|M_{\tau_1} \cap M_{\tau_2}|}{|M_{\tau_1} \cup M_{\tau_2}|}$$

- M_{τ_1}: Segmentation mask obtained with threshold τ_1.
- M_{τ_2}: Segmentation mask obtained with threshold τ_2.
- $|M_{\tau_1} \cap M_{\tau_2}|$: Number of pixels in the intersection of masks M_{τ_1} and M_{τ_2}.
- $|M_{\tau_1} \cup M_{\tau_2}|$: Number of pixels in the union of masks M_{τ_1} and M_{τ_2}.

In our method, we set the high threshold to 0.9 and the low threshold to 0.5.

TABLE 4.1
Predicted IoU and Stability Score Across Datasets

Dataset	Predicted IoU	Stability Score
Bodleian Library	0.9154	0.9630
Ryhiner-Sammlung	0.9292	0.9647
Bibliotheque Nationale de France	0.9183	0.9645
Biblioteca Digital Hispanica	0.9212	0.9643
John Carter Brown Library	0.9207	0.9656
Beinecke Library	0.9228	0.9653
David Rumsey	0.9173	0.9665
New York Public Library	0.9127	0.9656
Library Of Congress	0.9221	0.9643
Royal Museums Greenwich	0.9204	0.9647
Boston Public Library	0.9229	0.9657

For each dataset, we compute the average Predicted IoU and Stability Score across all segmented elements. The results are presented in Table 4.1. As shown, both metrics achieve consistently high values, exceeding 0.9 across all datasets. This indicates that the segmentation results are reliable, providing a solid foundation for the subsequent stages of the pipeline.

4.2 IMAGE CAPTION ANALYSIS

From the word cloud Figure 4.2, several key insights can be drawn by examining four distinct perspectives: Frequent Patterns, Artistic Styles, Urban Focus, and Regional Variation.

- **Frequent Patterns**: The word cloud reveals a strong emphasis on terms related to political and administrative geography, such as "state", "province", "region", "boundaries", and "capital". These

FIGURE 4.2
Word cloud created from captions generated from BLIP-2

patterns suggest that historical maps were primarily used for governance, territorial delineation, and administrative oversight. Additionally, the recurrence of temporal markers like "year", "century", and "1750" indicates the significance of historical timelines in these maps.

- **Artistic Styles**: Recurring references to colors ("blue", "white", "green") and geometric shapes ("rectangular", "circular", "curved") highlight consistent artistic conventions in mapmaking. These stylistic elements were likely employed not only for aesthetic purposes but also for enhancing clarity and differentiation of regions, landmarks, and annotations. The prominence of descriptive terms such as "labeled", "stylized", and "flat" further emphasizes the structured and intentional visual design of historical maps.
- **Urban Focus**: Terms such as "city", "town", "building", "castle", "street", and "house" suggest a significant focus on urban landscapes within historical maps. The recurring appearance of specific cities like "york", "kiev", "paris", and "london" indicates the prominence of these locations in historical cartography, either due to their political importance, economic significance, or cultural influence at the time.
- **Regional Variation**: The presence of diverse place names such as "france", "germany", "italy", and "kashmir" indicates a wide geographic scope in the dataset. Additionally, references to natural features like "river", "mountain", "forest", and "lake" suggest that historical maps often integrated both human settlements and natural landmarks. This diversity underscores the broad coverage and varied purposes of historical cartography, catering to both administrative needs and geographic exploration.

4.3 INDEX

In this part we try different k in the index to determine the cluster number and see the influence to the result. The cluster number is calculated as below. The formula is defined as:

$$\lfloor k \cdot \sqrt{n} \rfloor$$

where:

- k: A scaling factor or parameter.

- n: Represents the number of data points, expressed as $n =$ data.shape[0].
- $\sqrt{n}$: The square root of the number of data points.
- $\lfloor \cdot \rfloor$: The floor function, which rounds down to the nearest integer.

