International Journal of Computer Vision (2022) 130:1325–1339
https://doi.org/10.1007/s11263-022-01576-x

Spatially-Consistent Feature Matching and Learning for Heritage Image Analysis

Xi Shen[1] · Robin Champenois[1] · Shiry Ginosar[2] · Ilaria Pastrolin[3] · Morgane Rousselot[3] · Oumayma Bounou[8] · Tom Monnier[1] · Spyros Gidaris[9] · François Bougard[7] · Pierre-Guillaume Raverdy[8] · Marie-Françoise Limon[4] · Christine Bénévent[3] · Marc Smith[3] · Olivier Poncet[3] · K. Bender[6] · Béatrice Joyeux-Prunel[5] · Elizabeth Honig[2] · Alexei A. Efros[2] · Mathieu Aubry[1]

Received: 14 March 2021 / Accepted: 13 January 2022 / Published online: 25 March 2022

Abstract
Progress in the digitization of cultural assets leads to online databases that become too large for a human to analyze. Moreover, some analyses might be challenging, even for experts. In this paper, we explore two applications of computer vision to analyze historical data: watermark recognition and one-shot repeated pattern detection in artwork collections. Both problems present computer vision challenges which we believe to be representative of the ones encountered in cultural heritage applications: limited supervision is available, the tasks are fine-grained recognition, and the data comes in several different modalities. Both applications are also highly practical, as recognizing watermarks makes it possible to date and locate documents, while detecting repeated patterns allows exploring visual links between artworks. We demonstrate on both tasks the benefits of relying on deep mid-level features. More precisely, we define an image similarity score based on geometric verification of mid-level features and show how spatial consistency can be used to fine-tune out-of-the-box features for the target dataset with weak or no supervision. This paper relates and extends our previous works (Shen et al. in Discovering visual patterns in art collections with spatially-consistent feature learning, 2019; Shen et al. in Large-scale historical watermark recognition dataset and a new consistency-based approach, 2020). Our code and data are available at http://imagine.enpc.fr/~shenx/HisImgAnalysis/.

Keywords Feature learning · Self-supervised learning · Artwork analysis · Watermark recognition

1 Introduction

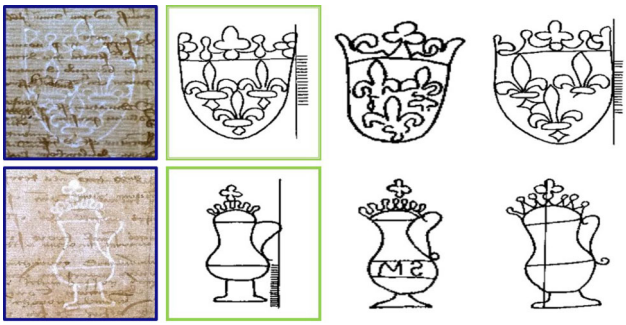
Learning global image features has been successful in many image recognition tasks, such as image classification (He et al., 2016), landmark retrieval (Gordoet al., 2017; Radenović et al., 2018), metric learning (Kim et al., 2020; Elezi et al., 2020; Teh et al., 2020) and few-shot classification (Vinyals et al., 2016; Snell et al., 2017). In this paper, we highlight that matching and learning mid-level features and taking into account spatial information is better adapted to two cultural heritage applications required to recognize exactly repeated patterns across wide appearance changes, namely, historical watermark recognition (Fig. 1a) and one-shot repeated pattern detection in artwork collections (Fig. 1b).

For watermark recognition (Fig. 1a), we aim at retrieving the exact drawing (green) corresponding to the query photograph (blue) of a watermark from a very large database,

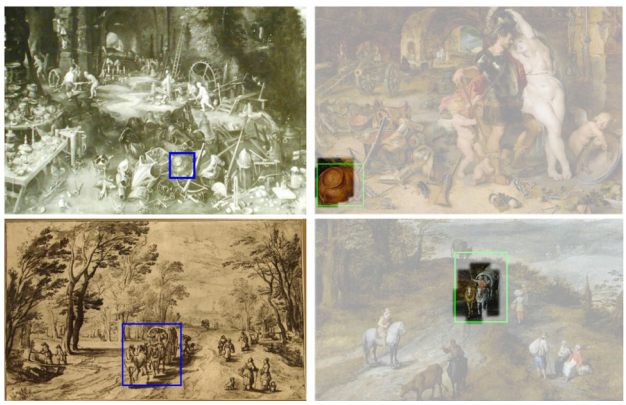
Communicated by Katsushi Ikeuchi.

✉ Xi Shen
Xi.Shen@enpc.fr

1 LIGM (UMR 8049), École des Ponts, CNRS, Univ. Gustave Eiffel, Marne-la-Vallee, France
2 University of California, Berkeley, USA
3 École Nationale des Chartes, Paris, France
4 Archives Nationales, Paris, France
5 University of Geneva, Geneva, Switzerland
6 Independent researcher, Gent, Belgium
7 IRHT, Paris, France
8 INRIA, Paris, France
9 Valeo AI, Paris, France



(a) Historical watermark recognition



(b) One-shot repeated pattern detection

Fig. 1 We present a unified approach to two problems: **a** historical watermark recognition, and **b** repeated-pattern detection in artwork collections

which includes many similar watermarks each corresponding to a different category. In artwork collections, we tackle one-shot repeated pattern detection (Fig. 1b), i.e. given a query detail (blue) we retrieve it in artworks with various styles (green). Both applications share similar Computer Vision challenges: (1) since annotations are very costly and require expert knowledge, there are only few category-level annotations available for training for watermark recognition and no training annotations for repeated detail detection in art collections; (2) the artwork details and the watermarks are represented in different visual domains; (3) both tasks require fine-grained separation between similar but distinct classes. We believe these challenges are common to many cultural heritage applications.

The two problems we tackle are also highly practical for historians and archivists. For watermark recognition, the drawings we recognize come from large catalogs (http://www.ksbm.oeaw.ac.at/_scripts/php/BR.php; Briquet, 1907) where they are associated with information such as the date and location of the paper fabrication, which are important clues to analyze historical documents. Similarly, repeated patterns in art collections are important to art historians and allow them to analyze influences, find provenance, and even establish authorship (Castellano et al., 2021).

For each application, we introduced datasets for training and evaluation adapted to modern deep learning approaches (Shen et al., 2019, 2020) and extend them in this paper. For historical watermark recognition, we first collected more than 6k new photographs for hundreds of classes, then searched and photographed systematically the exact original watermarks corresponding to hundreds of line drawings printed in Briquet's classic catalog of more than 16k (Briquet, 1907) watermarks. This allows us to tackle at scale for the first time the scenarios of practical interest for scholars: one-shot instance recognition and cross-domain one-shot instance recognition among more than 16k fine-grained classes. For one-shot detection of the artwork's details, we used a database of 1587 artworks from Brueghel's workshop (http://www.janbrueghel.net/; Honig, 2016), selected 10 of the most commonly repeated details, and annotated their 273 occurrences in the full dataset for evaluation. Note that these visual patterns are near-duplicated but with various styles as artworks are in different media (e.g. oil, ink, chalk, watercolor) and on different materials (e.g. paper, panel, copper). In this paper, we provide additional results on a collection of 25,681 depictions of Venus (Bender, 2015). It is more than an order of magnitude larger and more diverse than the Brueghel dataset, with works spreading over centuries and continents.

We demonstrate that both problems can be successfully addressed by leveraging the geometric consistency of mid-level feature matches both for training features and for scoring candidate correspondences. Our idea can be seen as revisiting the geometric verification of classical feature matches (Sivic & Zisserman, 2003) for mid-level deep feature training and matching. More precisely, we propose a local matching score to compare pairs of images, which combines spatial consistency and local feature similarity. We also present a feature fine-tuning strategy that can be used without any annotations or with weak category-level annotations: we mine positive and negative pairs using geometric verification and use them to optimize a standard triplet loss.

