


Self-Supervised Vision Transformers for Writer Retrieval

Tim Raven  [0009-0002-1528-2744], Arthur Matei^[0009-0009-6028-7502], and
Gernot A. Fink^[0000-0002-7446-7813]

TU Dortmund University
{tim.raven, arthur.matei, gernot.fink}@tu-dortmund.de

Abstract. While methods based on Vision Transformers (ViT) have achieved state-of-the-art performance in many domains, they have not yet been applied successfully in the domain of writer retrieval. The field is dominated by methods using handcrafted features or features extracted from Convolutional Neural Networks. In this work, we bridge this gap and present a novel method that extracts features from a ViT and aggregates them using VLAD encoding. The model is trained in a self-supervised fashion without any need for labels. We show that extracting local foreground features is superior to using the ViT’s class token in the context of writer retrieval. We evaluate our method on two historical document collections. We set a new state-of-the-art performance on the Historical-WI dataset (83.1% mAP), and the HisIR19 dataset (95.0% mAP). Additionally, we demonstrate that our ViT feature extractor can be directly applied to modern datasets such as the CVL database (98.6% mAP) without any fine-tuning.

Keywords: Writer Retrieval · Writer Identification · Historical Documents · Self-Supervised Learning · Vision Transformer

1 Introduction

Writer retrieval involves systematically extracting documents, written by the same author as a queried document, from a large corpus of handwritten texts with unidentified authors. Closely related to this, writer identification aims to identify the writer of a queried document by consulting a corpus of labeled documents. In historical research, these processes are crucial for categorizing and examining manuscripts based on authorship, particularly when manuscripts lack signatures [11]. In the forensic sciences, accurately identifying authors in handwritten notes or documents is essential in criminal investigations, such as verifying ransom notes, anonymous threatening letters, or fraudulent documents.

Recently, methods employing Vision Transformers (ViT) [12] have achieved state-of-the-art performance in various computer vision tasks, including handwritten text recognition [23]. However, ViTs are prone to overfitting even on large datasets, making them challenging to train [12]. Self-supervised learning can help to address this challenge, as no annotations are required for training and, even when annotations are available, self-supervised training followed

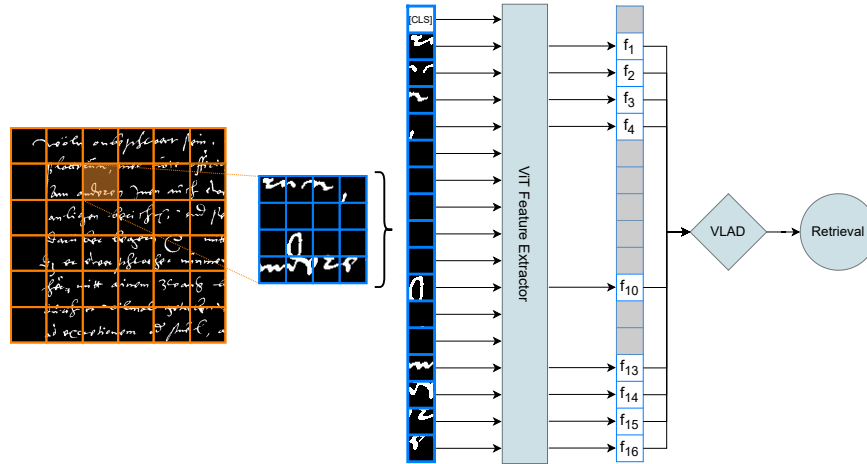


Fig. 1: Illustration of our proposed method. Document images are cut into windows in a regular grid. The windows are again cut into patches and form the input sequence. To extract local features, a self-supervised Vision Transformer (ViT) is used. We extract only foreground patch tokens from the ViT output sequence, i.e., only the patch tokens of input patches with sufficient handwriting. All foreground tokens of the document are aggregated using a VLAD encoding. These encodings are used for retrieval and reranking.

by supervised fine-tuning can outperform fully supervised training [16]. Popular self-supervised learning paradigms include contrastive learning [3, 4], self-distillation [2], masked image modeling [1, 16, 36] and similarity maximization [5]; or a combination of these [20, 37].

Current writer retrieval and identification methods still rely on local features extracted from Convolutional Neural Network (CNN) activations [8] or hand-crafted methods [22]. While there have been efforts to employ ViTs for writer retrieval [28], the results were not competitive. Motivated by the recent success of pre-training a ViT for handwritten text recognition using masked image modeling [35] on only IAM [25], we revisit the use of self-supervised ViTs in writer retrieval.

In this work, we introduce a fully self-supervised approach that employs a ViT as a feature extractor for writer retrieval and identification (see Fig. 1). Our method requires no labels and outperforms existing methods in historical document benchmarks. We train the ViT using a combination of Masked Image Modeling [16, 36] and self-distillation [2], utilizing a student-teacher network. The teacher’s self-attention is used to guide the masking process which in turn the student has to reconstruct in feature space. Contrary to previous attempts of utilizing a ViT as a feature extractor for writer retrieval [28], we do not use the class token of the ViT as feature representation. Instead, we extract fore-

ground patch tokens from the ViT’s output sequence, i.e., tokens corresponding to input patches with a sufficient amount of foreground pixels. Our evaluation demonstrates that encoding these features with a Vector of Locally Aggregated Descriptors (VLAD) [18] to obtain a global page descriptor enhances performance notably compared to using the class token with either sum-pooling or VLAD. However, even when using sum-pooling as encoding, our method surpasses previous methods, underscoring the quality of the extracted features.