The parameter k significantly affects the clustering quality, retrieval performance, and overall search efficiency in the IVF (Inverted File Index) search process. Experimental observations highlight the following key relationships:

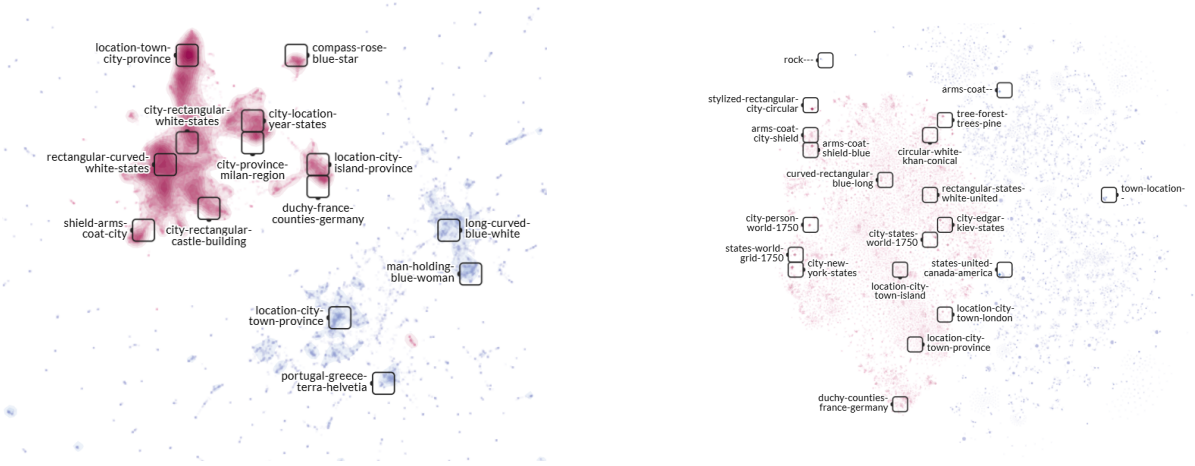
1. Inverse Relationship with Retrieved Elements:
    - The number of retrievable elements generally decreases as k increases.
    - At extreme values (e.g., $k = 0.5$ and $k = 5$), some categories, such as *animal*, fail to retrieve highly relevant images.
2. Impact on Query Quality:
    - Small k (e.g., 0.5–1):
        - Fewer clusters are generated.
        - The search results increase.
        - Index construction is faster, and memory usage is lower.
        - Each cluster contains a higher density of vectors, increasing intra-cluster noise and reducing search precision.
    - Large k (e.g., 2–4):
        - More clusters are generated, leading to finer granularity.
        - The search results decrease.
        - Index construction and query execution become more computationally intensive.
        - Excessive clustering (e.g., $k = 5$) may over-fragment the vector space, distributing relevant vectors across multiple clusters and degrading query results.

Poorly chosen values of k result in ineffective clustering, disrupting the balance between localization and coverage in the vector space. Both under-clustering (small k) and over-clustering (large k) can impair the effectiveness of the search. Based on the results in Table 4.2, we ultimately selected k = 1.

TABLE 4.2
The Number of Elements Query Across Different k Values

k **Value**	**Compass**	**Ship**	**Animal**	**Explanation**	**Cartouches**
$k = 0.5$	328	198	N/A*	262	238
$k = 1$	122	137	100	102	176
$k = 2$	78	85	37	83	147
$k = 3$	60	32	17	27	70
$k = 5$	41	39	N/A*	22	14

Note: * means the result is unreliable



(A) UMAP embedding visualization

(B) T-SNE embedding visualization

FIGURE 4.3

4.4 EMBEDDINGS VISUALIZATION

We try to use both UMAP (McInnes, Healy and Melville 2018) an T-SNE (Van der Maaten and Hinton 2008) to do dimensional reduction for the embedding, and the result is shown on 4.3. As the result, the text embedding is more separate compared to images, and UMAP is more density compared to T-SNE. Thus, in the following analysis, we utilize UMAP to examine the image embeddings of the dataset, enabling us to derive meaningful insights.

To generate appropriate contours for each grid, we utilize image captions as the basis for contour generation. However, to prevent interference from frequently occurring prepositions and meaningless words, we preprocess the captions beforehand.

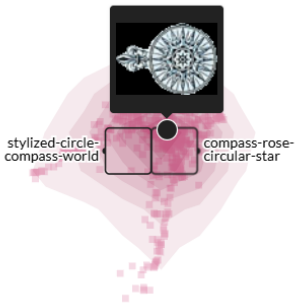
First, we convert all words to lowercase and remove punctuation marks. Next, we filter out commonly occurring stopwords such as "the", "shows", "image", and others. This preprocessing ensures cleaner and more relevant textual input for contour generation.

