An approach for dataset extension for object detection in artworks using open-vocabulary models

Tetiana Yemelianenko, Iuliia Tkachenko, Tess Masclef, Mihaela Scuturici, Serge Miguet

▶ **To cite this version:**

Tetiana Yemelianenko, Iuliia Tkachenko, Tess Masclef, Mihaela Scuturici, Serge Miguet. An approach for dataset extension for object detection in artworks using open-vocabulary models. Proceedings of the European Conference on Computer Vision (ECCV) Workshops, ECCV, Sep 2024, Milan (Italie), Italy. hal-04820558

HAL Id: hal-04820558
https://hal.science/hal-04820558v1

Submitted on 5 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An approach for dataset extension for object detection in artworks using open-vocabulary models

Tetiana Yemelianenko, Iuliia Tkachenko, Tess Masclef, Mihaela Scuturici, and Serge Miguet

Université Lumière Lyon 2, CNRS, Ecole Centrale de Lyon, INSA Lyon, Universite Claude Bernard Lyon 1, LIRIS, UMR5205, 69007 Lyon, France
tetiana.yemelianenko@univ-lyon2.fr

Abstract. While studying objects presented in paintings, art history specialists identify their significance, symbolic meaning and historical context. Analyzing big artistic collections can be very time-consuming for the specialists. The search could be relieved by using modern object detectors. However, object detectors have poor performance on artistic images. This problem could be solved by fine-tuning them on specialized annotated datasets. In this paper, we explore the possibilities of using open-vocabulary foundation models for dataset annotation in a semi-automated manner. We propose an approach for artistic dataset annotation for object detection task based on a small set of images annotated on image-level and using Vision Transformer for Open-World Localization (OWL-ViT2) model, the YOLO object detector and an approximate nearest neighbour oh yeah (ANNOY) algorithm. We extend the existing DEArt dataset by 97.2% and introduce the way of adding new classes without exhaustive annotation. With the extended version of the dataset, we achieve 12.2% increase of mAP0.5 metric on average on the test data compared to the model trained on the original dataset.

Keywords: Open-vocabulary object detection · Weekly supervised detection · OWL-ViT2 · YOLOv8 · Artwork analysis

1 Introduction

Object detection task can be solved using modern deep learning models pre-trained on large photographic datasets such as COCO [18], Open Images [17], and others. These models could potentially be used for object detection in artworks using the transfer learning technique. But the objects in paintings differ significantly from the photographs by the different styles of the artists and techniques used. This variability leads to a decrease in the precision of object detection. The art history specialists are interested in specific objects which very often have symbolic meanings and usually are not present in the modern datasets (*e.g.*, knight, skull, crucifixion) or represent imaginary beings (*e.g.*, centaur, unicorn, dragon). One of the possible solutions to deal with these problems is the fine-tuning of a pre-trained model on an artistic dataset.

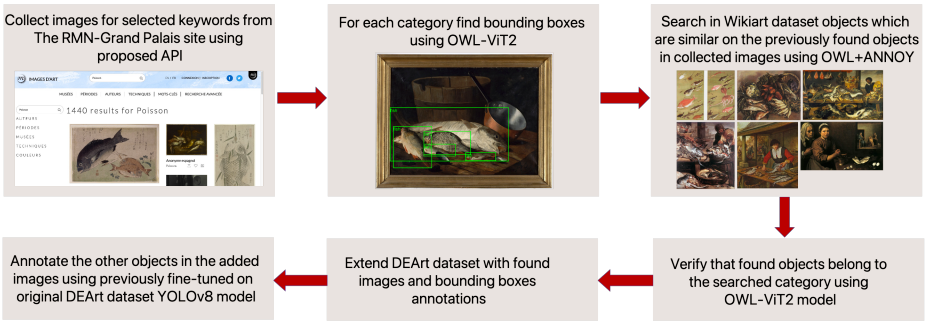

Fig. 1: Proposed pipeline to artistic dataset annotation using image-level annotations, open-vocabulary detector and ANN search.

The contributions of this work are the following:

- We propose a semi-automated approach to data annotation based on using open-vocabulary object detection foundation models illustrated in Fig. 1. We demonstrate that the proposed approach increases F1 scores, mAP50, and mAP50-95 metrics on average for 5 experiments for most extended classes.
- We work on the DEArt dataset presented in [35], considered as the most appropriate dataset for object detection in paintings. We extend the object detection dataset by more than 14700 images with an average augmentation of instances for each selected class of 268%. Also, we add 4 new classes 'candle', 'pomegranate', 'sail', and 'umbrella' which can be interesting for specialists in art history.
- In order to promote the reproducible research, we make the dataset and code publicly available[1].

This work is a part of an interdisciplinary project that involves the participation of specialists in computer vision and art history[2].

The paper is organized as follows: Sec. 2 deals with the current state of the art in open-vocabulary object detection and object detection in the cultural heritage domain. Sec. 3 presents datasets used and preliminary study of the models used. Sec. 4 focuses on the proposed approach. In Sec. 5 we discuss the results of experiments. Sec. 6 concludes the paper and outlines direction for further research.