We make the following contributions:

- We successfully apply a Vision Transformer to the task of writer retrieval through self-supervised learning.
- We demonstrate that features extracted from our ViT outperform both hand-crafted and CNN-based features.
- We show that encoding foreground tokens using VLAD is superior to encoding class tokens.
- We show that our method learns robust features directly applicable to historical and modern handwriting without the need for model fine-tuning.

The remainder of this paper is organized as follows. First, we cover related work in the field of writer retrieval in Section 2. Next, we outline our proposed method in more detail in Section 3. Then, we introduce the evaluation protocol for our experiments in Section 4. The conducted experiments are detailed in Section 5. Finally, we summarize our findings and give an outlook on future work in Section 6.

2 Related Work

Writer retrieval methods commonly follow the same pipeline of local feature extraction, page-level aggregation, and distance-based retrieval with optional reranking.

2.1 Feature Extraction

Local features are generally divided into handcrafted features and deep learning features. Examples of handcrafted features are SURF [17], Zernike moments [6] or a combination of SIFT and Pathlet features [22]. In contrast, deep learning features were first introduced by Fiel and Sablatnig [15], who used a supervised CNN as a feature extractor. Christlein *et al.* [8] propose an unsupervised method for CNN training for historical documents, in which pseudo labels are derived from clustering SIFT descriptors. In [28] an ImageNet pre-trained ViT is fine-tuned for document images in a self-supervised student-teacher approach operating on differently augmented views, using the ViT’s class tokens as feature representation. However, the method underperforms considerably compared to CNN-based or handcrafted methods.

2.2 Aggregation

Methods to aggregate local features into global page descriptors are generally categorized into codebook-free and codebook-based methods. Codebook-free methods include sum-pooling and generalized max-pooling [26]. Codebook-based methods used in writer retrieval/identification include Fisher Vectors [14], GMM supervectors [7] and VLAD-based methods [8, 22], where several VLAD encodings are computed and jointly decorrelated using PCA with whitening. The VLAD encoding is sometimes incorporated into the network architecture using NetVLAD [30, 31]. As an additional refinement step, Christlein *et al.* [8] train exemplar SVMs for each query, using the query as the only positive example and all training pages as negatives, exploiting the writer-disjointness of training and test sets. We find that a single VLAD encoding is sufficient with our features.

2.3 Retrieval and Reranking

Retrieval is commonly done using a distance measure, like cosine distance. Other metrics such as the l_1 distance, l_2 distance, or the Canberra distance have also been explored [29]. An optional step that has shown to be beneficial in retrieval tasks is reranking, which refines the retrieval list by exploiting the information within it. The authors of [31] use a k reciprocal nearest neighbor Query Expansion by averaging each descriptor with its k reciprocal nearest neighbors. The E-SVM approach of Christlein *et al.* is extended into a Pair/Triple SVM approach in [19], where similar documents from the test set are included as additional *positive* examples. In [30] (Similarity) Graph Reranking is explored. Here, a parameter-free graph network is constructed and used to obtain updated global descriptors using message propagation.

3 Method

Our method follows the common framework of local feature extraction, aggregation, distance-based retrieval and reranking used in previous work [8, 9, 22, 30]. Our method operates on binary images, obtained in a preprocessing step, if necessary. An illustration of the method is given in Figure 1. As the computational complexity of a ViT grows quadratically with the sequence length, our ViT uses a fixed input image size of 224×224 , and operates on windows extracted from the document. The ViT is trained in a self-supervised fashion (see Section 3.1) and used to extract patch tokens corresponding to handwriting (see Section 3.2). The extracted features are aggregated into a global page descriptor using a VLAD encoding (see Section 3.3). Finally, the global page descriptors are compared using cosine distance and optionally reranked (see Section 3.4).

3.1 Self-supervised Training

ViTs feature a large number of trainable parameters and are prone to overfit [12], requiring extensive amounts of annotated data to train a ViT successfully

with traditional, supervised methods. Thus, utilizing self-supervised training for ViTs is a logical conclusion. We chose AttMask [20] for self-supervised training. The method is an adaptation of iBOT [37], which incorporates Masked Image Modeling (MIM) into the DINO [2] framework.

DINO [2] employs self-distillation, a special form of knowledge distillation, i.e., training a student network to reproduce the output of a teacher network. In self-distillation, the teacher network is defined as an exponential moving average of the student. Student and teacher receive differently augmented views of the input, forcing the student to learn an invariance to the applied augmentations. Additionally, the method samples global and local views. While all views are shown to the student, only the global views are shown to the teacher, thus training a local-to-global correspondence and disentangling objects in feature space.

iBOT [37] integrates the MIM objective into DINO’s framework by masking a randomized selection of input patches from the student’s view while still showing them to the teacher. Afterwards, the student has to predict the teacher’s output for the masked patches.

Finally, AttMask [20] introduces a novel masking strategy that increases the complexity of feature reconstruction compared to the original iBOT masking, aiming to generate a more robust feature space. The final self-attention map generated by the teacher is used to mask the most attended input patches from the student, forcing the student to develop a deeper understanding of the input by masking the most important regions.