4.4.1 HIGH DENSITY ANALYSIS

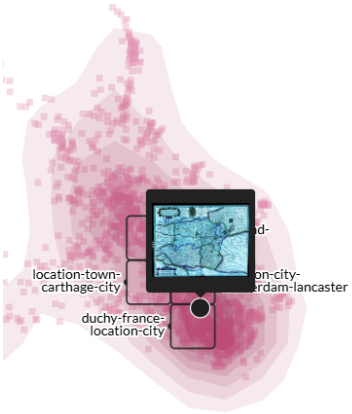
The density of feature maps indicates the number of similar patterns in a specific region. A denser region suggests that the patterns appear more frequently in the dataset. Analyzing high-density regions allows us to identify the types of patterns that frequently occur in historical maps, revealing commonalities across them. Additionally, the higher number of samples in these regions enables more in-depth analysis.

As shown in Figure 4.3a, three main clusters are visible in the feature space. The upper-right cluster (Figure 4.4a) is arranged in circular patterns, resembling a compass. Meanwhile, the lower-right cluster (Figure 4.4b) follows a rectangular pattern, resembling a complete map.

As for the largest cluster, the patterns within consist of irregular shapes. The densest section of the main part is illustrated in Figure 4.5a, highlighting an area with roads and landmark names. Additionally, Figure 4.5b depicts a region showcasing terrain features, such as mountains or trees, but without any accompanying text. Furthermore, Figure 4.5c defines the range of coat-of-arms, while Figure 4.5d focuses on the textual annotations present on the map.



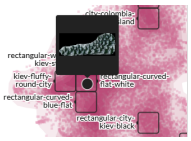
(A) A high-density cluster arranged in a circular pattern, resembling a compass layout



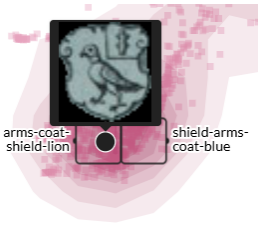
(B) A high-density cluster arranged in a rectangular pattern, resembling a full map

FIGURE 4.4

(A) A high-density cluster annotated with roads and landmark names



(B) A high-density cluster annotated with terrain features like mountains or trees



(C) A high-density cluster annotated with coat-of-arms

(D) A high-density cluster annotated with text illustration

FIGURE 4.5

(A) An outlier cluster annotated with a circular pattern and text labels on the pattern

(B) An outlier cluster annotated with portraits

(C) An outlier cluster annotated with a cityscape scenario in a rectangular layout

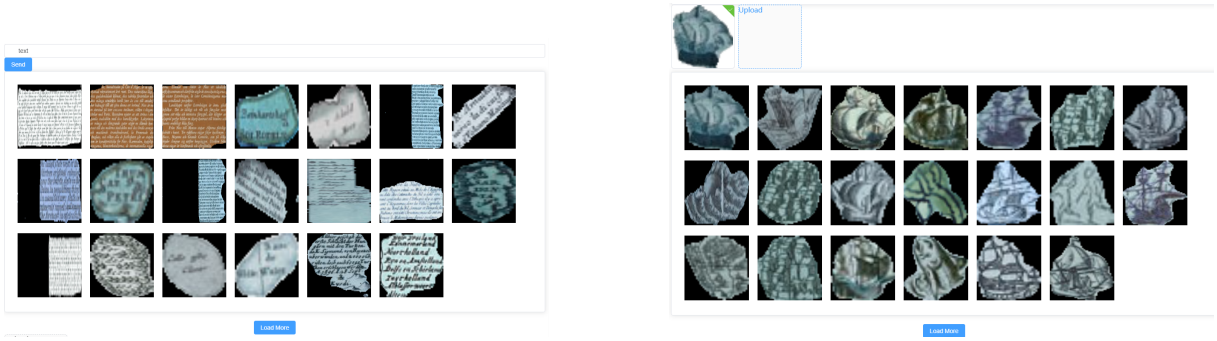
FIGURE 4.6

4.4.2 OUTLIER ANALYSIS

Outliers in feature maps reveal elements that rarely appear in the dataset. These anomalies can serve as special cases, offering valuable insights and contributing to a deeper understanding of the research subject.

The first cluster, shown in Figure 4.6a, highlights an outlier pattern with a circular layout accompanied by text annotations, suggesting a symbolic or emblematic representation. The second cluster, depicted in Figure 4.6b, showcases portraits, indicating a human-centric focus that stands out from other patterns. Lastly, Figure 4.6c illustrates an outlier cluster arranged in a rectangular layout, portraying a cityscape scenario with prominent river features. Each of these outlier clusters contributes to a deeper understanding of the dataset's diversity and exceptional elements.

