The Art of Detection

Elliot J. Crowley(✉) and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science,
University of Oxford, Oxford, UK
{elliot,az}@robots.ox.ac.uk

Abstract. The objective of this work is to recognize object categories in paintings, such as cars, cows and cathedrals. We achieve this by training classifiers from natural images of the objects. We make the following contributions: (i) we measure the extent of the *domain shift* problem for image-level classifiers trained on natural images vs paintings, for a variety of CNN architectures; (ii) we demonstrate that classification-by-detection (i.e. learning classifiers for regions rather than the entire image) recognizes (and locates) a wide range of small objects in paintings that are not picked up by image-level classifiers, and combining these two methods improves performance; and (iii) we develop a system that learns a region-level classifier *on-the-fly* for an object category of a user's choosing, which is then applied to over 60 million object regions across 210,000 paintings to retrieve localised instances of that category.

1 Introduction

"It is of the highest importance in the art of detection to be able to recognize out of a number of facts which are incidental and which vital. Otherwise your energy and attention must be dissipated instead of being concentrated."

– Sherlock Holmes, "The Reigate Puzzle"

The ability of visual classifiers to label the content of paintings is of great benefit to art historians, as it allows them to spend less time arduously searching through paintings looking for objects to study, and more time studying them. However, such visual classifiers are generally trained on natural images, for which there is a copious amount of annotation (and which is often lacking for paintings). Unfortunately, as Hall *et al.* observe [24], there is a drop in performance in training on natural images rather than paintings. So we ask, when it comes to classifying paintings using natural images as training data, *what are we missing?*

We investigate the answer to this question from two directions: first, by measuring quantitatively the *domain shift* problem for image-level classifiers, and second, by looking at what is missed by image-level classifiers, but not missed by *detectors*.

The task of interest here is image classification – classifying an image by the objects it contains. With increasingly powerful image representations provided by each generation of Convolutional Neural Networks (CNNs) there has been a

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part I, LNCS 9913, pp. 721–737, 2016.
DOI: 10.1007/978-3-319-46604-0_50

steady increase in performance over a variety of challenging datasets of natural, photographic images [17,31,34] (and for a variety of tasks [15,22,29,32]). It has been shown that these representations transfer well between domains such as between DSLR and webcam images [39], natural images and sketches [43] and of particular interest to us, between images and paintings [11,12].

Our first contribution is to compare image-level-classifiers (i.e. representing an entire image by a single vector) trained on natural images to those trained on paintings at the task of painting classification. This allows us to observe how severe the drop in performance is for different architectures when they have to cope with domain shift.

It transpires that a major shortcoming of natural image-trained, image-level-classifiers is their inability to retrieve very small objects in paintings. Small objects are particularly prevalent in paintings, for example: animals dotted across the countryside in landscape scenes; boats that are often small regions in a seascape; and aeroplanes that are sometimes little more than a speck in the sky. Our second contribution is to demonstrate that classification-by-detection (i.e. finding regions in an image and classifying them) finds such objects for a variety of classes. We also show that combining the two methods, image-level classification and classification-by-detection, leads to improved performance.

Finally, we build upon the detector by contributing a system that detects an object of a user's choosing in paintings *on-the-fly*. The system downloads natural images from the web and learns a region-level classifier. This classifier is applied to over 60 million regions across 210,000 paintings to retrieve a large range of objects with high precision. The detected objects are given in their paintings with a bounding box, allowing for easy comparison of objects. We evaluate this system for many different queries.

The paper is organized as follows: Sect. 3 describes the datasets of natural images and paintings used in this work. An evaluation of painting classification for different network architectures with natural image-trained and painting-trained image-level classifiers is carried out in Sect. 4. In Sect. 5 object detectors are utilised for the retrieval of small objects in paintings. Lastly, we describe the on-the-fly learning system for detecting objects in paintings in Sect. 6.

