# Object Retrieval and Localization in Large Art Collections using Deep Multi-Style Feature Fusion and Iterative Voting

Nikolai Ufer, Sabine Lang, and Björn Ommer

Heidelberg University, HCI / IWR, Germany
`nikolai.ufer@iwr.uni-heidelberg.de, ommer@uni-heidelberg.de`

**Abstract.** The search for specific objects or motifs is essential to art history as both assist in decoding the meaning of artworks. Digitization has produced large art collections, but manual methods prove to be insufficient to analyze them. In the following, we introduce an algorithm that allows users to search for image regions containing specific motifs or objects and find similar regions in an extensive dataset, helping art historians to analyze large digitized art collections. Computer vision has presented efficient methods for visual instance retrieval across photographs. However, applied to art collections, they reveal severe deficiencies because of diverse motifs and massive domain shifts induced by differences in techniques, materials, and styles. In this paper, we present a multi-style feature fusion approach that successfully reduces the domain gap and improves retrieval results without labelled data or curated image collections. Our region-based voting with GPU-accelerated approximate nearest-neighbour search [29] allows us to find and localize even small motifs within an extensive dataset in a few seconds. We obtain state-of-the-art results on the Brueghel dataset [2, 53] and demonstrate its generalization to inhomogeneous collections with a large number of distractors.

**Keywords:** Visual retrieval; Searching art collections; Feature fusion

## 1 Introduction

For art history, it is crucial to analyze the relationship between artworks to understand individual works, their reception process, and to find connections between them and the artists [28, 22, 23]. Hereby, the investigation of motifs and objects across different images is of particular importance since it allows more detailed analyses and is essential for iconographic questions. Digitization has produced large image corpora [4, 1, 3, 59], but manual methods are inadequate to analyze them since this would take days or even months. Computer-assisted approaches can dramatically accelerate and simplify this work. However, most of them consist of a simple text search through metadata and are not sufficient since text cannot capture the visual variety, and labels are often either

missing, incomplete, or not standardized. Therefore, there is a need for efficient algorithms, capable of searching through visual art collections not only based on textual metadata but also directly through visual queries. In this paper, we present a novel search algorithm that allows users to select image regions containing specific motifs or objects and find similar regions in an extensive image dataset.

While computer vision successfully developed deep learning-based approaches for visual instance retrieval in photographs [42, 43], artworks present new challenges. This includes a domain shift from real photos to artworks, unique and unknown search motifs, and a large variation within art collections due to different digitization processes, artistic media, and styles. This strongly highlights the need for specifically tailored algorithms for the arts [50, 53]. Solving visual instance retrieval across artworks is difficult and requires local descriptors, which are both highly discriminative to find matching regions and invariant regarding typical variations in art collections. Learning such descriptors in a supervised fashion is extremely time-consuming and requires annotating thousands of corresponding images [51]. Besides, the learned descriptors improve retrieval results only for very similar datasets. Alternative approaches based on self-supervision [53] show promising results. However, they are not stable against images without any repetitions in the dataset, and are very slow in large-scale scenarios since they require a pairwise comparison between all images. We circumvent these issues and present a new multi-style feature fusion, where we utilize generic pre-trained features and current style transfer models to improve their style invariance without any additional supervision. Given a dataset, we stylize all images according to a set of fixed style templates, and by mixing their feature representations, we project them into the same averaged style domain. This massively reduces the domain gap across artworks and improves overall retrieval results.

**Contributions.** Our main contributions are threefold. (1) We present an unsupervised multi-style feature fusion, which successfully reduces the domain gap and improves overall retrieval results in art collections. (2) The introduced iterative voting in combination with GPU-accelerated nearest-neighbour search [29] enables us to localize and find small regions in large datasets within a few seconds. (3) We demonstrate that the proposed method significantly outperforms current methods in terms of retrieval time and accuracy for object retrieval and localization in art collections.

## 2   Related work

In the following, we present the most relevant research related to our work and put our contributions in context.

**Computer vision in the arts.** For quite some time, there has been a mutual exchange between computer vision and the arts. This exchange ranges from the analysis of artworks using computer vision [6, 9, 51, 18] to the development of new methods in collaboration with the art community [48, 65, 60, 54] to generative algorithms that transfer normal photos into artworks [21, 16,

47] or direct attempts to create art [13, 20]. Concerning the analysis of art-works, collaborations are very natural because computer vision and art history are both concerned with the visuality and ask similar questions. In this context, researchers have transformed successful object detection and classification methods for photos to artworks [48, 9, 30, 58]. Some of these works detect gestures, people, or iconographic elements in paintings [48, 17, 67, 70, 18, 26], recognize object categories occurring in natural images [9, 10, 68], classify paintings in terms of their style, genre, material, or artist [30, 36, 46, 45, 61, 68, 59], or investigate the aesthetics of paintings [58, 5, 12]. Some approaches directly try to find visual relationships within art collections automatically [52, 53, 15] to relieve as much work as possible from art historians. However, these are very time-consuming since they compare all possible image pairs within the dataset. Therefore, they are limited to small collections, which could also be manually analyzed. In this work, we focus on large-scale instance retrieval and localization in the arts. We are convinced, with an efficient search system, art historians can find relevant visual links through several searches faster and more targeted compared to fully automated approaches.