4.5 INTERFACE



(A) Text Search Interface

(B) Image Search Interface

FIGURE 4.7

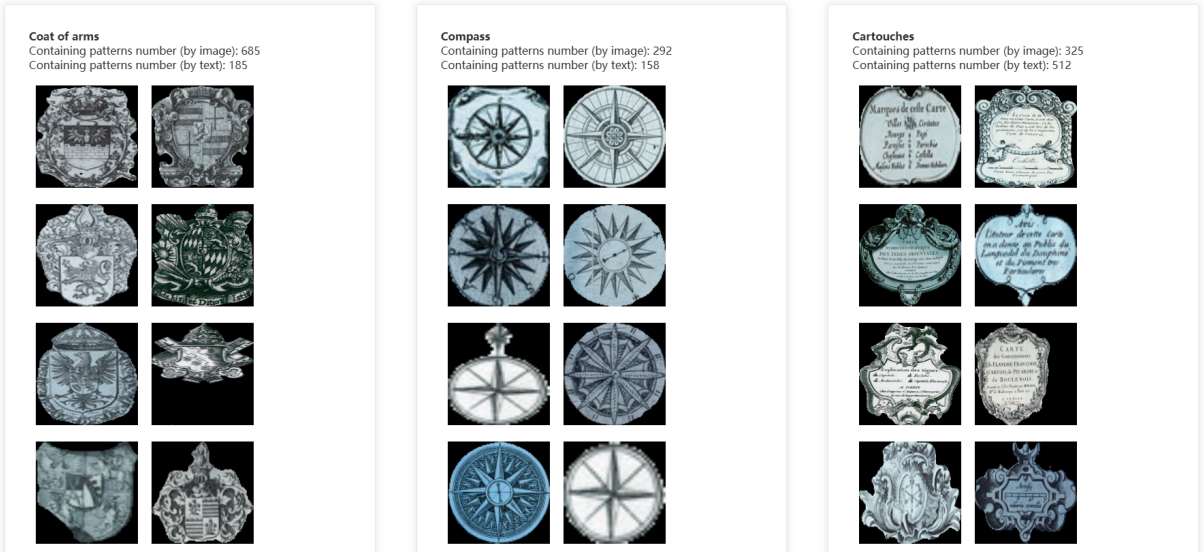
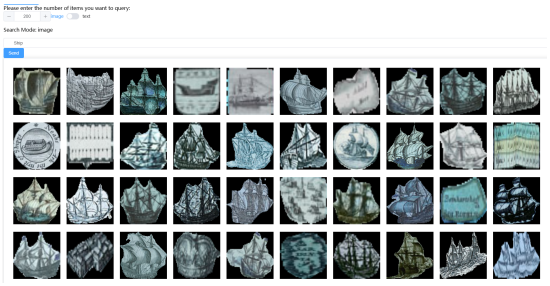


FIGURE 4.8
Categories Interface

The interface consists of several key modules. The first module comprises the embedding visualization (Figure 4.3a), described in the previous section, which allows users to interactively explore the existing embeddings in the dataset using Wizmap. Next is the text-based search (Figure 4.7a), enabling users to find similar images in the database based on textual queries. The image-based search (Figure 4.7b)

follows, where users can upload an image to retrieve visually similar ones from the database. Additionally, there are category windows (Figure 4.8), presenting representative elements commonly found in historical maps, along with their estimated frequency in the current dataset. These frequencies are calculated using normalized similarity distances, with an adjustable threshold available for users.

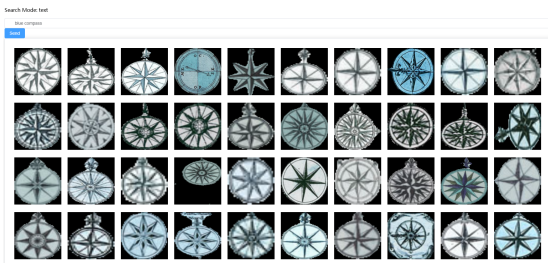


(A) Ship search results based on image embeddings

(B) Ship search results based on image caption embeddings

FIGURE 4.9

(A) Blue compass search results based on image embeddings



(B) Blue compass search results based on image caption embeddings

FIGURE 4.10

In terms of search modes, the unified feature space allows matching either with the image features or with generated captions. For example, when searching for the keyword "ship" (category search), matching using image embeddings produces results as shown in Figure 4.9a, while matching with text embeddings produces results as shown in Figure 4.9b. When using more complex commands, such as describing specific visual characteristics like color or shape, image embeddings yield more accurate search results. For instance, searching for "blue compass" using image embeddings produces results shown in Figure 4.10a, while matching with text embeddings gives results as shown in Figure 4.10b. The image embedding results are noticeably more accurate in this case. Therefore, both search modes are preserved to provide users with flexible options for obtaining optimal search results.

