Received: 28 November 2022 | Revised: 10 March 2023 | Accepted: 31 March 2023

DOI: 10.1049/ipr2.12806

IET Image Processing

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Toward cross-domain object detection in artwork images using improved YoloV5 and XGBoosting

Tasweer Ahmad[1,3] | **Maximilian Schich**[2,3]

[1] School of Digital Technology, Tallinn University, Tallinn, Estonia

[2] Baltic, Film, Media, and Arts School, Tallinn University, Tallinn, Estonia

[3] ERA Chair for Cultural Data Analytics, Tallinn University, Tallinn, Estonia

Correspondence
Tasweer Ahmad, School of Digital Technology, Tallinn University, Narva mnt 25, 10120 Tallinn, Estonia.
Email: tasveerahmad@gmail.com

Funding information
European Union's Horizon 2020, Grant/Award Number: 810961

Abstract

Object recognition in natural images has achieved great success, while recognizing objects in style-images, such as artworks and watercolor images, has not yet achieved great progress. Here, this problem is addressed using cross-domain object detection in style-images, clipart, watercolor, and comic images. In particular, a cross-domain object detection model is proposed using YoloV5 and eXtreme Gradient Boosting (XGBoosting). As detecting difficult instances in cross domain images is a challenging task, XGBoosting is incorporated in this workflow to enhance learning of the proposed model for application on hard-to-detect samples. Several ablation studies are carried out by training and evaluating this model on the StyleObject7K, ClipArt1K, Watercolor2K, and Comic2K datasets. It is empirically established that this proposed model works better than other methods for the above-mentioned datasets.

1 | INTRODUCTION

Now-a-days, machine learning methods are stunningly capable of art image generation, segmentation, and detection. Over the last decade, object detection has achieved great progress due to the availability of challenging and diverse datasets, such as MS COCO [1], KITTI [2], PASCAL VOC [3] and WiderFace [4]. Yet, most of the existing object detectors are domain-specific, as training and testing are typically carried out on the same data set. This situation may, for example, be rooted in the presumption that there exists some underlying non-trivial domain shift, due to which a model trained using one dataset may probably not work well on another kind of dataset. Object detection has achieved remarkable performance for natural images, but the detection task is less explored for stylized images including artworks; for example, object detection remains harder in paintings, watercolors, clipart, and comic images. This situation is partially driven by the relative scarcity of large-scale annotated datasets for object detection of stylized images, not even speaking of artworks that do not follow a Euclidean convention of space and object representation. Meanwhile, object detection in stylized conventional images, and eventually non-representational, such as cubist or semi-abstract images, would be an appreciated capability in art research, including for the appreciation of art market value, where our style-image object detection could improve meta-data and facilitate search for visual motifs.

Over the recent few years, computer vision has successfully developed deep-learning based methods for visual object detection and recognition in natural images, including photographs [5, 6] and videos. The latter includes a recent contribution by the first author, using a similar workflow pipeline to recognize difficult examples in drone videos. Meanwhile, within the domain of stylized images, including artworks using a more or less conventional mode of object representation, several challenges of "domain shift" remain, constituting a hurdle for object detection between natural images to stylized images. These challenges necessitate specifically tailored algorithms for artwork images [7, 8]. In this research work, we contribute to this area by performing a computer-aided study of non-natural stylized images, toward opening the way for more advanced tasks, such as a first iteration of automatic artwork image captioning applied to large digital image collections, [9], and further extensions of the presented method to less conventional artworks.

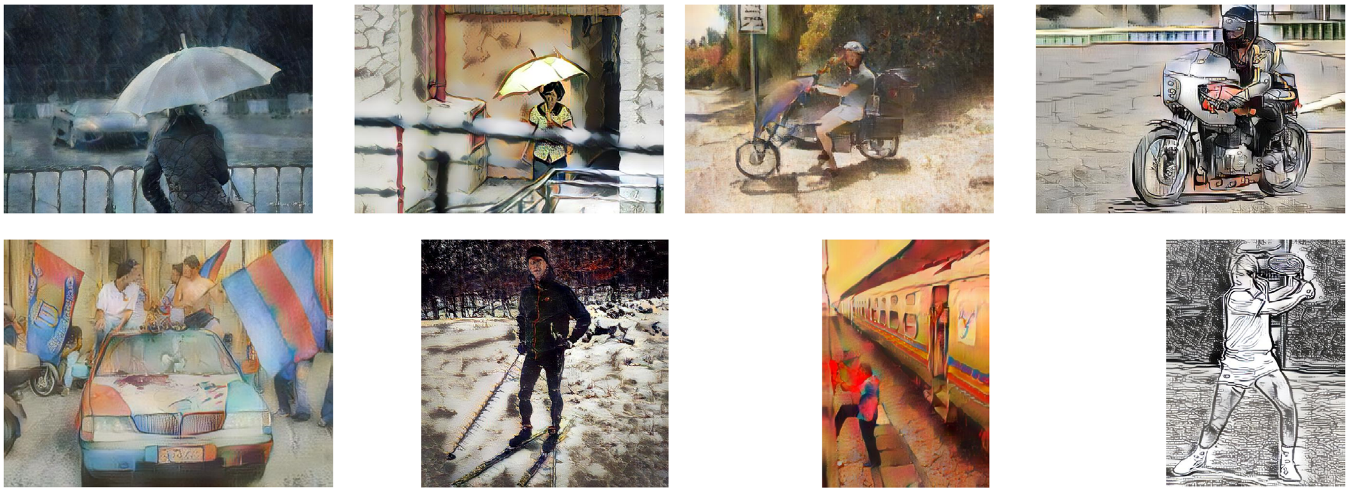


FIGURE 1 The StyleObject7K dataset includes images that are stylized in different ways, starting from "natural" images, as a proxy for conventional art styles, yet with a known underlying ground truth and visual features of object, which are depicted from a multitude of perspectives, do vary across different style transformations. The purpose of the dataset is to create a controlled challenge for our machine learning classifier.

In stylized images, including even fairly realistic artworks, object detection becomes particularly challenging as an object can be represented subject to what boils down to non-natural, often non-linear pictorial transformations or filters, for example, summarizing to conventional artistic styles such as Impressionism, and so on. The texture and visual features of the same object can furthermore drastically vary and change from one art style to another. In our study, we simulate such variance in a controlled way, using a set of image transformation over previously "natural" images, as shown in Figure 1. We specifically follow the style transfer work of [10], which establishes that neural networks prioritize image texture over its shape for classification of images.

For this work, our goal is to propose an implementation of YoloV5 with bootstrapping using gradient-boosting for cross-domain object detection in stylized art and non-photographic images. In general, the tasks associated with our research program may include:

- Artwork attribute prediction
- Object detection and recognition in artwork images
- Photo-realistic translation of artworks
- Fake artwork detection
- Emotion recognition in artwork images
- Visual Q&A and artwork captioning (iconographic enrichment)

Here, our work focuses on a single task of object detection in artwork images.

The target audience for our application are practitioners in the nexus of art research, visual culture research, yet also the deep learning and computer vision community more broadly. Section 2 of this paper summarizes relevant related literature. Section 3 explains our proposed methodology, while Section 4 discusses experimental details and results. Finally, in Section 5, we conclude our paper.

2 | RELATED WORK

Object detection refers to detecting and localizing objects in an image from pre-defined classes. As such, object detection tasks, like recognition and localization tasks, have widespread applications in real-world scenarios and can be considered to be an important sub-domain of computer vision. In this section, we briefly describe the related work and development in (i) Yolo object detection, (ii) object detection in artwork and stylized images, and (iii) boosting algorithms.