We demonstrate experimentally that the proposed local matching score provides important performance gains for both tasks compared to global features, and that our feature fine-tuning—without any annotations on the Brueghel dataset and with weak category-level supervision on the watermark dataset—further improves the results. For example, on the challenging cross-domain one-shot recognition over more than 16k fine-grained watermark categories, for which average pooling fails (0% accuracy), our local matching score using classification pretrained features achieves 45% accuracy, and our fine-tuning further improves it to 55%.

Our code and data are available at our project page http://imagine.enpc.fr/~shenx/HisImgAnalysis/.

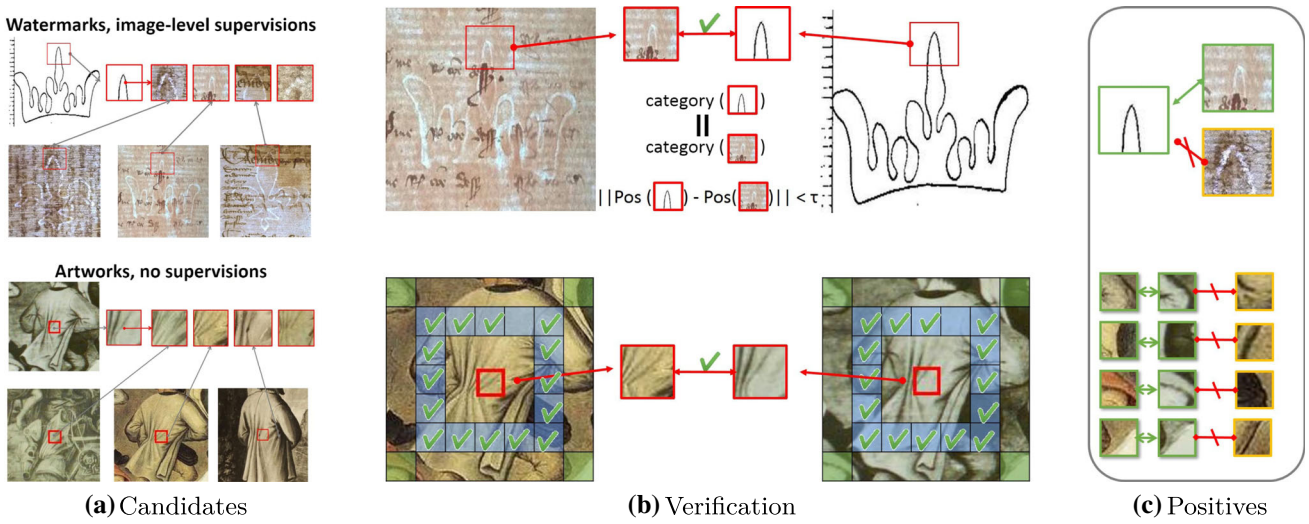


Fig. 2 Mid-level feature learning *with weak supervisions* or *without supervisions*. **a** Our approach relies on candidate correspondences obtained by matching the features of a proposal region (in red) to the full database. **b** In case of category-level supervisions (top), we verify whether matched pairs belong to the same category and their spatial difference is small. If no supervisions is available (bottom), the candidate correspondences are verified by matching the features of the verification region (in blue, note that, to make the figure visible, the blue regions are visualized as 5×5 squares while we use 10×10 squares in our algorithm.) of the query in the candidate images and checking for consistency. **c** Finally, we extract features from the positive (in green) and negative (in yellow) regions and use them to improve the features using a metric learning loss (Color figure online)

2 Related Work

In this section, we first discuss the use of mid-level features for image recognition, then works related to our two applications, artwork analysis and watermark recognition.

Mid-level features for image recognition. Recent work analyzing the performance of CNNs (Brendel & Bethge, 2019; Geirhos et al., 2019) suggests that they might ignore a large part of the spatial information present in the image, and rather work in a way similar to classical order-less bags-of-features methods (Wallraven et al., 2003; Grauman & Darrell, 2005; Zhang et al., 2007). This might not be suitable for recognizing repeated patterns such as watermarks and near-duplicated details in artworks, where the actual shape is key. To build on the local features learned by CNNs but consider spatial information, we follow an approach closely related to the classic spatial verification step introduced in Video Google (Sivic & Zisserman, 2003) with SIFT features (Lowe, 2004). Rather than using SIFTs, which we found were not adapted to non-photographic depictions, we use intermediary deep features, that can be thought of as mid-level image features. Mid-level features (Singh et al., 2012; Doersch et al., 2013, 2014) have been used in cross-domain matching (Aubry et al., 2014), video instance segmentation (Wang et al., 2019) and coarse parametric transformation estimation (Shen et al., 2020). Our feature fine-tuning also leverages the spatial structure of images: thus, it is related both to self-supervised feature learning methods that use spatial information to learn deep visual features in images (Doersch et al., 2015; Noroozi & Favaro, 2016) or videos (Wang et al., 2019; Jabri et al., 2020), and to Rocco et al., (2018) that uses neighborhood consensus to learn correspondences from the correlation map. We adapted (Rocco et al., 2018) to use as baseline.

Computer vision and art. There is a long-standing and fruitful collaboration between computer vision and art. On the synthesis side, promising results have been obtained for transferring artistic style to a photograph (Hertzmann et al., 2001; Gatys et al., 2016; Zhu et al., 2017), or even trying to create art (Elgammal et al., 2017; Hertzmann, 2018). On the analysis side, there are several efforts on the collection and annotation of large-scale art datasets (Karayev et al., 2013; Mensink & Van Gemert, 2014; Picard et al., 2015; Wilber et al., 2017; Strezoski & Worring, 2017; Mao et al., 2017; Paumard et al., 2018), and using them for genre and authorship classification (Karayev et al., 2013; Tan et al., 2016; Strezoski & Worring, 2017) or fragment alignment (Paumard et al., 2018). Others focus on applying and generalizing visual correspondence and object detection methods to paintings using both classical (Shrivastava et al., 2011; Ginosar et al., 2014; Crowley & Zisserman, 2013; Aubry et al., 2014; Seguin et al., 2017), as well as deep (Crowley et al., 2015; Crowley & Zisserman, 2016; Westlake et al., 2016; Gonthieret al., 2018) approaches. In particular, Yin et al., (2016) used the same Brueghel data (http://www.janbrueghel.net/; Honig, 2016) as us and annotate it to train detectors for five object categories (carts, cows, wind-

mills, rowboats, and sailboats). Beyond classification, several projects have built search engines for artworks (https://imgs.ai/; timemachine). An original and related approach was also proposed by Jenicek and Chum (2019) to relate artwork by estimating human poses rather than simply comparing abstract features. Our work is also related to Ubeda et al. (2019) which performs pattern spotting in historical documents but for patterns with little appearance change. To the best of our knowledge, our work is the first to demonstrate a method capable of detecting small copied details across different depiction styles: this fine-grained analysis is the most relevant for art historians, as it may allow them to discover influences, find provenance and establish authorship.

Historical watermark imaging and recognition. Manual tracing and back-lit photography are two standard ways to reproduce watermarks. Manual tracing (Picard, 1977; Briquet, 1907) consists in copying the watermark pattern on tracing paper and was tyically used to create the classical catalogs of watermarks, which are nowdays aggregated in online databases such as Briquet Online (http://www.ksbm.oeaw.ac.at/_scripts/php/BR.php) which includes specifically the drawings from Briquet (1907) we worked with. Back-lit photography is the most common technique to acquire an actual photograph of a watermark. While a watermark is often barely visible, placing it in front of a light source and looking at the transmitted light reveals it more or less clearly.

This cross-modality between the drawings and the photographs one would like to identify, is one of the main challenges of watermark recognition. Several studies have focused on localizing and extracting the pattern of a watermark from a photograph (Hiary & Ng, 2007; Hiary, 2008; Said & Hiary, 2016), which could potentially be helpful to match a photograph to a database of drawings (Rauber et al., 1997). However, these approaches are complex, not end-to-end and have not yet been demonstrated on a large scale. Some works on watermark recognition focuse on drawings. Rauber et al. (1997) uses histogram-based descriptors in a spirit similar to shape context (Belongie et al., 2001), while more recent works use machine learning approaches, such as dictionary learning (Picard et al., 2016) or neural networks (Pondenkandath et al., 2018). The study most similar to us is probably (Pondenkandath et al., 2018), that used a non-publicly available database (Frauenknecht & Stieglecker 2015) to train a convolutional neural network classifying 106,000 watermark into 12 coarse categories.