To facilitate the analysis of dataset characteristics and their relationships, a filtering function is also provided. Users can filter one or more datasets, and the corresponding displayed content will dynamically adjust based on the selected datasets.

4.6 ENTIRE MAP MATCHING

Figure 4.11 shows two matched maps, highlighting their shared artistic style and distinctive features. Both maps exhibit an intricate and highly detailed visual design, characterized by ornate decorations and elaborate illustrations.

A unique feature common to both maps is the prominent use of portraits. These portraits are not merely

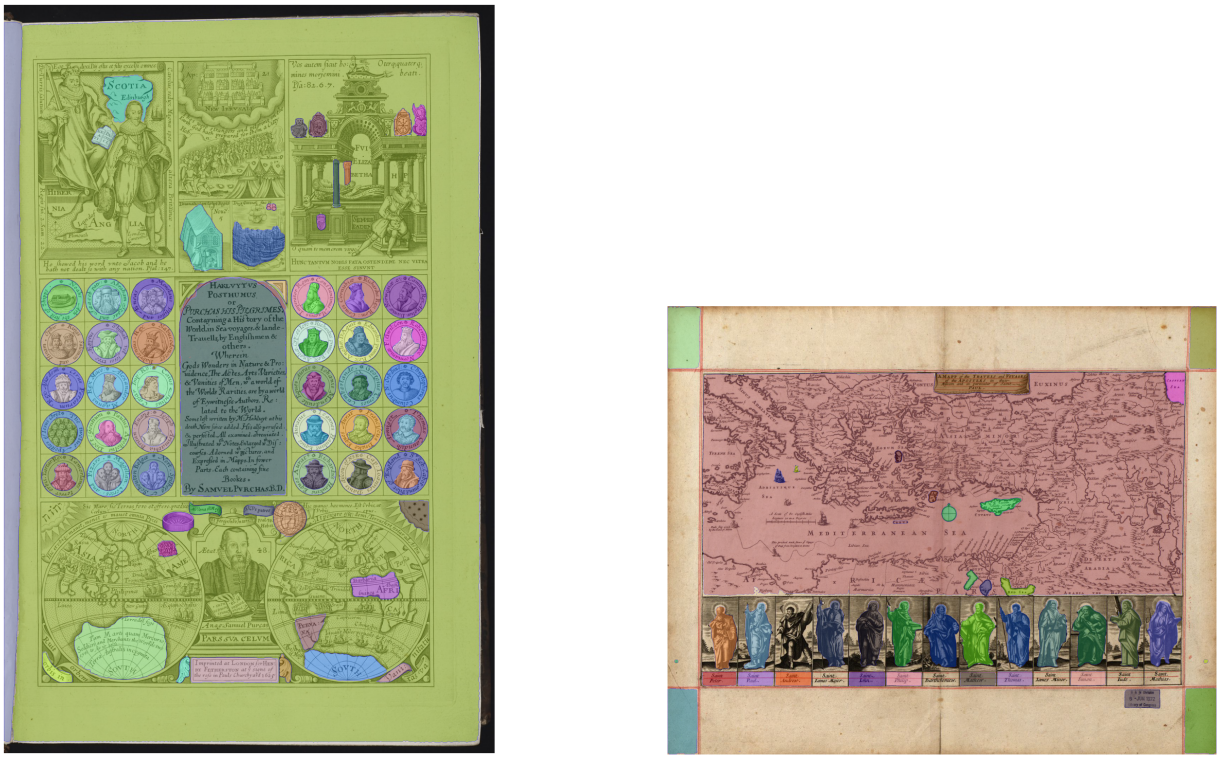


FIGURE 4.11
Example of two matched maps

decorative but play an integral role in the overall design and theme. In the first map, they are arranged in circular frames around the central text, showcasing historical figures or symbols. In the second map, portraits line the bottom edge.