2.1 | YOLO object detection

Object detection is broadly categorized into (i) single-staged and (ii) two-staged object detectors. Single-staged anchor-free object detectors include Yolo [11] and SSD [12] architectures, whilst, two-staged object detectors [13–16] include region-proposal networks in the first stage and object classification in the second stage. The very first version of YOLO (You Only Look Once) was proposed by [11] as an end-to-end neural network for predicting bounding boxes and class labels at once. The proposed Yolo architecture was considered to be much faster over its contemporary methods and operating at up to 45 fps. The Yolo method divides the image into $N \times N$ grids, where each grid contributes to detecting and localizing the object it contains. Then on top of such grids, B number of bounding boxes are created. So-called non-maximum suppression is operating at the next stage, where the smaller bounding boxes are suppressed, and we are left with the only boxes which contain the entirety of recognized objects in an image. Yolo architecture is inspired by GoogleNet [17] and contains 24 convolutional layers and two fully-connected layers.

A second version of Yolo architecture was released as YoloV2 by [18] in order to overcome the limitations of the Yolo framework for small object detection and better localization accuracy.

A special architecture of YoloV2 was released as YOLO9000 which could detect more than 80 objects of the large benchmark MS COCO dataset [19] of around 330 k images of everyday objects and humans. YoloV2 also introduces batch normalization which increases mean Average Precision (mAP). YoloV3 [20] includes the 106-layered Darknet-53 as backbone architecture, with other modifications, such as having residual network and skip connections. YoloV3 detects features at three different scales and performs better than YoloV2 and Yolo in terms of small object detection. YoloV4 proposed by [21] resulted in a further improvement of YoloV3, claiming novelty by including Weighted Residual Connections, Cross Mini-batch Normalization, and Self-Adversarial Training. The YoloV4 tiny version operates at 65 frames per second, with a slight decrease in prediction accuracy. YoloV5 released by Ultralytics consists of a family of Yolo object detection models pre-trained on the MS COCO dataset and with Pytorch implementation. The versions higher than YoloV5 were not released at the time experimentation and this research work was carried out.

In literature, researchers have modified the original YoloV5 and proposed different variants for YoloV5 for various applications. Guo et al. [22] addressed the problem of road damage detection such that the backbone architecture of YoloV5 was replaced with MobileNetV3 for road damage detection. A very challenging task of Human action detection in drone images was carried out by [23] using YoloV5. Yan et al. [24] devised a practical application of YoloV5 where a small-scaled YoloV5s was deployed on an NVIDIA Jetson Xavier appliance for automatic pavement crack detection. A tiny-YoloV5 is proposed by [25] as light-weight deep learning model for Intelligent Edge Surveillance and the Internet of Things.

2.2 | Object detection in stylized images

Cai et al. [26] first attempted to detect cross-domain objects using a CNN model pre-trained on a large dataset of natural images and then fine-tuned on a smaller labelled artwork dataset. Crowley et al. [27] and [28] attempted to detect objects in paintings. Westlake et al. [29] proposes a new "People-Art" dataset for detecting human individuals in photos, cartoons, and different artwork images. The authors propose a CNN architecture and fine-tune it on the People-Art dataset. In [30], a weakly supervised object detector is proposed for paintings where only the image-level annotations are provided during training. The authors also propose an "IconArt" dataset where the model is trained to learn new classes which were not provided before or during the training. Smirnov et al. [31] used VGG-19 as network architecture for object detection in fine-art paintings.

2.3 | eXtreme gradient boosting

XGBoosting efficiently implements an ensemble of stochastic gradient boosting algorithms and has been a winning solution for several different machine learning competitions. The procedure of XGBoosting has been combined in different ways with convolutional neural networks. Khan et al. [32] introduced the new idea of channel-boosting in CNNs for better exploiting the transfer learning. Wu et al. [33] proposed boosted CNNs for enhancing the performance of pedestrian detection. Kalaivani et al. [34] implemented ensembles of boosted CNNs for segmentation of infected lungs in x-ray images. Memon et al. [35] established the superiority of XGBoosting over artificial neural networks for classification of urban and covered *areas* in satellite images.

3 | PROPOSED METHODOLOGY

For this research work, we collected a dataset of 7000 style images from the People-Art dataset [26], and we annotated these images for 10-different objects. We further train our proposed model for cross-domain object detection in the StyleObject images, going on testing the model on datasets of clipart, watercolors, and comic images, as proposed by [36]. At the first stage of the pipeline, our work includes YoloV5 architecture for detecting objects, while in the second stage True-negatives and False-positives are suppressed using "gradient boosting," which gives more emphasis on the difficult samples. We establish the validity of our method by using the above-mentioned datasets. Taken together, our work makes an attempt to take a step forward in developing computer vision algorithms that could learn more general representations of objects, leading to robust object detection in images and videos. The pipeline and flow of work for our proposed method is explained in Figure 2.

3.1 | YoloV5 for styleobject detection

YoloV5 is considered to be one of the most efficient models in the Yolo family with faster speed and less memory size. We will briefly discuss the key aspects of YoloV5, which makes it an excellent choice over other object detectors. YoloV5 is largely divided into three key components, including (i) a Backbone architecture, (ii) a feature-pyramid (PANet) as the Neck layer and (iii) prediction head layers for final object detection. A Cross Stage Partial Darknet (CSPDarknet53) as backbone architecture maximizes the functionality for feature aggregation. The resulting aggregated features are then passed on to the PANet in the neck layer. Meanwhile, the head convolutional layers generate predictions from the anchor boxes for object detection. CSPDarknet53 is efficient in solving the problem of repeated gradient information in backbone architecture by integrating gradient changes with feature maps and thus reducing the model parameters and FLOPs (floating-point operations per second). On the one hand, this parameter reduction decreases the storage size of the model, while on the other hand, it avoids the overfitting problem, thus increasing the average precision. The Path Aggregation Netwok (PANet) smoothly propagates low-level features in the neck layer by adopting a new Feature Pyramid Network (FPN) structure. At the same time in the lower layers of the feature pyramid, PANet accurately utilizes the