2 Related works

2.1 Open-vocabulary object detection

It is possible to define three main groups among the traditional object detection methods [5]: (i) region-based methods, such as Mask R-CNN [16], Feature Pyra-

[1] https://gitlab.liris.cnrs.fr/anr-aaa/eccv-ai4dh-2024
[2] https://icar.cnrs.fr/aaa

mid networks [23], Faster R-CNN [34], (ii) pixel-based methods, such as SSD [25], YOLOs [1,33,38], FCOS [37], and (iii) query-based methods, such as DETR [4], Deformable DETR [43]. Using traditional object detection methods with custom datasets is usually limited to a relatively small number of classes in the datasets because the process of data annotation is costly and time-consuming.

Today, there is an increasing attention to solve the object detection task for custom datasets using open-vocabulary object detection [41]. Open-vocabulary object detection models enable the detection of objects beyond pre-defined classes. These models are based on using image-text pre-training. Vision-language models like CLIP [32] are trained for the representation of image-text pairs in a multimodal embedding space, which allows for a given sample from one modality to find a corresponding sample of the other modality. In the last years, a variety of open-vocabulary methods is proposed among the recent ones are Grounding DINO [42], OWL-ViTs [30, 31], YOLO-World [5], and Florence-2 [40]. We describe them shortly in the following paragraphs.

Grounding DINO is built upon an end-to-end transformer-based detector DINO [42] by performing vision-language modality fusion at different phases, including a feature enhancer, a language-guided query selection module, and a cross-modality decoder. This model has a dual-encoder-single-decoder architecture with an image and a text backbone for image and text feature extraction respectively, a feature enhancer for image and text feature fusion, a language-guided query selection module, and a cross-modality decoder [24]. Image features are extracted with an image backbone like Swin Transformer [26], and text features are extracted with a text backbone like BERT [8]. Grounding DINO takes the given image-text pair and returns multiple pairs of object boxes and noun phrases describing the content of the boxes.

The OWL-ViT models [30, 31] use a standard vision transformer as the image encoder and a transformer architecture as the text encoder. The OWL-ViT model contrastively pre-trains image and text encoders on large-scale image-text pairs, then adds detection heads, and is fine-tuned on detection data.

YOLO-World model is the open-vocabulary object detector pre-trained on large-scale datasets. YOLO-World is a real-time object detection model which has a high inference speed, unlike previously discussed models. The architecture of the YOLO-World model consists of a YOLO detector, a text encoder, and a re-parameterizable Vision-Language Path Aggregation Network [5].

Florence-2 is a vision foundation model with a unified, prompt-based representation which takes text-prompts as task instructions and generates results in text forms [40]. This model stands out from the previously mentioned one due to its ability to tackle a broader range of tasks, including captioning, object detection, grounding, and segmentation. The model uses a vision encoder to extract image embeddings, which are then concatenated with text embeddings and processed by a transformer-based multi-modal encoder-decoder to generate the response.

Open-vocabulary object detection models could be used for auto-labeling object detection datasets automatically or in a weekly supervised manner us-

ing image-level annotations. This approach is commonly used for photographic datasets though it is not effective for artistic datasets because the modern open-vocabulary models were mostly trained on photographs but object representation in artistic datasets differs from real-life objects by style and technique used by the artist (we demonstrate some practical results in Sec. 3.2). Moreover, adding new classes requires significant time because of relatively big inference time of open-vocabulary models.

2.2 Object detection in artworks

Modern object detectors have demonstrated significant success in object detection with fixed vocabulary for natural image datasets. Meanwhile, the task of object detection in artistic datasets remains quite challenging and has been less studied. Most existing artistic datasets are extensively used in classification and retrieval tasks, among them Wikipaintings [19], Painting-91 [20], MultitaskPainting100k [27], Rijksmuseum [28], OmniArt [36], VGG Paintings [6], ArtBench-10 [22], and others. Artworks in these datasets are annotated on image-level with art attributes or/and with textual descriptions like SemArt [10]. There are not so many specialized artistic datasets oriented on object detection. In the following paragraphs, we discuss them and how they are used for solving object detection task in artworks.

The work [11] is focused on people detection in cubist paintings. In this work, four detection methods are compared: Dalal and Triggs [7], deformable part-based models [9], Poselets [2, 3], and R-CNN [13]. The object detectors are trained on a set of 218 Picasso paintings that have titles indicating that they depict people, the paintings in this dataset are annotated with the single class 'person'. In [39], authors introduce People-Art dataset which contains photos, cartoons and images from 41 different artwork movements. The artistic dataset is annotated with bounding boxes for the single class 'person'. This dataset is considered challenging because of the high variability in styles and techniques. For solving object detection task, authors fine-tune Fast R-CNN model [12] on the introduced dataset.