2 Related Work

Domain Adaptation. There is a wealth of literature on adapting hand-crafted (i.e. shallow) features between domains (from a source domain to a target domain): Daumé [14] augments the feature space of source and target data. Others [27,30] have re-weighted source samples based on target sample similarity. Saenko *et al.* [35] map source and target data to a domain-invariant subspace; several later works build upon this idea [19,23,26,38]. For deep learning, Ganin and Lempitsky [20] incorporate a branch in their network architecture that classifies an input sample as being from one of two domains. The resulting loss is back-propagated, and then reversed before being passed to the original network to maximise domain confusion – the idea being that this should create

a domain-invariant network. Tzeng *et al.* [39] learn domain invariant representations by adding two losses to their network architecture: (i) a loss based on a domain classifier similar to [20], and (ii) a 'domain confusion' loss that forces samples from different domains to appear similar to the network. Aljundi and Tuytelaars [3] propose a method that identifies those filters in the first convolutional layer of a network that are badly affected by domain shift. These filters are then reconstructed from filters less affected by the domain shift in order to achieve domain adaptation.

Natural Images to Paintings. In the vast majority of the domain adaptation literature, the source and target data both consist of natural images. Evaluation is mainly carried out on the 'Office Dataset' [35] where the domains in questions are images taken with a DSLR camera, a webcam, and images from the Amazon website. There is however, work on the specific problem of learning from natural images and retrieving paintings: Shrivastava *et al.* [36] use an Exemplar-SVM [28] to retrieve paintings of specific buildings. Aubry *et al.* [5] improve on this by utilising mid-level discriminative patches, the patches in question demonstrating remarkable invariance between natural images and paintings. Our previous work [13] demonstrates that this patch-based method can be extended to object categories in paintings beyond the instance matching of [5]. Others [41,42] have considered the wider problem of generalising across many depictive styles (e.g. photo, cartoon, painting) by building a depiction-invariant graph model. Cai *et al.* [6] utilise query expansion to refine a DPM [18] model learnt on natural images with confident detections found on artwork.

3 Datasets

In this section we describe the datasets used in the paper: one of natural images, that will be used for training the *source* image classifiers and detectors; and the other of paintings that will be used to provide the *target* training images, and also a test set. The statistics for these datasets are given in Table 1.

3.1 Paintings

The `Paintings Dataset` introduced in [13], and available at the website [2], is a subset of the publicly available 'Art UK' dataset [1] – over 200,000 paintings from British galleries, of different styles and eras (formerly known as 'Your Paintings') – for which each painting is annotated for the occurrence of 10 classes – aeroplane, bird, boat, chair, cow, dining-table, dog, horse, sheep, train. The annotation is complete in the PASCAL VOC [16] sense – in that every occurrence is annotated at the image-level. These classes were chosen because they are all present in PASCAL VOC (used for the natural image dataset below), allowing us to assess the domain shift problem between the two datasets directly by class. Example images of the dataset are shown in Fig. 1.

Fig. 1. Example class images from the `Paintings Dataset`. From top to bottom row: aeroplane, cow, sheep. Notice that the dataset is particularly challenging: objects can be large or minuscule, may often be occluded, and are depicted in a large variety of styles such as photo-realistic, abstract and impressionist.

The entire 'Art UK' dataset [1] is used in the on-the-fly system (Sect. 6) to provide the variety required for general searches. This dataset consists of over 210,000 oil paintings.

3.2 Natural Images

The `VOC12` dataset is the subset of PASCAL VOC 2012 [16] TrainVal images that contain any of the 10 classes. Only 10 of the 20 VOC classes are used, because the 'Art UK' dataset does not have a sufficient number of annotated examples of the other 10 classes.

4 Domain Adaptation

In this section, we compare image-level classifiers trained on features from natural images (`VOC12`) to classifiers trained on features from paintings (the training set of the `Paintings Dataset`). In both cases these classifiers are evaluated on the test set of the `Paintings Dataset`. The classifiers trained on paintings are representative of the 'best-case scenario' since there is no domain shift to the target domain. Performance is assessed using Average Precision (AP) per class, and also precision at rank k (Pre@k) – the fraction of the top-k retrieved paintings that contain the object – as this places an emphasis on the accuracy of the highest classification scores. To evaluate the domain shift problem we examine the 'mAP gap' – the change in mean (over class) AP between natural image and painting-trained classifiers.

Table 1. The statistics for the datasets used in this paper: each number corresponds to how many images contain that particular class. Note, because each image can contain multiple classes, the total across the row does not equal the total number of images. Train/Validation/Test splits are also given.