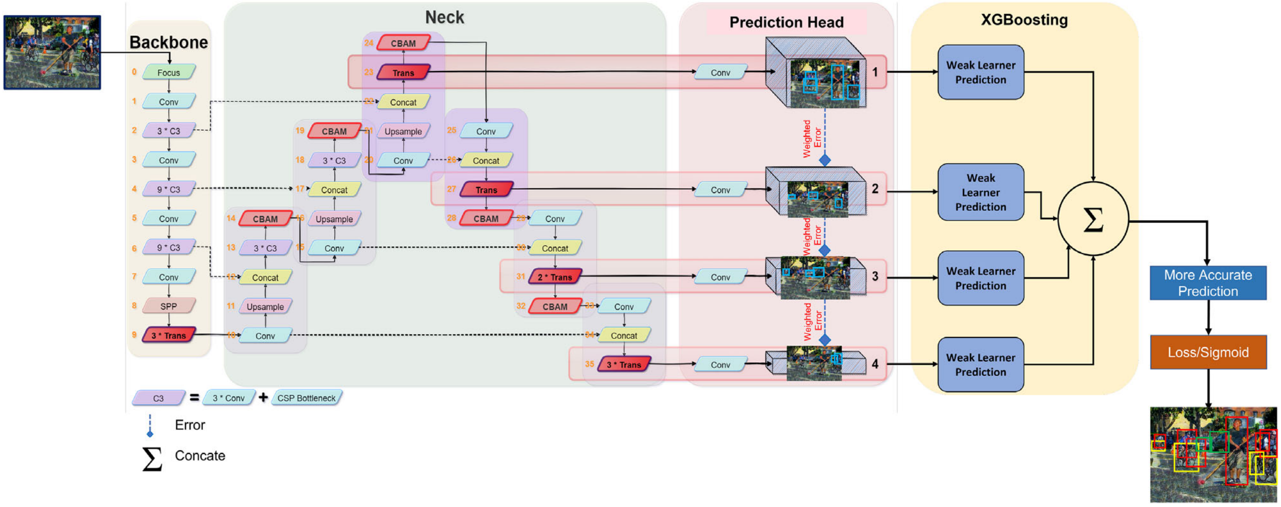


FIGURE 2 Our proposed methodology comprises of four blocks: (1) CSPDarkNet as backbone architecture; (2) Neck: PANet; (3) Prediction Head: Yolo layer; (4) XGBoosting. The data is first input to CSPDarkNet for extracting features, then these features are fused using PANet. The Prediction head computes the class prediction and localization scores. Finally, XGBoosting computes the loss and boosts the weight for different samples, while down-weighting easy samples.

localization signals which can be helpful to better locate objects in the image. Typically, shallow layers have higher feature dimension and are thus good at detecting low-level features, for example, edges and textures; meanwhile, deeper layers with a small-sized feature map detect high-level features as complex texture and shapes. In YoloV5, prediction heads generate four different size feature maps, and thus the model can predict small, medium and large objects at multiple scales, [20]. While dealing with artwork images using central perspective, for example, it could happen that distant objects look smaller while nearby objects look larger in artwork image, and therefore, a multi-scale prediction head can be beneficial to better handle such images. YoloV5 leverages Stochastic Gradient Decent (SGD) and ADAM for network optimization while harnessing binary cross-entropy as a loss-function during training. YoloV5 is an improvement to YoloV4 and has several advantages over previous Yolo versions for easy Pytorch setup installation, simpler directory structure and smaller storage size, [37]. In YoloV5, a genetic algorithm automatically learns the sizes of anchor-boxes while the previous Yolo versions take fixed size anchor boxes only.

There are five different versions available for YoloV5, which are YoloV5s, YoloV5n, YoloV5m, YoloV5l and YoloV5x. The working principle for these architectures are the same but they differ in their memory storage size. YoloV5x claims the largest memory size and YoloV5n is the smallest in storage size. For this research work, we experiment with three variants, including YoloV5s, YoloV5m and YoloV5l; as for now, there is no such storage constraint for object detection in artwork image applications.

3.2 | XGBoosting classifier

XGBoosting includes parallel computation by using parallel tree building. This makes sense as trees can easily be implemented using parallel processing in CPU and GPU machines. Contrary to other methods like Gradient Boosting Machines (GBM), XGBoosting implements “max-depth” which results in an improvement of computational performance. XGBoosting is robust to missing data values during training and can better handle sparse data [38]. Caruana et al. [39] carried out a comparative study for different machine learning algorithms like random forest, logistic regression, SVM and found that XGBoosting performs better than all algorithms used in comparison. These above-mentioned descriptions constitute the rationale for using XGBoosting for this research work.

The underlying principle of XGBoosting is that it assigns different weights to different observations, emphasizing weights for difficult to classify samples while assigning less weights to easily handled samples. Weak-learners are sequentially added in a manner for focusing the training on difficult to classify samples. XGBoosting comprises (i) a loss function for optimization, (ii) a weak learner classifier, and (iii) an additive model to add up the weak learners. XGBoosting can accommodate squared error for regression loss of localization and logarithmic loss for classification in object detection. We treated the above-mentioned YoloV5 as a weak classifier because its performance was slightly higher than 50% (slightly higher than random guess). Therefore, a weak detector (i.e. YoloV5) is combined with XGBoosting which reduces the detection loss. XGBoosting uses gradient descent for minimizing the loss function and subsequently updates the weights of each weak detector.

3.3 | Handling of difficult samples using XGBoosting

Training hard and complex samples using hard-mining is carried out by [40–42] using complex sampling and a re-weighting scheme. Lin et al. [43] proposed focal loss for mitigating the weights of well-classified samples and for focusing on hard samples during training.

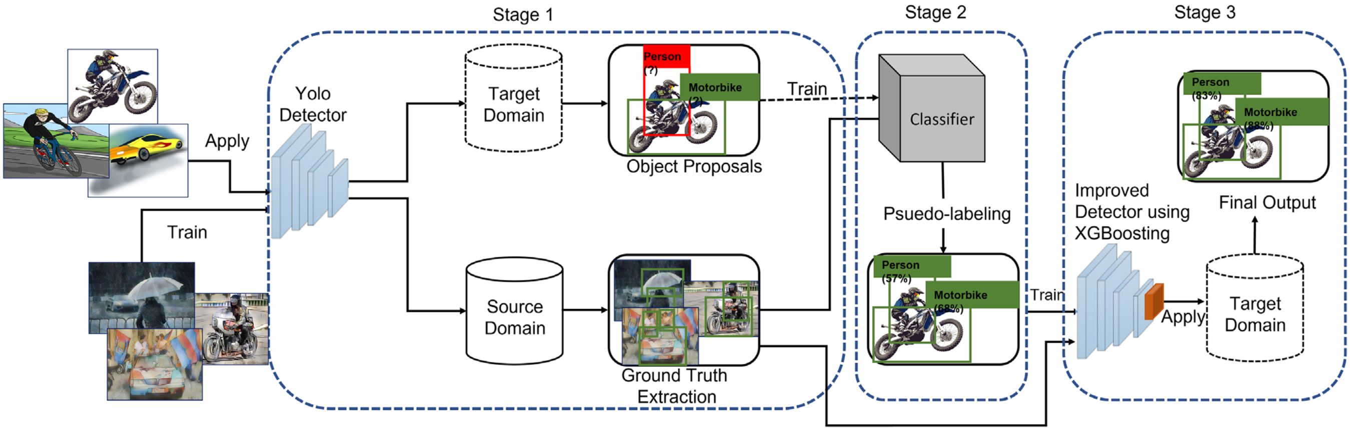


FIGURE 3 We explain our pipeline of work for the cross-domain object detection. At the first stage, object proposals are generated for the Target domain using a detector, which are fed to a classifier in the second stage for Pseudo-labelling. In the third stage, an improved detector predicts the final labels for the Target domain.

XGBoosting builds a new model which reduces the errors and residuals of previously built models. Using a bag-of-samples approach, XGBoosting computes the detection loss for difficult samples and then tries to build better learning by increasing the frequency of such difficult samples for better detection.

For difficult samples, it is hypothesized that they can be handled better by combing XGBoosting with weak learners. XGBoosting handles the difficult samples by imposing the constraints on a weak learner built to choose only a smaller number of features, f_{min}. Using the method this way, each such weak learner serves as the feature selection unit. A weak learner minimizes the mis-classification examples by determining the optimal threshold value. For a weak classification learner, $k_j(x)$, having a set of features, f_j, a threshold θ_j and a polarity p_j for the direction of inequality sign,

$$k_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0, & otherwise \end{cases} \tag{1}$$

where x is the size of feature map of the detector.

