

On self-supervision in historical handwritten document segmentation

Josef Baloun^{1,2*}, Martin Prantl^{1,2}, Ladislav Lenc^{1,2}, Jiří Martínek^{1,2}, Pavel Král^{1,2}

¹Department of Computer Science and Engineering, University of West Bohemia, Univerzitní, Pilsen, 30100, Czech Republic.

²NTIS - New Technologies for the Information Society, University of West Bohemia, Univerzitní, Pilsen, 30100, Czech Republic.

*Corresponding author(s). E-mail(s): balounj@kiv.zcu.cz;

Contributing authors: perry@kiv.zcu.cz; llenc@kiv.zcu.cz; jimar@kiv.zcu.cz; pkral@kiv.zcu.cz;

Abstract

Historical document analysis plays a crucial role in understanding and preserving our past. However, this task is often hindered by challenges such as limited annotated training data and the diverse nature of historical handwritten documents. In this paper, we explore the potential of self-supervised learning (SSL) in historical document analysis, with a particular focus on historical handwritten document segmentation, to overcome the need for extensive annotated data while enhancing efficiency and robustness. We present an overview of SSL methods suitable for historical document analysis and discuss their potential applications and benefits. Furthermore, we present an approach for SSL in the document domain, considering various setups, augmentations, and resolutions. We also provide experimental results that demonstrate its feasibility and effectiveness. Our findings indicate that most document segmentation tasks can be effectively addressed using SSL features, highlighting the potential of SSL to advance historical document analysis and pave the way for more efficient and robust document processing workflows.

Keywords: Historical handwritten document, self-supervised learning, document digitization, semantic segmentation

1 Introduction

Historical documents housed in various national archives contain a wealth of valuable data and insights into our history, including details about past weather conditions, disease epidemics and many other pieces of information. Extensive efforts have been invested in digitizing these documents to facilitate electronic access. However, the utilization of these digitized documents remains

somewhat constrained due to limitations in search functionalities available in the archives.

Efficient searching is possible only after processing the documents, analyzing them and associating them with metadata. Manual processing is not realistic and thus machine learning has a large potential in this field. Although a significant progress has been made in the field of historical document analysis, leveraging deep learning techniques to achieve impressive results, there are

still many challenges in generalization capabilities. This is attributed to the distinct and often unique characteristics of each document set. Additionally, the scarcity of annotated training data poses a significant obstacle, as it is both time-consuming and costly to produce. For instance, a model trained on one particular chronicle may not be applicable to another one due to the differing characteristics of them. As a result, efforts are primarily focused on processing individual chronicles, with minimal opportunity for model reuse across different document sets.

To address the challenge of limited training data, artificial training data is frequently generated to mimic the characteristics of the target domain. However, this approach requires considerable effort and may introduce unnecessary complexity, essentially shifting the problem to generating training data. Moreover, pre-training on synthetic data may result in synthetic features that are not optimal and may induce problems. As an alternative, we are exploring the potential of self-supervised learning (SSL) for document analysis, since the data without annotations are available and cheap.

Our findings demonstrate that SSL features are highly effective for most document segmentation tasks, achieving performance comparable to fine-tuning methods such as DiT [16]. Using principal component analysis (PCA), we successfully identify major document categories without the need for annotated data. For more complex segmentation tasks, we apply a linear transformation approach to learn the mapping of features to specific classes, achieving competitive results with fully supervised methods. This underscores the effectiveness of SSL in few-shot learning scenarios and datasets with limited annotated data. Additionally, we show that fine-tuning self-supervised models enhances segmentation quality, particularly by mitigating challenges such as faint or noisy text and document artifacts.

2 Related work

SSL is a machine learning approach where models generate their own training signals from the data, eliminating the need for human-provided labels. The model leverages inherent data structures by solving tasks designed to capture key features. This often involves augmenting data to

form related sample pairs, where one sample acts as input and the other as the supervisory signal. SSL is often used for model pre-training, followed by fine-tuning on a downstream task, as the designed task typically differs from the final downstream task.

Pre-training techniques have become a standard in various tasks, particularly in natural language processing (NLP). These approaches demonstrate the significant potential of leveraging large-scale learning to enhance model performance on downstream applications without requiring costly annotations. However, in document analysis and segmentation, many pre-training approaches rely on annotations or techniques to simulate them, such as object detection or optical character recognition (OCR) systems and they are thus prone to errors of these dependencies that can be reflected in the model.

LayoutLM [30] is a pre-trained model inspired by BERT [7]. It combines visual and textual modalities by employing an OCR system and an object detector, enhancing the textual embeddings by adding corresponding image embeddings. Representations are pre-trained using document classification and masked visual-language model (MVLM). In this process, some input tokens are randomly masked while their corresponding 2D position embeddings are preserved. The MVLM then aims to predict the masked tokens based on the given context. LayoutLM inputs are revised in [24] to enable multi-page processing. The pre-training tasks of document classification and MVLM are extended by incorporating document shuffle prediction and document topic modelling.

SelfDoc [17] also employs an object detector and OCR, but the textual and image modalities are firstly encoded separately before being combined using a cross-modality encoder. Masking is then applied to the individual image or textual embeddings, with the goal of reconstructing them.

Text region-level masking strategy is presented in [31]. The suggested approach involves the random masking of image regions using text word bounding box coordinates. During pre-training, the objective is to simultaneously reconstruct both the pixels within the masked regions of the image and the corresponding masked tokens.

Document understanding transformer (Donut) [14] is an end-to-end framework for visual document understanding that uses an encoder to

process the image and a decoder to generate textual output. It maps an input document image into a desired structured output which can be converted to formats like JSON, for example. Initially pre-trained for OCR, it subsequently undergoes fine-tuning for tasks such as document classification, document information extraction, and document visual question answering.

SSL is closing in on the performance of supervised methods on major computer vision benchmarks. A typical approach aims to learn representations that stay consistent despite input deformations. However, this can result in trivial solutions, that current methods avoid by various techniques such as SimCLR [4], Bootstrap Your Own Latent (BYOL) [12], and Barlow Twins [32].

SimCLR builds on contrastive learning by using augmentations to create positive pairs, while negative pairs are not explicitly sampled but are instead taken from other examples within a mini-batch. The similarity of positive pairs is trained to be maximized, while the similarity of negative pairs is minimized.

BYOL relies on two neural networks that learn from each other. Using image augmentation, the *online* network is trained to predict the representations of the *target* network. Concurrently, the *target* network is updated using a slow-moving average of the *online* network’s weights. The authors suggest that BYOL’s dynamics are analogous to those in generative adversarial networks (GANs) [11], and they hypothesize that no single loss function exists to simultaneously minimize the parameters of both the online and target networks. The slow-moving average mechanism helps avoid undesirable equilibria and, consequently, trivial solutions.

Barlow Twins, similarly to SimCLR, operates with a single network that processes two augmented variants of an image. Within a batch, an objective function forces the cross-correlation matrix of these two outputs close to the identity matrix. Compared to BYOL, it does not require moving average or asymmetry between network twins. It neither requires explicit negative pairs.

General SSL methods in computer vision, like DINOv2 [22], struggle with text-related tasks and mainly perform document page segmentation in the scene. Their lack of specialization for textual documents limits their effectiveness in handling

degradation, diverse writing styles, and complex layouts.

SSL for document image classification is presented in [6]. The self-supervised pre-training involves predicting flips and solving jigsaw puzzles. The network is pre-trained to predict if the input was flipped or the position of the jigsaw puzzle patch. The authors state that patch-based pre-training performs poorly on document images because of their different structural properties.