3.2 Local Feature Extraction

We directly use the self-supervised Vision Transformer (ViT) g as a local feature extractor without any additional fine-tuning. A document image I is cut into N windows $\{w_1, \dots, w_N\}$ in a regular grid. The ViT further cuts each window w into a sequence of flattened patches $\{p_1^w, \dots, p_L^w\}$, where each patch p_i^w is then of length P^2 . A learnable class token [CLS] is prepended, forming the input for the ViT as $x = \{\text{[CLS]}, p_1^w, \dots, p_L^w\}$. Thus, the output of the ViT g is given as the token sequence

$$g(x) = \{f_{\text{[CLS]}}^w, f_1^w, \dots, f_L^w\}. \quad (1)$$

We find that for aggregating local features using VLAD, retaining the local information of the patch tokens is crucial. However, handwriting images are often sparse due to the horizontal and vertical spacing between words. As a result, many ViT patches may only contain background information, contributing little to the analysis of handwriting characteristics. To address this, we filter out patch tokens that lack sufficient foreground pixels. The set of foreground tokens $FG(I)$ is then given as:

$$FG(I) = \{f_i^w \mid \sum_{j=1}^{P^2} p_{i,j}^w \geq t_{\text{fg}} \text{ for } i \in \{1, \dots, L\} \text{ and } w \in I\}, \quad (2)$$

where $p_{i,j}^v$ is the j -th pixel in the flattened patch and t_{fg} a threshold on the number of contained foreground pixels.

3.3 Aggregation

To construct the VLAD codebook Θ , we cut all training documents into windows of size 224×224 with stride 224, i.e., non-overlapping, and gather all foreground tokens from the entire training set. These are jointly clustered using minibatch k -Means [34] with C centroids, which are used as the VLAD codebook $\Theta = \{\mu_1, \dots, \mu_C\}$. During inference, the test documents are cut into windows with an adjustable stride of S_{eval} . For each test document d the set of foreground tokens $FG(d)$ is gathered and encoded by assigning each token $f \in FG(d)$ to the closest centroid and aggregating the residuals between the centroids and their assigned features. For a centroid μ_k , this yields

$$v_k^d = \sum_{\{f | NN_{\Theta}(f) = \mu_k\}} (f - \mu_k), \quad (3)$$

where $NN_{\Theta}(f)$ is the nearest centroid to f in codebook Θ . The resulting VLAD encoding \hat{v}^d of a document d is the concatenation of all such residuals:

$$\hat{v}^d = \text{concat}(v_1^d, \dots, v_N^d). \quad (4)$$

We apply power normalization with power 0.5 followed by l_2 -normalization. Finally, principal component analysis (PCA) with whitening is used for decorrelation and dimensionality reduction to D dimensions, resulting in the global page descriptor v^d for document d . The parameters of the PCA are fitted on the training set.

3.4 Retrieval and Reranking

For retrieval, we use the cosine distance measure. A low distance indicates that documents are similar. The cosine distance d_{cos} between two global page descriptors v^a and v^b extracted from documents a and b is given as

$$d_{cos}(v^a, v^b) = 1 - \frac{v^a \cdot v^b}{\|v^a\| \cdot \|v^b\|}. \quad (5)$$

We evaluate different reranking strategies from previous methods [30,31] in conjunction with our method, i.e., k RNN, Graph reranking and SGR.

4 Evaluation Protocol

In this section, we first introduce the metrics for our evaluation in Section 4.1. Next, we describe the utilized datasets in Section 4.2. Lastly, we outline the hyperparameters of our baseline implementation in Section 4.3.

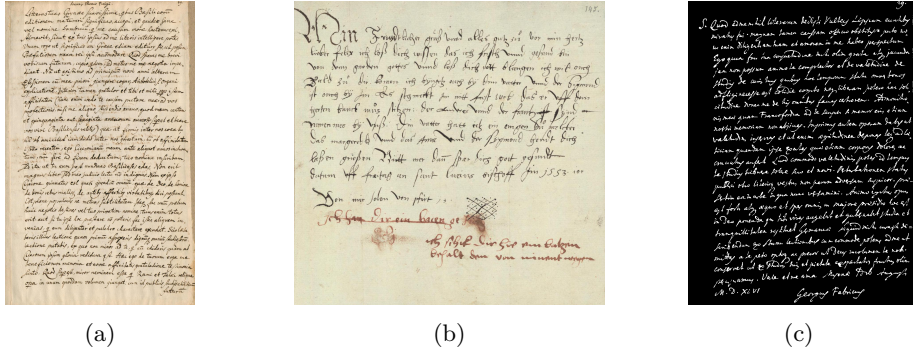


Fig. 2: Visualization of sample document images from the Historical-WI dataset: (a) and (b) show color images, (c) shows a binarized image provided in the dataset.



Fig. 3: Visualization of sample document images from the HisIR19 dataset.

4.1 Metrics

The evaluation is done in a leave-one-out fashion, i.e., every document in the test set is used once as a query image. The remaining documents are ranked by their distance to the query such that the documents with the lowest distance rank highest. The relevant documents, i.e., documents written by the same author, should ideally be at the top of this ranking. A common measure to describe the quality of a retrieval list is the mean average precision (mAP). To assess the writer identification performance, the Top1 accuracy is commonly considered, i.e., the percentage of query images for which the highest ranking result is a relevant document.

4.2 Datasets

We evaluate our method on two benchmark datasets of historical document images.