3.4 | Cross-domain StyleObject detection

We extend our work for cross-domain object detection where we train our model on the source-domain of the StyleObject7K dataset and then detect objects in the target-domains of the Clipart, Watercolor, and Comic2K datasets using pseudo-labelling. Domain adaptation attempts to align the source and target feature distributions such that the difference between two distributions is minimum in the high-dimensional feature space. The pipeline of our work is shown in Figure 3. For the StyleObject7K dataset as source-domain S, we provide images and annotations, but for the target domain datasets T only the images are provided (no labels in T).

In Figure 3, the first stage comprises two-parallel networks which detect object proposals for the target domain and ground truth for the source domain which mathematically is defined as,

$$P_S = f(S, W) \tag{2}$$

$$P_T = f(T, W) \tag{3}$$

The detection network f is identical and initialized with the same weight distribution W, for two different datasets as X_1 and X_2.

In the second stage, a classification model is trained using the ground truth prediction of S. This classifier generates the pseudo-labels for object proposals in T. The main reason for employing this classifier is to rely on representations different from the first stage detector, which may help the third stage detector.

4 | EXPERIMENTAL SETUP

4.1 | Datasets

StyleObject7K dataset: We trained our model on the StyleObject7K dataset, which contains 7000 stylized images. The dataset is annotated for 10 different object categories, including person, train, umbrella, car, horse, bike, motorbike, laptop, bus, and sheep. Among these 7000 images, 5000 images were used for training, while the remaining 2000 images were equally divided into the validation and test set. To provide an impression of this dataset, the number of instances for each category is listed in Table 1. There is a class imbalance in this dataset, with the person class dominating over other classes, as this dataset was originally derived from the People-Art dataset [26].

ClipArt1K: The Clipart1k dataset was devised by [44] with images collected from CMPlaces [45] and two other image search engines, Openclipart2 and Pixabay3. This dataset

TABLE 1 **The StyleObject7K dataset** contains 10 object classes ranging in frequency from hundreds to thousands of instances.

All	Person	Car	Umbrella	Motorbike	Bus	Horse	Laptop	Bike	Train	Sheep
19268	8910	2857	1472	1410	1342	1196	688	668	594	131

TABLE 2 The Detection Performance using the baseline model—This table presents the average and best values of **Precision**, **Recall** and **mAP** as the performance metrics for StyleObject7K dataset.

	Precision (%)	Recall (%)	mAP (%)
Average value	68.3±1.81	55.8±2.07	58.9±2.17

contains 1000 images, including a resemblance of eight object categories with the categories of the StyleObject7K dataset, including the classes of person, train, car, horse, bike, motorbike, and sheep which are found in both datasets.

Comic2k and Watercolor2k: Collected from a large dataset of 2.5 million images, Behance Artistic Media (BAM) [46], the Comic and Watercolor datasets were also created by [44]. They contain 17,814 watercolor images and 52,790 comic images, respectively. There is an overlap between three class categories (person, car, and bike) in the StyleObject7k, the Comic2k and the Watercolor2k datasets. From this, [44] have "randomly" collected and annotated 2000 images to constitute the Comic and Watercolor datasets.

4.2 | Implementation details

We run our experiments using the Pytorch machine learning library [47] with the Python 3.7 version. The experiments were run for a maximum of 200 epochs with an initial learning rate of 0.001 which was reduced by one-tenth after one-third of an epoch. The batch-size was set as 32 and sub-division as 2, whilst stochastic gradient descent (SGD) was used as optimization solver with a momentum of 0.9 and weight-decay of 0.0005. We trained our model with a Tesla P100-PCIE GPU and CUDA version 11.2, [48]. XGBoosting was trained using 200 trees of maximum depth of 5 and a learning rate of 0.0001. We implemented XGBoosting using the Scikit-learn python library.

We used Precision, Recall and mean Average Precision (mAP) as the metrics for measuring the performance of our proposed model for different datasets. mAP is the average of a series of scores at different IoU thresholds from 0.50 to 0.95 with a uniform step size of 0.05 for all the categories.

5 | RESULTS AND DISCUSSION

5.1 | Baseline results

We define our baseline model as trained on a randomly sampled training set and testing set. We excite our baseline model with 3-channel RGB-images. We report the performance for our baseline model in terms of Precision, Recall and mean average precision (mAP) in Table 2. Moreover, the baseline model is excited with the same hyper-parameters as for the rest of experiments. We present the visualization of detection results using the baseline model in Figure 4. At the same time, we also list average precision for each class using the baseline model in Table 3. The performance of our proposed model varies by varying the confidence score and this change in performance with the confidence score is shown in Figure 5. The highest class precision (67.2%) for all classes was obtained for a confidence score of 0.437.

5.2 | Testing on difficult samples of the StyleObject7K dataset

We evaluate the performance of our proposed model on difficult samples and manually identified the hard instances from the StyleObject7K dataset and put them in the test set. These hard samples were even difficult to correctly detect and recognize by humans. We trained our model on simple and easy to detect samples from the StyleObject7K dataset. In this ablation study, we explain that by training our model on easy samples, can even work well and detect objects in a test set comprising difficult samples. The quantitative results of this study are listed in Table 6. Meanwhile, the qualitative detection results on difficult samples of the StyleObject7K dataset are shown in Figure 8-a.

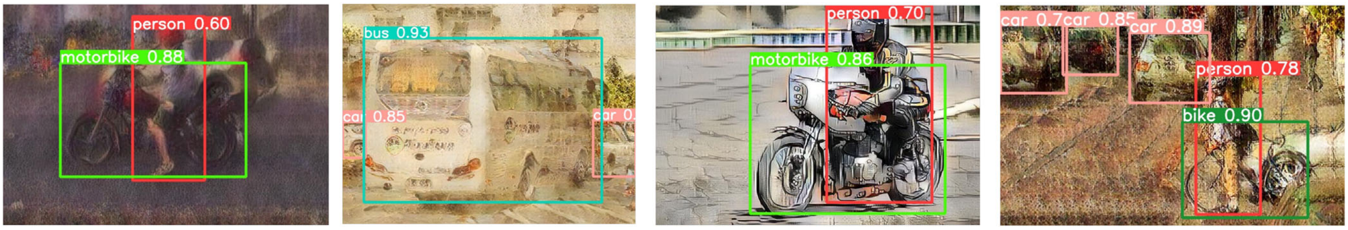


FIGURE 4 This figure presents Positive detection using the baseline model for the StyleObject7K dataset.

TABLE 3 Per-class Detection Performance—This table presents mAP for each class using the baseline model for the StyleObject7K dataset.

	mAP (%) for each class									
Person	Car	Train	Umbrella	Bus	Bike	M.Bike	Horse	Sheep	Laptop	mAP
0.82	0.68	0.72	0.65	0.78	0.64	0.59	0.57	0.63	0.75	0.683