SelfDocSeg [19] extends BYOL for self-supervised visual pre-training by incorporating layout masks generated through classical image processing techniques as visual guidance. In this extended BYOL framework, these masks act as ground truth for the online network, which is trained using a focal loss. Following this pre-training phase, the encoder is fine-tuned for use as an object detector.

Document image transformer (DiT) [16] is a vision transformer based on BERT pre-training of image transformers (BEiT) [3]. Just as text can be tokenized into discrete textual tokens, an image can be represented as a sequence of discrete tokens produced by an image tokenizer. In this process, a discrete variational auto-encoder acts as the tokenizer, predicting tokens that index an embedding table, from which the image can be decoded back to its original form. DiT is pre-trained on document images in a self-supervised manner using image patch masking. The image is first divided into patches according to the tokenizer, then some patches are masked, and the task is to predict the corresponding token retrieved by the discrete variational auto-encoder tokenizer.

Finally, the paper Learning to Read by Spelling [13] introduces an intriguing self-supervised OCR method. The approach breaks OCR down into two sub-problems: segmenting and clustering characters, and solving a 1:1 substitution cipher. A GAN model is seeded with a text image to generate real text. However, there are several constraints, such as hardcoded character positions, which, along with the variations between handwritten and printed characters, pose challenges for adapting this method to handwritten text recognition.

3 Dataset

Currently, there are numerous datasets available for various tasks in historical document analysis. However, their characteristics, languages, annotations, and targeted tasks often differ significantly, making them challenging to use for large-scale traditional supervised learning.

DIVA-HisDB [26] dataset aims at semantic segmentation, binarization, text line segmentation, and detection. It consists of three sub-datasets: CSG18, CSG863 and CB55 written in Latin and Italian language. Each contains pixel-level annotation for the main text body, comments, and background. Each subset has consistent characteristics. Train, validation, and test splits contain 20, 10, and 20 samples respectively.

The primary focus of the cBAD dataset [8] is baseline detection. It includes annotations in PAGE format [23] for text regions, text lines, and baselines across a variety of documents. These documents are categorized into two subsets based on their layout complexity: simple and complex.

ChronSeg [2] is another diverse dataset for semantic segmentation of text and images. Its main part is written in the Czech language and contains double-sided pages from five chronicles and annotations in both PAGE format and PNG masks.

GRPOLY-DB [10] is a collection of machine-printed and handwritten old Greek documents. It contains PAGE annotations for text region, text line, word, and word-level text transcription. Therefore, it may be used for a wide range of document analysis tasks.

Bentham [27] is an English dataset containing scanned pages, lines, and transcriptions. PAGE format annotations are provided. Pages were written by the philosopher Jeremy Bentham and his secretaries on the topic of law and moral philosophy.

READ dataset [28] contains pages and annotations in PAGE format. The baselines and line bounding polygons are available with transcription. It was written in Early Modern German from 1470 to 1805 by an unknown number of writers. An interesting feature of the dataset is the presence of bleed-through text.

The IAM dataset [20] is a well-known dataset for handwriting English text recognition and contains 1,539 pages of scanned text by 657 authors.

It contains text annotation on page, line, and word-level. There are several splits as described in [21].

CVL-DataBase [15] contains XML annotations for writer retrieval, writer identification, and word spotting. The dataset contains one German and six English texts and 311 writers.

The dataset for the historical writer identification task is presented in [9]. It contains 3,600 handwritten pages from 720 different writers originating from 13th to 20th century. This dataset has been further expanded to include up to 20,000 pages in [5].

3.1 Data preparation

For *pre-training*, we used all publicly available datasets we could find. These datasets exhibit a wide range of characteristics, including grayscale and color pages, handwritten and printed text, graphics, various writing styles and formats, single and double pages, page degradation, and different languages. Specifically, we used:

- 90 samples from the training and validation splits of the DIVA-HisDB dataset [26].
- 10 samples from the training and validation splits and 40 samples from the experimental part of the ChronSeg dataset [2].
- 2035 samples from the cBAD dataset [8].
- 46 samples from the GRPOLY-DB dataset [10].
- 433 samples from the Bentham dataset [27].
- 450 samples from the READ dataset [28].
- 1539 samples from the IAM dataset [20].
- 1604 samples from the CVL-DataBase [15].
- 4782 samples from the dataset for historical writer identification [9].
- 1200 samples from the validation split of [5].

This adds up to a total of 12,229 scanned pages, regardless of whether they are single-sided or double-sided. Although the distribution of data is not ideal, it is important to note that most datasets are highly diverse, containing several writers and styles, or are composed of multiple datasets or subsets.

For the pre-training phase, we did not require annotations, simplifying the process as there was no need to unify the classes or format, which is often unique to each dataset (e.g., PAGE format, binary masks, bitwise encoded masks). On the other hand, there are variations in resolution

Table 1 Number of pages for each evaluation dataset and its corresponding splits.

| | cBAD | | DIVA-HisDB | | | ChronSeg |
|------------|--------|---------|------------|------|-------|----------|
| | simple | complex | CB55 | CS18 | CS863 | |
| train | 87 | 99 | 20 | 20 | 20 | 6 |
| validation | 22 | 25 | 10 | 10 | 10 | 4 |
| test | 107 | 123 | 20 | 20 | 20 | 8 |

and scan quality. While different scan processes should be acceptable for robust feature extraction, variations in resolution may pose a problem (e.g., different resolutions even within a single dataset like DIVA-HisDB, where CB55 has a resolution of 4872x6496 and other subsets have a resolution of 3328x4992). This can result in significantly different text sizes, which may be problematic for the encoder.

To analyze this, we used two variants: the original *full* resolution and a downsampled resolution where the page height is limited to *1024* pixels. Although there may be some outliers, this approach generally works well, yielding a similar text pixel size for the majority of pages. No additional preprocessing was applied.

We aimed to select a diverse set of datasets covering various styles and segmentation classes—such as text regions, text lines, baselines, images, decorations, backgrounds, and comments—to ensure a comprehensive evaluation. For *probing* and *evaluation*, we selected the cBAD (both simple and complex subsets), DIVA-HisDB (all CB55, CS18, and CS863 subsets), and ChronSeg datasets. Specifically, we consider cBAD and ChronSeg to be diverse, as they encompass multiple sources, while DIVA-HisDB is comparatively less diverse, consisting of three relatively homogeneous subsets. This selection allows us to evaluate our approach under different conditions. While we believe the chosen samples are representative, we acknowledge that, as computer scientists rather than historians, we are not in a position to definitively confirm their historical representativeness.

The details of the splits are shown in Table 1. ChronSeg and DIVA-HisDB have a predefined train, test, and validation split. In contrast, for cBAD, annotations were only partially available for the training part. Therefore, we created splits

by sequentially assigning samples to train, validation, and test splits in a 0.4, 0.1, and 0.5 ratio, respectively.

4 Approach

In the document image domain, self-supervised learning faces several challenges. Beyond the substantial memory requirements and high-resolution data, it necessitates an effective model and approach to learn meaningful features.

Additionally, these features must be interpretable, which is particularly difficult without annotated training data. However, interpretability can be enhanced if we utilize certain hints, primarily through augmentations. Therefore, it is crucial to engineer these augmentations carefully to guide the learning process towards the desired outcome.

The resulting features can be visualized using principal component analysis (PCA). The linear transformation obtained through PCA can then be thresholded to distinguish between two main categories. However, since there are typically more than two categories or these categories are solely defined by statistical significance, a separate linear transformation can be learned for individual classes within a specific dataset. To address this, we draw inspiration from the probing techniques used with large language models, as demonstrated in [29], where a probing model is trained on top of a frozen language model for a downstream task. Utilizing annotated data, we learn a simple linear transformation as a probing model to map features into corresponding classes, allowing us to evaluate and analyze the relevance of the learned features for various objectives.