Historical-WI The Historical-WI dataset, a collection of historical document images, was released for the *ICDAR 2017 Competition on Historical Document*

Writer Identification [13]. This dataset is available in both binarized and color image formats. It includes a predefined train-test split: the training set comprises 1,182 document images authored by 394 writers, with each writer contributing three documents. The test set is more extensive, containing 3,600 document images from 720 writers, each contributing five documents. Spanning from the 13th to the 20th century, these documents feature texts in German, Latin, and French. Figure 2 shows three examples of contained document images.

HisIR19 The HisIR19 dataset was released for the *ICDAR 2019 Competition on Image Retrieval for Historical Handwritten Documents* [11]. Contrary to the Historical-WI dataset, there is no predefined training split. The authors of the challenge suggest using the Historical-WI test dataset for training. Additionally, a validation dataset is included containing 1200 images of 520 writers, with 300 writers contributing a single page, 100 writers contributing three pages, and 120 writers contributing five pages. The test set is considerably larger than the Historical-WI dataset and contains a total of 20000 documents authored by 10068 writers. 7500 writers contributed one page each, while the others contributed three to five documents. The dataset contains images from manuscript books from the European Middle Ages (9th to 15th century), letters from the 17th and 18th centuries, as well as charters and legal documents. Figure 3 shows two examples of contained document images.

4.3 Implementation Details

We use the following parameters as baseline implementation unless explicitly stated differently. Our model is a ViT-small/16 [12] with an input image size of 224. For HisIR19 the document images are only available in color format, thus we binarize them as a preprocessing step using Sauvola Binarization [33] with a window size of 51. For our evaluations on the HisIR19 dataset, we directly use the ViT trained on the Historical-WI training set and do not perform any additional fine-tuning. We use the HisIR19 validation set only to construct the VLAD codebook.

We generate training data by sampling windows of size 256 in a regular grid with stride 32 from the Historical-WI training set, resulting in roughly 1.75 million training windows. On these, we train the model for 20 epochs. We apply a cosine learning rate schedule with a linear warmup during the first two epochs and a peak learning rate of 0.005. The last layer is frozen during the first epoch. We use a MultiCrop augmentation [2] with two global crops (size 224, scale $\in [0.4, 1]$) and eight local crops (size 96, scale $\in [0.05, 0.4]$). Since we operate on binary images, all color-related augmentations are dropped. Instead, following Peer *et al.* [28], we apply Dilation and Erosion with random kernels to all crops independently with 50% probability.

During feature extraction, we extract windows with stride $S_{\text{eval}} = 224$ and apply a foreground threshold of $t_{\text{fg}} = 10$ pixels for extracting patch tokens. To save computation, we only use input windows with more than 2.5% foreground

Table 1: Evaluation of the performance when extracting class tokens ([CLS]) versus foreground tokens (FG) at different foreground thresholds t_{fg} as features. We evaluate sum-pooling (Sum) and VLAD for aggregation. Results are given for the Historical-WI and HisIR19 test datasets.

		Historical-WI				HisIR19			
		Sum		VLAD		Sum		VLAD	
Features	t_{fg}	mAP	Top1	mAP	Top1	mAP	Top1	mAP	Top1
[CLS]	-	78.5	90.0	64.3	80.3	92.0	96.9	75.8	87.8
FG	0	75.1	88.3	80.1	90.5	89.8	95.8	90.1	96.0
FG	1	77.1	90.0	81.4	91.1	91.5	96.7	92.9	97.2
FG	10	76.7	89.3	81.1	90.5	92.4	97.1	93.6	97.5
FG	20	76.3	89.1	81.0	90.6	92.5	97.1	93.7	97.6
FG	50	76.0	89.3	80.9	90.5	92.3	97.1	93.5	97.5

pixels during inference and discard the rest. For aggregation, we use $C = 100$ centroids in the VLAD codebook and reduce the final dimensionality of our VLAD encodings to $D = 384$ dimensions.

5 Experiments

We evaluate the different parts of our method separately. First, in Section 5.1, we investigate the performance of our foreground tokens in conjunction with different aggregation methods. Second, in Section 5.2, we compare different training paradigms to train our ViT feature extractor. Third, in Section 5.3, we evaluate different reranking algorithms. Fourth, in Section 5.4, we investigate the effect of the feature extraction and aggregation parameters. Finally, in Section 5.5, we compare our results with previous work.

5.1 Feature Extraction and Aggregation

In our proposed method, we extract all foreground patch tokens as features for a given window instead of using the class token. To evaluate this strategy, we compare the performance when using the class tokens as features versus using our foreground patch tokens at various threshold values t_{fg} . For aggregating all local features extracted from a document, we consider sum-pooling and VLAD.

The results given in Table 1 show that using class tokens works well with sum-pooling, whereas a significant drop in performance is observed with VLAD. In contrast, when using all patch tokens ($t_{fg} = 0$) VLAD outperforms sum-pooling. A likely explanation for this is the low number of class features extracted compared to patch tokens. Filtering empty ViT patches ($t_{fg} = 1$) improves the performance of both encodings compared to using all tokens. Again, VLAD

Table 2: Evaluation of different training paradigms for training the ViT feature extractor. We evaluated different masking strategies for iBOT and AttMask.