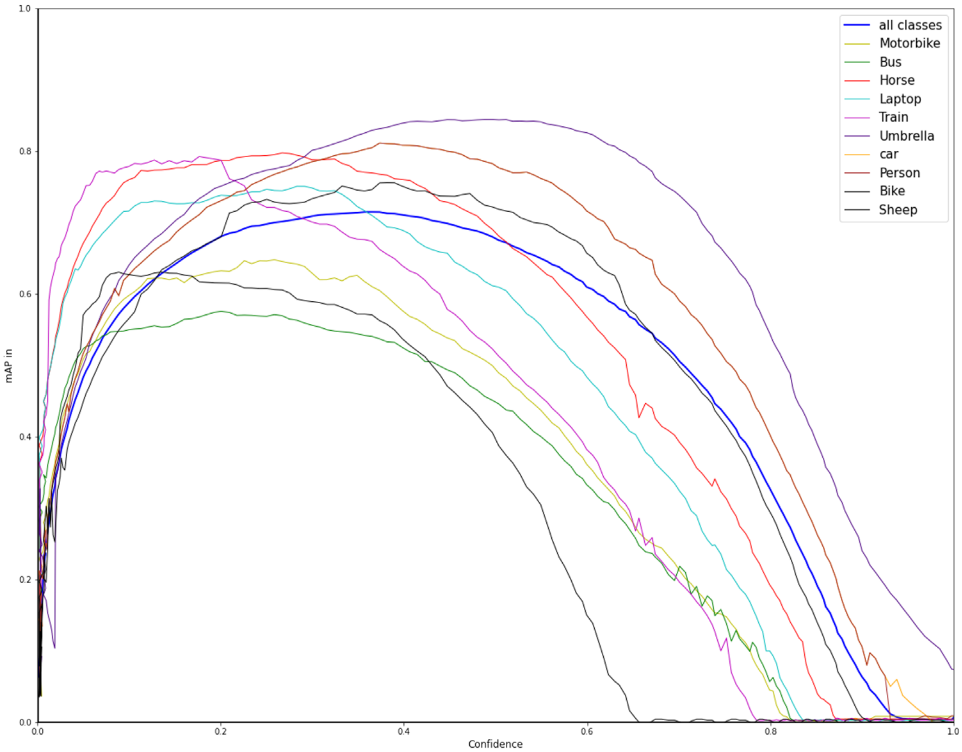


FIGURE 5 Confidence-versus-Precision plot—The detection performance varies for different values of confidence score for different classes in StyleObject7K dataset.

5.3 | Data augmentation

In deep learning, data augmentation techniques are helpful for improving the performance of datasets, which are small-sized and lacking diversity, thus resulting in high-bias and low-variance. Data augmentation makes sense in our case, since the datasets used throughout this study are probably not comparable to other commonly-used datasets in standard deep learning research, such as, the MNIST, the CIFAR dataset or the ImageNet dataset. Due to the nature of our datasets, data augmentation could be very helpful toward low-bias and high-variance, thus resulting in better generalization of the model for our test-set images. As images contain objects in different orientations shown in Figure 6, we identified and sorted out certain types of data transformations, such as rotation, translation, scaling, horizontal and vertical flip. Image rotation helped to augment the data by perturbing the angle in the range between -90° to +90° with an offset of 10°, and a small translation offset up to 5% of the patch size was added. The image was scaled up to to 50% of the image size, meanwhile the horizontal flip was more frequently exercised, since vertical flips appear less frequently (i.e. turning a human or a bus upside down completely.) The quantitative results using data augmentation are listed in Table 4.

5.4 | Cross-dataset generalization

In this experiment, we first train the model on the StyleObject7k dataset and then test its performance on the same and other datasets. In cross-dataset generalization, we realized that the StyleObject7K dataset performed better for the ClipArt1K dataset since it most closely resonates with the StyleObject7K dataset in terms of features, and moreover, there were seven classes similar in both datasets. The ClipArt1K and Comic2K datasets have good generalization for the Watercolor2K dataset due to visual similarity of images in both datasets. Training on Watercolor2K showed better generalization performance for the ClipArk1K dataset. It was quite obvious that our proposed method exhibited best results when trained and tested on the

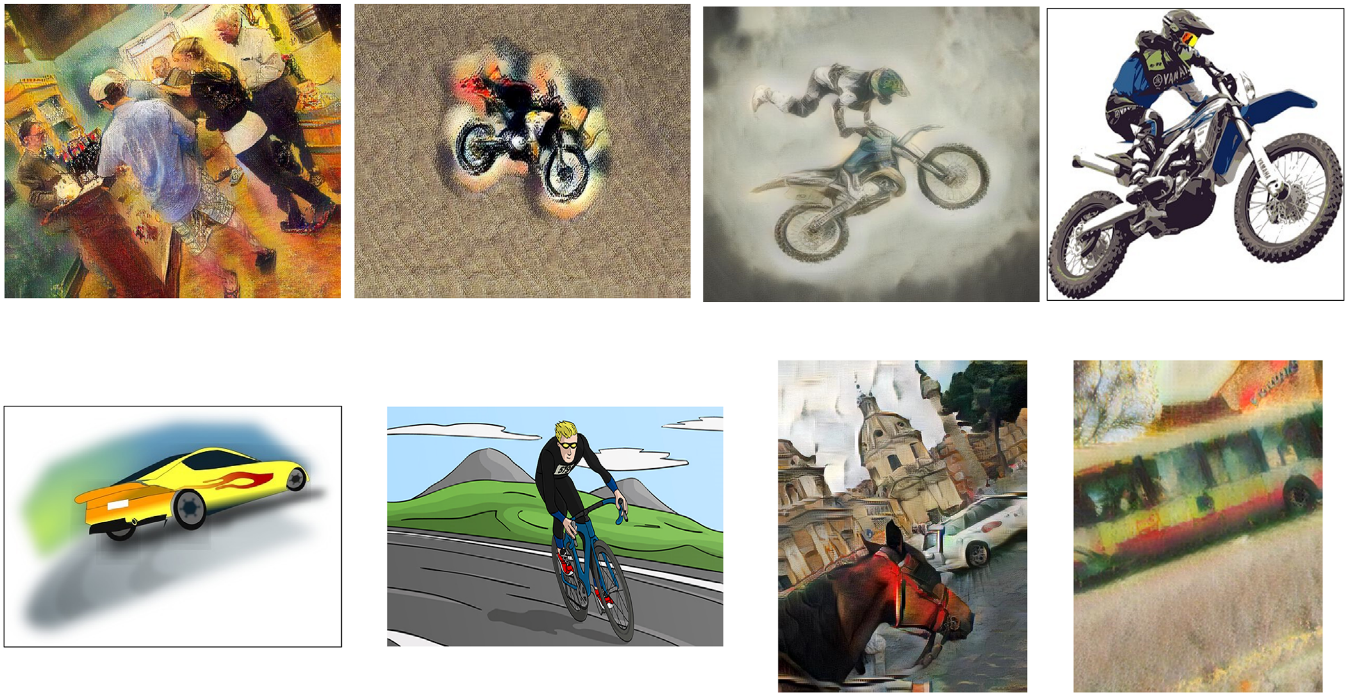
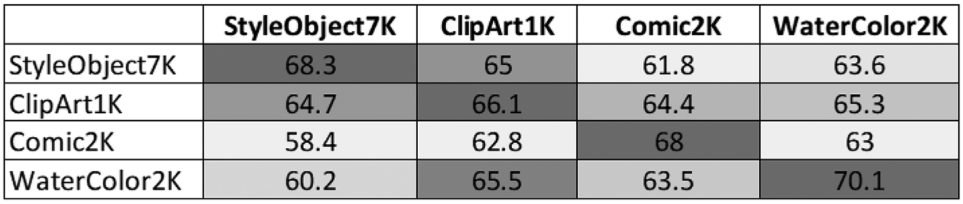


FIGURE 6 Data augmentation for various orientations—The data augmentation is required because artwork images may contain objects in different orientations.

TABLE 4 Detection performance for data augmentation—We present a performance comparison without and with data augmentation. We noticed a rise in performance metrics when we use data augmentation.