Dataset	Split	Aero	Bird	Boat	Chair	Cow	Din	Dog	Horse	Sheep	Train	Total
VOC12	Train	327	395	260	566	151	269	632	237	171	273	3050
	Val	343	370	248	553	152	269	654	245	154	271	3028
	TrainVal	670	765	508	1119	303	538	1286	482	325	544	6078
`Paintings Dataset`	Train	74	319	862	493	255	485	483	656	270	130	3463
	Val	13	72	222	140	52	130	113	127	76	35	865
	TrainVal	87	391	1084	633	307	615	596	783	346	165	4328
	Test	113	414	1059	569	318	586	549	710	405	164	4301

The classifiers used are linear one-vs-rest SVMs, and the features are produced using a CNN. In Sect. 4.1 we determine how the mAP gap is affected by the CNN architecture used to produce the feature, and in Sect. 4.2 we discuss train and test augmentations, and the per class performance. Implementation details are given at the end of the section.

4.1 Networks

Three networks are compared, each trained on the ILSVRC-2012 image dataset with batch normalisation: first, the **VGG-M** architecture of Chatfield *et al.* [8] that consists of 8 convolutional layers. The filters used are quite large (7×7 in the first layer, 5×5 in the second). The features produced are 4096-D. Second, the popular 'very deep' model of Simonyan and Zisserman [37] **VD-16** that consists of 16 convolutional layers with very small 3×3 filters in each layer of stride 1. The features produced are again 4096-D. Third, the ResNets of He *et al.* [25] that treat groups of layers in a network as residual blocks relative to their input. This allows for extremely deep network architectures. The 152-layer ResNet model **RES-152** is selected for this work. The features extracted are 2048-D.

Network comparison. Table 2 gives the mAP performance for the three networks trained on `VOC12` or the `Paintings Dataset`. Three things are clear: first, and unsurprisingly, for features from the same network, classifiers learnt on paintings are better at retrieving paintings than classifiers learnt on natural images; second, RES-152 features surpass VD-16 features, which in turn surpass VGG-M features; and finally, that the mAP gap decreases as the network gets better – from a 14.9 % difference for VGG-M to a 12.7 % for RES-152. Thus improved classification performance correlates with increased domain invariance.

From here on only ResNet features are used for image-level classifiers.

Table 2. mAP for retrieval using image-level classifiers trained on VOC12 vs the Paintings Dataset. Both the networks used to generate the features and the augmentation schemes are varied. 'Net' refers to the network used. 'none', 'f5', 'f25' and 'Stretch' are augmentation schemes and each column gives the corresponding mAP. Augmentation schemes are described further in Sect. 4.2. The last column shows the gap in mAP between natural image and painting-trained classifiers for 'Stretch' augmentation.

Net	Training set	None	f5	f25	Stretch	mAP gap
VGG-M	VOC12	50.8	51.9	52.9	52.9	14.9
VGG-M	`Paintings Dataset`	65.1	67.8	67.8	67.8	
VD-16	VOC12	54.8	56.2	56.7	56.8	14.0
VD-16	`Paintings Dataset`	68.7	71.2	71.2	70.8	
RES-152	VOC12	60.5	61.6	62.0	62.3	12.7
RES-152	`Paintings Dataset`	72.5	74.6	74.6	75.0	

4.2 Augmentation

Four augmentation schemes available in the MatConvNet toolbox [40] are compared, and are applied to each image to produce N crops. In all cases the image is first resized (with aspect ratio preserved) such that its smallest length is 256 pixels. Crops extracted are ultimately 224×224 pixels. The schemes are: **none**, a single crop ($N = 1$) is taken from the centre of the image; **f5**, crops are taken from the centre and the four corners. The same is done for the left-right flip of the image (N=10); **f25**, an extension of f5. Crops are taken at 25 % intervals in both width and height, this is also carried out for the left-right flip ($N = 50$); and finally, **Stretch**, a random rectangular region is taken from the image, linear interpolation across the pixels of the rectangle is performed to turn it into a 224×224 crop, there is then a 50 % chance that this square is left-right flipped. This is performed 50 times ($N = 50$). Note that the same augmentation scheme is applied to both training and test images.

Table 2 shows that the type of augmentation is important: 'stretch' generally produces the highest performance – a 2 % or more increase in mAP over 'none', and equal to or superior to 'f5' and 'f25'. This is probably because the stretch augmentation also mimics foreshortening caused by out-of-plane rotation for objects.