In [14, 15], authors work on weekly supervised object detection in artworks using only image-level annotations. The authors propose a model to solve the multiple instance problem for weekly supervised object detection and introduce the IconArt dataset which contains 5955 paintings from Wikicommons, the artworks are dated from the 11^{th} to the 20^{th} century. The dataset is annotated with 7 classes on the image-level, test dataset of 1480 images is annotated on object-level with classes 'angel', 'child Jesus', 'crucifixion', 'Mary', 'nudity', 'ruins', and 'Saint Sebastian'. In [18] a dataset comprising 58,672 artistically styled images is created using AdaIn style transfer applied to images from the COCO dataset. This dataset is used for fine-tuning a Faster R-CNN object detection model [34]. The fine-tuned model is evaluated on the People-Art test dataset, demonstrating an improvement over the existing state-of-the-art. Despite being the largest artistic dataset for object detection, StyleCOCO includes only classes presented in the COCO dataset. Art history specialists, however, are interested

in classes depicting imaginary beings, such as pegasus and centaur, or religious objects which are commonly not present in photographic datasets. In [17] the dataset for medieval musicological studies is introduced. This dataset contains 693 samples in five classes: 'book', 'folio', 'phylactery', 'lectern', and 'altar'. The authors propose the technique for performing few-shot object detection based on bi-stage training, in which the first stage tries to improve on the object localization process for the new classes and the second stage aims to improve the image classification and fine-tuning of the pre-located coordinates. Authors [29] use pre-trained GLIP [21] vision-language model for the generation of bounding boxes for the objects' classes from the COCO dataset. The feasibility of the approach is evaluated on the People-Art test set which contains a unique class – 'person'.

In [35] the DEArt dataset is introduced. This dataset is oriented on the detection of iconographic elements in artworks that are specific to art history. The DEArt is focused on European art and contains more than 15000 paintings from the 12^{th} to the 18^{th} centuries. The dataset is annotated with 69 classes (in the current version of the dataset the new 70^{th} class 'fish' is added). The total number of each class is presented in supplemental materials. 10k images from the dataset are annotated manually, the remaining images are annotated in a semi-supervised manner by using a Faster R-CNN model trained on 10k images and manually corrected after training. This dataset is highly unbalanced with around 46990 instances for the class 'person' (maximum) and 29 instances for the class 'mouse' (minimum). The achieved precision for the trained object detection model (mAP0.5) equals 31.2%. We selected this dataset as the base in our work due to its diverse range of cultural heritage objects and observed the possibility to extend it especially the classes with the small number of instances (*e.g.*, 'bear', 'orange', 'fish').

3 Preliminary study

3.1 Preparation of data and description of the datasets used

In our research we use three datasets: DEArt, The RMN-Grand Palais, and WikiArt. The DEArt dataset is introduced in the Sec. 2.2. In [35] detailed dataset description and statistics can be found. In our work we also use paintings annotated on image-level from the photography collection of the French Museum consortium Réunion des Musées Nationaux Grand Palais[3] (RMN-Grand Palais). The RMN-Grand Palais proposes an API for full access to the images and their metadata from the photography fund. The collection has 550,000 works from over hundreds of museums, the API gives access to the image-level annotations of the artworks in French and English. The lists of keywords in French are more detailed in comparison with keywords in English, so in this work for the image-level annotation search, we used the translation of class names in French. We create a collection of image-level annotated artworks using keyword search

[3] https://art.rmngp.fr/

with the API on the French Museum consortium site. Then, this collection is manually cleaned from the unrelated to the selected keyword images. After this pre-selection stage, we have 5269 paintings annotated on image-level in average 181 paintings for each class selected for extension. The third dataset which is used in our work is the publicly available WikiArt[4] dataset. The WikiArt dataset contains more than 130000 digitized paintings from the 15^{th} to 20^{th} centuries, metadata of the dataset includes 27 different styles (Romanticism, Baroque, Impressionism, *etc.*), 10 painting genres (landscape, portrait, still life, *etc.*), and more than 1000 artists. The WikiArt dataset is used for searching paintings in which the objects from selected classes are present, this part is explained in details in Sec. 3.3. The DEArt dataset contains only 687 paintings from WikiArt dataset, so there are less than 5% of images which potentially could be duplicated in the final extended dataset.

3.2 Qualitative comparison of open-vocabulary detectors

In this section we study the possibility of using open-vocabulary detectors for automated dataset annotation. In literature [5,30,31,40,42], a quantitative comparison of open-vocabulary detectors is conducted. However, this comparison is oriented predominantly on natural image datasets, specifically those containing classes from the COCO dataset. In our research, we are focused on specific classes of objects which could be interesting for the art history specialists. For the artistic images and more rare classes, such as angel, crozier, and crucifixion, to the best of our knowledge, this type of comparison does not exist. In this section, we realize a qualitative comparison of four state-of-the-art open-vocabulary detectors on artistic images. An exhaustive comparison of the open-vocabulary detectors is beyond the scope of this paper and could be a direction for further research.

First, we start by verifying how well Grounding DINO-T, OWL-ViT2 B/16, YOLO-World-L, and Florence-2-B models can detect selected classes of objects on the images. We select for our analysis the classes which do not have a big number of instances in the DEArt dataset and potentially could be detected by open-vocabulary detectors (that is why we exclude such very specific classes as 'Judith', 'saturno', 'zucchetto', and others), then we add some classes which are well presented in the DEArt dataset as well as in photographic datasets. Our final list of classes consists of $n_c = 27$ DEArt classes (the full list of classes is presented in Tab. 2). For the detectors comparison we randomly choose $k = 10$ images for each selected class from the DEArt dataset, then we detect objects using four open-vocabulary detectors. For each image i among k selected we visually evaluate how many objects of the class c are detected correctly and calculate the proportion of the number of correctly detected objects $obj_{c,i}$ to all objects of this class presented in the ground truth annotations $obj_{c,i}^{(gt)}$ for this image. Then we calculate the average of these values for each class. After

[4] https://www.wikiart.org

Table 1: Qualitative comparison results.