**Visual instance retrieval.** Visual instance retrieval deals with the task of identifying matching regions in other images within a dataset, given a query region or image. This is a well-established research field in computer vision with successful classical [24, 9], as well as deep learning-based [62, 69, 38] approaches. Early methods were based on classical feature point descriptors like SIFT [33] combined with a Bag-of-Words approach [24, 9]. Over the years, numerous improvements have been made for different parts of this approach [55]. More recently, Convolutional Neural Networks (CNN) showed remarkable results in many areas of computer vision, including visual instance retrieval [7]. However, the primary retrieval research was always focused on photographs, either of the same place [39, 40] or the same object [37]. If researchers dealt with art databases, then often only on an image level without allowing to search for regions [56, 60, 41, 8, 35], which is a central requirement to find similar motifs and objects in art collections. Just recently, Shen et al. [53] introduced the first benchmark with annotations for finding and localizing objects and motifs in artworks on a region level, which is also the primary dataset we use for our evaluation. Most closely related to our work is Shen et al. [53] and Seguin et al. [50], which also deal with instance retrieval and finding visual relationships in art collections. In contrast to our work, Seguin et al. [50] use off-the-shelf CNNs that have been fine-tuned in a supervised fashion and thus does not apply to other datasets. The approach of Shen et al. [53] learns dataset-dependent features in a self-supervised manner by mining correspondences between image pairs, similar to [63, 64]. However, their approach does not generalize to inhomogeneous collections with many distractors, and their retrieval system is intractable in large-scale scenarios. In contrast, our method improves retrieval results by successfully reducing the domain gap without labelled data or curated image collections and enables us to find even small motifs within an extensive dataset in a few seconds.
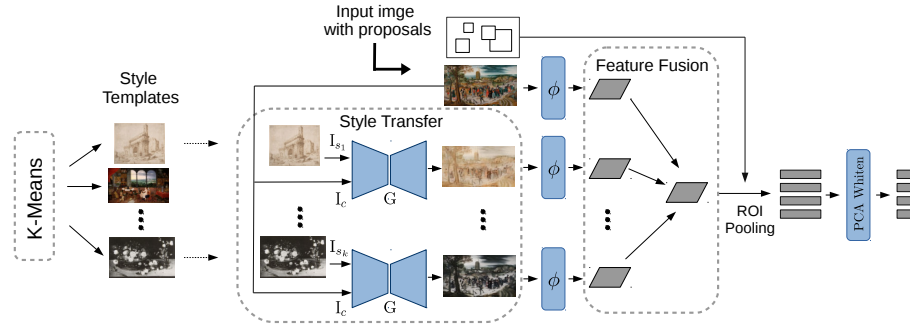
**Fig. 1.** Overview of our multi-style feature fusion. It consists of three main steps: First, we extract image level features of all dataset images, apply K-means and select the cluster centers as style templates. Given an input image and region proposals, the input image is stylized based on the style templates and features are generated using the pre-trained feature extraction and style transfer network. The image features are fused to create a final image representation, and through ROI pooling, we obtain region descriptors for all proposals with a fixed dimension. Finally, we apply principle-component analysis (PCA) and whitening to reduce the feature dimension.

## Method

The following requirements on the image representation and the retrieval system are essential. First, it should be possible to search for any motif in a diverse art collection. This poses particular challenges to the underlying feature descriptors since off-the-shelf models are trained on photographs and are not invariant to colours and artistic styles. We address this problem in the first part of this section and introduce a new multi-style feature fusion, which considerably reduces the domain gap. Second, it should be possible to search for any image region with an arbitrary size across a large dataset, and the search should take only a few seconds and deliver exact retrieval results. These are challenging demands since a single feature descriptor is not capable of capturing multiple objects or motifs in an image accurately, and encoding all regions is not tractable due to memory constraints. We address this problem in the second part of this section. For the query region and all images in the dataset, we extract a moderate number of local descriptors and formulate the search as a voting procedure of local patches within the query region.

### Multi-style feature fusion

Our multi-style feature fusion is based on the following hypothesis. The dataset contains images or regions that are similar to each other but are depicted in different styles. Generic pre-trained descriptors are capable of finding similar regions if the style differences are small. Based on this assumption, our main idea is to use current style transfer models to project all images into the same

averaged style domain to reduce the domain gap and simplify the retrieval task. Therefore, we stylize each image based on multiple fixed style instances and fuse their extracted features to generate a single robust representation. The approach consists of three main steps, which we describe in the following. See Fig. 1 for an overview.

**Finding style template.** To find a diverse set of style templates, we proceed as follows. We denote the collection of images as $\mathcal{I} = \{I_i | 1 \leq i \leq N\}$ and assume that an ImageNet pre-trained CNN, which maps images into an $m$-dimensional deep feature space, i.e. $\phi \colon \mathcal{X} \to \mathbb{R}^m$, is given. We group all images into $k_s$ clusters according to their pairwise distance in the embedding space using K-means. Since the CNN $\phi$ is trained on real photos, it is not invariant with respect to styles, and hence different clusters contain different content depicted in different artistic styles. We select $k_s$ images which are closest to the cluster centers in the feature space and obtain a diverse set, which serves as our style templates $\mathcal{S} = \{I_s | 1 \leq s \leq k_s\}$. This selection is sufficient because our style transfer method is independent of the image content, and only the depicted style is important. In our experiments, we set $k_s = 3$ since this has proven to be a good trade-off between performance and computational cost.

**Style transfer.** Our style transfer module is a CNN, $G \colon \mathcal{X} \times \mathcal{X} \to \mathcal{X}$, which takes a content image $I_c$ and style image $I_s$ as input to synthesize a new image $G(I_c, I_s)$ with the content from the former and style from the latter. It is based on the network architecture of Li et al. [31] and consists of three main parts: an encoder-decoder, a transformation, and a loss module. The encoder consists of the first layers of the ImageNet pre-trained VGG-19 model [11, 57] and the decoder comprises of its symmetrical counterpart. The transformation module consists of two small CNNs which receive the encoded feature maps of the content and style image as input, and provide a transformation matrix as output, respectively. Given a content and style image, the style is transferred by multiplying the encoder's content feature with the two transformation matrices and applying the decoder on the output, which produces the stylized image. Since the network is a pure feed-forward convolution neural network, it allows converting an image into an arbitrary style in milliseconds.