3 Method

In this section, we start by introducing our similarity score named "local matching score" which combines spatial consistency and local feature similarity in Sect. 3.1. Then, in Sect. 3.2, we detail our feature learning strategies for both the weakly supervised and the unsupervised scenario, which correspond to the two applications focused in this work. Precisely, for watermark recognition, a small number of image-level annotations are accessible and watermarks are approximately centered. For artworks, no annotations are available. The overview of our feature learning strategies is shown in Fig. 2. Finally, we provide some implementation details in Sect. 3.3.

3.1 Local Matching Score

Let us assume we have access to a trained convolutional neural network that we want to use to compare two images I_1 and I_2. The simplest solution is to use the similarity between aggregated features, typically the cosine similarity between average-pooled features for a ResNet network. However, such features will completely discard location information, which is likely not well adapted for fine-grained differentiation between similar classes. Another natural solution is then to consider convolutional features and use the similarity between the resulting descriptors. However, this gives the same importance to all local features, while some might be discriminative and others might not be, and this amounts to comparing only convolutional features at the same spatial location, i.e., assuming that the images are exactly aligned. This is a strong assumption and is not realistic in the cases we study: precisely centering and scaling the watermarks is difficult and the copied details in artworks might have undergone some amount of local deformation.

Thus, we propose a similarity metric to compare a pair of images I_1 and I_2 which relies on matching densely mid-level CNN features (e.g., *conv4* in ResNet He et al., (2016)) computed at different scales. To select only discriminative features, we check that they correspond to reciprocal matches and discard them otherwise. More formally, for every local feature f_1^i in I_1, we first find its best match in I_2, f_2^i. We then check that it is a reciprocal match, i.e. that the best match i.e. the best match in I_1 of f_2^i is f_1^i and we write $\mathcal{M}$ the set of all reciprocal matches. We also write x_1^i the position of f_1^i in I_1, and x_2^i the position f_2^i in I_2.

Assuming that the two images are approximately aligned, which is true for example in the watermark dataset, we define the local matching score S by combining a *Spatial Consistency* score (SC) measuring the similarity between the positions x_1^i and x_2^i, and a *Feature Similarity* (FS) measuring the distance between f_1^i and f_2^i:

$$S(I_1, I_2) = \sum_{i \in \mathcal{M}} \underbrace{e^{-\frac{\|x_1^i - x_2^i\|^2}{2\sigma^2}}}_{SC} \underbrace{s(f_1^i, f_2^i)}_{FS} \tag{1}$$

where s is a feature level similarity—for which we use cosine similarity in all of our experiments—and σ is a tolerance parameter.

If the images are not coarsely aligned, which is the case for repeated details detection in artworks, we need to further model the alignment between the two images. We do so by estimating a parametric transformation $\mathcal{T}$ between the images, using RANSAC on the set of reciprocal matches $\mathcal{M}$. We then compute the score:

$$S(I_1, I_2) = \sum_{i \in \mathcal{M}} \underbrace{e^{-\frac{\|\mathcal{T}(x_1^i) - x_2^i\|^2}{2\sigma^2}}}_{SC} \underbrace{s(f_1^i, f_2^i)}_{FS} \tag{2}$$

$\mathcal{T}$ is an Affine transformation and therefore has 6 parameters. Classical RANSAC is employed to estimate $\mathcal{T}$: for each iteration, we sample 3 matches to compute a candidate transformation ; a match is considered as an inlier if its difference with the candidate transformation is smaller than 2 in the feature space; the final transformation is the one with maximum number of inliers after 1k iterations.

3.2 Spatially-Consistent Feature Learning

The overview of our feature learning strategies is shown in Fig. 2, which can be separated into three parts: (1) searching candidates of training samples (Fig. 2a); (2) verification of candidates with weak category-level supervision or no supervision (Fig. 2b); (3) training on selected samples (Fig. 2c). We first explain the objective function we use to improve our features in Sect. 3.2.1, then present how we sample candidate matches in Sect. 3.2.2. Finally, we detail how to filter training samples in the candidate matches in Sect. 3.2.3.

3.2.1 Hypothesis and Objective Function

Our goal is to improve local features for matching. Starting with a standard pretrained deep feature extractor, we extract matching regions from the dataset that we can then circle back and use to improve the features. Our two key hypotheses are that: (1) our dataset includes large parts of images that contain repeated patterns, and (2) the initial feature descriptor is good enough to extract some positive matches. We follow a metric learning approach. Assuming we have a set of positive pairs $\mathcal{P}$ and a set of negative pairs $\mathcal{N}$, we learn our feature extractor f by minimizing a standard triplet loss:

$$\begin{aligned}\mathcal{L}(f) = &\sum_{(n_1,n_2)\in\mathcal{N}} \max(1-\lambda, s(f(n_1), f(n_2))) \\ &- \sum_{(p_1,p_2)\in\mathcal{P}} \min(\lambda, s(f(p_1), f(p_2))\end{aligned} \tag{3}$$

where the similarity measure s is the cosine similarity and λ is a hyper-parameter.

The main challenge for learning such features is defining the sets of positive and negative pairs. In the following sections, we show how to find these pairs in both weakly supervised and unsupervised cases.

3.2.2 Candidates

We randomly sample several query features from each image in the dataset and match them densely at every scale to all the images in the dataset using cosine similarity. This can be done efficiently and in parallel for many queries using the normalized query features as the weights of a convolution applied to the normalized features. For each query, we select its top K matches in the entire dataset as candidate correspondences (Fig. 2a).

3.2.3 Verification

We now explain how we filter the positive and negative pairs from these candidate local feature pairs both in the case with and without image-level supervision.

Image-level supervisions In the case where image level category labels are available, they can be used as a first cue to select positive matches: they can only exist between images belonging to the same category. Furthermore, if we assume that the images are coarsely aligned, as is the case for our watermark recognition problem, we can additionally filter only matches that are spatially consistent up to a given spatial threshold τ. This threshold allows to take into account misalignment between the source and target images and we study its influence in our experiments. Once a positive pair has been identified, we look for hard negatives by matching one of the positive features with all the photographs of watermarks from other categories, and select the the most similar feature as negative (Fig. 2b top).

Unsupervised setting In the more challenging case where no image-level label is available and spatial alignment cannot be assumed, as in the case of repeated details in art collections, we rely on spatial consistency to verify the quality of candidates: a match will be considered valid if its neighbors agree with it. More precisely, let us assume we have a candidate match between features from the proposal region p_1 in image I_1 and a corresponding region p_2 in image I_2, visualized in red in Fig. 2b bottom. We define a *verification region* around p_1, visualized in blue. Every feature in this region is individually matched to image I_2, and votes for the candidate match if it matched consistently with p_2. Summing the votes of all features in the verification region allows us to rank the candidate matches. A fixed percentage of the candidates are then considered verified.

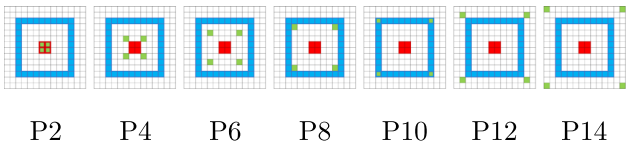


Fig. 3 Different region configurations. red: query regions, blue: verification regions, green: positive regions (Color figure online)

The choice of the verification region is important to the verification step. The key aspect is that the features in the verification region should be, as much as possible, independent of the features in the proposal region. On the other hand, having them too far apart would reduce the chances of the region being completely matched. For our experiments, we used the 10x10 feature square centred around the query region.