		Historical-WI				HisIR19			
Method	Features	Sum		VLAD		Sum		VLAD	
		mAP	Top1	mAP	Top1	mAP	Top1	mAP	Top1
Supervised (Writer)	[CLS]	52.7	72.1	27.9	40.4	82.2	90.4	53.5	69.3
Supervised (Writer)	FG	67.5	84.8	61.5	79.1	87.8	94.6	82.0	91.1
Supervised (Page)	[CLS]	58.1	75.9	43.0	60.1	85.5	92.7	61.4	75.8
Supervised (Page)	FG	66.4	83.5	66.8	83.4	87.8	94.7	87.0	94.2
DINO	FG	74.9	89.0	80.0	90.3	91.7	96.8	92.6	97.3
iBOT (rand)	FG	75.9	88.7	80.7	90.3	91.3	96.7	92.7	97.2
iBOT (block)	FG	75.5	88.9	81.0	90.4	91.2	96.5	93.2	97.3
AttMask (Hint)	FG	75.7	88.6	81.0	90.3	91.8	96.8	93.3	97.3
AttMask (High)	FG	76.7	89.3	81.1	90.5	92.4	97.1	93.6	97.5

yields better results than sum-pooling. Importantly, it also yields better results than sum-pooling of the class tokens on both datasets. While further increasing t_{fg} harms performance on the Historical-WI dataset, we observe a peak at $t_{fg} = 20$ on the HisIR19 dataset. This is likely caused by noise in the automated binarization process which is not present in the curated binarized version of Historical-WI.

5.2 Vision Transformer Training

For feature extraction, we train a ViT in a self-supervised approach using AttMask. In this section, we evaluate other self-supervised training approaches, as well as supervised approaches.

Self-Supervised Training In this section, we evaluate other related self-supervised training approaches. We compare AttMask [20] to its predecessors, DINO [2] and iBOT [37], and evaluate different masking strategies. Both iBOT and AttMask allow to configure the masking process. Choosing *rand*, the ViT’s input patches are masked randomly, whereas when choosing *block* patches for masking are selected, such that they form consecutive block shapes in the original image. In the case of AttMask, we evaluate the masking strategies *high* and *hint*. The masking strategy *high* masks the most highly attended patches in the input image, while *hint* reveals some highly attended patches again. We use the *high* masking strategy as default option in the remaining experiments. Table 2 shows that DINO, iBOT and AttMask slightly improve upon each other. For all methods, the best results are obtained from encoding our foreground tokens using VLAD, with AttMask achieving slightly higher mAP and Top1 than the others.

Table 3: Evaluation of different reranking methods in combination with class tokens ([CLS]) and foreground tokens (FG), as well as sum-pooling (Sum) and VLAD to compute a global page descriptor.

		Historical-WI				HisIR19			
		Sum		VLAD		Sum		VLAD	
Method	Features	mAP	Top1	mAP	Top1	mAP	Top1	mAP	Top1
None	[CLS]	78.5	90.0	64.3	80.3	92.1	96.9	79.8	90.7
k RNN	[CLS]	80.5	89.0	66.6	78.8	93.1	96.6	83.1	90.3
Graph	[CLS]	80.9	89.0	65.6	74.8	93.0	95.2	83.2	88.1
SGR	[CLS]	80.2	89.0	65.5	75.1	93.0	96.2	81.1	87.0
None	FG	76.7	89.3	81.1	90.5	92.4	97.1	93.6	97.5
k RNN	FG	78.7	88.3	82.0	90.1	93.1	96.6	94.2	97.3
Graph	FG	78.7	87.3	81.9	89.1	93.1	95.2	93.9	95.6
SGR	FG	78.2	87.4	81.7	89.4	92.9	97.1	93.8	96.2

Supervised Training Given the availability of writer identities in our training dataset, a straightforward training approach for the ViT is to use the writer identity as the classification target. We also experiment with using the page id as a classification target. As illustrated in Table 2, both supervised training strategies underperform when compared to self-supervised methods. Interestingly, contrary to our findings in Section 5.1, the foreground tokens yield better performance with sum-pooling than the class tokens.

5.3 Reranking

In this section, we evaluate the impact of several reranking methods on the performance of our baseline implementation. We evaluate the k RNN reranking used in [31], Graph reranking (Graph) [30], and Similarity Graph Reranking (SGR) [30]. We evaluate the impact of reranking in combination with both the class tokens and foreground tokens ($t_{fg} = 10$), and both sum-pooling and VLAD as encoding. To save computation, we don't optimize the reranking hyperparameters for each combination on the training set but use fixed values which we found to work well across all combinations. For k RNN we set $k = 2$. For Graph-reranking we set $k_1 = 4, k_2 = 2, L = 3$ following [30]. For SGR we use $k = 2, \gamma = 0.1$. The $\gamma = 0.4$ suggested in [30] heavily reduced our results, likely due to our higher baseline performance.