**Feature fusion.** Based on the style transfer module $G$ and style templates $\mathcal{S}$, we obtain the multi-style feature representation as follows. Given an input image $I$ and a set of proposals, we stylize the image with respect to all style templates $\mathcal{S}$, and fuse them by taking their mean, i.e.

$$\phi_{ms}(I, \mathcal{S}) = \frac{1}{1 + |\mathcal{S}|} \left( \phi(I) + \sum_{I_s \in \mathcal{S}} \phi(G(I, I_s)) \right). \tag{1}$$

Here we also take the original image feature into account since it contains fine-grained information that can be useful for the retrieval task but are lost during the transformation process. Given the proposals and the new image representation, we apply Precise ROI Pooling [27] to obtain local feature descriptors for all proposals with a fixed feature dimension. Finally, we apply principle-component analysis (PCA) and whitening to reduce their feature dimension.
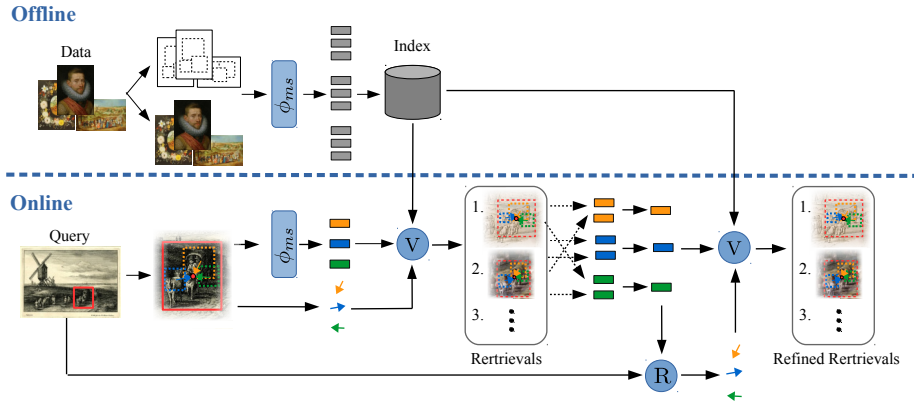
**Fig. 2.** Overview of our retrieval system. The retrieval system consists of an offline and an online stage. During the offline stage, the search index is initialized before the actual search. For each image, discriminative local patches are determined, and their feature descriptors are extracted, compressed, and stored in the search index. During the online stage, most discriminative local patch descriptors within the marked query region are extracted, and their k-nearest neighbours are determined using the search index. Our voting procedure aggregates these local matches and predicts well-localized retrieval bounding boxes for the whole query. The results are further improved using local query expansion and re-voting.

### Retrieval system using iterative voting

To find and localize objects and motifs of arbitrary size in an extensive art collection, we introduce an iterative voting approach based on local patch descriptors. Therefore, we decompose all images into a set of quadratic patches on multiple scales in a sliding-window manner, which are encoded by the multi-style feature extraction network described previously. The actual search consists of finding k-nearest neighbours of local patches within the selected query region across the whole dataset, which are afterwards aggregated to well-localized retrieval bounding boxes. This approach has three main advantages. First, the search is performed on a region level and enables searching for small motifs that cannot be found when images are represented with a single feature descriptor. Second, it allows us to predict well-localized retrieval bounding boxes independent of the local patches we use for the image encoding. Third, we combine several search queries in combination with a spatial verification, which provides better search results.

Our retrieval system consists of an offline and online stage. In the offline stage, the search index is initialized. Therefore, local image descriptors are extracted, compressed, and stored in the index. This step only needs to be done once. In the online stage, most discriminative local feature descriptors within the query region are extracted, and our voting procedure is applied to their k-nearest neighbours to obtain the first search results. Through local query expansion and re-voting,

the retrievals results are further improved. In the following, we describe each step in more details. We also provide an overview in Fig. 2.

**Image encoding using local patch descriptors.** In our experiments classic region proposal algorithms [34, 66, 71] and networks [44] showed insufficient results for artworks. Our strategy for finding suitable regions to extract local patch descriptors consists of two steps. First, we generate a broad set of local patches by dividing the space into quadratic regions on multiple scales in a sliding window manner. Second, we filter them for selecting the most discriminative as follows. We sample a subset of all local patches and assign them into $k_e$ groups using the $L_2$ distance in the multi-style feature space and K-means. We store the $k_e$ cluster centers, compute the distance of all local patches to these centers, and select those with the highest mean distance. By this, we obtain local patches on different scales that are more discriminative and more suitable for the retrieval task than others. We generate a maximum of 4000 local patches for each image $I$ in this way and extract their multi-style feature descriptors $\mathcal{D}(I)$ as described previously. To keep the search index compact, we reduce the number of extracted proposals per image linearly with the dataset size.

**Search index.** We build a search index containing all local patch descriptors of all images $\mathcal{D}$ for fast approximate nearest neighbor search. For the search index, we use the Inverted File Index (IVF) and the Product Quantization (PQ) algorithm from [29]. They provide a high-speed GPU-parallelized variant that allows to search for multiple queries in an extensive database in seconds. The IVF algorithm [25] clusters the feature vectors into groups and calculates their centroids using K-means. Now, given a query vector, the distances to all centroids are determined, and only the feature vectors assigned to the closest centroids are considered for the k-nearest neighbor search, which massively accelerates the search. The actual search uses product quantization. Here the feature vectors are sliced into subvectors, and a codebook for each of these slices is learned. Based on these codebooks, the feature vectors can be stored efficiently using their ids, and a look-up table containing all codebook centroids distances allows a fast approximated nearest neighbor search with a query vector. We calculate the IVF clustering and PQ codebooks for each dataset separately. Therefore, we take all regions from 1000 randomly selected images and train on their multi-style features.