This extensive use of portraits sets these maps apart from typical historical cartography, where human figures are often minimal or purely decorative. Here, they act as focal points, adding narrative depth. Such maps are rare in datasets and hold significant value for research.

By using this method, it becomes easier to efficiently identify a series of maps with similar unique characteristics, such as prominent portraits or specific design elements, enabling quick and accurate retrieval from large datasets.

4.7 ANALYSIS

4.7.1 STATISTICS

To estimate the number of specific elements on the map, we set a similarity threshold to identify similar images or image captions within the dataset.

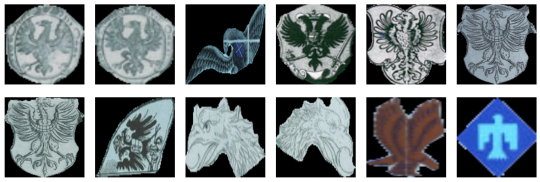
The result is shown in Table 4.3. Given that image embeddings are more prone to errors in certain contexts, we primarily rely on image caption embeddings to evaluate and refine the statistical count of these elements. This approach enhances the accuracy and reliability of our element estimation process.

4.7.2 ELEMENT FUNCTION

- **Animals**

Element Name	Number
Coat of arns	185
Compass	158
Cartouches	512
Legend	5
Ship	171

TABLE 4.3
Statistics about the number of elements



(A) Eagles in the dataset

(B) Camels in the dataset

(C) An eagle example

(D) A camel example

(E) A kangaroo example

(F) A virtual example

FIGURE 4.12
Examples of dataset analysis showing various animals.

(A) A castle depicted on a historical map.

(B) A castle represented as a coat of arms.

(C) A castle integrated into a landscape illustration.

(D) A castle depicted as an iconic church structure.

FIGURE 4.13
Visual Representations of Castles in Different Contexts.

Maps often feature a diverse array of animals, each serving distinct functions within the cartographic context. These functions vary based on the type of animal and their symbolic or representational roles on the map.

For example, animals such as lions or eagles (Figure 4.12a) are frequently used as heraldic symbols in coats of arms or cartouches to convey power, authority, or nobility, as illustrated in Figure 4.12c. In contrast, animals like camels (Figure 4.12b) or fish often serve as geographical markers, symbolizing specific terrains or ecosystems such as deserts or oceans, exemplified in Figure 4.12d.

Additionally, certain animals act as regional identifiers, representing local fauna unique to a specific area. For instance, the kangaroo is emblematic of Australia (Figure 4.12e), serving as both a cultural and geographical signifier.

Moreover, some animals are included purely for decorative purposes, contributing to the artistic embellishment of maps. A notable example is the presence of mythical sea creatures in oceanic regions, as shown in Figure 4.12f.

- **Prominent Buildings**

  Prominent buildings serve multiple purposes in historical and artistic maps, each representation conveying unique meanings and fulfilling specific functions. These depictions are not merely decorative but play essential roles in navigation, symbolism, and visual storytelling, reflecting both the practical needs and artistic intentions of the mapmakers.

  In many maps, distinctive buildings often act as landmarks for navigation (Figure 4.13a), where structures such as castles or prominent architectural features are used to indicate key locations. These landmarks provide clear visual cues, helping viewers identify specific regions or important points on a journey. The representation of these landmarks is usually precise and highlights recognizable architectural details, ensuring they can be easily interpreted.

  Beyond their functional role, salient buildings are also commonly depicted as symbols in coats of arms (Figure 4.13b), where they represent authority, ownership, and heritage. These stylized representations are rich in cultural and political significance, often signifying power or territorial control. The use of castles, in particular, within coats of arms underscores their historical association with protection and governance.

  Additionally, well-known buildings often appear as integral parts of landscape illustrations (Figure 4.13c), blending seamlessly into painted scenery on maps. These artistic representations emphasize the aesthetic qualities of the region, capturing both the natural beauty and architectural character of the landscape. This form blurs the boundary between cartography and fine art, transforming maps into evocative visual narratives.

  In some instances, prominent buildings are portrayed as iconic structures (Figure 4.13d), where stylized depictions prioritize symbolism over architectural accuracy. Churches, castles, and other monumental buildings are simplified into iconic forms, ensuring immediate recognition by viewers. These representations are not only symbolic but also help establish a sense of place and identity within the map.