Open-vocabulary detector	$Q^{(det)}$
OWL-ViT2	0.6027
Florence-2	0.4609
Grounding DINO	0.1830
YOLO-World	0.0130

that, we calculate the average for all selected classes for each open-vocabulary detector. The final formula for quality evaluation is

$$Q^{(det)} = \frac{1}{n_c \cdot k} \sum_{c=1}^{n_c} \sum_{i=1}^{k} \frac{obj_{c,i}}{obj_{c,i}^{(gt)}}. \tag{1}$$

The calculated $Q^{(det)}$ for each detector is present in Tab. 1 (bigger value - better the model found objects in the selected paintings). According to the analysis, we choose the OWL-ViT2 zero-shot detector, which is the most successful during our comparison. It is worth mentioning that for some classes OWL-ViT2 tends to propose several bounding boxes for one object, this problem could be solved by adapting the threshold for each class. OWL-ViT2 detector gives better results for the classes specific to art history analysis, such as 'crozier', 'crucifixion', 'nude', and others. Florence-2 and Grounding DINO detectors in some cases give more accurate bounding boxes for the classes which are present in the COCO dataset, such as 'cat', 'dog' and others. The Grounding DINO detector demonstrates a tendency to detect and label objects according to a given prompt even when they are not present in the paintings.

Figure 2 demonstrates the detection results for the OWL-ViT2, Florence-2, and Grounding DINO models (shown in the first, second, and third columns, respectively). With the prompt 'angel' all three models correctly find the object in the image, although, this class is not present in the classes of the most popular object detection datasets. With the prompt 'crozier' only with the OWL-ViT2 model we receive satisfactory results. We observe similar results for other classes, such as 'crucifixion', 'nude', and 'halo'. YOLO-World zero-shot detector out of the box is not capable to correctly detect objects on the paintings during our evaluation even for the classes present in COCO dataset. Perhaps, the results for the Grounding DINO and YOLO-World detectors could be improved by using prompt engineering technique. Anyway, the capabilities of open-vocabulary detectors (mostly pre-trained on photographic images) to detect objects in the artistic images should be investigated more formally and in a quantitative manner, but this is beyond the scope of this research.

As well, we can conclude that in the case when we additionally have annotations on image level, it is better to use text prompts with a single class in comparison with prompts which contain all classes that we want to use for the

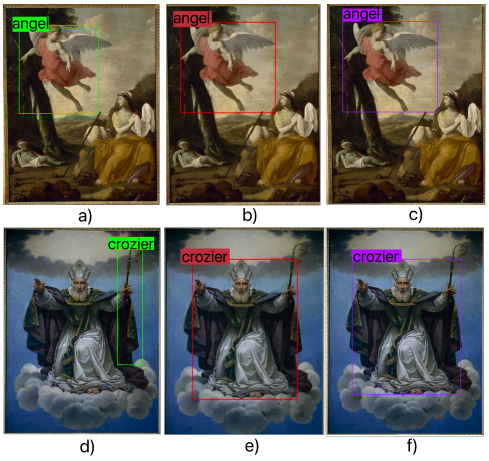

Fig. 2: Comparison of detection results with prompt 'angel' (a, b, c) and 'crozier' (d, e, f) by OWL-ViT2, Florence-2, Grounding DINO respectively.

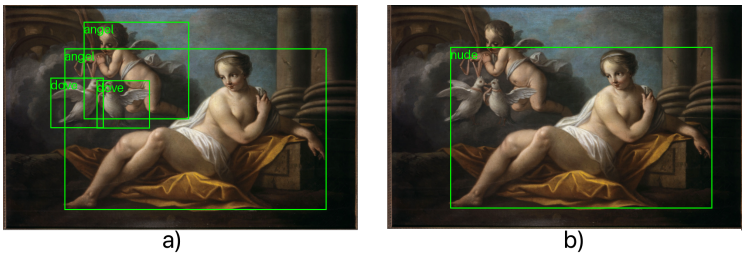

Fig. 3: Comparison of detection results by OWL-ViT2 with prompt which includes all selected classes (a) and a single word prompt (b). For the first prompt classes 'angel' and 'dove' were detected.

annotation. In Fig. 3, for the first query (a) as a prompt we use all pre-selected classes, as a result, the 'nude' object is defined as 'angel'. When we use a specific prompt for searching 'nude' (b), this object is successfully found.

3.3 Similar objects retrieval with OWL-ViT2 and approximate nearest neighbor (ANN) search

Both OWL-ViT2 and Florence-2 models look suitable for the data annotation of artistic images. But the OWL-ViT2 model introduces image conditional detection which allows open-vocabulary detectors to detect objects when even their names are unknown. This property looks prominent for object detection of cultural heritage-specific objects which are not commonly present in the photographic datasets.