**Query reformulation using local patch descriptors.** In the online stage, the user selects an image and marks a rectangle $q$ as the query region. Besides the selected region itself, we additionally use the most discriminative local patches within the query region for the actual search. We find these local query patches as follows. We extract local patches on multiple scales and select the most discriminative using two criteria. First, we filter all local patches with less than 90 percent overlap with the query region. Second, we apply non-maximum suppression concerning their feature activation, which we obtain by summing over the feature channel and taking the mean within the proposal region. For the selected patches, we extract their features and store their voting vectors. This set of local

query patch descriptors $\mathcal{D}(q)$ for a given query region $q$ are used in the following for our voting strategy.

**Voting based on local matches.** As described previously, we encode each dataset image $I \in \mathcal{I}$ with a set of local patch descriptors $\mathcal{D}(I)$, where we denote the set of all descriptors in the dataset by $\mathcal{D}$. For a given query rectangle $q$, we also extract a set of local patch descriptor $\mathcal{D}(q)$. In the following, we do not distinguish between the local patches itself and their feature descriptors, but it should always be clear what is meant from the context. Our voting consist of the following two main steps.

In the first step, we determine the k-nearest neighbours $NN_k(f, \mathcal{D})$ for each local query patch $f \in \mathcal{D}(q)$ using the search index with $L_2$ distances, where we denote a local query patch with one of its k-nearest neighbours as a local match. Based on the $L_2$ distances, we define a local matching score via

$$s_f(g) = \exp\big(-\|g - f\|_2^2 \,/\, \|\hat{g} - f\|_2^2\big), \tag{2}$$

where $\hat{g} \in NN_k(f, \mathcal{D})$ with a fixed rank and provides a reference distance. For the image ranking, we utilize a majority based voting, where we determine the most promising images. To do this, for a given image $I$, we look at its local matches and sum their local matching scores, where we consider at most one hit for each local query feature. We select the images with the highest scores and restrict the following voting based on local matches on this subset. Due to this pre-selection, the computational costs of the following steps are independent of the dataset size. However, in contrast to other approaches [49], the pre-selection is conducted on a part, not image level, and hence we do not miss similar small regions in the final search results.

In the second step, we apply our voting scheme on the local matches of the most promising images to predict well-localized retrieval bounding boxes. For the voting, we assume persistent aspect ratios and neglect object rotations to reduce the voting space and accelerate the search. Let us consider a local match $(f, g)$, i.e. $g \in NN_k(f, \mathcal{D}) \cap \mathcal{D}(I)$ for a local query patch $f \in \mathcal{D}(q)$, then this match is voting for a specific location and scale of a rectangle $r$ in image $I$. If we denote $\mathbf{v}(f) = \mathbf{c}_q - \mathbf{c}_f$ as the vector from the center of the local query patch $\mathbf{c}_f$ to the query rectangle $\mathbf{c}_q$ and $d_f$, $d_g$ and $d_q$ are the diagonal lengths of $f$, $g$ and $q$, respectively. Then this match votes for a rectangle with center $\mathbf{c}_r = \mathbf{c}_g + \mathbf{v}(f) \cdot d_g/d_f$ and diagonal $d_d = d_q \cdot d_g/d_f$. We aggregate these votes and create a voting map in which each point votes for the center of a box at the corresponding position similar to [55]. To keep the voting map compact, we quantize the image space so that the voting map is much smaller than the actual image. Here, we reduce quantization errors by voting for a $5 \times 5$ window with a Gaussian kernel for each local match. For the voting score, we use the local matching similarity defined in Equ. 2. To find the correct diagonal lengths for the retrievals, we do not increase the voting space by an additional dimension but average the diagonals of all votes pointing to the same center. This keeps the voting space compact and accelerates the search. For determining the center and diagonal of the best retrieval box in the image $I$, we take the position of

the maximum in the voting map and the corresponding averaged diagonal. This voting approach allows object retrieval and localization at the same time without an exhaustive sliding-window search as a post-processing step, like in [50, 53].

**Local query expansion and re-voting.** After finding our first retrievals across the dataset, we improve our search results using local query expansion and re-voting. Concerning the local query expansion, we consider the first ten nearest neighbours in different images for each local query patch and fuse their feature descriptors by taking their mean and $L_2$ normalization over the feature channels. By this, we obtain new and more generalized local patch descriptors, containing more diverse information by including multiple instances. The aggregated patches no longer have the same coordinates in the query image because of the combination of local patch descriptors and possible shifts in local matches. Therefore, we also update the voting vectors based on the new patch representations. We do this by determining their nearest neighbours in the query image and measuring the voting vector to the query bounding box center. The generalized local patch queries and updated voting vectors have the same structure as in the first voting stage, and we can apply the previously described voting procedure again, which leads to better search results.

### Implementation details

Concerning the multi-style feature fusion, we use VGG16 with batch normalization [57], pre-trained on ImageNet [11], and truncated after the fourth layers' RELU as backbone architecture. We rescale images to 640 pixels concerning the smallest image side and pad them by 20 pixels on each side. We add a max-pooling layer and hence obtain a ratio of 16 between image and feature space. For the style transfer model, we use the pre-trained model of Li et al. [31], which was trained on the MS-COCO [32] dataset. We also experimented with training on each target dataset separately, but this did not improve the performance and heavily increased the initialization time. We generate the stylized images during the offline stage so they only need to be loaded for the feature extraction.

Concerning the iterative voting, our proposal algorithm uses six different scales with a scaling factor of $2^{-1/2}$. The patch size ranges from $1/12$ up to $1/2$, where we use a stride of $1/50$ regarding the largest image side. For the selection of discriminative patches, we set $k_e = 200$. To keep the search index compact, we linearly decrease the number of proposals per image with increasing dataset size. First, we extract 4000 proposals and reduce the number by 375 for each additional 20k images, starting with a dataset size of 20k images. The OPQ algorithm utilizes 96 sub-quantizers with 8 bits allocated for each sub-quantizer. For the IVF algorithm, we generate 1024 clusters and use 30 for the nearest neighbor search.