Table 3 shows that all reranking methods increase mAP at the cost of Top1 accuracy. On the Historical-WI dataset, sum-pooling of class tokens still produces better results (80.9% mAP) than foreground tokens (78.7% mAP). On the HisIR19 dataset, both the class tokens and the foreground tokens achieve equal mAP of 93.1%, closing the slight gap in the un-reranked results (see Table 1). Even with reranking, VLAD computed on the class tokens still heavily

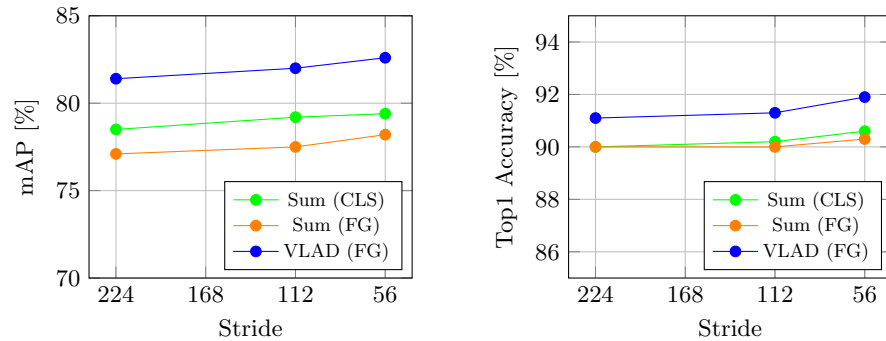


Fig. 4: Evaluation of S_{eval} , i.e., the stride with which windows are sampled from the test documents during inference on the Historical-WI dataset. We compare different combinations of features and aggregating methods. The left plot shows mAP and the right plot shows Top1 accuracy.

underperforms compared to other combinations. The best results overall are still achieved with VLAD on the foreground tokens (82.0% on Historical-WI, 94.2% on HisIR19). While all reranking approaches yield similar mAP results, k RNN produces slightly better results on both datasets, likely due to retaining the best Top1 accuracy.

5.4 Parameter Evaluation

In this section, we evaluate the remaining parameters of our pipeline on the Historical-WI dataset. We do not consider encoding class tokens using VLAD as the previous experiments have shown this combination to not yield competitive results.

Evaluation Stride During inference, we sample windows in a regular grid with stride S_{eval} . In the previous experiments, we used a baseline value of $S_{eval} = 224$. Figure 4 shows that reducing the stride enhances performance in all cases. Lowering $S_{eval} = 224$ to 56 improves the performance of VLAD on the foreground tokens to 82.6% mAP (+1.5%). We did not evaluate smaller strides for computation reasons as halving the stride produces four times more input windows.

Number of VLAD Cluster Centers Our baseline constructs a codebook of size $C = 100$ for the VLAD encoding. Figure 5 shows that both mAP and Top1 are relatively stable regardless of the number of clusters. Even with as few as 10 clusters performance only deteriorates slightly, and still considerably improves on sum-pooling.

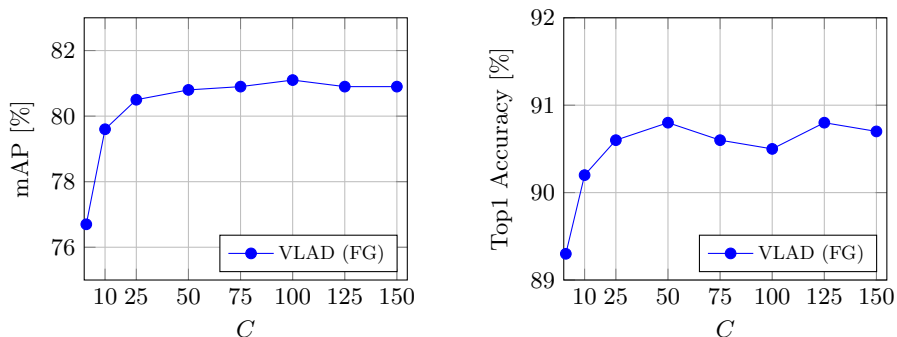


Fig. 5: Evaluation of parameter C , i.e., the number of cluster centers used to compute the VLAD codebook Θ . The left plot shows mAP and the right plot shows Top1 accuracy.

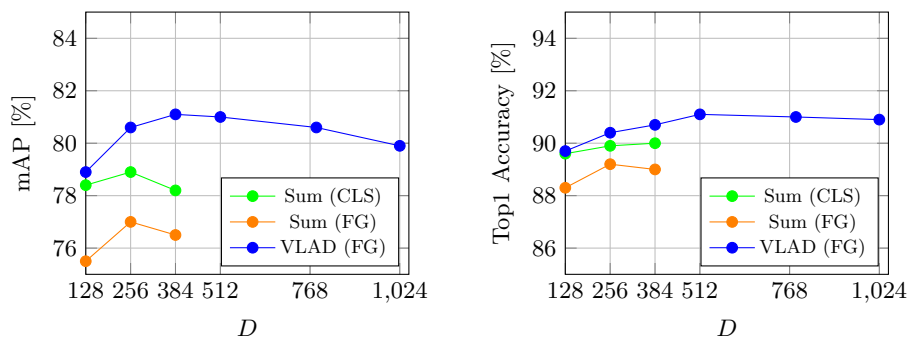


Fig. 6: Evaluation of the parameter D , i.e. the number of dimensions kept during principal component analysis. We evaluate different combinations of features and aggregation methods. The left plot shows the mAP and the right plot shows the Top1 accuracy.

PCA dimensionality After aggregation, we use principal component analysis with whitening and dimensionality reduction to $D = 384$ dimensions. As Figure 6 shows, retrieval performance with VLAD peaks at $D = 384$ dimensions, whereas Top1 accuracy peaks at $D = 512$. With sum-pooling, the dimensionality of the final page descriptor is equal to the ViT’s embedding dimensionality, in our case 384. As such, larger values can not be evaluated. For both class tokens and foreground tokens, mAP and Top1 peak around 256 dimensions but still fall short compared to VLAD.