OWL-ViTs models provide not only zero-shot text-conditioned, but one-shot image-conditioned object detection also, this means that it is possible to find

Fig. 4: The similarity search results for the selected region in the query image. Images are sorted according to the cosine distance between query embeddings and pre-calculated embeddings of the images in the WikiArt dataset.

similar to the query image objects using query image embeddings. Moreover, with OWL-ViTs the objectness of region of interest and its embeddings can be obtained. According to [30], objectness predicts the likelihood that an output actually represents an object. The high value of objectness means that the objects could potentially be present in this region. Using this property we pre-calculate and save embeddings for the top 50 regions of interest (based on their objectness) for each image in the non-annotated WikiArt dataset. Then, given the query region, we can use its embeddings to calculate the distances between the query region and the saved embeddings for the images from the WikiArt dataset. We decide to use the ANN search instead of the exhaustive search which could be very time-consuming. As the ANN algorithm, we choose ANNOY[5] proposed in 2015 by Erik Bernhardsson for the Spotify platform. ANNOY permits to build the index only once and after that re-use it. ANNOY algorithm is fast and efficient in high dimensional spaces, in our case for each region the length of the embeddings is 3600.

Figure 4 illustrates the results of an image-conditioned search for the top 8 similar objects. First, we select the region of interest in the query image, then we calculate the embeddings for the selected region and search similar regions in the embeddings database indexed by the ANNOY algorithm.

3.4 Object detection with fine-tuned YOLOv8 model

We fine-tune YOLOv8[6] model on the original DEArt dataset. The dataset is split into training, validation, and test datasets with 80%, 10%, and 10% of images

[5] https://github.com/spotify/annoy
[6] https://github.com/ultralytics/ultralytics

respectively. Among all YOLOv8 models we chose YOLOv8m, we tried to fine-tune the bigger YOLOv8l model, but we observed quick over-fitting, which could be explained by the relatively small size of the DEArt dataset. The YOLOv8m model is fine-tuned during 50 epochs with hyperparameters proposed by default. Figure 5 illustrates fine-tuning results. From this figure we can conclude that training and validation losses for box, classification, and distribution focal losses decrease over epochs, indicating the model's improving performance, precision and recall metrics show an overall increasing trend. The increasing mAP metrics confirm that the model improves accuracy in detecting and classifying objects. For the fine-tuned model metric mAP0.5 is equal to 0.42 on the test dataset (in average for the five experiments with different split on training, validation and test data). We choose fine-tuned model with the maximum value of mAP0.5 for further using during our semi-automated annotation.

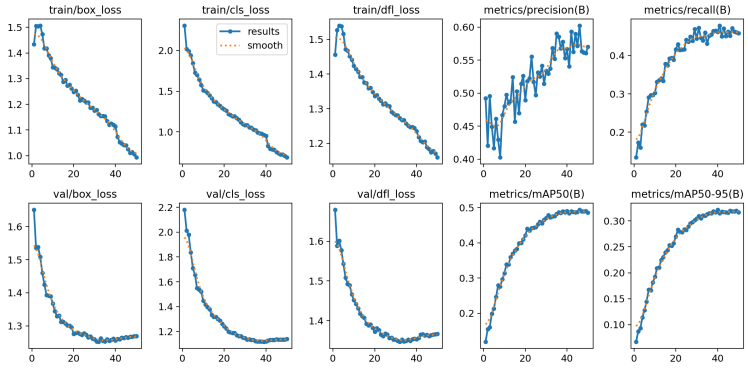

Fig. 5: Fine-tuning results for YOLOv8m model on DEArt dataset.

4 Proposed semi-automated dataset annotation approach

We start by preparing dataset with images annotated on the image-level. For this, we query the RMN-Grand Palais API for each class pre-selected for extension. Then, we visually evaluate the retrieved images and exclude non-relevant ones. This process is described in details in Sec. 3.1. Next, we pre-calculate embeddings of the regions of interest for the all images from the WikiArt dataset as described in Sec. 3.3 and fine-tune YOLOv8 model on the DEArt dataset as described in Sec. 3.4.

In the proposed semi-automated approach, for each class from the list of classes selected for the extension, we define the value of threshold for zero-shot object detection by OWL-ViT2 model. Empirically as the starting value we choose 0.4. Next, for each image in the list of images annotated on image-level with current class, we detect current class objects with the currently defined

threshold. If there are detected objects of the current class, we add this image with all bounding boxes to the list of annotated images. Next, for each detected box we obtain a query embedding vector using the OWL-ViT2 model and find using ANNOY, embeddings similar to this query embedding vector. Each embedding corresponds to the object in the painting. Next, we verify that found objects belong to the current class using OWL-ViT2 with the current threshold. If the image contains an object of the current class, we add this image with all its bounding boxes to the list of annotated images. If the number of annotated images is smaller than a pre-defined minimum number of annotated images and the threshold is bigger than or equal to 0.2, we decrease the value of the threshold and repeat all steps. When the final list of annotated images with the current class is ready, for each image in this list we annotate other objects present in the image using the fine-tuned YOLOv8 model. Now the images of the current class are fully annotated and we add them to the final list.