### Experiments

In this section, we present comparative evaluations on challenging benchmark datasets and diagnostic experiments.

## Datasets

To show the efficiency and generalization capability of our system, we evaluate on five different benchmark datasets.

**Brueghel.** Our main evaluation is based on a collection of Brueghel paintings [2] with annotations from Shen et al. [53]. To the best of our knowledge, no other dataset for instance retrieval and localization in the arts with annotations is available at the moment. The dataset consists of 1,587 paintings, including a variety of different techniques, materials, and depicted scenes. It includes ten annotated motifs with 11 up to 57 instances of each motif, which results in 273 annotations overall. We follow the evaluation protocol of [53], and count retrievals as correct, if the intersection over union (IoU) of predicted with ground-truth bounding boxes is larger than 0.3. For each query, we compute the Average Precision (AP), average these values per class, and report the class level mean Average Precision (mAP).

**Brueghel5K and Brueghel101K.** We are particularly interested in the large-scale scenario, where the algorithm has to deal with an extensive and inhomogenous image collection. This is a much more common use case for art historians. For this purpose, we introduce the Brueghel5K and Brueghel101K dataset, where we extend the previously described dataset with an additional 3,500 and 100,000 randomly selected images from the Wikiart dataset [4] as distractors, respectively. To avoid false negatives, we used the annotations from Wikiart and excluded all Brueghel paintings from the selection. The evaluation uses the same annotations and evaluation protocol as the Brueghel dataset.

**LTLL.** We also evaluate our algorithm on the Large Time Lags Location (LTLL) dataset, which was collected by Fernando et al. [14]. It consists of historical and current photos of 25 cities and towns spanning over a range of more than 150 years. The main goal is to recognize the location of an old image using annotated modern photographs, where the old and new images can be considered as belonging to two different domains. In total, the dataset contains 225 historical and 275 modern images. Since our retrieval system assumes that users mark image regions he is interested in, we provide and utilize additional query bounding boxes for our and all baseline models. Analogously to the evaluation protocol [14] we report the accuracy of the first retrieval.

**Oxford5K.** We also evaluate our approach on the Oxford5K datasets, which was collected by Philbin et al. [39]. It consists of 5,062 photos with 11 different landmarks from Oxford and five different query regions for each location. The occurrence of each landmark ranges from 7 up to 220. We follow the evaluation protocol of [39] and compute the Average Precision (AP) for each query, average them per landmark and report the mean Average Precision (mAP).

## Effect of multi-style feature fusion

To validate our multi-style feature fusion, we compare the performance of our algorithm with different feature representations. As baselines, we use VGG16 features truncated after the conv-4 layer, which are either pre-trained on ImageNet

**Table 1.** Retrieval performance comparison of our multi-style feature fusion (Ours) and features generated from VGG16, which are either pre-trained on ImageNet [11] (ImageNet pre-training) or additionally fine-tuned using [53] (Artminer)

| Features | Brueghel | Brueghel5K | Brueghel101K | LTLL | Oxford5K |
|---|---|---|---|---|---|
| ImageNet pre-training | 79.1 | 76.7 | 67.3 | 88.1 | 87.9 |
| Artminer [53] | 80.6 | 37.9 | 34.5 | 89.0 | 79.4 |
| Ours | **85.7** | **84.1** | **76.9** | **90.9** | **89.8** |

**Table 2.** Ablation study of our voting approach. We measure the performance for searching only with the selected query region (wo/voting), and for restricting ourselves to the first round of voting (wo/it.voting) and our full system. We also report the performance on the Brueghel dataset for different IoU thresholds

| Methods | Brueghel | | | LTLL | Oxford5K |
|---|---|---|---|---|---|
| | IoU@0.3 | IoU@0.5 | IoU@0.7 | | |
| Ours wo/voting | 72.3 | 48.5 | 5.9 | 73.2 | 72.7 |
| Ours wo/it.voting | 74.1 | 54.4 | 19.1 | 90.4 | 87.8 |
| Ours | **85.7** | **63.3** | **21.7** | **90.9** | **89.8** |

[11] (ImageNet pre-training) or additionally fined-tuned with the self-supervised approach of Shen et al. [53] (Artminer).

From Tab. 1 it can be seen that our approach improves the results on all benchmarks. The improvement compared to pre-trained features is especially high for the art datasets since there is a particular large domain gap due to differences in styles between queries and targets. Even for datasets without any domain shift, like Oxford5K, our algorithm improves the search results due to the aggregation of diverse feature representations. We also achieve significantly better results compared to the Artminer [53] fine-tuned variants. Their method has particular problems for image collections containing many distractors as well as on the Oxford5K dataset.

**Effect of iterative voting**

We analyze the impact of our voting procedure on the visual search. For this purpose, we measure the performance of our method for searching only with the selected query region (wo/voting), restricting ourselves to the first round of voting (wo/it.voting) and our full system. To better understand the impact on the localization of retrievals, we also report the performance for different IoU thresholds on the Brueghel dataset.