5.5 Comparison with State-of-the-Art

For our comparison with other methods, we distinguish between the performance without additional reranking steps and the performance when reranking is ap-

plied. We use the baseline parameters outlined in Section 4.3, with the exception of reducing the evaluation stride, i.e., setting $S_{\text{eval}} = 56$. The results are given in Table 4.

Historical-WI On the Historical-WI dataset, our method surpasses existing methods considerably in terms of mAP. We achieve 82.6% mAP and 91.9% Top1 score without reranking, beating the best previous method of Lai *et al.* (77.1% mAP, 90.1% mAP) by 5.5% mAP and 0.8% Top1 score. Notably, our method also exceeds previous methods when using sum-pooling as an encoding in conjunction with the ViTs class token. We achieve 79.4% mAP with this configuration, beating previous methods by more than 2% mAP. This is especially noteworthy as methods based on CNN-based features perform much worse with sum-pooling: Christlein *et al.* [8] report a drop of over 30% mAP compared to their *m*VLAD encoding. When applying additional reranking, our method still beats previous methods. We achieve 83.1% mAP with reranking, improving over the method of Peer *et al.* (80.6% mAP) [30] by 2.5% mAP. Still, even the sum-pooling of class tokens outperforms previous methods slightly.

HisIR19 On the HisIR19 dataset, our method also outperforms previous methods considerably. Without reranking, we achieve 94.4% mAP and 97.8% Top1 accuracy, beating the previously best method (92.8% mAP, 97.4% Top1 [22]) by 1.6% mAP and 0.4% Top1 accuracy. When also applying reranking, our method achieves 95.0% mAP and 97.6% Top1 accuracy, improving over the best previous method by 1.8% mAP and 0.9% Top1 accuracy. Similar to our findings for the Historical-WI dataset, the sum-pooled class tokens achieve competitive performance, with and without reranking.

CVL Additionally, we evaluate our method on the CVL database [21], a dataset containing modern handwriting. We directly use the ViT feature extractor trained on the Historical-WI dataset without any fine-tuning and construct the VLAD codebook from the training set of the CVL dataset. With reranking, we achieve 98.6% mAP and 99.4% Top1 accuracy, matching the results of previous methods [10,31]. Even without reranking, our method achieve a mAP of 97.1%, highlighting the robustness of the extracted features, despite only training on a relatively small set of historical documents.

6 Conclusion

In this work, we presented a novel method using a Vision Transformer as a feature extractor. The model is trained in an unsupervised fashion. Patch tokens containing foreground are extracted as local features and subsequently encoded with VLAD. Retrieval is done using the cosine distance, with optional reranking. Our method achieved a new state-of-the-art performance on the historical

Table 4: Comparison of our method with the state of the art on the Historical-WI, HisIR19 and CVL test datasets. We evaluate two configurations of our method: aggregating class tokens with sum-pooling and aggregating foreground tokens with VLAD. For all datasets, we use the same evaluation parameters, i.e., $D = 384$, $C = 100$, $t_{fg} = 10$, $k = 2$, $S_{eval} = 56$.

Method	Encoding	Reranking	Historical-WI		HisIR19		CVL	
			mAP	Top1	mAP	Top1	mAP	Top1
Peer <i>et al.</i> [30]	NetVLAD	-	73.4	88.5	91.6	96.1	-	-
Christlein <i>et al.</i> [8]	mVLAD	-	74.8	88.6	-	-	-	-
Lai <i>et al.</i> [22]	bVLAD	-	77.1	90.1	92.5	97.4	-	-
Ours (CLS)	Sum	-	79.4	90.6	92.8	97.3	94.2	98.9
Ours (FG)	VLAD	-	82.6	91.9	94.4	97.8	97.1	99.4
Christlein <i>et al.</i> [10]	VLAD	E-SVM	-	-	-	-	98.4	99.5
Rasoulzadeh [31]	NetVLAD	k RNN	-	-	-	-	98.6	99.2
Christlein <i>et al.</i> [8]	mVLAD	E-SVM	76.2	88.7	91.2	97.0	-	-
Christlein <i>et al.</i> [8]	mVLAD	P/T-SVM [19]	78.2	89.4	-	-	-	-
Peer <i>et al.</i> [30]	NetVLAD	SGR	80.6	91.1	93.2	96.7	-	-
Ours (CLS)	Sum	k RNN	81.2	90.6	93.2	96.9	97.6	98.8
Ours (FG)	VLAD	k RNN	83.1	90.9	95.0	97.6	98.6	99.4

benchmark datasets Historical-WI and HisIR19, improving over previous methods by 2.5% mAP and 1.8% mAP respectively. We additionally showed that our method is versatile and also works well on modern datasets, achieving 98.6% mAP on the CVL database without requiring any fine-tuning of the ViT.

Further research could be done to evaluate other SSL methods for model training and different model architectures. In terms of SSL methods, DINOv2 [27] introduced several improvements to the iBOT framework which might be interesting for writer retrieval, e.g. the KoLeo regularizer [32]. Regarding architectures, Swin-Transformers [24] have shown promising results in domains with limited data, which might help to boost performance and training time.