The described annotation process is summarized in Algorithm 1 and in Fig. 1. It is recommended to verify manually the final annotated list after annotation and fine-tuning the YOLOv8 model on the augmented dataset for the classes for which metrics after augmentation decrease. For example, we correct in this way the final list for the 'elephant' class. It is obvious that in the final annotated dataset could be presented artefacts with wrong bounding boxes and classes but due to the significant augmentation of quantity of instances for classes, these mistakes could be ignored.

The proposed approach can be used for any other pairs of datasets - small annotated on the image-level and big without any annotations. Anyway, we recommend verifying that all images in the extended dataset contain the searched classes. This verification has been done for the final version of the presented extended dataset. The process of verification is significantly simpler than a complete annotation for object detection.

5 Results

5.1 Detection performance on extended dataset

To demonstrate that the proposed approach works, we conduct 5 experiments for fine-tuning the YOLOv8m model before dataset extension and after. For each experiment, we divide the original DEArt dataset on the train, validation, and test datasets with 70%, 10% and 20% images respectively. Then, we fine-tune the YOLOv8m model during 50 epochs with hyperparameters proposed by default and evaluate it on the test dataset, this is our result before dataset extension. After that, we add images from the classes selected for the extension (images are found by the approach described in Sec. 4) to train and validation data. Then, we fine-tune the YOLOv8m model with the same hyperparameters on the extended data and evaluate detection results on the same test dataset used for the evaluation before dataset extension.

Table 2 demonstrates evaluation results averaged by 5 experiments before and after extension for all classes and for the classes for which we extended data

Algorithm 1 Proposed semi-automated annotation approach.

Require: Pre-calculated embeddings of the regions of interest for the images from the WikiArt dataset, fine-tuned on the DEArt dataset YOLOv8 model, *selected_classes*, *query_images* grouped by classes

min_images_count ← 20
for all *class* from *selected_classes* **do**
    threshold ← 0.4
    all_annotated_images ← empty list
    images_count ← 0
    while *threshold* ≥ 0.2 or *images_count* < *min_images_count* **do**
        annotated_images ← empty list
        for all *image* from *query_images* for the *class* **do**
            Find *b_boxes* with OWL-ViT2(*image*, *class*, *score* > *threshold*)
            if *count*(*b_boxes*) > 0 **then**
                images_count ← *images_count* + 1
                Add *image* to *annotated_images* list
                for all *b_box* from *b_boxes* **do**
                    Get *query_embedding* for *b_box* with OWL-ViT2
                    Find using ANNOY in pre-calculated embeddings objects similar to *query_embedding*
                    for all similar objects **do**
                        Verify that found objects belong to the *class* using OWL-ViT2
                        Add images found with ANNOY to the *annotated_images* list
                        images_count ← *images_count* + 1
                    end for
                end for
            end if
            if *images_count* < *min_images_count* **then**
                threshold ← *threshold* − 0.1
                annotated_images ← empty list
                images_count ← 0
            end if
        end for
    end while
    Automatically annotate *annotated_images* using fine-tuned YOLOv8 model for all *selected_classes* ≠ *class*
    Add *annotated_images* to *all_annotated_images* list
end for

by the proposed approach. For the evaluation, we use F1 score - the measure that balances precision and recall, mAP0.5 metric - mean average precision at the intersection over union (IoU) threshold 0.5, and mAP0.5-0.95 metric - mean average precision across multiple IoU thresholds from 0.5 to 0.95.

We can observe the increase of all metrics for the augmented dataset as a whole and almost for all classes (except of 'orange') for which we have done extension. The decreasing of metrics for class 'orange' can be explained as follows.

Table 2: The average F1, mAP0.5, mAP0.5-0.95 metrics on the test data for the fine-tuned model before dataset extension and after using the proposed approach.

Class	F1	F1 (ours)	mAP0.5	mAP0.5 (ours)	mAP0.5-0.95	mAP0.5-0.95 (ours)
All	0.439	**0.512**	0.422	**0.473**	0.268	**0.306**
Dog	0.554	**0.654**	0.574	**0.658**	0.363	**0.434**
Angel	0.657	**0.711**	0.695	**0.744**	0.447	**0.491**
Cat	0.294	**0.415**	0.256	**0.401**	0.172	**0.291**
Eagle	0.435	**0.556**	0.468	**0.539**	0.349	**0.417**
Lion	0.407	**0.551**	0.452	**0.548**	0.309	**0.397**
Nude	0.644	**0.696**	0.678	**0.728**	0.467	**0.524**
Donkey	0.315	**0.466**	0.329	**0.472**	0.198	**0.303**
Cow	0.565	**0.657**	0.598	**0.692**	0.373	**0.450**
Horse	0.578	**0.645**	0.580	**0.646**	0.360	**0.419**
Apple	0.340	**0.487**	0.343	**0.442**	0.206	**0.279**
Butterfly	0.582	**0.632**	0.606	**0.688**	0.476	**0.545**
Halo	0.746	**0.762**	0.779	**0.797**	0.521	**0.545**
Swan	0.354	**0.457**	0.320	**0.456**	0.195	**0.299**
Deer	0.409	**0.570**	0.427	**0.530**	0.261	**0.354**
Sheep	0.410	**0.532**	0.398	**0.495**	0.236	**0.305**
Crucifixion	0.721	**0.744**	0.744	**0.806**	0.493	**0.543**
Serpent	0.248	**0.326**	0.195	**0.277**	0.129	**0.197**
Skull	0.650	**0.726**	0.676	**0.748**	0.469	**0.527**
Crozier	0.324	**0.425**	0.322	**0.420**	0.180	**0.248**
Rooster	0.366	**0.469**	0.317	**0.474**	0.234	**0.345**
Monkey	0.397	**0.549**	0.417	**0.550**	0.244	**0.351**
Trumpet	0.141	**0.163**	0.090	**0.136**	0.050	**0.072**
Dove	0.580	**0.668**	0.579	**0.669**	0.305	**0.375**
Orange	**0.196**	0.160	0.104	**0.182**	0.073	**0.135**
Elephant	0.548	**0.552**	0.531	**0.558**	0.372	**0.413**
Bear	0.421	**0.557**	0.488	**0.595**	0.386	**0.498**
Fish	0.050	**0.303**	0.036	**0.252**	0.025	**0.153**