	Without augmentation			With augmentation		
	Precision	Recall	mAP	Precision	Recall	mAP
ClipArt1K	66.1±0.80	54.5±0.7	57.0±2.01	66.9±1.40	55.1±1.37	58.0±0.39
Comic2K	68.0±1.20	58.5±0.4	61.2±2.73	70.3±1.90	59.0±1.98	62.1±0.24
Watercolor2K	70.1±0.67	65.5±0.9	67.0±1.87	72.2±0.94	67.1±1.95	67.5±0.56



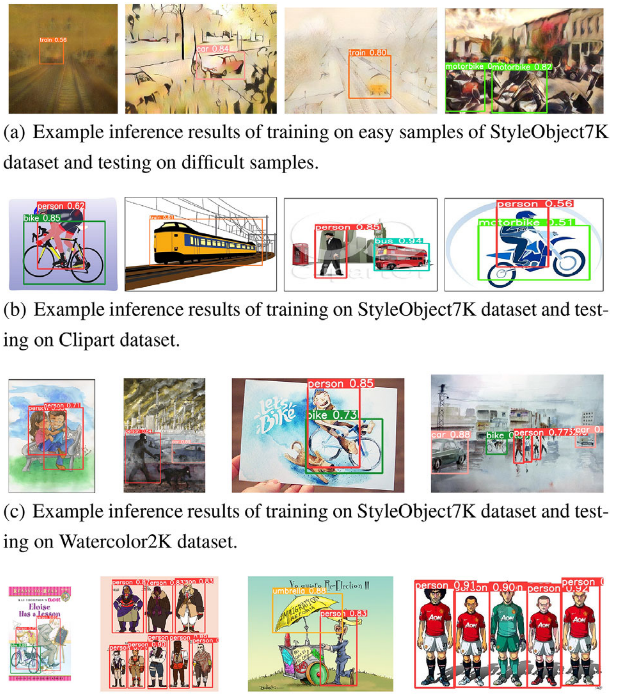
	StyleObject7K	ClipArt1K	Comic2K	WaterColor2K
StyleObject7K	68.3	65	61.8	63.6
ClipArt1K	64.7	66.1	64.4	65.3
Comic2K	58.4	62.8	68	63
WaterColor2K	60.2	65.5	63.5	70.1

FIGURE 7 Cross-dataset performance comparison—Our proposed deep learning model was trained on one dataset and then validated on all other datasets.

same dataset. A cross-dataset generalization performance for aforementioned datasets is shown in Figure 7.

5.5 | Full StyleObject12K dataset

We move a step further and aggregate the images from all the aforementioned datasets and then divided the images into training, validation and testing sets with a ratio of 70%, 15% and 15% respectively. The main objective of this experiment is to mix and incorporate the features from different style image datasets and then train the model for achieving a better generalization on different datasets, since each one of them contains some particular genre of artwork images. This unified dataset contains 12,000 images in total, with 7000 images from StyleObject7K, 1000 images from ClipArt1K, 2000 images from Comic2K and 2000 images from Watercolor2K. During training, feature aggregation was carried out by shuffling the input mini-batch based



(a) Example inference results of training on easy samples of StyleObject7K dataset and testing on difficult samples.

(b) Example inference results of training on StyleObject7K dataset and testing on Clipart dataset.

(c) Example inference results of training on StyleObject7K dataset and testing on Watercolor2K dataset.

(d) Example inference results of training on StyleObject7K dataset and testing on Comic dataset.

FIGURE 8 We trained the model on StyleObject7K dataset, while tested on different artwork image datasets.

TABLE 5 The detection performance on the full dataset—The network is trained on a holistic dataset by mixing all the images from the StyleObject7K, ClipArt1K, Comic2K and WaterColor2k datasets, followed by randomly sampling images in each mini-batch. The detection performance is reported in terms of **Precision**, **Recall** and **mAP**.

	Precision (%)	Recall (%)	mAP (%)
Average value	69.7±1.54	57.8±4.47	60.6±3.71

TABLE 6 The detection performance on difficult samples—The network is trained using easy samples of the StyleObject7K dataset while tested on the difficult samples of the StyleObject7K dataset. The detection performance is calculated in terms of **Precision**, **Recall** and **mAP**.

	Precision (%)	Recall (%)	mAP (%)
Average	67.2±2.87	53.2±3.26	57.3±3.99

on attribute labels and then randomly selecting samples from the input and shuffled mini-batches. Our proposed method performed well, and the results are listed in Table 5.

5.6 | Ablation study

We carry out ablation studies where first we compare the performance of different YoloV5 architectures, and then the second study entails the significance of XGBoosting architecture in our proposed methodology.

Comparison between YoloV5 architectures:
We trained our model on three different Yolo versions, that is, YoloV5n, YoloV5m and YoloV5l. The YoloV5n architecture is the smallest in size and is an ideal choice for real-time embedded applications. YoloV5m performs slightly better than YoloV5n, but it requires larger model storage and computational power. YoloV5l performs even better than YoloV5m, but definitely requires larger storage size and computational resources. As far as the current circumstances are concerned, use cases for real-time applications of art-work image detection or segmentation may be rare, but in the future may be required as dynamic art and computer animation is increasingly automated. Consequently, more compact Yolo models, which could be applied in such real-time applications are likely to become desirable. Indeed, here lies a major possible application for our approach, as generative art may produce images with objects, whose creation, performance, and appreciation may depend on fast automated object detection as introduced here. A comparison using different Yolo architectures is presented in Table 7.

Comparison with/without XGBoosting:
We further make a copy of our proposed architecture by excluding XGBoosting and carry out an ablation study by comparing the performance with and without the XGBoosting block. By running our model, we noticed that using XGBoosting results in the rise in average precision because it helps to suppress false-positives and thus results in the rise in performance. A quantitative comparison for the rise in accuracy is presented in Table 8.

5.7 | Comparison with the state-of-the-art

The StyleObject7K dataset can function as a newly proposed public benchmark dataset for the evaluation of deep learning models in style images, including artworks. We compare our proposed XGBoosting-YoloV5 method other methods for the StyleObject7K dataset and other ClipArt1K, Comic2K and Watercolor2K datasets in Table 9. Bilen et al. [49] proposed a weakly supervised object detector, which was originally pre-trained on the PASCAL VOC dataset when fine-tuned for the ClipArt1K, Watercolor2K and Comic2K datasets. They produce preliminary results of mean average precision. Context-aware deep models were proposed by [50], which performed slightly better than [49], due to additive and contrastive models that leverage the surrounding context regions of the objects to improve localization. More recently, adversarial models have performed very well. The authors in[51], for example, have devised Adversarial Discriminative Domain Adaptation, which combines discriminative modelling, untied weight sharing, and GAN loss. This methodology resulted in a significant increase in detection performance for the ClipArt1K, Watercolor2K and the Comic2K datasets. Inoue et al. [44] implemented a weakly-supervised object detection using domain-adaptation, arguing that source and target domains differ due to their low-level features, such as color and texture. This problem is overcome by generating images similar to the target domain using CycleGAN

TABLE 7 The detection performance for different Yolo architectures on the Full StyleObject12K dataset—The parameters are measured in millions (M), average precision is measured in %, training time is measured in hours, and model size is measured in MB. In the names of YoloV5, the subscripts "n," "m," and "l" refer to nano, medium, and large networks.