Finally, given a set of verified correspondences, we have to decide which features to use as positive training pairs (Fig. 2c). One possibility would be to directly use features from the proposal region, since they have been verified. However, since the proposal region has already been "used" once (to verify the matches), it does not bring enough independent signal to make quality hard positives. Instead, we propose to sample positives from a different *positive region*. We evaluated different configurations for the positive region, as visualized in Fig. 3 (in green). We choose to keep only 4 positive pairs per verified proposal, positioned at the corners of a square and denote the different setups as P2 to P14, the number corresponding to the size of the square. We will show in the experiments (Sect. 5.2) that P12 and P14 perform better than the alternatives. The features from the positive regions (Fig. 2b bottom in green) are then used as hard positives for feature fine-tuning (Fig. 2c bottom).

For a given positive pair, we extract top-N closest features from the same image as negatives. This selects hard negatives and avoids any difference in the distribution of the depiction styles in our positive and negative samples. We chose a relatively high number (N = 20) of negatives to account for the fact that some of them might actually correspond to matching regions, for example, in the case of repeated elements, or to locations near the optimal match.

3.3 Implementation Details

In all of our experiments, we used *conv4* features of the ResNet-18 (He et al., 2016) architecture and train the models with the Adam (Kingma & Ba, 2014) optimizer with learning rate 1e-5 and $\beta = [0.9, 0.99]$. For both applications, we use $K = 10$ candidate matches for each query.

For fine-tuning on watermark datasets, the initial features are pretrained on our classification dataset (see Sect. 4.1). The hyper-parameter λ is set to 1. Since the watermarks might be flipped and rotated with respect to the references, we consider matches with four rotated (0°, 90°, 180°, 270°) reference images and their flipped versions. Each source image was resized to 352×352 so that it was represented by 22×22 features. To be robust to scale discrepancies, we matched the source features to features extracted from the target image resized at five scales, corresponding to square features of widths 16, 19, 22, 25 and 28. Using a single Geforce GTX 1080 Ti GPU, training converged in approximately 2 h

For learning features on art collections, the initial features are pretrained on ImageNet (Deng et al., 2009). The hyper-parameter λ is set to 0.8. Since repeated patterns might have large scale differences in different artworks, we used 7 different scales ranging from 20 to 80 features in the largest dimension, regularly sampled on two octaves with 3 scales per octave. From these candidates, the top 10% with the most votes from neighbours were considered verified. Note that these parameters might need to be adjusted depending on the diversity and size of the dataset, but we found that they performed well for the Brueghel data (http://www.janbrueghel.net/). Using a single GPU Geforce GTX 1080 Ti, the training converged in approximately 10 h

4 Experiments on Watermark Recognition

In this section, we provide experimental results on watermarks recognitions. We first introduce the datasets we used, then present detailed results and analysis of the proposed local matching score and the proposed feature learning strategies.

4.1 Datasets

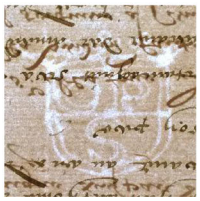
In this section, we explain how we created datasets to evaluate (A) one-shot and (B) cross-domain watermark recognition in the case where the watermarks are coarsely aligned. In the supplementary material available on our project page, we explain in detail the definition of classes and preprocessing procedures. More samples can be found in the project page.

Dataset A The goal of this dataset is to train and evaluate methods for one-shot fine-grained watermark recognition from photographs. It thus only includes photographs and is split into two parts: a first part that can be used for feature training or meta-training, with many examples of each watermark, and a second part, with few examples, to evaluate one-shot recognition.

We first created **dataset A-classification** by collecting 50 training and 10 validation images for 100 watermarks, which we found was large enough to perform pre-training / meta-training of CNNs. Examples of images from the same class are given in Fig. 4. We then collected **dataset A-one-shot** with 100 other test classes, each made of 3 photographs:

(a) Samples from the same class in A-classification

(b) Clean references (top) and associated test sample (bottom) in A-one-shot



(c) Photograph

(d) Drawing

(e) Synthetic

Fig. 4 Images in watermark datasets: **a** samples from the same class in A-classification; **b** three categories in A-one-shot. **c**–**e** Photograph, Drawing and Synthetic in B-cross-domain

one 'clean' image without any writing and two standard test images. We used 'clean' images as references for fine-grained one-shot recognition, as they are representative of what archivists typically collect as reference images and allow us to ensure that recognition is not related to the writing style of the document. We show examples from three different categories in Fig. 4 of references and and associated test images. Note that some challenges can be directly seen from the examples: (1) many of the training and testing examples are cluttered and in poor condition, and (2) some testing categories are highly similar.

Dataset B An important challenge is to use the drawings from existing watermark catalogs to perform recognition. Indeed, collecting watermarks and information for such catalogs is a very tedious and expensive work. We focused on the Briquet catalog (Briquet, 1907; http://www.ksbm.oeaw.ac.at/_scripts/php/BR.php). The available images include additional information, such as IDs of watermarks, paper line positions, or complementary marks that can be found at another position on the paper sheet. Such information is valuable to experts, but cannot easily be used in an automatic recognition system based on a single photograph. We thus extracted the main part of the watermarks whenever it was clear and ended up with 16,753 drawings that could be used as references for photograph recognition. Examples of these drawings can be seen on the project page.

The challenge that we want to evaluate on this dataset is to recognize photographs in a drawing database. We thus collected **dataset B** by searching the original archives that provided the material for the Briquet catalog (Briquet, 1907), in a specific city (Paris) and collected photographs for 240 classes (see examples in Fig. 4). We use 140 classes with a total of 463 photographs as training and keep the remaining 100 classes with 2 photographs in each class for testing. Because comparing photographs (Fig. 4) directly with a line drawing (Fig. 4) as a reference is very challenging, we also report results using a synthetic image generated from the drawing simply by using the average watermark color as background and making the drawing pattern lighter (Fig. 4). In our experiments, we first compare the methods for 100-class one-shot cross-domain classification and then give the results for the even more challenging 16,753-class classification. In the following sections, we refer the 100-class one-shot cross-domain classification as the task on the dataset B. The 16,753-class classification is the task on Briquet.

4.2 One-Shot Cross-Domain Recognition

In this section, we present results on the one-shot recognition on the dataset A-one-shot and one-shot cross-domain recognition on the dataset B. In all experiments, we use a network pretrained on A-classification (100 classes, 50 training and 10 validation images for each class). The best performances were obtained with a strong dropout (0.7 ratio), which is not surprising given the relatively small size of our dataset. The validation accuracy is 98.8%, the mis-classified images being only very difficult or ambiguous cases, which shows that our 6k images dataset was large enough to train a good network for fine-grained watermark classification.

Comparison to one-shot recognition methods

On dataset A-one-shot, which does not include any domain shift, we compare our method with some one-shot recognition methods:

- *Baseline* directly using the features learned by training a network on the classification task and computing the dot-product as the distance between classification weights and features.
- *Cosine Classifier* (Gidaris & Komodakis, 2018; Qi et al., 2018) have shown that the performance of the baseline can be improved if the dot-product operation (between classification weights and features) in the last linear layer

Table 1 Comparison with one-shot recognition approaches on dataset A-one-shot (200 images to classify in 100 categories unseen during training and described by a single 'clean' image)

Method/features	AvgPool		Concat		Local. Sim.	
Resolution	256	352	256	352	256	352
Baseline	69	78	74	86	75	81
Cosine classifier (Qi et al., 2018; Gidaris & Komodakis, 2018)	84	83	82	80	84	82
Matching networks (Vinyals et al., 2016)	74	76	76	78	82	84
Weights prediction (Gidaris & Komodakis, 2018)	86	83	84	-	85	-
Ours resolution 256	85					
Ours resolution 352	**90**					

Accuracy in %. Our score based on local matches clearly outperforms all baselines

of the network is replaced with the cosine similarity operation.
- *Matching Networks* we tried the metric-learning approach of Matching Networks (Vinyals et al., 2016), performing meta-training to solve one-shot recognition tasks using a differentiable nearest-neighbor-like classifier.
- *Weights Prediction* the one-shot recognition approach of Gidaris and Komodakis (2018). The key insight is that the novel class weights have two components: (1) average features of annotated samples; (2) linear combinations of the base class weights, which are meta-learned during the training. It uses a feature extractor learned with a cosine-similarity-based classifier which remains frozen during the meta-training procedure.