There are only 113 instances with class 'orange' in DEArt dataset, and among them there are lemons annotated as oranges, but in the extended version of the dataset the objects of class 'orange' are mostly oranges, and in extended dataset there are oranges painted more abstractly or loosely in comparison with the paintings in the original version of the dataset. Probably, images of this class in both versions of the dataset should be manually verified.

5.2 Extended dataset statistics

In total the DEArt dataset is augmented by 14736 images, that is, by 97.2% relative to the original size. Table 3 demonstrates a number of added instances for each class and the percent of extension relative to the number of instances in the original DEArt dataset. Using the proposed approach we add to the dataset

Table 3: Number of instances for the extended classes of DEArt dataset and the percent of extension for each class.

Class	Number (%)	Class	Number (%)	Class	Number (%)
Dog	3449 (154)	Donkey	1432 (274.9)	Deer	847 (239.9)
Angel	3548 (71.5)	Cow	6494 (455)	Sheep	3590 (274)
Fish	883 (735.8)	Horse	8773 (258.6)	Crucifixion	847 (150.7)
Cat	925 (528.6)	Apple	3831 (444.9)	Serpent	656 (149.4)
Eagle	910 (343.4)	Butterfly	969 (199.8)	Skull	651 (130.5)
Lion	981 (197.8)	Halo	1869 (37.7)	Crozier	348 (95.3)
Nude	6592 (129)	Swan	1157 (697)	Rooster	714 (466.7)
Monkey	775 (166.3)	Trumpet	446 (107.5)	Dove	778 (196.5)
Orange	409 (361.9)	Elephant	128 (69.6)	Bear	574 (438.2)

4 new classes which potentially could be interesting for art history specialists, among them 'candle' with 165 images, 'pomegranate' - 46, 'sail' - 561, and 'umbrella' - 449. The new version of the dataset contains images from the 12^{th} to 20^{th} centuries in contrast with the original DEArt dataset with images from the 12^{th} to 18^{th} centuries. If it is necessary it is possible to restrict the period of the paintings by filtering images in the WikiArt dataset before dataset extension.

6 Conclusion and future work

We have proposed a semi-automated approach for image annotation in artistic datasets. We extended the DEArt dataset by 14736 images, *i.e.* by 97.2%. Our experimental results demonstrate the increase in F1 score, mAP0.5, and mAP0.5-0.95 metrics after dataset extension in comparison with metrics obtained for the original DEArt dataset. We also added to the existing dataset 4 new categories which could be potentially interesting to the art history specialists: 'candle', 'sail', 'pomegranate', and 'umbrella' with 165, 561, 46, and 449 images respectively. In order to promote the reproducible research, the extended dataset and the code will be publicly available. The limitation of the proposed approach is the necessity of choosing the value of the threshold for zero-shot object detection by the OWL-ViT2 model when you add new classes to the dataset. Future work will focus on the further studying the properties of open-vocabulary detection models presented in the paper, the possibilities of their fine-tuning and prompt engineering techniques for improving the process of annotation.

Acknowledgments

This work was funded by french national research agency with grant ANR-20-CE38-0017. We would like to thank the PAUSE ANR-Program: Ukrainian scientists support to support the scientific stay of T. Yemelianenko in LIRIS laboratory.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. ArXiv **abs/2004.10934** (2020)
2. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 1365–1372 (09 2009). https://doi.org/10.1109/ICCV.2009.5459303
3. Bourdev, L.D., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: European Conference on Computer Vision (2010)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 213–229. Springer International Publishing, Cham (2020)
5. Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y.: Yolo-world: Real-time open-vocabulary object detection. ArXiv **abs/2401.17270** (2024)
6. Crowley, E.J., Zisserman, A.: In search of art. In: Workshop on Computer Vision for Art Analysis, ECCV (2014)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 886–893 vol. 1 (2005). https://doi.org/10.1109/CVPR.2005.177
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019)
9. Felzenszwalb, P., Girshick, R., Mcallester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**, 1627–45 (09 2010). https://doi.org/10.1109/TPAMI.2009.167
10. Garcia, N., Vogiatzis, G.: How to read paintings: Semantic art understanding with multi-modal retrieval. ArXiv **abs/1810.09617** (2018)
11. Ginosar, S., Haas, D., Brown, T., Malik, J.: Detecting people in cubist art. In: SIGAI (2014)
12. Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (11 2013). https://doi.org/10.1109/CVPR.2014.81
14. Gonthier, N., Gousseau, Y., Ladjal, S., Bonfait, O.: Weakly supervised object detection in artworks. In: Computer Vision – ECCV 2018 Workshops. pp. 692–709. Springer International Publishing, Cham (2019)
15. Gonthier, N., Ladjal, S., Gousseau, Y.: Multiple instance learning on deep features for weakly supervised object detection with extreme domain shifts. Computer Vision and Image Understanding **214**, 103299 (2022). https://doi.org/https://doi.org/10.1016/j.cviu.2021.103299
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
17. Ibrahim, B.I.E., Eyharabide, V., Le Page, V., Billiet, F.: Few-shot object detection: Application to medieval musicological studies. Journal of Imaging **8**(2) (2022). https://doi.org/10.3390/jimaging8020018