The following sections detail our approach and model for self-supervised pre-training, the augmentation strategies employed. We also explore

the direct application of these features using PCA and conclude with probing to map the features.¹

4.1 Self-supervised pre-training

We utilize Barlow Twins [32] for its straightforward concept. It does not require annotations and multiple versions of the model. On the other hand, it benefits from a large batch size and projector size, making it memory-intensive.

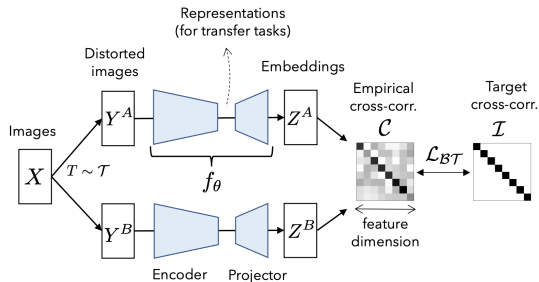


Fig. 1 Barlow Twins method for SSL [32]: It passes two augmented images through a single encoder-projector model, generating two embeddings. Forcing empirical cross-correlation matrix to identity causes the augmented sample embeddings to be similar to each other and distinct from embeddings of other samples within the same batch.

As shown in Figure 1, during SSL, the model learns to undo the augmentations applied to the image, or more precisely, it learns that the features should closely align with those of the original image. It should be evident that this is the characteristic of robust features.

4.1.1 Feature encoder

The feature encoder (FE) is a critical component of our approach, as it generates the essential features. We employed a straightforward yet effective arrangement of convolutional layers, max-pooling, and batch normalization layers. This setup is common in computer vision (CV) and forms the core of general fully convolutional networks (FCNs), widely used for semantic segmentation tasks.

Similarly to the contracting path of the U-Net [25] or dhSegment [1], the FE is composed of multiple blocks. As illustrated in Figure 2, each

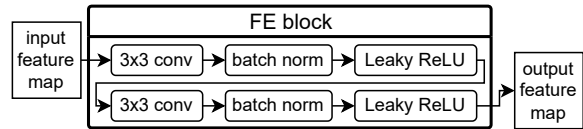


Fig. 2 Feature encoder block

block includes a 3x3 convolutional layer without padding, followed by batch normalization and a Leaky ReLU activation function, all repeated twice

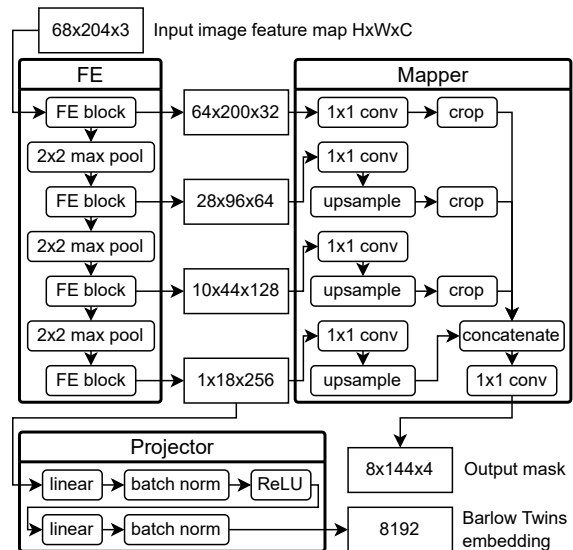


Fig. 3 Overall architecture with four feature encoder (FE) blocks and an example 68x204 input image: the FE processes the input image, producing a feature map at each FE block. During the self-supervised pre-training phase, the FE is trained alongside a projector, which takes the feature map from the final FE block as input. After pre-training, a mapper linearly maps the features of individual pixels to class logits (four classes in this case).

We implemented two variants of the FE, featuring four or five blocks, with max-pooling layers separating the blocks and an increasing number of filters in each subsequent block, as illustrated in Figure 3. This design results in receptive fields of 68x68 pixels for the 4-block variant and 140x140 pixels for the 5-block variant. For instance, in the context of downsampled data (see Section 3.1), 140 pixels roughly correspond to approximately 6 lines of text, depending on the dataset.

¹The source code, complete results, and example models are available at <https://gitlab.kiv.zcu.cz/balounj/semsegssl>

4.1.2 Projector

The projector processes the FE outputs into embeddings for the Barlow Twins SSL method. As illustrated in Fig. 3, it comprises two linear layers followed by a batch normalization layer and a ReLU activation function in between.

4.1.3 Augmentation

Self-supervision essentially learns to reverse the augmentation process. Therefore, appropriate augmentations are necessary to obtain high-quality features. However, identifying these augmentations can be challenging, often requiring substantial effort and expertise. As highlighted in [12], this challenge poses a major obstacle to extending self-supervised methods to other modalities.

In computer vision, self-supervision frequently employs random crops, assuming that the object of interest will appear in both crops. This technique helps the network to learn that different parts of the same object, like zebra’s head and leg, share similar features. It may also aid in associating the zebra with its typical environment.



Fig. 4 Input images (first row) and their corresponding augmented versions.

However, in document segmentation, it is important to avoid associating text with images or empty spaces, as these elements should have distinct features. Furthermore, we may want to ensure that different texts do not share overly similar features. Similarly to random crops, patch-based approaches also perform poorly due to little context [6]. Therefore, we believe it is helpful to maintain the same text content in both

cropped images. As augmentation of the crop (see Figure 4), we prefer techniques such as slight shifting, binarization, perspective warp, color jitter, or horizontal and vertical line masking, where the lines (with a width randomly set between 8 and 12 pixels) are assigned a random color.

4.2 Visualization of pre-trained features

Once we have multidimensional feature representation of each pixel or block of pixels (e.g. 8x8 tile), we need to utilize them effectively. If the features are meaningful, there should ideally be several categories of features. However, identifying these categories is not straightforward. While clustering is an option, inspired by DINOv2 [22], PCA appears to be a more straightforward approach.

For documents, the primary categories include text and background, and they may also include images. These categories tend to be relatively balanced, which can be advantageous for thresholding the first principal component (PC1), likely separating one of these primary categories.

After extracting the features, we apply PCA to determine the linear transformation for PC1, and then apply a default threshold of $t = 0$, aiming to distinguish the two major categories on the page. Following this, we select the thresholded features and perform PCA again to obtain the first three principal components, which are used for visualization as RGB channels.

As illustrated in Figure 5, the primary categories are truly distinguished by PC1. The visualization of the first three principal components reveals that, depending on the setup, the features may emphasize different aspects. Some setups may focus more on text or images, highlighting details such as font style, bleed-through text, or baselines.

Generally, the visualization outcome is seldom related to the target task, as defined by the dataset’s annotations, and this relationship is difficult to establish. However, this does not mean the features are useless; it simply suggests that we currently lack an effective method to extract the relevant information, as illustrated in Figure 6.