For each feature, we report three different similarities:

- *AvgPool* cosine similarity using the average pooled features.
- *Concat* cosine similarity on the descriptor formed by the concatenation of all the spatial features.
- *Local Sim* computing the cosine similarity over each local feature individually, then averaging.

For all baselines, we report the best performance over *conv4*/*conv5* features. The optimal parameters and training strategy for each baseline and a qualitative analysis of our results are reported on the project page.

The results are in Table 1, our local matching score leads to 85% accuracy for 256 × 256 images, which is close to the best one-shot approach, Weights Prediction (Gidaris & Komodakis, 2018), but without any specific feature learning. This demonstrates the interest of our local matching score for one-shot fine-grained watermark recognition. The performance can further be boosted to 90% by resizing images to a larger resolution, 352 × 352 pixels, which was not possible for Weights Prediction for very large features due to computational cost. This is clearly above the performances of the other methods.

Table 2 Comparison of our local matching score (Eq. 1) with alternative feature similarities

Method	A	B-4	B-4	Time (s/query)
Exact features comparison				
AvgPool	78	4	12	1
Concat	86	55	61	2
Local Sim.	81	56	65	2
Our local matching score				
Ours	**90**	**66**	**72**	15

Best results are in bold
Accuracy in % for one-shot recognition on dataset A-one-shot (A column) and one-shot cross-domain recognition on dataset B using either the drawing (B-4) or our synthetic image (B-4)

Comparison of feature similarities for one-shot cross-domain recognition In Table 2, we compare our local matching score (Eq. 1) with alternative feature similarities on our two datasets. On dataset B, we use either the drawing (B-4) or our synthetic image (B-4) as reference. We always used the features trained for classification on dataset A-classification, and compare on each dataset to the similarities described in the previous paragraph (AvgPool, Concat, and LocalSim.). Our local matching score consistently outperforms all these baseline similarities. In our naive implementation, our approach is slower than direct feature comparison, but both can be mixed to obtain fast results on very large datasets.

4.3 Learning Features for Cross-Domain Recognition

We now focus on cross-domain recognition. We first compare our approach with different feature training strategies on our dataset B. We scale our watermark recognition method to the full Briquet catalog, showing we can perform classification with more than 16k classes.

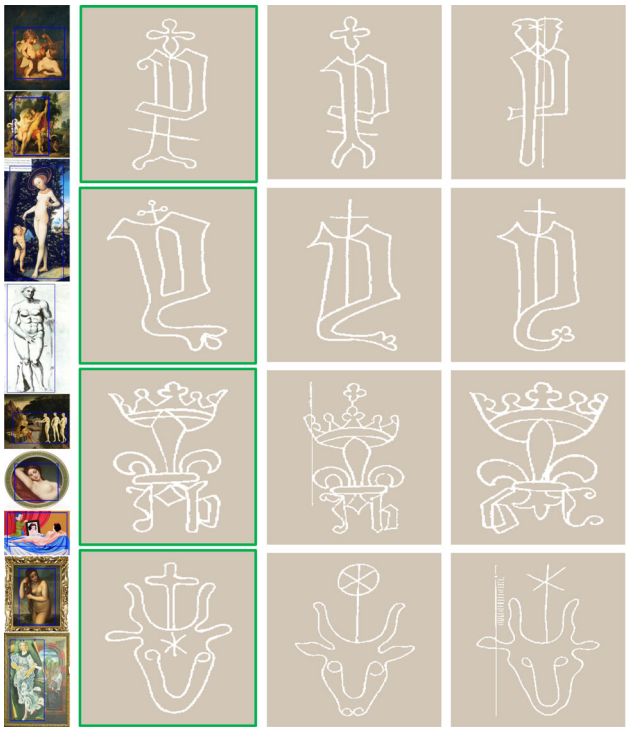
Feature fine-tuning In Table 3, we compare the results from our fine-tuning strategy to different baselines:

Table 3 Accuracy (in %) on one-shot cross-domain recognition with different feature fine-tuning approaches

Method	B-4	B-4
w/o Fine-tuning	66	72
Unsupervised (translation) (Sun et al., 2016)	63	70
Supervised (affine) (Massa et al., 2016; Rad et al., 2018)	64	72
Randomization (Su et al., 2015; Loing et al., 2018)	53	75
Triplet-loss	64	65
NC-Net (Rocco et al., 2018)	61	65
Ours (unsupervised)	60	71
Our weakly supervised fine-tuning		
$\tau = 0$	65	72
$\tau = 1/22$	75	81
$\tau = 3/22$	**75**	**83**
$\tau = 5/22$	73	77
$\tau = \text{inf}$	61	74

Best results are in bold

τ is the hyper-parameter to constraint the misalignment of matches (Sect. 3.2.3). $\tau = 3/22$ indicates we tolerates a difference of 3 features while the entire feature map is 22×22. We experiment with different reference images ("B-4" : synthetic image ; "B-4" : drawing). Note that all the approaches except NC-Net (Rocco et al., 2018) use local matching score



(a) Query **(b)** 1st Match **(c)** 2nd Match **(d)** 3rd Match

Fig. 5 Top-3 matches retrieved with our approach with synthetic reference (Fig. 4) on Briquet dataset (16,753 categories). For all the four queries, our approach is able to recognize the correct categories

– *Unsupervised (translation)* in a spirit similar to Sun et al., (2016), we translated the features in our target domain so that, on our training set, they have the same mean as the features from the target domain. We then use our score to perform nearest-neighbor classification.
– *Supervised (affine)* since we have aligned images from both domains, we can adapt features from the source and target domains in a supervised way, similarly to Massa et al. (2016 and Rad et al. 2018. We found that a simple affine adaptation gave the best results, likely because of the small size of our dataset.
– *Randomization* we also report results using randomized generated synthetic images from the drawings. The randomized synthetic image S are generated by computing $S = B + R \times (G * E)$, where B is a background sampled from photographs of paper without watermarks, G is a Gaussian filter, R is a random image and E is the binary watermark pattern extracted from the drawing. Generated images can be found in the supplementary material. We trained a standard classifier on the generated images. Such an approach has been shown to be very successful for example for 3D pose estimation (Su et al., 2015; Loing et al., 2018).
– *Triplet-loss* similarly to our method, we tried to improve the features using a triplet loss on local features, but using as positive all aligned features in the images from the same category.
– *NC-Net* (Rocco et al., 2018) while it was not initially designed for domain adaptation, we trained NC-Net on our database because of the intuition that, similarly to our method, it is able to learn to leverage spatial information. We use our pre-trained ResNet for the feature extractor and freeze it during the training. The other parts are kept the same as the category level matching model proposed in Rocco et al. (2018). The positive pairs are composed with one image from each domain, which results in 463 pairs in the training set. The training converges in 20 epochs. We then consider the sum of the scores over all correspondences as the score between a pair of images.

We detail these approaches and the training details in the supplementary material available in our project website. All results except for NC-Net are reported for matching done with our local matching score—using other metrics leads to worse performance. The effectiveness of this metric might be the reason why standard domain adaptation approaches only marginally improve performances over the baseline. Another possible reason is the small size of our training set (463 pho-

Table 4 Top-1 and top-1000 accuracy on our Briquet dataset with different models ("Briquet-4" : model trained on classification on dataset A-classification; "Briquet-4+Fine-tuning": fine-tuned model)

Method	Briquet-4		Briquet-4+Fine-tuning	
	acc.@1	acc.@1000	acc.@1	acc.@1000
AvgPool	0	16	0	21
Concat	27	77	29	82
Local Sim.	28	80	28	83
Ours N = 1000	**45**	80	54	83
Ours N =inf	44	**86**	**55**	**91**

Best results are in bold
Our approach first applies Local Similarity to obtain N top ranked references and then uses our score to re-rank the N references

tographs, 140 references). On the contrary, our fine-tuning strategy boosts performances by a clear margin.