The results are summarized in Tab. 2. We see that the first round of voting especially improves the results on the LTLL and the Oxford5K dataset, the performance gain on the Brueghel dataset for the IoU of 0.3 is smaller. The reason is that the query regions for the Brueghel dataset are, on average, much smaller. Therefore, there are fewer voting regions, and the effect of voting becreases. However, the results for higher IoU thresholds show that the localization

**Table 3.** Comparison of the retrieval time of our method (Ours) and [53] (Artminer) for different dataset sizes, where we also report the size of our search index (last row)

| Method | 5K | 20K | 40K | 60K | 80K | 100K |
|---|---|---|---|---|---|---|
| Artminer [53] | 12.9 min | 50.4 min | 1.7 h | 2.8 h | 3.8 h | 4.6 h |
| Ours | 8.5 s | 9.0 s | 9.3 s | 9.7 s | 10.1 s | 10.5 s |
| Ours | 1.7 GB | 6.8 GB | 13.7 GB | 20.6 GB | 27.4 GB | 34.2 GB |

**Table 4.** Retrieval results of our method and state-of-the-art methods on the Brueghel [2, 53], Brueghel5K, Brueghel101K, LTLL [14] and Oxford5K [39] dataset. We also report the underlying network architecture (Net), and what dimension the underlying features have (Dim)

| Methods | Net | Dim | Brueghel | | | LTLL | Oxford |
|---|---|---|---|---|---|---|---|
| | | | [2, 53] | 5K | 101K | | |
| ImageNet, image level | VGG16 | 512 | 24.0 | 22.5 | 17.7 | 47.8 | 25.6 |
| Radenović et al. [43], wo/ft | VGG16 | 512 | 15.5 | 12.7 | 5.9 | 59.3 | 53.4 |
| Radenović et al. [43] | VGG16 | 512 | 15.8 | 12.8 | 5.7 | 76.1 | 87.8 |
| Artminer [53], wo/ft | ResNet18 | 256 | 58.1 | 56.0 | 50.2 | 78.9 | 84.9 |
| Artminer [53] | ResNet18 | 256 | 76.4 | 46.5 | 37.4 | 88.5 | 85.7 |
| Artminer [53], wo/ft | VGG16 | 512 | 54.4 | 50.5 | 44.1 | 81.8 | 85.0 |
| Artminer [53] | VGG16 | 512 | 79.9 | 39.5 | 36.4 | 88.9 | 81.5 |
| Ours | VGG16 | 96 | 85.7 | 83.9 | **76.9** | 90.9 | **89.9** |
| Ours | VGG16 | 128 | 87.2 | 85.6 | —- | 90.9 | 89.8 |
| Ours | VGG16 | 256 | **88.1** | **86.7** | —- | **91.3** | 89.8 |

is significantly improving. The second round of voting with local query expansion improves the results on all datasets further. This has a particularly strong influence on the retrieval results for the Brueghel dataset.

**Computational cost**

We also investigate the search speed of our Python implementation. All measurements are conducted on the same machine with 3 GPUs (Nvidia Quadro P5000). In Tab. 3, we summarize the results and compare them with [53], where we assumed a perfect implementation of [53] on multiple GPUs by dividing their search times by a factor of 3. Besides the search speed, we also report the index size after storing to disk.  It shows that our method is much faster, and its speed depends only moderately on the number of images. The index size mainly determines the size of the image collection that can be searched. Its size primarily depends on the number of proposals extracted per image, which is an important factor in finding small regions. Since the required retrieval accuracy for small regions, available hardware, and dataset size vary from application to application, the number of proposals should be adjusted according to the actual use case.
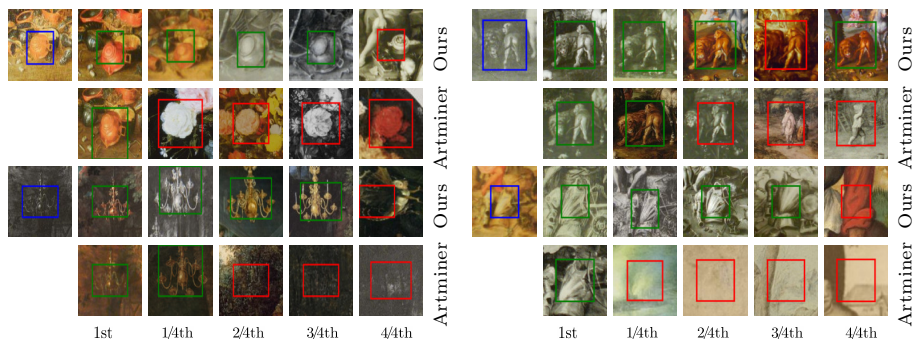
**Fig. 3.** Qualitative comparison. Retrieval examples of our approach (Ours) and [53] (Artminer) on the Brueghel dataset. We show queries on the left in blue and its nearest-neighbor, as well as four additional retrievals with an equidistant distance given by the number of ground truth annotations for the query divided by four. We draw green bounding boxes if the intersection over union is larger than 0.3 and red otherwise.

### Retrieval performance on benchmarks

In the following, we give quantitative and qualitative results on the previously introduced benchmark datasets.

**Quantitative evaluation.** We compare our results with max-pooled pre-trained features on image level (ImageNet, image level) as well as the state-of-the-art results of Shen et al. [53] (Artminer) and Radenović et al. [43] on Brueghel, LTLL, and Oxford5K. For a fair comparison, we select the same backbone architecture for all methods and report also the original numbers from [53] with their fine-tuned ResNet18 model. Our approach utilizes marked query regions within the image. This is not the case for the discovery mode of [53]. According to their publication and our experiments, they obtain the best results using full images as query, which allows their algorithm to utilize more context. The reported numbers refer to their discovery mode.