CHAPTER 5

CONCLUSION

In this study, we introduced an integrated pipeline for analyzing historical map layout elements using state-of-the-art machine learning models, including SAM2, BLIP-2, and ImageBind. Our results demonstrate consistent segmentation quality, validated through high Predicted IoU and Stability Scores across various datasets. The integration of multimodal embeddings enabled efficient cross-modal querying, while dimensionality reduction methods such as UMAP provided intuitive insights into embedding distributions. This approach successfully bridges traditional historical cartography analysis with modern AI-driven techniques, offering scalable and interactive tools for researchers and historians to uncover new insights from historical cartographic data.

However, challenges remain. The BLIP-2 model exhibits slow inference speeds, limiting real-time analysis capabilities. Future work will explore the adoption of lightweight models to reduce computational overhead and improve efficiency. Additionally, integrating metadata, including temporal annotations, will allow filtering and analysis of maps across different historical periods, unlocking new dimensions for temporal cartographic studies.

With these enhancements, the pipeline has the potential to become an even more powerful tool for historical map analysis, facilitating deeper exploration and richer insights into the cultural and historical narratives embedded in these cartographic artifacts.

BIBLIOGRAPHY

Petitpierre, Rémi (2021). 'Neural networks for semantic segmentation of historical city maps: Cross-cultural performance and the impact of figurative diversity'. In: *arXiv preprint arXiv:2101.12478*.

Xia, Xue, Magnus Heitzler and Lorenz Hurni (2022). 'CNN-Based template matching for detecting features from historical maps'. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43, pp. 1167–1173.

Sertel, Elif, Can Michael Hucko and Mustafa Erdem Kabadayı (2024). 'Automatic Road Extraction from Historical Maps Using Transformer-Based SegFormers'. In: *ISPRS International Journal of Geo-Information* 13.12, p. 464.

Wu, Sidi et al. (2023). 'Cross-attention Spatio-temporal Context Transformer for Semantic Segmentation of Historical Maps'. In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–9.

Li, Junnan, Dongxu Li, Caiming Xiong et al. (Feb. 2022). *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. arXiv:2201.12086. URL: http://arxiv.org/abs/2201.12086 (visited on 19th Nov. 2024).

Li, Junnan, Dongxu Li, Silvio Savarese et al. (June 2023). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. arXiv:2301.12597 version: 3. URL: http://arxiv.org/abs/2301.12597 (visited on 19th Nov. 2024).

Han, Jiaming et al. (2023). 'Imagebind-llm: Multi-modality instruction tuning'. In: *arXiv preprint arXiv:2309.03905*.

Jiang, Ting et al. (2024). 'E5-v: Universal embeddings with multimodal large language models'. In: *arXiv preprint arXiv:2407.12580*.

Ravi, Nikhila et al. (2024). 'Sam 2: Segment anything in images and videos'. In: *arXiv preprint arXiv:2408.00714*.

Douze, Matthijs et al. (2024). 'The Faiss library'. In: arXiv: 2401.08281 [cs.LG].

McInnes, Leland, John Healy and James Melville (2018). 'Umap: Uniform manifold approximation and projection for dimension reduction'. In: *arXiv preprint arXiv:1802.03426*.

Van der Maaten, Laurens and Geoffrey Hinton (2008). 'Visualizing data using t-SNE.' In: *Journal of machine learning research* 9.11.

Radford, Alec et al. (2021). 'Learning transferable visual models from natural language supervision'. In: *International conference on machine learning*. PMLR, pp. 8748–8763.

Wang, Zijie J, Fred Hohman and Duen Horng Chau (2023). 'Wizmap: Scalable interactive visualization for exploring large machine learning embeddings'. In: *arXiv preprint arXiv:2306.09328*.

You, Evan (2014). *Vue.js: The Progressive JavaScript Framework*. https://vuejs.org/. Accessed: 2024-01-06.

Harris, Rich (2016). *Svelte: Cybernetically Enhanced Web Apps*. https://svelte.dev/. Accessed: 2024-01-06.

Foundation, Django Software (2005). *Django: The Web Framework for Perfectionists with Deadlines*. https://www.djangoproject.com/. Accessed: 2024-01-06.

Ramírez, Sebastián (2018). *FastAPI: Fast and Highly Performant Web Framework for Building APIs*. https://fastapi.tiangolo.com/. Accessed: 2024-01-06.