4.3 Mapping of pre-trained features

For the evaluation, a linear mapping of features can be trained using 1x1 convolutions. Because

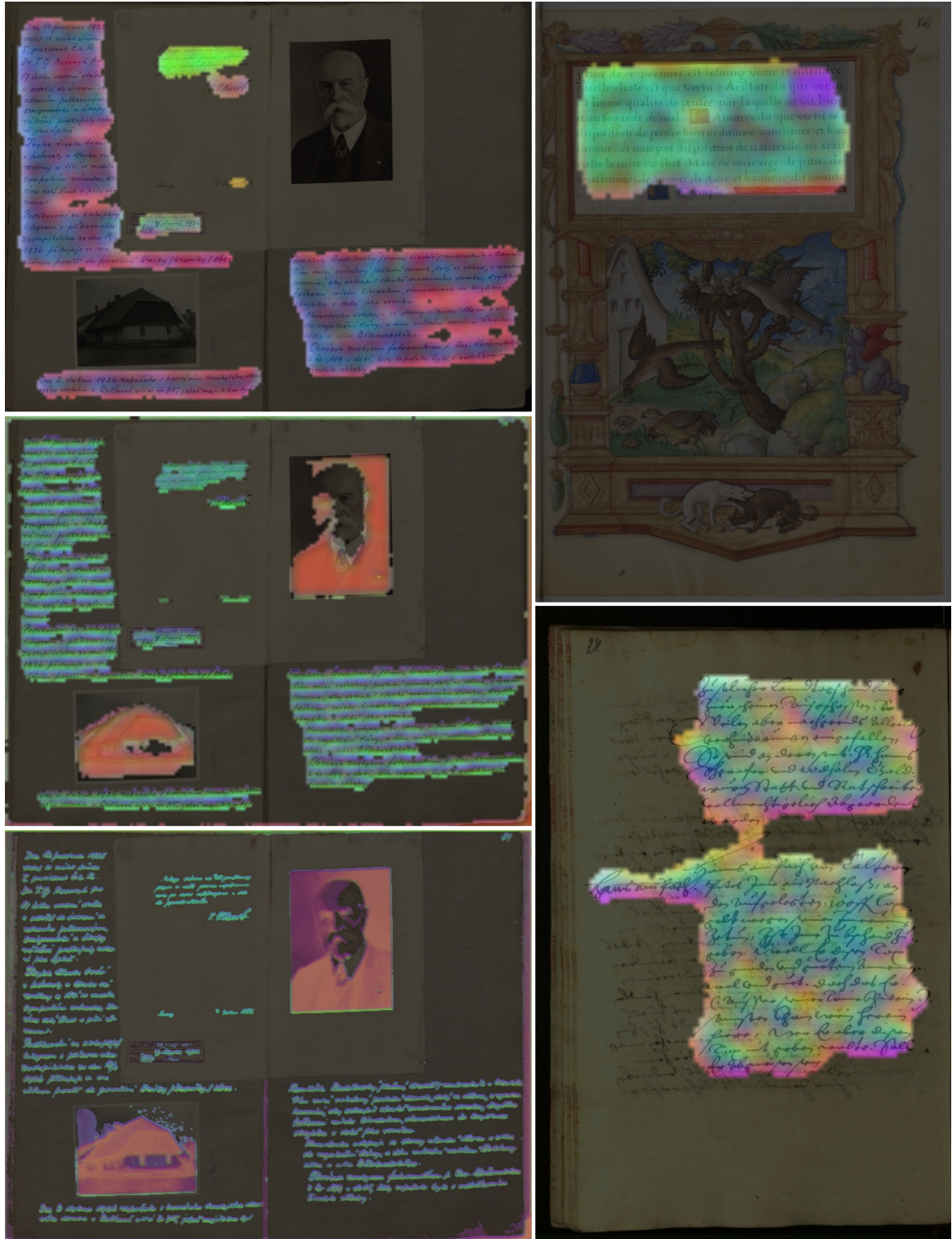


Fig. 5 Examples of thresholding the first principal component: The left images are from ChronSeg [2], the upper right from the READ dataset [28], and the lower right from the ICDAR 2019 Writer Identification Competition [5]. Features are trained with various setups on downsampled images, except for the lower left, which is trained on full resolution.

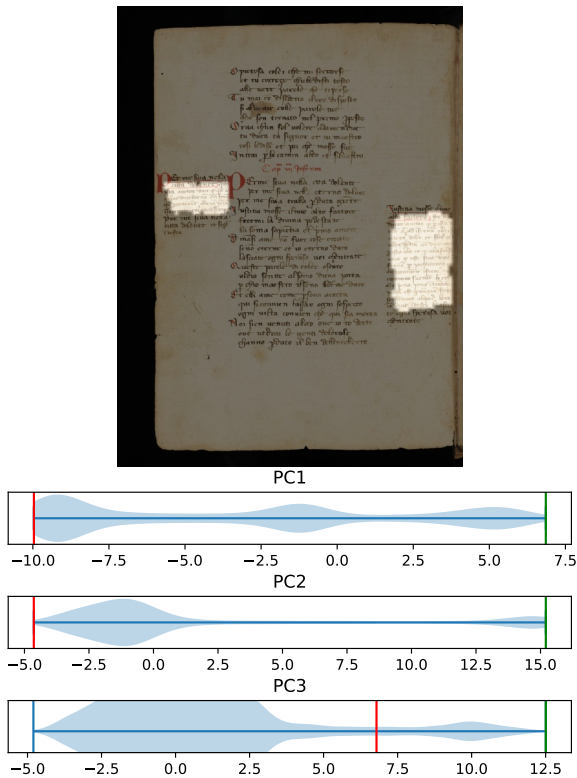


Fig. 6 The comment class of DIVA-HisDB [26] is determined using the third principal component (PC3) to select features, with the filter accepting values between the red and green lines.

the feature encoder includes max-pooling layers, the output resolution is downsampled, leading to imprecise results due to tiling. To address this, we employ a decoding path to map the features into target classes of a given dataset as outlined in Figure 3. This process includes upsampling and center cropping to align with max-pooling and convolutional layers of FE. The linear transformation of individual pixel features is ensured by omitting the activation function in the mapper, and by consistently using a 1x1 context for the mapper.

5 Experiments

As previously mentioned, our study targets semantic segmentation in historical handwritten documents. We emphasize feature analysis and probing rather than training the model for optimal performance. While we also present fully supervised *fine-tuning* and training *from scratch*

results, a direct comparison on individual datasets is possible but less meaningful due to expected impaired performance.

To the best of our knowledge, similar research in this field is limited. Consequently, we compared our approach to the pre-trained document image transformer² (DiT) [16]. The full image is originally down-sampled to 224x224, resulting in an output resolution of 14x14 tiles, with each tile being 16x16 pixels. This resolution is too coarse for tasks such as baseline detection. Therefore, we also experimented with using the full resolution and a page height limit of 1024 pixels, employing a sliding window approach to achieve higher resolution outputs. This approach aligns with the pre-training process, which included random resized cropping.

The following sections cover the training and evaluation setup, present both quantitative and qualitative results, and include a discussion that highlights strengths while critically assessing weaknesses.

5.1 Pre-training setup

Unless otherwise specified, the *default* configuration corresponds to Figure 3 and includes:

- FE consisting of 4 FE blocks and 32 convolutional filters in the first layer.
- Page images downsampled to a height of 1024 pixels.
- Color crops of the image, a batch size of 512, and the AdamW optimizer were used.
- Combined augmentation used:
 - Random shifting of the crops with a horizontal and vertical limit of 30 and 12 pixels, respectively.
 - Color jitter on the first crop.
 - The second crop includes OTSU binarization (probability 0.1), perspective warping (probability 0.8), horizontal line masking (probability 0.5), and vertical line masking (probability 0.5).