Large-scale recognition on Briquet We finally evaluate one-shot cross-domain recognition using the test photographs of our dataset B and our full curated version of the Briquet dataset as reference. This comparison with 16,753 fine-grained classes is very challenging, but also corresponds to a realistic scenario for watermark recognition. We use our synthetic images (Fig. 4) to represent the drawings. Table 4 shows the top-1 and top-1000 accuracy using the different baseline similarities described in Sect. 4.2. Since our local matching score is computationally more expensive than the baselines, we evaluate a two-step procedure for recognition: for each test photograph, we first select the top-N candidate classes using Local Sim., then re-rank them using our local matching score. Since the local similarity—i.e. averaging the cosine distance between the local features over the images—yields the best results, we use it to perform the first step of selection. Using $N = 1000$ leads to performances similar to directly applying our method to the full 16,753 classes.

Ranking the reference drawings with Local Similarity takes approximately 3s and reranking the top-1000 with our local matching score takes 37s on a single GPU Geforce GTX 1080 T. This is acceptable for practical use, and we thus believe the application of our algorithm will be a game-changer and widen considerably the potential use of watermark analysis, which until now has been limited to a small number of experts.

Web application The ultimate goal of watermark recognition is to develop an application to simplify the search of watermarks. We developed a first version of such a web application (Bounou et al., 2020) which can be accessed at https://filigranes.inria.fr/.

Qualitative results Typical examples of top-3 matches with synthetic references (Fig. 4) are shown in Fig. 5. Although the task is challenging, as some queries are cluttered and some other categories are highly similar to the ground-truth, our approach allows to recognize the correct categories. Note that more visual results can be found in the project page.

5 Experiments on One-Shot Art Pattern Detection

In this section, we present results on one-shot art pattern detection. We first introduce the datasets we used, then present detailed results on one-shot detection on (http://www.janbrueghel.net/) dataset. We finally show qualitative results on the Venus dataset.

5.1 Datasets

Brueghel The Brueghel (http://www.janbrueghel.net/) dataset contains 1587 artworks done in different media (e.g. oil, ink, chalk, watercolour) and on different materials (e.g. paper, panel, copper), describing a wide variety of scenes (e.g. landscape, religious, still life). This dataset is especially adapted for our task since it assembles paintings from artists related to the same workshop, who thus had many interactions with each other, and includes many copies, preparatory drawings, and borrowed details.

With the help of our art history collaborators, we selected 10 of the most commonly repeated details in the dataset and annotated the visual patterns in the full dataset using the VGG Image Annotator tool (Dutta et al., 2016). The 10 annotated patterns can be seen in Fig. 6 as queries (blue boxes), and our full annotations can be found in the project page. We were careful to select diverse patterns, and for each of them to annotate only duplicates, and not full object classes. Note for example, that for the horses and lion classes, we annotated separately two variants of the details (front and back facing lion, front and right facing horse). This resulted in 273 annotated instances, with a minimum of 11 and a maximum of 57 annotations per pattern.

These annotations allow us to evaluate one-shot duplicate detection results. In our evaluation, we use an IoU threshold of 0.3 for positives, because precise annotations of the bounding boxes in different environment is difficult and approximate detection would be sufficient for most applications. In practice, our detected bounding boxes, visualised in Fig. 6 (green boxes) often appear more consistent than

Fig. 6 Detection example with our trained features on the Brueghel dataset. We show the top 5 matches (in transparency) as well as the annotations (green bounding box) for one example of query from each of our 10 annotated categories. Notice how the style of the matches can be different from the one of the query (Color figure online)

the annotations. We compute the Average Precision for each query, average them per class and report class level mean Average Precision (mAP).

Venus To demonstrate the generality of our approach, we show results on a very different database, containing 25,681 depictions of Venus (Bender, 2015). This database is much larger than the Brueghel database and more diverse since it includes depictions from many different periods and very diverse styles. Because it is thematic, it however includes many cases of revisited artwork and artistic citations. It is thus much better suited to our purpose than standard artwork databases. The size and diversity of the database makes annotation unpractical, but we report qualitative results which demonstrate matches of interest beyond the case of copied details.

5.2 One-Shot Detection

We evaluated our feature learning strategy using one-shot detection mainly on the Brueghel dataset which was small enough to be manually annotated. We performed one-shot detection simply by computing dense features on the dataset and computing their cosine similarity with the features corresponding to the query. The query was resized so its largest dimension in the feature map would be 8. Note that unlike standard deep detection approaches (Girshick et al., 2014; Ren et al., 2015), we do not use region proposals because we want to be able to match regions which do not correspond to objects.

Comparison and analysis We compare one-shot detection performance with different features using both cosine similarity and the score described in Eq. 1. In Fig. 8 we show an example of the top-6 matches for the same query using different approaches. In the first line, the matches obtained with the cosine similarity between features trained on ImageNet are all from styles similar to the style of the query, and while three of them include horses, they are not in the same pose as the query. On the contrary, the matches with our trained features, shown on the second line, mainly associate horses in the exact same position, including in depictions with different styles. Finally, the results with the same trained features but using the local matching score, shown on the third line, show a slight improvement.

Table 5 Experimental results on Brueghel, IoU > 0.3

Feature \method	Cosine similarity	Local matching
ImageNet pre-taining	58.0	54.8
Context prediction (Doersch et al., 2015)	58.8	64.3
Ours fine-tuning	**75.3**	**76.4**

The corresponding quantitative results are presented in Table 5 and confirm these observations. Indeed, learning our features improves the score by approximately 30%. The discovery procedure and score provide an additional boost.

Positive region configuration We now focus on evaluating the different positive region settings described in Sect. 3.2.3 and Fig. 3. For each of them, we analyse the performance of the features on one-shot learning on the Brueghel dataset and its evolution during training. The results can be seen in Fig. 7. Interestingly, the performance always initially

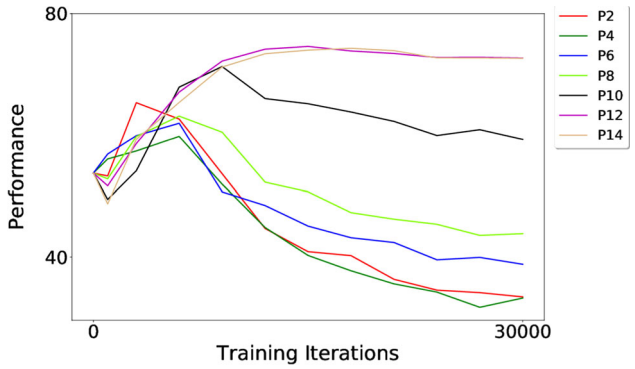


Fig. 7 Evolution of the mean Average Precision for one-shot matching on the Brueghel dataset during training. Performance decreases after a few iterations for settings where we extract positive regions correlated with the proposal region

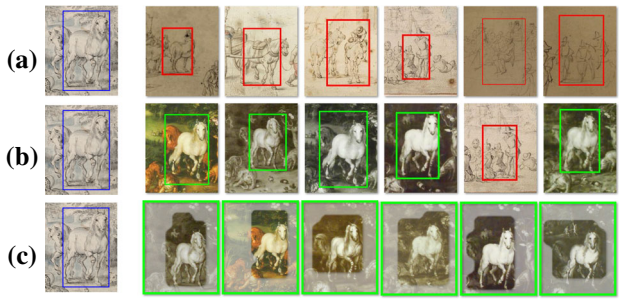


Fig. 8 From a single query, shown on the left, we show the detection results obtained with cosine similarity with ImageNet feature (**a**) and our trained features (**b**) as well as the ones (**c**) obtained with our features and the local matching score presented in Sect. 3.1