We summarize all results in Tab 4. We outperform all methods on all benchmark datasets without fine-tuning on the retrieval task and with a much smaller feature dimension. The results on the Brueghel datasets with additional distractors show that the self-supervised method of Shen et al. [53] is not stable against images without corresponding regions in the dataset. The main reason is that, the probability of selecting regions without correspondences is very high, which results in few and potentially spurious matches for training. This effect can already be seen for Brueghel5K, where their fine-tuned network leads to worse retrieval results compared to their initial model. In contrast, our method is much more robust against such distractors. Furthermore, it can be seen that we even outperform [43] on the Oxford5K dataset, although, their approach is explicitly designed for geo-localization by fine-tuning on an extensive image collection of various landmarks using ResNet101 [19]. However, since their model is optimized for this task, their search results on art collections are rather weak.
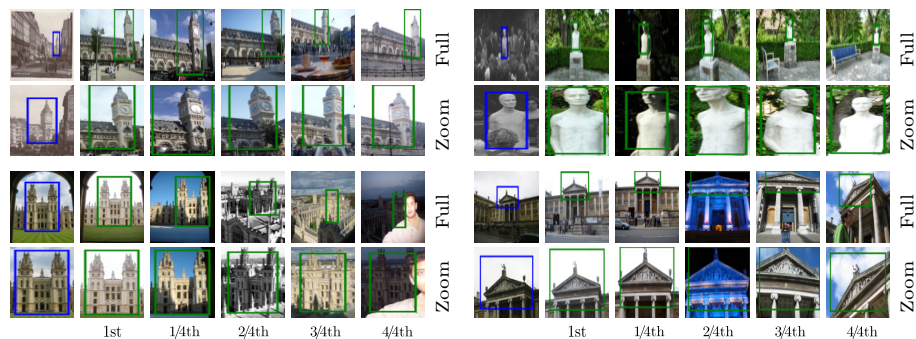
**Fig. 4.** Retrieval examples. We show examples in rows 1-2 and 3-4 for the LTLL and Oxford5K dataset. The first row shows full images (Full) and the second zoomed-in versions (Zoom). The queries are visualized in blue on the left and its nearest-neighbor, as well as four additional retrievals with an equidistant distance, given by the number of ground truth annotations for the query divided by four on the right.

**Qualitative evaluation.** In Fig. 3, we provide a qualitative comparison with the state-of-the-art of Shen et al. [53] (Artminer). It shows that their method can find first retrievals quite well. However, these become significantly worse for higher ranks, where our approach gives much better results. In Fig. 4, we show some qualitative examples for the other datasets. The retrieval results show that our system is capable of finding similar objects despite differences in colour and style. Furthermore, we see that objects can be precisely located despite changes in perspective and partial occlusions, which is also the case for small regions.

## Conclusion

We have presented a novel search algorithm to find and localize motifs or objects in an extensive art collection. This enables art historians to explore large datasets to find visual relationships. Our algorithm is based on a new multi-style feature fusion, which reduces the domain gap and thus improves instance retrieval across artworks. In contrast to previous methods, we require neither object annotations, image labels, nor time-consuming self-supervised training. The presented iterative voting with recent GPU-accelerated approximate nearest-neighbor search [29] enables us to find and localize even small motifs within an extensive database in a few seconds. We have validated the performance of our model on diverse benchmark datasets, including art collections [2, 53] and real photos [39]. We have also shown that our method is much more stable against distractors compared to the current state-of-the-art.

## Acknowledgement

# References

1. Artuk. https://artuk.org, accessed: 2020-05-31
2. "brueghel family: Jan brueghel the elder." the brueghel family database. university of california, berkeley. http://www.janbrueghel.net, accessed: 2018-10-16
3. Wga. https://wga.hu, accessed: 2020-05-31
4. Wikiart. https://www.wikiart.org, accessed: 2018-10-16
5. Amirshahi, S.A., Hayn-Leichsenring, G.U., Denzler, J., Redies, C.: Jenaesthetics subjective dataset: analyzing paintings by subjective scores. In: European Conference on Computer Vision. pp. 3–19. Springer (2014)
6. Arora, R.S., Elgammal, A.: Towards automated classification of fine-art painting style: A comparative study. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). pp. 3541–3544. IEEE (2012)
7. Collins, E., Süsstrunk, S.: Deep feature factorization for content-based image retrieval and localization. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 874–878. IEEE (2019)
8. Crowley, E.J., Parkhi, O.M., Zisserman, A.: Face painting: querying art with photos. In: BMVC (2015)
9. Crowley, E.J., Zisserman, A.: In search of art. In: Workshop at the European Conference on Computer Vision. pp. 54–70. Springer (2014)
10. Crowley, E.J., Zisserman, A.: The art of detection. In: European Conference on Computer Vision. pp. 721–737. Springer (2016)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Denzler, J., Rodner, E., Simon, M.: Convolutional neural networks as a computational model for the underlying processes of aesthetics perception. In: European Conference on Computer Vision. pp. 871–887. Springer (2016)
13. Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M.: Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. arXiv preprint arXiv:1706.07068 (2017)
14. Fernando, B., Tommasi, T., Tuytelaars, T.: Location recognition over large time lags. Computer Vision and Image Understanding **139**, 21–28 (2015)
15. Garcia, N., Renoust, B., Nakashima, Y.: Contextnet: representation and exploration for painting classification and retrieval in context. International Journal of Multimedia Information Retrieval **9**(1), 17–30 (2020)
16. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
17. Ginosar, S., Haas, D., Brown, T., Malik, J.: Detecting people in cubist art. In: European Conference on Computer Vision. pp. 101–116. Springer (2014)
18. Gonthier, N., Gousseau, Y., Ladjal, S., Bonfait, O.: Weakly supervised object detection in artworks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
20. Hertzmann, A.: Can computers create art? In: Arts. vol. 7, p. 18. Multidisciplinary Digital Publishing Institute (2018)

21. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 327–340 (2001)
22. Hristova, S.: Images as data: cultural analytics and aby warburg's mnemosyne. International Journal for Digital Art History (2) (2016)
23. Impett, L., Süsstrunk, S.: Pose and pathosformel in aby warburg's bilderatlas. In: European Conference on Computer Vision. pp. 888–902. Springer (2016)
24. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European conference on computer vision. pp. 304–317. Springer (2008)
25. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence **33**(1), 117–128 (2010)
26. Jenicek, T., Chum, O.: Linking art through human poses. arXiv preprint arXiv:1907.03537 (2019)
27. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 784–799 (2018)
28. Johnson, C.D.: Memory, metaphor, and Aby Warburg's Atlas of images. Cornell University Press (2012)
29. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)
30. Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemoeller, H.: Recognizing image style. arXiv preprint arXiv:1311.3715 (2013)
31. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast arbitrary style transfer. arXiv preprint arXiv:1808.04537 (2018)
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
33. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
34. Manen, S., Guillaumin, M., Van Gool, L.: Prime object proposals with randomized prim's algorithm. In: Proceedings of the IEEE international conference on computer vision. pp. 2536–2543 (2013)
35. Mao, H., Cheung, M., She, J.: Deepart: Learning joint representations of visual arts. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1183–1191 (2017)
36. Mensink, T., Van Gemert, J.: The rijksmuseum challenge: Museum-centered visual recognition. In: Proceedings of International Conference on Multimedia Retrieval. pp. 451–454 (2014)
37. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 2161–2168. Ieee (2006)
38. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE international conference on computer vision. pp. 3456–3465 (2017)
39. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)

40. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2008)
41. Picard, D., Gosselin, P.H., Gaspard, M.C.: Challenges in content-based image indexing of cultural heritage collections. IEEE Signal Processing Magazine **32**(4), 95–102 (2015)
42. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. arXiv preprint arXiv:1803.11285 (2018)
43. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1655–1668 (2018)
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
45. Saleh, B., Abe, K., Arora, R.S., Elgammal, A.: Toward automated discovery of artistic influence. Multimedia Tools and Applications **75**(7), 3565–3591 (2016)
46. Saleh, B., Elgammal, A.: Large-scale classification of fine-art paintings: Learning the right metric on the right feature. arXiv preprint arXiv:1505.00855 (2015)
47. Sanakoyeu, A., Kotovenko, D., Lang, S., Ommer, B.: A style-aware content loss for real-time hd style transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 698–714 (2018)
48. Schlecht, J., Carqué, B., Ommer, B.: Detecting gestures in medieval images. In: Image Processing (ICIP), 2011 18th IEEE International Conference on. pp. 1285–1288. IEEE (2011)
49. Seguin, B.: The replica project: Building a visual search engine for art historians. XRDS: Crossroads, The ACM Magazine for Students **24**(3), 24–29 (2018)
50. Seguin, B., diLenardo, I., Kaplan, F.: Tracking transmission of details in paintings. In: DH (2017)
51. Seguin, B., Striolo, C., Kaplan, F., et al.: Visual link retrieval in a database of paintings. In: European Conference on Computer Vision. pp. 753–767. Springer (2016)
52. Seguin, B., Striolo, C., Kaplan, F., et al.: Visual link retrieval in a database of paintings. In: European Conference on Computer Vision. pp. 753–767. Springer (2016)
53. Shen, X., Efros, A.A., Aubry, M.: Discovering visual patterns in art collections with spatially-consistent feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9278–9287 (2019)
54. Shen, X., Pastrolin, I., Bounou, O., Gidaris, S., Smith, M., Poncet, O., Aubry, M.: Large-scale historical watermark recognition: dataset and a new consistency-based approach. arXiv preprint arXiv:1908.10254 (2019)
55. Shen, X., Lin, Z., Brandt, J., Avidan, S., Wu, Y.: Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3013–3020. IEEE (2012)
56. Shrivastava, A., Malisiewicz, T., Gupta, A., Efros, A.A.: Data-driven visual similarity for cross-domain image matching. In: Proceedings of the 2011 SIGGRAPH Asia Conference. pp. 1–10 (2011)
57. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

58. Spratt, E.L., Elgammal, A.: Computational beauty: Aesthetic judgment at the intersection of art and science. In: European Conference on Computer Vision. pp. 35–53. Springer (2014)
59. Strezoski, G., Worring, M.: Omniart: multi-task deep learning for artistic data analysis. arXiv preprint arXiv:1708.00684 (2017)
60. Takami, M., Bell, P., Ommer, B.: An approach to large scale interactive retrieval of cultural heritage. In: GCH. pp. 87–95 (2014)
61. Tan, W.R., Chan, C.S., Aguirre, H.E., Tanaka, K.: Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In: 2016 IEEE international conference on image processing (ICIP). pp. 3703–3707. IEEE (2016)
62. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)
63. Ufer, N.: Deep semantic feature matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6914–6923 (2017). https://doi.org/10.1109/CVPR.2017.628
64. Ufer, N., Lui, K.T., Schwarz, K., Warkentin, P., Ommer, B.: Weakly supervised learning of dense semantic correspondences and segmentation. In: German Conference on Pattern Recognition. pp. 456–470. Springer (2019)
65. Ufer, N., Souiai, M., Cremers, D.: Wehrli 2.0: An algorithm for "tidying up art". In: European Conference on Computer Vision. pp. 532–541. Springer (2012)
66. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**(2), 154–171 (2013)
67. Westlake, N., Cai, H., Hall, P.: Detecting people in artwork with cnns. In: European Conference on Computer Vision. pp. 825–841. Springer (2016)
68. Wilber, M.J., Fang, C., Jin, H., Hertzmann, A., Collomosse, J., Belongie, S.: Bam! the behance artistic media dataset for recognition beyond photography. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1202–1211 (2017)
69. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision. pp. 467–483. Springer (2016)
70. Yin, R., Monson, E., Honig, E., Daubechies, I., Maggioni, M.: Object recognition in art drawings: Transfer of a neural network. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2299–2303. IEEE (2016)
71. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European conference on computer vision. pp. 391–405. Springer (2014)