We experimented with various setups for comparison and further analysis. In the Barlow Twins pre-training phase (see Section 4.1), we determined the projector size of 8192 neurons and batch

²<https://huggingface.co/microsoft/dit-base>

Table 2 The Jaccard index results for probing after different FE pre-training setups. Compared to the *default* setup: *16b768* indicates a higher batch size of 768, with 16 convolutional filters in the first layer. *5block* refers to the use of 5 FE blocks. *Gray* denotes the use of grayscale input. *LARSscheduler* specifies that the LARS optimizer with a learning rate scheduler was used, instead of the default AdamW optimizer.

| setup | resolution | cBAD-complex | | DIVA-HisDB-CB55 | | ChronSeg | |
|---------------------|------------|--------------|-------------|-----------------|-------------|-------------|--|
| | | baseline | comments | decorations | image | text | |
| default | 1024 | 0.28 | 0.74 | 0.15 | 0.47 | 0.80 | |
| 16b768 | 1024 | 0.25 | 0.69 | 0.14 | 0.42 | 0.77 | |
| 5block | 1024 | 0.28 | 0.74 | 0.11 | 0.63 | 0.79 | |
| 5blockGray | 1024 | 0.29 | 0.78 | 0.01 | 0.52 | 0.79 | |
| 5blockLARSscheduler | 1024 | 0.23 | 0.63 | 0.04 | 0.41 | 0.78 | |
| Gray | 1024 | 0.30 | 0.80 | 0.04 | 0.49 | 0.82 | |
| LARSscheduler | 1024 | 0.24 | 0.65 | 0.08 | 0.44 | 0.76 | |
| default | full | 0.27 | 0.56 | 0.69 | 0.44 | 0.72 | |
| 16b768 | full | 0.25 | 0.51 | 0.57 | 0.41 | 0.72 | |
| 5block | full | 0.29 | 0.62 | 0.57 | 0.43 | 0.77 | |
| 5blockGray | full | 0.29 | 0.58 | 0.14 | 0.45 | 0.81 | |
| Gray | full | 0.28 | 0.51 | 0.15 | 0.40 | 0.75 | |
| LARSscheduler | full | 0.22 | 0.50 | 0.68 | 0.43 | 0.62 | |

Table 3 Mean Jaccard index results for probing and different augmentations applied to the 1024 page height limit variant.

| augmentation | cBAD | | DIVA-HisDB | | | ChronSeg |
|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | complex | simple | CB55 | CS18 | CS863 | |
| combined | 0.54 | 0.58 | 0.65 | 0.65 | 0.62 | 0.73 |
| binarization | 0.51 | 0.52 | 0.57 | 0.54 | 0.51 | 0.62 |
| hline masking | 0.47 | 0.45 | 0.48 | 0.53 | 0.44 | 0.58 |
| shift | 0.46 | 0.43 | 0.68 | 0.61 | 0.55 | 0.62 |
| perspective warp | 0.46 | 0.45 | 0.62 | 0.65 | 0.56 | 0.64 |
| vline masking | 0.46 | 0.44 | 0.50 | 0.58 | 0.45 | 0.59 |
| color jitter | 0.50 | 0.48 | 0.52 | 0.55 | 0.47 | 0.59 |
| none | 0.46 | 0.44 | 0.48 | 0.53 | 0.45 | 0.56 |
| FE random initialization | 0.49 | 0.48 | 0.52 | 0.55 | 0.46 | 0.61 |

Table 4 Mean Jaccard index results for comparison of probing with fully supervised training.

| training | encoder | resolution | cBAD | | DIVA-HisDB | | | ChronSeg |
|--------------|---------|------------|---------|--------|------------|------|-------|----------|
| | | | complex | simple | CB55 | CS18 | CS863 | |
| from scratch | FE | 1024 | 0.56 | 0.60 | 0.73 | 0.78 | 0.61 | 0.70 |
| from scratch | FE | full | 0.57 | 0.59 | 0.85 | 0.82 | 0.70 | 0.66 |
| fine-tune | DiT | 1024 | 0.59 | 0.64 | 0.73 | 0.74 | 0.59 | 0.84 |
| fine-tune | DiT | full | 0.58 | 0.64 | 0.69 | 0.75 | 0.60 | 0.81 |
| fine-tune | FE | 1024 | 0.57 | 0.60 | 0.76 | 0.78 | 0.69 | 0.67 |
| fine-tune | FE | full | 0.58 | 0.59 | 0.84 | 0.81 | 0.70 | 0.64 |
| probe | DiT | 1024 | 0.57 | 0.61 | 0.66 | 0.65 | 0.56 | 0.70 |
| probe | DiT | full | 0.56 | 0.55 | 0.55 | 0.63 | 0.53 | 0.64 |
| probe | FE | 1024 | 0.54 | 0.58 | 0.65 | 0.65 | 0.62 | 0.73 |
| probe | FE | full | 0.55 | 0.55 | 0.69 | 0.71 | 0.62 | 0.66 |

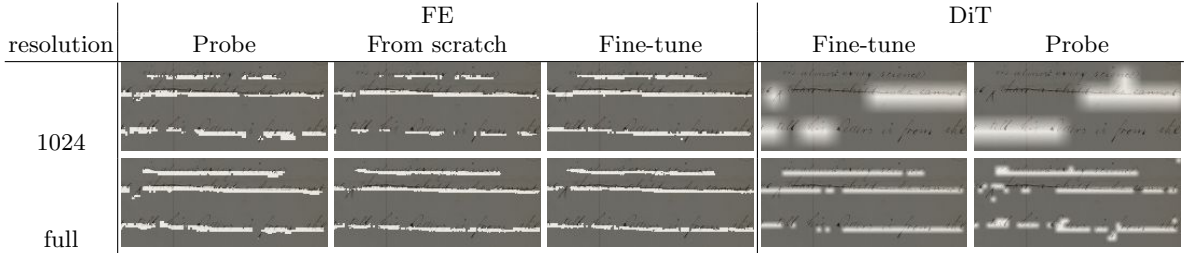


Fig. 7 Details on baseline segmentation for cBAD-simple, including comparisons of models and resolutions used.

sizes of 512 and 768 based on the analysis in [32]. In addition to using the LARS optimizer with a learning rate scheduler, as mentioned in the paper, we also tested the AdamW optimizer [18] with a learning rate of 10^{-4} .

For the FE, we explored several variants that processed both colored and grayscale images, utilized different number of convolutional filters and FE blocks, and applied different resolutions and augmentations.

5.2 Evaluation setup

Several evaluation schemes and metrics are used across different datasets. However, some metrics are not ideal because they require post-processing, potentially affecting feature analysis. Hence, we focus on pixel-wise metrics like Jaccard index, which handles class imbalances and is widely used evaluation metric in semantic segmentation. Additional metrics, including accuracy, F1 score, precision, and recall, can be found in the Git repository.

$$Jaccard\ index = \frac{TP}{TP + FP + FN} \quad (1)$$

The evaluation is conducted on individual images. Metrics are calculated for each class and then averaged. Finally, the results are averaged over all samples in the evaluation split.

5.3 Results and discussion

Given the number of experiments, we selected the most important or particularly interesting results, which are presented in Tables 2, 3, and 4.

5.3.1 Pre-training setup

According to Table 2, the pre-training setup of FE does influence the results, but the impact is

not particularly significant. While different pre-training configurations, such as the use of a 4-block versus a 5-block setup, show some variation in performance, these differences are not substantial across the board. Specifically, the 5-block configuration may offer an advantage for handling inputs in full resolution, providing a bigger context. However, in general, the distinction between the two setups is minimal, indicating that the view context provided by feature extraction is generally sufficient across various setups without causing significant performance differences. Using grayscale images is competitive overall and may be better for text-relevant classes, but certain classes, such as decorations in the DIVA-HisDB dataset, appear to benefit from color. The AdamW optimizer yields better results compared to the default LARS optimizer with scheduler used in Barlow Twins.