Moreover, we showed that only considering patch tokens containing sufficient foreground information is beneficial. Here, future research could investigate other strategies for filtering out patch tokens, for instance by utilizing the self-attention of the ViT to identify relevant patches.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT Pre-Training of Image Transformers. In: International Conference on Learning Representations (2022)
2. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In: Advances in Neural Information Processing Systems. vol. 33, pp. 9912–9924 (2020)

3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
4. Chen, X., He, K.: Exploring Simple Siamese Representation Learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
5. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
6. Christlein, V., Bernecker, D., Angelopoulou, E.: Writer identification using VLAD encoded contour-Zernike moments. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 906–910 (2015)
7. Christlein, V., Bernecker, D., Hönig, F., Angelopoulou, E.: Writer Identification and Verification using GMM Supervectors. In: IEEE Winter Conference on Applications of Computer Vision. pp. 998–1005. IEEE (2014)
8. Christlein, V., Gropp, M., Fiel, S., Maier, A.: Unsupervised Feature Learning for Writer Identification and Writer Retrieval. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 991–997. IEEE (2017)
9. Christlein, V., Maier, A.: Encoding CNN Activations for Writer Recognition. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 169–174. IEEE (2018)
10. Christlein, V., Maier, A.: Encoding cnn activations for writer recognition. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 169–174 (2018). <https://doi.org/10.1109/DAS.2018.9>
11. Christlein, V., Nicolaou, A., Seuret, M., Stutzmann, D., Maier, A.: ICDAR 2019 Competition on Image Retrieval for Historical Handwritten Documents. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1505–1509. IEEE (2019)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houshy, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2021)
13. Fiel, S., Kleber, F., Diem, M., Christlein, V., Louloudis, G., Nikos, S., Gatos, B.: ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1377–1382. IEEE (2017)
14. Fiel, S., Sablatnig, R.: Writer Identification and Writer Retrieval using the Fisher Vector on Visual Vocabularies. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 545–549. IEEE (2013)
15. Fiel, S., Sablatnig, R.: Writer Identification and Retrieval using a Convolutional Neural Network. In: Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015, Proceedings, Part II 16. pp. 26–37. Springer (2015)
16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
17. Jain, R., Doermann, D.: Combining Local Features for Offline Writer Identification. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. pp. 583–588 (2014)

18. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(9), 1704–1716 (2011)
19. Jordan, S., Seuret, M., Král, P., Lenc, L., Martínek, J., Wiermann, B., Schwinger, T., Maier, A., Christlein, V.: Re-ranking for Writer Identification and Writer Retrieval. In: *Document Analysis Systems: 14th IAPR International Workshop*. pp. 572–586. Springer (2020)
20. Kakogeorgiou, I., Gidaris, S., Psomas, B., Avrithis, Y., Bursuc, A., Karantzalos, K., Komodakis, N.: What to Hide from Your Students: Attention-Guided Masked Image Modeling. In: *European Conference on Computer Vision*. pp. 300–318. Springer (2022)
21. Kleber, F., Fiel, S., Diem, M., Sablatnig, R.: 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. IEEE (2013)
22. Lai, S., Zhu, Y., Jin, L.: Encoding Pathlet and SIFT Features With Bagged VLAD for Historical Writer Identification. *IEEE Transactions on Information Forensics and Security* **15**, 3553–3566 (2020)
23. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 37, pp. 13094–13102 (2023)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 10012–10022 (2021)
25. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for off-line handwriting recognition. *International Journal on Document Analysis and Recognition* **5**, 39–46 (2002)
26. Murray, N., Perronnin, F.: Generalized max pooling. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2473–2480 (2014)
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023)
28. Peer, M., Kleber, F., Sablatnig, R.: Self-supervised Vision Transformers with Data Augmentation Strategies Using Morphological Operations for Writer Retrieval. In: *International Conference on Frontiers in Handwriting Recognition*. pp. 122–136. Springer (2022)
29. Peer, M., Kleber, F., Sablatnig, R.: Writer retrieval using compact convolutional transformers and netmvlad. In: *2022 26th International Conference on Pattern Recognition (ICPR)*. pp. 1571–1578 (2022). <https://doi.org/10.1109/ICPR56361.2022.9956155>
30. Peer, M., Kleber, F., Sablatnig, R.: Towards Writer Retrieval for Historical Datasets. In: *International Conference on Document Analysis and Recognition*. pp. 411–427. Springer (2023)
31. Rasoulzadeh, S., BabaAli, B.: Writer identification and writer retrieval based on NetVLAD with Re-ranking. *IET Biometrics* **11**(1), 10–22 (2022)
32. Sablayrolles, A., Douze, M., Schmid, C., Jégou, H.: Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198* (2018)

33. Sauvola, J., Seppanen, T., Haapakoski, S., Pietikainen, M.: Adaptive Document Binarization. In: Proceedings of the Fourth International Conference on Document Analysis and Recognition. vol. 1, pp. 147–152. IEEE (1997)
34. Sculley, D.: Web-Scale K-Means Clustering. In: Proceedings of the 19th International Conference on World Wide Web. pp. 1177–1178 (2010)
35. Souibgui, M.A., Biswas, S., Mafla, A., Biten, A.F., Fornés, A., Kessentini, Y., Lladós, J., Gomez, L., Karatzas, D.: Text-DIAE: A Self-Supervised Degradation Invariant Autoencoders for Text Recognition and Document Enhancement. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2330–2338 (2023)
36. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: a Simple Framework for Masked Image Modeling. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9643–9653. IEEE (2022)
37. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: Image BERT Pre-training with Online Tokenizer. In: International Conference on Learning Representations (2022)