	Layers	Parameters	Precision (%)	Recall (%)	mAP (%)	Time	Size
YoloV5n	213	1.75	64.3	55.0	61.7	5.5	3.7
YoloV5m	290	20.1	69.7	58.8	63.4	12.2	71.0
YoloV5l	367	44.5	72.1	60.1	65.0	17.1	92.1

TABLE 8 The detection performance without/with XGBoosting—Using XGBoosting with YOLOV5 results in substantial rise in Precision, Recall and mAP.

	Precision (%)	Recall (%)	mAP (%)
Without XGBoosting	68.3	55.8	58.9
With XGBoosting	78.5	57.8	63.4

[52], and fine-tuning the model on such images. This novel idea works and results in significant rise in performance for the ClipArt1K, Watercolor2K and Comic2K datasets. Our proposed work inherently incorporates advantages of the aforementioned methods, as the YoloV5 framework implicitly includes adversarial training, while XGBoosting trains difficult samples by giving them more weights, compelling the model to learn better. This methodology suppresses false-positives and correspondingly increases true-positives. Consequently, our method improves overall detection accuracy for the above-mentioned datasets.

Applying a weakly supervised object detector [49] on the StyleObject7K dataset results in 55.6% mAP, which was surpassed with 60.7% when a context-aware deep network [50] was used. Domain adaptation combined with weakly supervised learning resulted in a further rise and raised the mAP to 62.9%. Through quantitative comparison in Table 9, it is evident that our proposed XGBoosting-YoloV5 performs better than other methods for the newly proposed StyleObject7K dataset.

5.8 | Applications

Our research on object detection in stylized images, as presented here, can be helpful toward art image understanding, as well as for art image to text generation and captioning. The

TABLE 9 A comparison of our proposed methodology with the contemporary methods on Stylized and Artwork image datasets.

	Clipart1K - Average Precision (%) for each class										
	Person	**Car**	**Train**	**Bus**	**Bike**	**Motor-Bike**	**Horse**	**Sheep**	**Umbrella**	**Laptop**	**mAP**
Weakly-Sup DDN, [56]	14.4	4.5	1.2	11.7	3.6	0.1	0.9	4.5	–	–	5.1
Contextlocnet, [57]	12.5	17.5	8.0	4.8	22.3	0.6	4.7	14.1	–	–	10.6
Adversarial Domain Adapt., [58]	46.6	34.9	23.6	40.5	50.2	53.6	31.7	18.0	–	–	37.4
X-domain weakly sup, [59]	61.1	44.0	38.4	53.0	60.1	62.2	40.4	20.9	–	–	47.5
Proposed	82.0	68.3	72.4	75.8	63.0	64.4	78.7	59.9	–	–	**70.6**
	Watercolor2K—Average precision (%) for each class										
Weakly-sup DDN, [56]	33.3	14.6	–	–	1.5	–	–	–	–	–	16.5
Contextlocnet, [57]	31.4	19.6	–	–	4.5	–	–	–	–	–	18.5
Adversarial Domain Adapt., [58]	65.1	39.5	–	–	79.9	–	–	–	–	–	61.5
X-domain weakly sup, [59]	62.5	40.2	–	–	82.8	–	–	–	–	–	61.8
Proposed	75.0	72.1	–	–	63.2	–	–	–	–	–	**70.1**
	Comic2K—Average precision (%) for each class										
Baseline, [59]	42.6	19.4	–	–	43.9	–	–	–	–	–	35.3
X-domain Weakly-sup, [59]	48.3	30.2	–	–	43.6	–	–	–	–	–	40.7
Proposed	73.4	69.6	–	–	61.0	–	–	–	–	–	**68.0**
	StyleObject7K—Average precision (%) for each class										
Weakly-sup DDN, [56]	73.7	55.8	52.4	41.4	42.7	39.8	47.8	38.2	59.9	49.1	50.1
Context-locnet, [57]	78.0	65.5	60.1	49.0	61.1	45.8	43.3	41.7	58.6	47.9	55.3
X-domain weakly sup, [59]	80.6	63.2	68.4	65.0	62.0	55.6	54.5	55.0	60.6	64.8	62.9
Proposed	82.4	68.0	72.2	78.1	64.4	59.6	57.0	62.7	63.6	74.6	**68.5**

proposed object detection and localization in non-photographic images can extract metadata, that promises to be useful in the domain of cultural heritage and museums [53, 54], the art market and online collections [55]. The proposed object detection in stylized images can be extended to include different object categories, as found in the MS COCO dataset. However, [26] noticed that human individuals are over-represented in art images, and sometimes, it may become challenging to search for other object categories with sufficient numbers of examples. Our work can further be improved via boot-strapping, for example through harnessing large-scale (art) image collections that are annotated via crowd-sourcing or through visual resource librarians, feeding into final machine learning model adjustments and corrections. The results from studying computer animation and video game content, while using our method, further promises advances, which in turn can be further applied toward a deeper understanding of more static visual products of cultural heritage and art.

6 | CONCLUSION

In this research work, we proposed a refined method for cross-domain object detection in stylized images including stylized natural images, clipart, watercolor, and comic images. We investigate YoloV5 for this purpose, where the Yolo model is trained on the StyleObject7K dataset using gradient-boosting and then evaluated on the ClipArt1K, Watercolor2K, and Comic2K datasets for cross-domain detection. We performed thorough experimentation, carried out a detailed ablation study and compared our results with other state-of-the-art methods, which establishes that the proposed model performs better. In future work, we aim to extend the StyleObject7K dataset and will include more images in it. Furthermore, we plan to extend our studies to include artwork images, such as a 65K benchmark abstract art of Art500K and WikiArt, as used in another stream of work within our research group [60]. Finally, we also plan to advance our investigation through the inclusion of more recent and emerging deep learning models.

AUTHOR CONTRIBUTIONS

This research responds to a research challenge of object detection in stylized images, toward eventually making sense of historical artworks at scale. This challenge was initially raised by Maximilian Schich (M.S.). Tasweer Ahmad (T.A.) designed and performed the research for this study, including preparation of data, literature survey, performance of data analysis, experiments, figure creation, and writing of the manuscript. M.S. provided conceptual guidance and comments regarding research design, data analysis, and figure design. M.S. has proofread the manuscript.

ACKNOWLEDGEMENTS

The authors thank Mar Canet for support, and all members of the CUDAN Research Group for useful comments. The work of Tasweer Ahmad and Maximilian Schich is fully supported as part of the CUDAN ERA Chair project for Cultural Data Analytics, funded through the Horizon 2020 research and innovation program of the European Commission (Grant No. 810961). Thumbnail images are taken from the datasets as cited in the manuscript, and depicted throughout the paper claiming fair use.

CONFLICT OF INTEREST STATEMENT

All authors declare that they have no conflicts of interest and grant permission to re-produce the results.

DATA AVAILABILITY STATEMENT

Code and data are available from the first author upon request.