5.3.2 Input resolution

The input image crop size showed no significant influence on performance. However, input image resolution and view context do have an impact, especially for classes with large regions (such as text areas and images) or intricate details (like decorations). Pages with a 1024-pixel height limit offer more context, benefiting comments, text regions, and images. This resolution may also help standardize text sizes, as discussed in Section 3.1.

On the other hand, full resolution excels at capturing fine details, making it more effective for decorations. As shown in Figure 7, higher resolution proves advantageous for classes requiring finer detail, such as decorations or the baseline, where DiT struggles due to tiling effects, even at 1024 resolution. Consequently, the 224 variant of DiT falls short of delivering satisfactory results.

DiT has greater context and shows strong results when fine-tuned, particularly on datasets

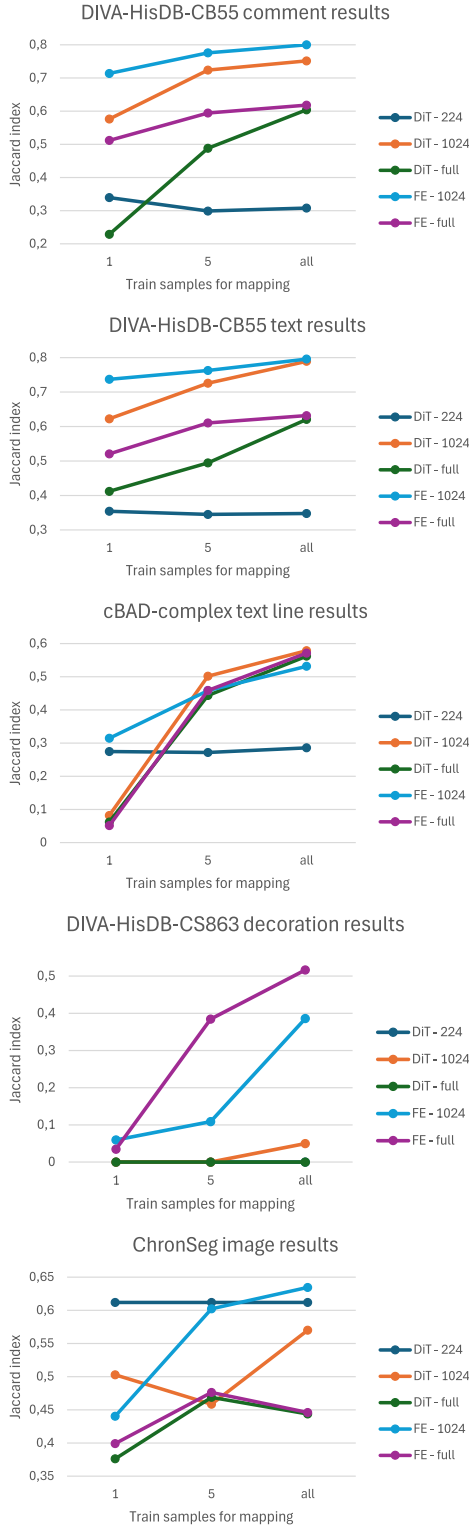


Fig. 8 Results of the probing approach using different models and varying numbers of training samples across various classes.

like ChronSeg, which contain larger text and image areas, as highlighted in Table 4. However, its tiling approach is less effective for capturing finer details.

5.3.3 Augmentation

According to Table 3, combined augmentations consistently outperform random initialization, yielding better results across all classes. Specifically, augmentations like binarization, shift, color jitter, and perspective warp generally provide better outcomes compared to random initialization.

On the other hand, methods such as none, horizontal line masking, and vertical line masking are less effective as standalone techniques because they do not sufficiently challenge the model; they allow it to match pixel values directly in most areas. Although binarization and color jitter do not shift the pixel positions, they still involve unpredictable changes in thresholds and colors.

Qualitatively, shift and perspective warp augmentations are the most effective for distinguishing text, background, or images through PC1 thresholding, highlighting their effectiveness in this context. Therefore, augmentations that shift pixel positions and introduce complex distortions, such as perspective warp, appear to be the most beneficial. However, combining augmentations does not degrade the results, indicating that even simpler augmentations can contribute positively when used in conjunction with others.

5.3.4 Few shot

In the context of probing, DiT and FE are fairly comparable, with FE showing a slight edge and providing more details. As expected, increasing the number of training samples improves the mapping of features across both models, as illustrated in Figure 8. Notably, FE performs better with minimal training samples, offering more useful features even with just one sample for mapping.

5.3.5 Fully supervised comparison

A model trained from scratch in a fully supervised manner is generally expected to outperform probing a pre-trained model in a self-supervised setting. While this holds true in most cases, as shown in Table 4, the difference is not substantial, and for ChronSeg, probing performs better.

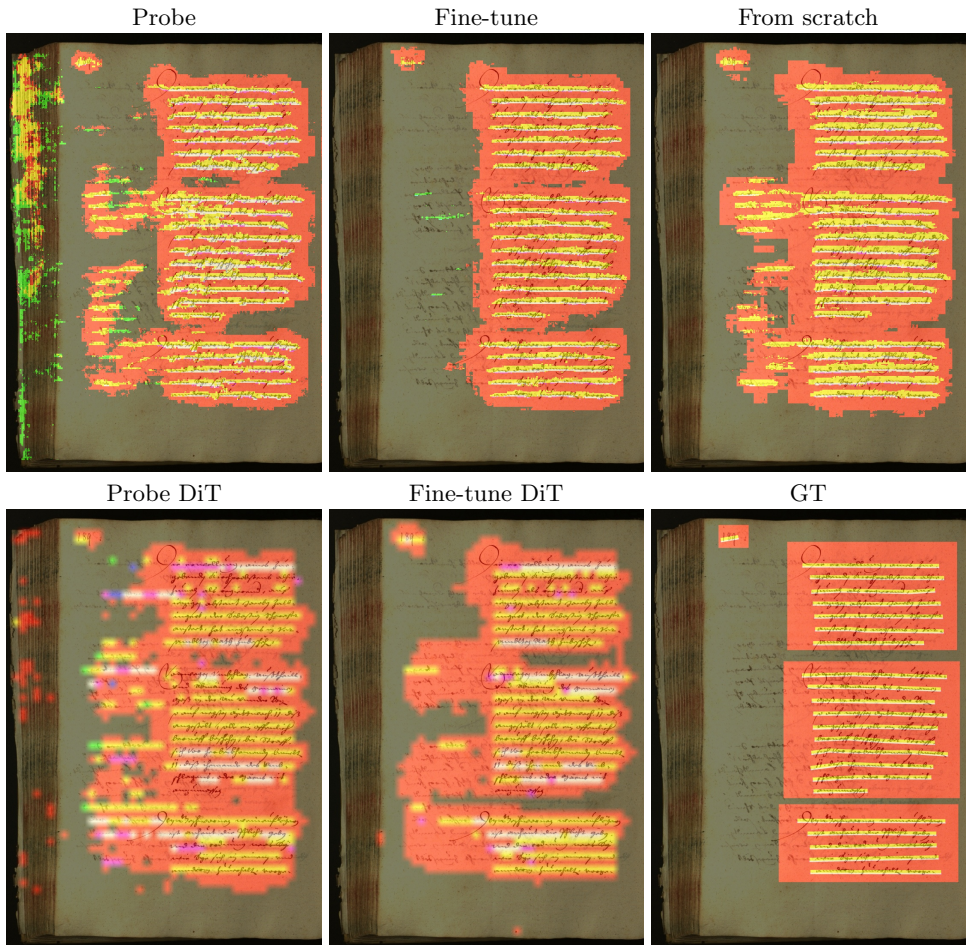


Fig. 9 The input image is overlaid with a mask where the *text region*, *text line*, and *baseline* are encoded in the R, G, and B channels, respectively. Noise in the prediction is mitigated only after fine-tuning the FE model.