18. Kadish, D., Risi, S., Løvlie, A.S.: Improving object detection in art images using only style transfer. 2021 International Joint Conference on Neural Networks (IJCNN) pp. 1–8 (2021)
19. Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemoeller, H.: Recognizing image style. ArXiv **abs/1311.3715** (2013)
20. Khan, F., Beigpour, S., Weijer, J., Felsberg, M.: Painting-91: A large scale database for computational painting categorization. Machine Vision and Applications **25**, 1385–1397 (08 2014). https://doi.org/10.1007/s00138-014-0621-6
21. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., Gao, J.: Grounded language-image pre-training. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10955–10965 (2021)
22. Liao, P., Li, X., Liu, X., Keutzer, K.: The artbench dataset: Benchmarking generative models with artworks. arXiv preprint arXiv:2206.11404 (2022)
23. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 936–944 (2016)
24. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., yue Li, C., Yang, J., Su, H., Zhu, J.J., Zhang, L.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. ArXiv **abs/2303.05499** (2023)
25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision (2015)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9992–10002 (2021)
27. Mao, H., Cheung, M., She, J.: Deepart: Learning joint representations of visual arts. In: Proceedings of the 25th ACM International Conference on Multimedia. p. 1183–1191. MM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3123266.3123405
28. Mensink, T., van Gemert, J.: The rijksmuseum challenge: Museum-centered visual recognition. In: Proceedings of International Conference on Multimedia Retrieval. p. 451–454. ICMR '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2578726.2578791
29. Meyer, L., Aaen, J.E., Tranberg, A.R., Kun, P., Freiberger, M., Risi, S., Løvlie, A.S.: Algorithmic ways of seeing: Using object detection to facilitate art exploration. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. CHI '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3613904.3642157
30. Minderer, M., Gritsenko, A.A., Houlsby, N.: Scaling open-vocabulary object detection. ArXiv **abs/2306.09683** (2023)
31. Minderer, M., Gritsenko, A.A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., Houlsby, N.: Simple open-vocabulary object detection with vision transformers. ArXiv **abs/2205.06230** (2022)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)

33. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 779–788 (2015)
34. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 1137–1149 (2015)
35. Reshetnikov, A., Marinescu, M.C.V., López, J.M.: Deart: Dataset of european art. In: ECCV Workshops (2022)
36. Strezoski, G., Worring, M.: Omniart: A large-scale artistic benchmark. ACM Transactions on Multimedia Computing, Communications, and Applications **14**, 1–21 (10 2018). https://doi.org/10.1145/3273022
37. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9626–9635 (2019). https://doi.org/10.1109/ICCV.2019.00972
38. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7464–7475 (2022)
39. Westlake, N., Cai, H., Hall, P.: Detecting people in artwork with cnns. In: ECCV Workshops (2016)
40. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. ArXiv **abs/2311.06242** (2023)
41. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14393–14402 (June 2021)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2023)
43. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. ArXiv **abs/2010.04159** (2020)

A. Dataset Details

Table 4: Number of instances for each class in DEArt dataset.

Class	Number	Class	Number	Class	Number	Class	Number
Person	46966	Crown	1108	Dragon	406	Swan	166
Tree	11409	Prayer	1004	Palm	405	Rooster	153
Nude	5109	Monk	939	Dove	396	Pegasus	138
Angel	4965	Devil	886	Key of Heaven	388	Saturno	138
Halo	4961	Apple	861	Mitre	377	Zucchetto	134
Horse	3392	Shield	772	Crozier	365	Zebra	132
Boat	3253	Scroll	747	Tiara	355	Bear	131
Bird	3036	Chalice	662	Deer	353	Unicorn	129
Book	2765	Crucifixion	562	Crown of thorn	345	Fish	120
Dog	2239	Donkey	521	Hands	284	Horn	119
Helmet	2052	Skull	499	God the Father	279	Stole	118
Lance	1768	Lion	496	Eagle	265	Orange	113
Knight	1765	Butterfly	485	Shepherd	244	Holy shroud	91
Sword	1751	Monkey	466	Head	240	Judith	85
Jug	1494	Serpent	439	Camauro	210	Banana	32
Cow	1426	Lily	437	Centaur	200	Mouse	29
Banner	1344	Arrow	422	Elephant	184		
Sheep	1310	Trumpet	415	Cat	175		