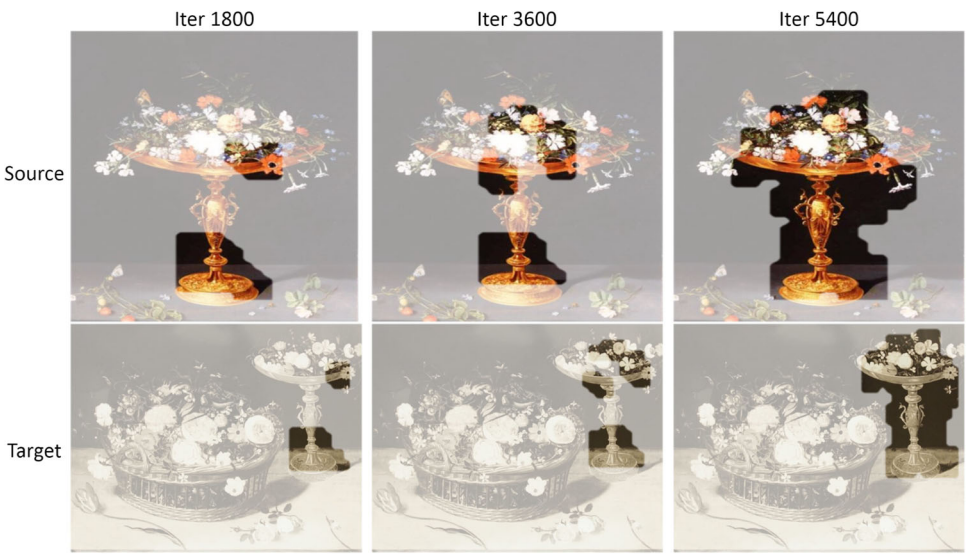
improves over ImageNet features. However, when the positive region is close to the proposal region, the performance decreases after some iterations of our training procedure and ends up with worse performance than the initial features. However, if the positive region is far enough from the query (P12 and P14), the performance improves much more and does not subsequently deteriorate. We thus use P12 for all our other experiments.

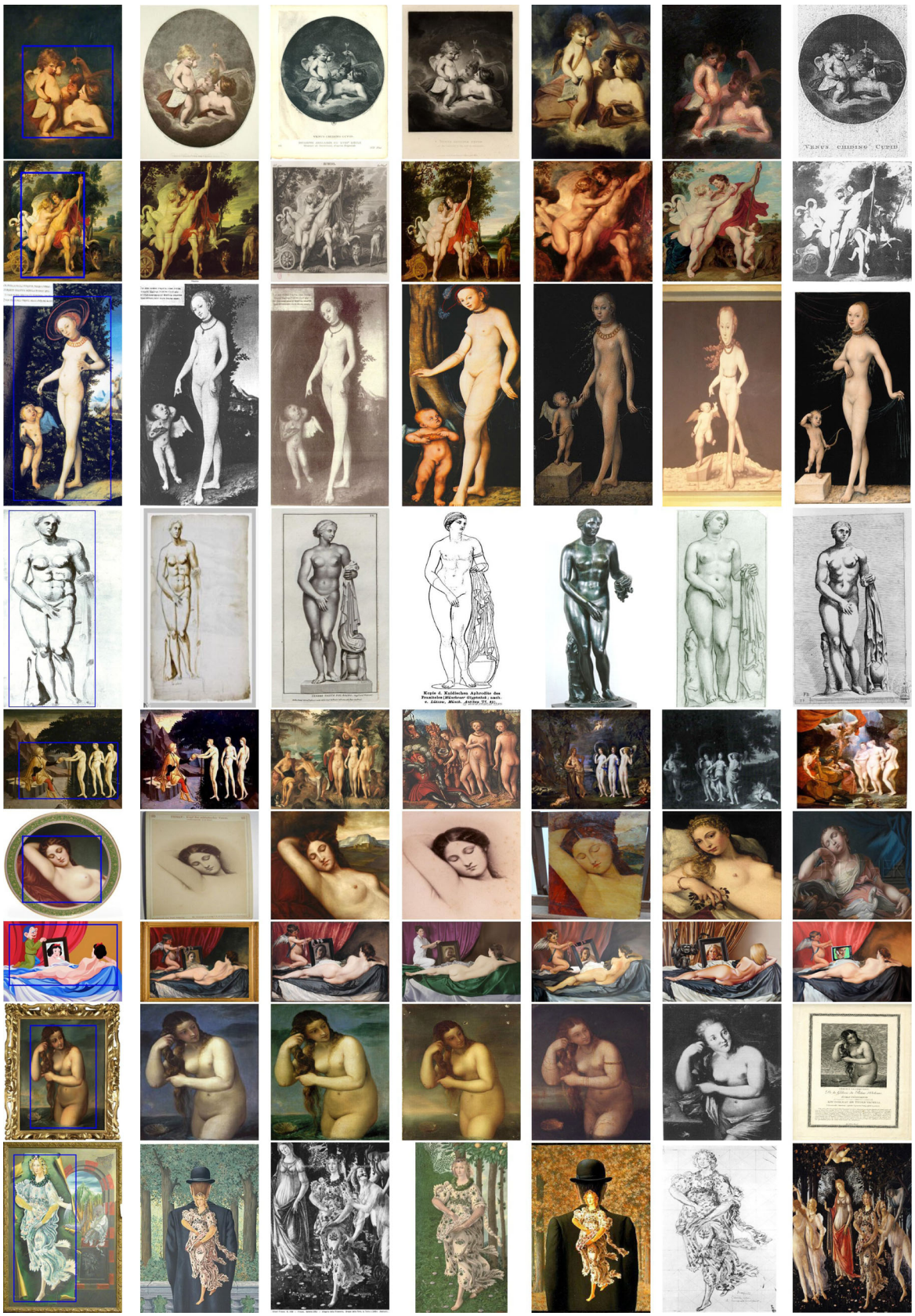
Training visualization To visualise the influence of the feature training for our discovery task, we selected a pair of matching images from the Brueghel dataset and visualised the the inlier set $\mathcal{I}$ (defined in Eq. 1) at different steps of training in Fig. 9. During training, larger and larger parts of the images can be matched in a consistent way. This shows both the efficiency of our feature training and its relevance for our task.

Qualitative results Qualitative results using our approach for each of the 10 details we annotated on the Brueghel dataset are shown in Fig. 6. It gives a sense of the difficulty of the task we target and the quality of the results we obtain. Note for example how the matches are of different styles, and how the two types of lions (top row) and the two types of horses (bottom row) are differentiated. In the following, we compare these results with baselines and analyze the differences.

In Fig. 10, we show qualitative results on the Venus dataset. In this case, the retrieved images might not be exact copies but sets of works that revisit at different periods the same model, sometime borrowing only part of the works and keeping details, such as Flora from the *Primavera* from Botticelli in the last row, sometime focusing on the composition such as for *La Venus del espejo* from Velázquez in the 7th row. Note how our method is able to retrieve relevant results in all of these cases, despite strong variations. These results showcase the interest of our method for Art Historians to explore such collections where one can hardly expect exhaustive annotations and meta-data.

Fig. 9 Visualizing matched correspondences between a pair of images during training





(a) Query **(b)** Top six detection results

Fig. 10 Each row shows the top 6 matches on the Venus database given queries outlined in blue on the left images

Limitations Our method has several limitations. First, the approach need to compare image pairs, which is not possible for fast retrieval applications. Second, in both applications, our feature learning strategy requires a good feature initialization. For watermark recognition, we start from features learned from A-classification. For artwork detection, we start from ImageNet (Deng et al., 2009) pretrained features.

6 Conclusion

In this paper, we have introduced two cultural heritage applications and associated datasets: watermark recognition and one-shot repeated art pattern detection. We have presented a unified mid-level feature training and matching method for both applications. We showed that the proposed local matching score clearly improves over standard feature similarities and that our mid-level feature fine-tuning approach can further boost performances on both tasks.

References

Aubry, M., Russell, B.C., & Sivic, J. (2014). Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (ToG)*, 33, 1–14.

Belongie, S., Malik, J., & Puzicha, J. (2001). Shape context: A new descriptor for shape matching and object recognition. In *NeurIPS*.

Bender, K. (2015). Distant viewing in art history. A case study of artistic productivity. *International Journal for Digital Art History*, 1, 100–110.

Bounou, O., Monnier, T., Pastrolin, I., Shen, X., Benevent, C., Limon-Bonnet, M.F., et al. (2020). A web application for watermark recognition. *Journal of Data Mining and Digital Humanities*.

Brendel, W., & Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*.

Briquet online. http://www.ksbm.oeaw.ac.at/_scripts/php/BR.php

Briquet, C. M. (1907). Les filigranes.