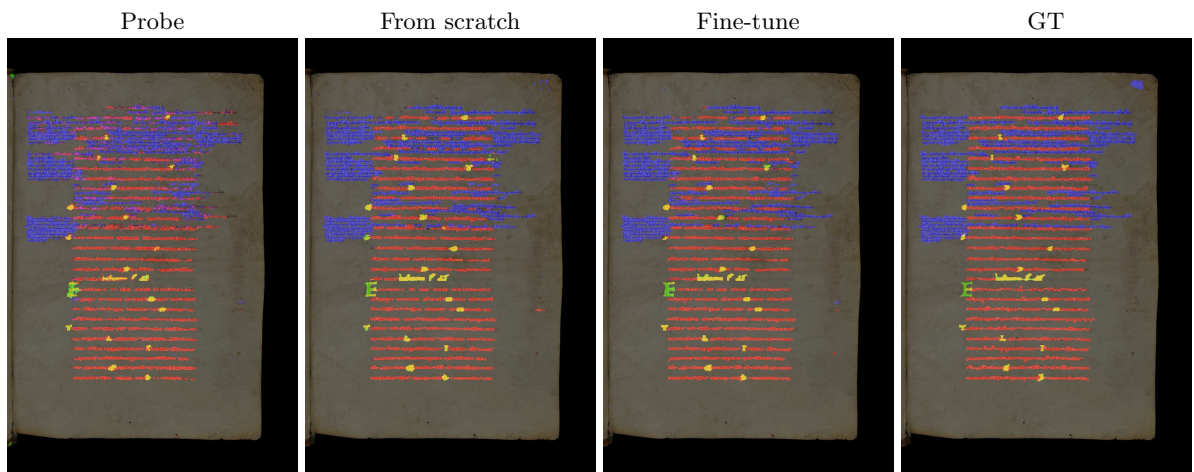


Fig. 10 The input image is overlaid with mask, where *text*, *decorations*, and *comments* are encoded in the R, G, and B channels, respectively. When probing, text in the upper part is being misclassified as comments.

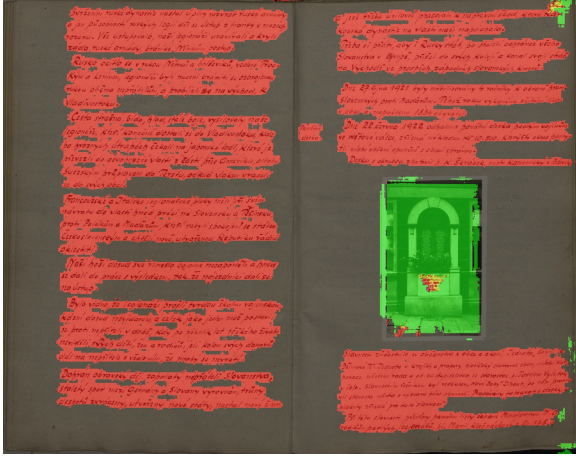


Fig. 11 The input image is overlaid with the FE probing result as a mask, where *text* and *image* are encoded in the R and G channels, respectively.

This may be due to the dataset’s limited training samples, where a simple linear transformation of pre-trained features proves more effective.

However, the numerical results do not fully capture the benefits of fine-tuning a pre-trained model, which helps address issues like noise or faint text. This is visually demonstrated in the Figures 7, 9, and 10.

5.3.6 Error analysis

Probing often yields impressive results, but several challenges remain. For instance, in Figure 11, noise at the edges is incorrectly classified as part of the image class, and text within the image region may not be accurately processed.

Noise is a common issue with probing, as demonstrated in Figure 9. This issue arises because visually similar parts, though distinct, may share similar features. Another challenge involves view context, as shown in Figure 10. Here, a large amount of comments between text lines leads to misclassification of text as comments, likely due to the prevalence of comments in that area, which distorts the feature representation. In both cases, these distinctions may not be accurately captured during training when relying solely on a simple linear transformation. This issue affects both the FE and DiT models.

As shown in Figure 7, probing at full resolution indicates that the features align more closely with the lower bound of the text rather than the

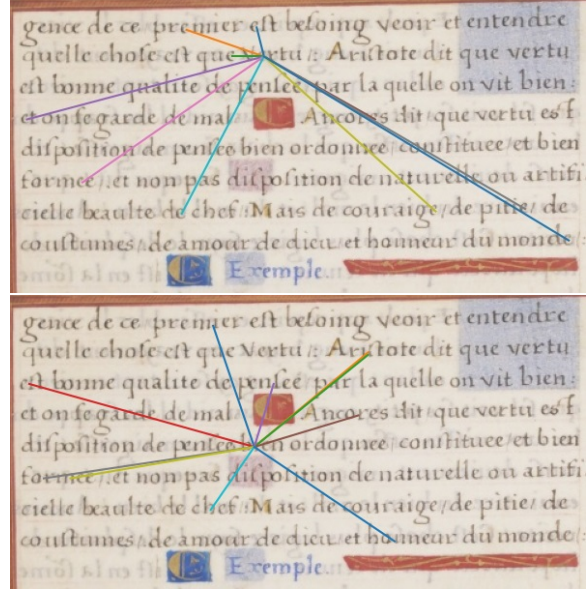


Fig. 12 The ten closest features for the query characters *e* and *i*: The character *e* is graphically distinct, resulting in a perfect match. In contrast, the character *i* appears as a vertical line, a common feature found in many other characters. Consequently, the matches for *i* are not accurate in the linguistic context, but they are graphically correct.

baseline. This suggests that not all dataset classes can be effectively mapped using a simple linear transformation, potentially leading to suboptimal results.

However, fine-tuning a pre-trained model mitigates these issues, yielding more robust results than training from scratch. This improvement applies to both FE and DiT models. While FE delivers more detailed and less noisy outcomes for semantic segmentation, DiT, being a more complex model with greater capacity, is probably better suited for fine-tuning on complex tasks when sufficient training data is available. Even though the FE could potentially be used as a feature encoder and may improve fine-tuning results when integrated with more complex models.

Furthermore, Figure 12 suggests that the pre-trained features contain useful information about the shapes of characters, even though they lack language-specific information. This is expected since language data was not directly included, and the visual masking model alone is likely insufficient given the diverse languages and writing styles used for training.

6 Conclusion

In this paper, we introduced a self-supervised learning approach for historical document segmentation, leveraging the Barlow Twins framework for feature extraction. Our results demonstrate that self-supervised features are highly effective for segmenting handwritten documents, even without external annotations. Using PCA, we showed that key document components (e.g., text, background, images) can be identified without manual labeling, making SSL a promising tool for historical document analysis.

We evaluated the effectiveness of probing techniques, which apply a simple linear transformation to pre-trained features. Our experiments indicate that probing achieves comparable performance to fully supervised methods and excels in few-shot learning scenarios where labeled data is scarce. Additionally, we analyzed the impact of input resolution, finding that a 1024-pixel height provides better context for text regions, while full resolution is preferable for capturing fine details like decorations.

Our comparison with fully supervised models (e.g., fine-tuned DiT) suggests that while supervised training is generally stronger, SSL-based approaches offer robust, high-quality features that are useful for document segmentation tasks. Furthermore, we demonstrated that fine-tuning a pre-trained model mitigates issues like noise and faint text, improving segmentation accuracy in degraded documents.