ORCID

Tasweer Ahmad https://orcid.org/0000-0002-8108-7915

REFERENCES

1. 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 Proc. European Conf. on Computer Vision, pp. 740–55. Springer, Cham (2014)
2. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE, Piscataway, NJ (2012)
3. Everingham, M., Eslami, S.M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision 111(1), 98–136 (2015)
4. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: a face detection benchmark. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5525–5533. IEEE, Piscataway, NJ (2016)
5. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: large-scale image retrieval benchmarking. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5706–5715. IEEE, Piscataway, NJ (2018)
6. Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. 41(7), 1655–1668 (2018)
7. Seguin, B., diLenardo, I., Kaplan, F.: Tracking Transmission of Details in Paintings. In DH (2017)
8. Shen, X., Efros, A.A., Aubry, M.: Discovering visual patterns in art collections with spatially-consistent feature learning. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 9278–9287. IEEE, Piscataway, NJ (2019)
9. Milani, F., Pinciroli Vago, N.O., Fraternali, P.: Proposals Generation for Weakly Supervised Object Detection in Artwork Images. J. Imag. 8(8), 215 (2022)
10. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 779–788. IEEE, Piscataway, NJ (2016)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In Proc. European Conf. on Computer Vision, pp. 21–37. Springer, Cham (2016)
13. Girshick, R.: Fast r-cnn. In Proc. IEEE Int. Conf. on Computer Vision, pp. 1440–1448. IEEE, Piscataway, NJ (2015)
14. 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, vol. 28. Curran Associates, Red Hook, NY (2015)

15. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Red Hook, NY (2016)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In Proc. IEEE Int. Conf. on Computer Vision, pp. 2961–2969. IEEE, Piscataway, NJ (2017)
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–9. IEEE, Piscataway, NJ (2015)
18. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 7263–7271. IEEE, Piscataway, NJ (2017)
19. 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 Proc. European Conf. on Computer Vision, pp. 740–755. Springer, Cham (2014)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
22. Guo, G., Zhang, Z.: Road damage detection algorithm for improved YOLOv5. Sci. Rep. 12(1), 1–2 (2022 Sep 15)
23. Ahmad, T., Cavazza, M., Matsuo, Y., Prendinger, H.: Detecting human actions in drone images using YOLOv5 and stochastic gradient boosting. Sensors 22(18), 7020 (2022)
24. Yan, X., Shi, S., Xu, X., He, Z., Zhou, X., Wang, C., Lu, Z.: An automatic pavement crack detection system with FocusCrack Dataset. In Proc. IEEE Vehicular Technology Conference, pp. 1–5. IEEE, Piscataway, NJ (2022)
25. Zhao, Y., Yin, Y., Gui, G.: Lightweight deep learning based intelligent edge surveillance techniques. IEEE Trans. Cognit. Commun. Networking 6(4), 1146–1154 (2020 Jun)
26. Cai, H., Wu, Q., Corradi, T., Hall, P.: The cross-depiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. arXiv preprint arXiv:1505.00110 (2015)
27. Crowley, E.J., Zisserman, A.: In search of art. In Proc. European Conf. on Computer Vision, pp. 54–70. Springer, Cham (2014)
28. Crowley, E.J., Zisserman, A.: The art of detection. In Proc. European Conf. on Computer Vision, pp. 721–737. Springer, Cham (2016)
29. Westlake, N., Cai, H., Hall, P.: Detecting people in artwork with CNNs. In Proc. European Conf. on Computer Vision, pp. 825–841. Springer, Cham (2016)
30. Gonthier, N., Gousseau, Y., Ladjal, S., Bonfait, O.: In European Conf. on Computer Vision Workshops. Lecture Notes in Computer Science, pp. 692–709. Springer, Cham (2019)
31. Smirnov, S., Eguizabal, A.: Deep learning for object detection in fine-art paintings. In: 2018 Metrology for Archaeology and Cultural Heritage (MetroArchaeo), pp. 45–49. IEEE, Piscataway, NJ (2018, October)
32. Khan, A., Sohail, A., Ali, A.: A new channel boosted convolutional neural network using transfer learning. arXiv preprint arXiv:1804.08528 (2018)
33. Wu, C.H., Gan, W., Lan, D., Kuo, C.C.J.: Boosted convolutional neural networks (BCNN) for pedestrian detection. In Proc. IEEE Winter Conf. on Applications of Computer Vision, pp. 540–549. IEEE, Piscataway, NJ (2017)
34. Kalaivani, S., Seetharaman, K.: A three-stage ensemble boosted convolutional neural network for classification and analysis of COVID-19 chest x-ray images. Int. J. Cogn. Comput. Eng. (3), 35–45 (2022)
35. Memon, N., Patel, S.B., Patel, D.P.: Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification. In Proc. Int. Conf. Pattern Recognition and Machine Intelligence, pp. 452–460. Springer, Cham (2019)
36. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 5001–5009. IEEE, Piscataway, NJ (2018)
37. Wang, H., Zhang, S., Zhao, S., Wang, Q., Li, D., Zhao, R.: Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++. Comput. Electron. Agric. 192, 106512 (2022)
38. Memon, N., Patel, S.B., Patel, D.P.: Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification. In Proc. Int. Conf. on Pattern Recognition and Machine Intelligence, pp. 452–460. Springer, Cham (2019)
39. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. Proc. Int. Conf. on Machine Learning, pp. 161–168. ACM, New York (2006)
40. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 761–769. IEEE, Piscataway, NJ (2016)
41. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. I–I. IEEE, Piscataway, NJ (2001)
42. Rota Bulo, S., Neuhold, G., Kontschieder, P.: Loss max-pooling for semantic image segmentation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2126–2135. IEEE, Piscataway, NJ (2017)
43. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In Proc. IEEE Int. Conf. on Computer Vision, pp. 2980–2988. IEEE, Piscataway, NJ (2017)
44. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5001–5009. IEEE, Piscataway, NJ (2018)
45. Castrejon, L., Aytar, Y., Vondrick, C., Pirsiavash, H., Torralba, A.: Learning aligned cross-modal representations from weakly aligned data. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2940–2949. IEEE, Piscataway, NJ (2016)
46. Wilber, M.J., Fang, C., Jin, H., Hertzmann, A., Collomosse, J., Belongie, S.: Bam! the behance artistic media dataset for recognition beyond photography. In Proc. IEEE Int. Conf. on Computer Vision, pp. 1202–1211. IEEE, Piscataway, NJ (2017)
47. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Red Hook, NY (2019)
48. CUDA Toolkit-Develop, Optimize and Deploy GPU-Accelerated Apps. https://developer.nvidia.com/cuda-toolkit. Accessed 24 April 2022
49. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2846–2854. IEEE, Piscataway, NJ (2016)
50. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: Contextlocnet: Context-aware deep network models for weakly supervised localization. Proc. European Conf. on Computer Vision, pp. 350–365. Springer, Cham (2016)
51. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 7167–7176. IEEE, Piscataway, NJ (2017)
52. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proc. IEEE Int. Conf. on Computer Vision, pp. 2223–2232. IEEE, Piscataway, NJ (2017)
53. RIJKS Museum. https://www.rijksmuseum.nl/en. Accessed 15 November 2022
54. Louvre Museum. https://www.louvre.fr/en. Accessed 15 November 2022
55. Art made by Artificial Intelligence. https://aimade.art/ accessed 17 November 2022
56. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In Proc. IEEE conf. on Computer Vision and Pattern Recognition, pp. 2846–2854. IEEE, Piscataway, NJ (2016)
57. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: Contextlocnet: Context-aware deep network models for weakly supervised localization. In Proc. European Conf. on Computer Vision, pp. 350–365. Springer, Cham (2016)
58. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 7167–7176. IEEE, Piscataway, NJ (2017)

59. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5001–5009. IEEE, Piscataway, NJ (2018)
60. Karjus, A., Solá, M.C., Ohm, T., Ahnert, S.E., Schich, M.: Compression ensembles quantify aesthetic complexity and the evolution of visual art. arXiv preprint arXiv:2205.10271 (2022)

How to cite this article: Ahmad, T., Schich, M.: Toward cross-domain object detection in artwork images using improved YoloV5 and XGBoosting. IET Image Process. 17, 2437–2449 (2023). https://doi.org/10.1049/ipr2.12806