Brueghel family: Jan brueghel the elder." the brueghel family database. University of California, Berkeley. http://www.janbrueghel.net/. Accessed 2018 October 16

Castellano, G., Lella, E., & Vessio, G. (2021). Visual link retrieval and knowledge discovery in painting datasets. *Multimedia Tools and Applications*, 80, 6599–6616.

Crowley, E. J., & Zisserman, A. (2013). Of gods and goats: Weakly supervised learning of figurative art. In *BMVC*.

Crowley, E. J., & Zisserman, A. (2016). The art of detection. In *ECCV*.

Crowley, E. J., Parkhi, O. M., & Zisserman, A. (2015). Face painting: Querying art with photos. In *BMVC*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

Doersch, C., Gupta, A., & Efros, A. A. (2013). Mid-level visual element discovery as discriminative mode seeking. In *NeurIPS*.

Doersch, C., Gupta, A., & Efros, A. A. (2014). Context as supervisory signal: Discovering objects with predictable context. In *ECCV*.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *ICCV*.

Dutta, A., Gupta, A., & Zissermann, A. (2016). VGG image annotator (VIA). http://www.robots.ox.ac.uk/vgg/software/via/

Elezi, I., Vascon, S., Torcinovich, A., Pelillo, M., & Leal-Taixé, L. (2020). The group loss for deep metric learning. In *ECCV*.

Elgammal, A., Liu, B., Elhoseiny, M., & Mazzone, M. (2017). Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. arXiv

Frauenknecht, E., Stieglecker, M. (2015). Wzis – wasserzeichen-informationssystem: Verwaltung und präsentation von wasserzeichen und ihrer metadaten. Kodikologie und Paläographie im Digitalen Zeitalter 3: Codicology and Palaeography in the Digital Age 3

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *CVPR*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*.

Gidaris, S., & Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. In *CVPR*.

Ginosar, S., Haas, D., Brown, T., & Malik, J. (2014). Detecting people in cubist art. In *Workshop at ECCV*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

Gonthier, N., Gousseau, Y., Ladjal, S., & Bonfait, O. (2018). Weakly supervised object detection in artworks. arXiv

Gordo, A., Almazan, J., Revaud, J., & Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval. In *IJCV*.

Grauman, K., & Darrell, T. (2005). Pyramid match kernels: Discriminative classification with sets of image features. In *ICCV*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Hertzmann, A. (2018). Can computers create art? In *Arts*. Multidisciplinary Digital Publishing Institutes

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001). Image analogies. In *SIGGRAPH*.

Hiary, H. (2008). Paper-based watermark extraction with image processing. Ph.D. thesis.

Hiary, H., & Ng, K. (2007). A system for segmenting and extracting paper-based watermark designs. *International Journal on Digital Libraries*, 6, 351–361.

Honig, E. (2016). *Jan Brueghel and the Senses of Scale*. University Park: Pennsylvania State University Press.

imgs.ai. https://imgs.ai/

Jabri, A., Owens, A., & Efros, A. A. (2020). Space-time correspondence as a contrastive random walk. In *NeurIPS*.

Jenicek, T., & Chum, O. (2019). Linking art through human poses. In *ICDAR*.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2013). Recognizing image style. arXiv

Kim, S., Kim, D., Cho, M., & Kwak, S. (2020). Proxy anchor loss for deep metric learning. In *CVPR*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv

Loing, V., Marlet, R., & Aubry, M. (2018). Virtual training for a real application: Accurate object-robot relative localization without calibration. In *IJCV*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. In *IJCV*.

Mao, H., Cheung, M., & She, J. (2017). Deepart: Learning joint representations of visual arts. In: ACM Multimedia

Massa, F., Russell, B. C., & Aubry, M. (2016). Deep exemplar 2d–3d detection by adapting from real to rendered views. In *CVPR*.

Mensink, T., & Van Gemert, J. (2014). The rijksmuseum challenge: Museum-centered visual recognition. In *ICMR*.

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*.

Paumard, M. M., Picard, D., & Tabia, H. (2018). Jigsaw puzzle solving using local feature co-occurrences in deep neural networks. In *ICIP*.

Picard, D., Gosselin, P. H., & Gaspard, M. C. (2015). Challenges in content-based image indexing of cultural heritage collections. *IEEE Signal Processing Magazine*, 32, 95–102.

Picard, D., Henn, T., & Dietz, G. (2016). Non-negative dictionary learning for paper watermark similarity. In *ACSSC*.

Piccard, G. (1977). Die Wasserzeichenkartei Piccard im Hauptstaatsarchiv Stuttgart: Wasserzeichen Buchstabe P.

Pondenkandath, V., Alberti, M., Eichenberger, N., Ingold, R., & Liwicki, M. (2018). Identifying cross-depicted historical motifs. arXiv

Qi, H., Brown, M., Lowe, D. G. (2018). Low-shot learning with imprinted weights. In *CVPR*.

Rad, M., Oberweger, M., Lepetit, V. (2018). Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *CVPR*.

Radenović, F., Tolias, G., & Chum, O. (2016). Fine-tuning cnn image retrieval with no human annotation. In *TPAMI*.

Rauber, C., Tschudin, P., & Pun, T. (1997). Retrieval of images from a library of watermarks for ancient paper identification. In *Electronic Visualisation and the Arts*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., & Sivic, J. (2018). Neighbourhood consensus networks. In *NeurIPS*.

Said, J., & Hiary, H. (2016). Watermark location via back-lighting modelling and verso registration. *Multimedia Tools and Applications*, 75, 5673–5688.

Seguin, B., diLenardo, I., & Kaplan, F. (2017). Tracking transmission of details in paintings. In *DH*.

Shen, X., Darmon, F., Efros, A. A., & Aubry, M. (2020). Ransac-flow: Generic two-stage image alignment. In *ECCV*.

Shen, X., Efros, A. A., & Aubry, M. (2019). Discovering visual patterns in art collections with spatially-consistent feature learning. In *CVPR*.

Shen, X., Pastrolin, I., Bounou, O., Gidaris, S., Smith, M., Poncet, O., & Aubry, M. (2020). Large-scale historical watermark recognition: Dataset and a new consistency-based approach. In *ICPR*.

Shrivastava, A., Malisiewicz, T., Gupta, A., & Efros, A. A. (2011). Data-driven visual similarity for cross-domain image matching. In *SIGGRAPH ASIA*.

Singh, S., Gupta, A., & Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *ECCV*.

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV*.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *NeurIPS*.

Strezoski, G., & Worring, M. (2017). Omniart: Multi-task deep learning for artistic data analysis. arXiv

Su, H., Qi, C.R., Li, Y., & Guibas, L.J. (2015). Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*.

Sun, B., Feng, J., & Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *AAAI*.

Tan, W. R., Chan, C. S., Aguirre, H. E., & Tanaka, K. (2016). Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In *ICIP*.

Teh, E. W., DeVries, T., & Taylor, G. W. (2020). Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*.

timemachine. diamond.timemachine.eu

Úbeda, I., Saavedra, J. M., Nicolas, S., Petitjean, C., & Heutte, L. (2019). Pattern spotting in historical documents using convolutional models. In *Proceedings of the 5th international workshop on historical document imaging and processing* (pp. 60–65).

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D., et al. (2016). Matching networks for one shot learning. In *NeurIPS*.

Wallraven, C., Caputo, B., & Graf, A. (2003). Recognition with local features: The kernel recipe. In *ICCV*.

Wang, X., Jabri, A., & Efros, A. A. (2019). Learning correspondence from the cycle-consistency of time. In *CVPR*.

Westlake, N., Cai, H., & Hall, P. (2016). Detecting people in artwork with cnns. In *ECCV*.

Wilber, M. J., Fang, C., Jin, H., Hertzmann, A., Collomosse, J., & Belongie, S. J. (2017). Bam! the behance artistic media dataset for recognition beyond photography. In *ICCV*.

Yin, R., Monson, E., Honig, E., Daubechies, I., & Maggioni, M. (2016). Object recognition in art drawings: Transfer of a neural network. In *ICASSP*.

Zhang, J., Marszałek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. In *ICCV*.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.