Moving forward, this work opens several avenues for research. Future studies could explore hybrid models that combine self-supervised learning with few-shot or semi-supervised approaches, enabling efficient segmentation with minimal labeled data. Additionally, further analysis is needed to extend these methods to more complex tasks such as handwritten text recognition, where linguistic information plays a crucial role.

Our findings highlight the potential of self-supervised learning to revolutionize historical document processing, paving the way for more efficient, scalable, and annotation-free segmentation methods.

Acknowledgements. The work of Josef Baloun has been supported by the Grant No. SGS-2025-022 – New

Data Processing Methods in Current Areas of Computer Science. The work of the other authors has been supported by the project R&D of Technologies for Advanced Digitalization in the Pilsen Metropolitan Area (DigiTech) No. CZ.02.01.01/00/23_021/0008436.

References

- [1] Ares Oliveira S, Seguin B, Kaplan F (2018) dhSegment: A generic deep-learning approach for document segmentation. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp 7–12, <https://doi.org/10.1109/ICFHR-2018.2018.00011>
- [2] Baloun J, Král P, Lenc L (2021) Chron-Seg: Novel dataset for segmentation of handwritten historical chronicles. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,, INSTICC. SciTePress, pp 314–322, <https://doi.org/10.5220/0010317203140322>
- [3] Bao H, Dong L, Piao S, et al (2022) BEiT: Bert pre-training of image transformers. [2106.08254](https://arxiv.org/abs/2106.08254)
- [4] Chen T, Kornblith S, Norouzi M, et al (2020) A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. JMLR.org, ICML'20
- [5] Christlein V, Nicolaou A, Seuret M, et al (2019) Icdar 2019 competition on image retrieval for historical handwritten documents. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp 1505–1509, <https://doi.org/10.1109/ICDAR.2019.00242>
- [6] Cosma A, Ghidoveanu M, Panaitescu-Liess M, et al (2020) Self-supervised representation learning on document images. In: Bai X, Karatzas D, Lopresti D (eds) Document Analysis Systems. Springer International Publishing, Cham, pp 103–117, https://doi.org/10.1007/978-3-030-57058-3_8

- [7] Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186, <https://doi.org/10.18653/v1/N19-1423>, URL <https://aclanthology.org/N19-1423>
- [8] Diem M, Kleber F, Fiel S, et al (2017) cBAD: Icdar2017 competition on baseline detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp 1355–1360, <https://doi.org/10.1109/ICDAR.2017.222>
- [9] Fiel S, Kleber F, Diem M, et al (2017) Icdar2017 competition on historical document writer identification (historical-wi). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp 1377–1382, <https://doi.org/10.1109/ICDAR.2017.225>
- [10] Gatos B, Stamatopoulos N, Louloudis G, et al (2015) GRPOLY-DB: An old greek polytonic document image database. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp 646–650, <https://doi.org/10.1109/ICDAR.2015.7333841>
- [11] Goodfellow I, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, et al (eds) Advances in Neural Information Processing Systems, vol 27. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
- [12] Grill JB, Strub F, Alché F, et al (2020) Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems 33:21271–21284
- [13] Gupta A, Vedaldi A, Zisserman A (2020) Learning to read by spelling: Towards unsupervised text recognition. In: Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing. Association for Computing Machinery, New York, NY, USA, ICVGIP '18, <https://doi.org/10.1145/3293353.3293386>, URL <https://doi.org/10.1145/3293353.3293386>
- [14] Kim G, Hong T, Yim M, et al (2022) Ocr-free document understanding transformer. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII. Springer-Verlag, Berlin, Heidelberg, p 498–517, https://doi.org/10.1007/978-3-031-19815-1_29, URL https://doi.org/10.1007/978-3-031-19815-1_29
- [15] Kleber F, Fiel S, Diem M, et al (2013) CVL-DataBase: An off-line database for writer retrieval, writer identification and word spotting. In: 2013 12th International Conference on Document Analysis and Recognition, pp 560–564, <https://doi.org/10.1109/ICDAR.2013.117>
- [16] Li J, Xu Y, Lv T, et al (2022) DiT: Self-supervised pre-training for document image transformer. In: Proceedings of the 30th ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, MM '22, p 3530–3539, <https://doi.org/10.1145/3503161.3547911>, URL <https://doi.org/10.1145/3503161.3547911>
- [17] Li P, Gu J, Kuen J, et al (2021) Self-Doc: Self-supervised document representation learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 5648–5656, <https://doi.org/10.1109/CVPR46437.2021.00560>
- [18] Loshchilov I, Hutter F (2017) Fixing weight decay regularization in adam. CoRR abs/1711.05101. URL <http://arxiv.org/abs/1711.05101>, 1711.05101
- [19] Maity S, Biswas S, Manna S, et al (2023) SelfDocSeg: A Self-supervised

- Vision-Based Approach Towards Document Segmentation, Springer Nature Switzerland, p 342–360. https://doi.org/10.1007/978-3-031-41676-7_20, URL http://dx.doi.org/10.1007/978-3-031-41676-7_20
- [20] Marti UV, Bunke H (2002) The IAM-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition* 5:39–46
- [21] Michael J, Labahn R, Grüning T, et al (2019) Evaluating sequence-to-sequence models for handwritten text recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, pp 1286–1293
- [22] Oquab M, Darcet T, Moutakanni T, et al (2024) DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research* URL <https://openreview.net/forum?id=a68SUt6zFt>
- [23] Pletschacher S, Antonacopoulos A (2010) The PAGE (page analysis and ground-truth elements) format framework. In: 2010 20th International Conference on Pattern Recognition, pp 257–260, <https://doi.org/10.1109/ICPR.2010.72>
- [24] Pramanik S, Mujumdar S, Patel H (2020) Towards a multi-modal, multi-task learning based pre-training framework for document representation learning. *arXiv preprint arXiv:200914457*
- [25] Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, et al (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, Cham, pp 234–241, https://doi.org/10.1007/978-3-319-24574-4_28
- [26] Simistira F, Seuret M, Eichenberger N, et al (2016) DIVA-HisDB: A precisely annotated large dataset of challenging medieval manuscripts. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp 471–476, <https://doi.org/10.1109/ICFHR.2016.0093>
- [27] Sánchez JA, Romero V, Toselli AH, et al (2014) ICFHR2014 competition on handwritten text recognition on transcriptorium datasets (htrts). In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp 785–790, <https://doi.org/10.1109/ICFHR.2014.137>
- [28] Sánchez JA, Romero V, Toselli AH, et al (2016) ICFHR2016 competition on handwritten text recognition on the READ dataset. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp 630–635, <https://doi.org/10.1109/ICFHR.2016.0120>
- [29] Wallace E, Wang Y, Li S, et al (2019) Do NLP models know numbers? probing numeracy in embeddings. In: Inui K, Jiang J, Ng V, et al (eds) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pp 5307–5315, <https://doi.org/10.18653/v1/D19-1534>, URL <https://aclanthology.org/D19-1534>
- [30] Xu Y, Li M, Cui L, et al (2020) LayoutLM: Pre-training of text and layout for document image understanding. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, KDD '20, p 1192–1200, <https://doi.org/10.1145/3394486.3403172>, URL <https://doi.org/10.1145/3394486.3403172>
- [31] Yu Y, Li Y, Zhang C, et al (2023) Structxtv2: Masked visual-textual prediction for document image pre-training. [2303.00289](https://arxiv.org/abs/2303.00289)
- [32] Zbontar J, Jing L, Misra I, et al (2021) Barlow twins: Self-supervised learning via redundancy reduction. In: *International conference on machine learning*, PMLR, pp 12310–12320