Results and discussion. Table 3 shows the per class AP and Pre@k for the best performing case (ResNet with stretch augmentation), with the corresponding PR curves given in Fig. 2. The datasets are not class balanced, and the ratio of number of TrainVal samples between natural images and paintings varies considerably over classes, but there does not seem to be an obvious correlation with performance – aeroplane classifiers learnt on paintings significantly outperform those learnt on natural images despite being trained with far fewer positive samples (87 vs. 670); and the 'chair', 'dining table' and 'dog' classes have similar numbers in the painting dataset, but with 'dining table' only having half the

Table 3. Retrieval performance comparison on the test set of the `Paintings Dataset` for classifiers trained using ResNet features. The images have been augmented using 'Stretch'. 'Set' refers to the training set used and the performance metric is given under 'Metric': Average Precision (AP) or Precision at rank k (Pre@k).

Set	Metric	Aero	Bird	Boat	Chair	Cow	Din	Dog	Horse	Sheep	Train	Avg
VOC	AP	69.4	42.0	88.7	57.3	62.4	48.4	50.5	73.5	48.7	81.9	62.3
	Pre@k=50	94.0	94.0	100.0	72.0	84.0	92.0	100.0	100.0	98.0	100.0	93.4
	Pre@k=100	61.0	82.0	99.0	72.0	89.0	84.0	98.0	100.0	86.0	98.0	86.9
Paint	AP	77.1	54.1	94.3	78.7	68.3	76.3	62.7	83.5	68.8	85.7	75.0
	Pre@k=50	96.0	100.0	100.0	98.0	92.0	94.0	100.0	100.0	100.0	100.0	98.0
	Pre@k=100	65.0	100.0	99.0	97.0	90.0	92.0	98.0	100.0	91.0	100.0	93.2

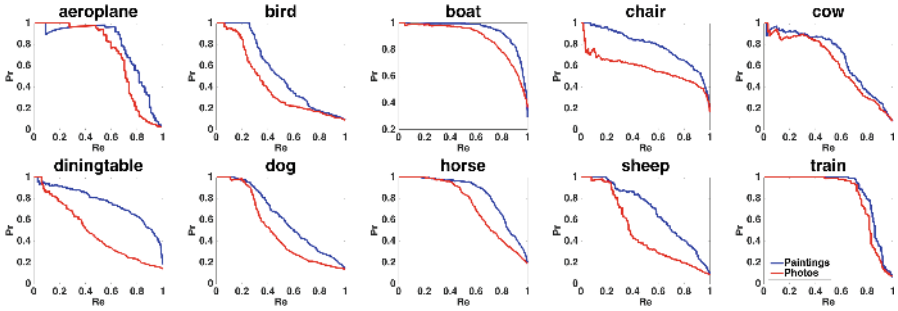

Fig. 2. Precision-Recall curves for different classes, comparing natural image-trained (red) and painting-trained (blue) classifiers learnt on ResNet features. Notice for 'sheep' that the gap in the curves is very significant even at low recall. (Color figure online)

TrainVal images of the other two in VOC12, yet (based on AP) their relative performance does not reflect these ratios at all.

One clear observation though is that the performance of natural image-trained classifiers is inferior to painting-trained classifiers, with an AP gap of around 0.1 for most classes. Pre@k sees a similar decrease. There are some particularly bad cases: 'sheep' has a colossal 20 % decrease in AP, and the furniture classes ('chair' and 'dining table') endure a significant drop. There are several reasons for this inferior performance: first, a few of the paintings are depicted in a highly abstract manner, understandably hindering classification; second, some objects are depicted in a particular way in paintings that isn't present in natural images, e.g. aeroplanes in paintings can be WWII spitfires rather than commercial jets. A third reason is size in the painting; in spite of many paintings being depicted in quite a natural way small objects are missed. Some examples of paintings containing small objects that have been 'missed' (i.e. received a low classifier score) are given in Fig. 3. We investigate this problem in Sect. 5.

Fig. 3. Examples of paintings where a small object has been 'missed' (i.e. given a low score) by a classifier. In each case, the object under consideration is brought to attention with a red box. From left to right: aeroplane, dog, sheep, chair. These small objects are found with confidence by a detector. (Color figure online)

4.3 Implementation Details

Each image (both training and test) undergoes augmentation to produce N crops. The mean RGB values of ILSVRC-2012 are subtracted from each crop. These crops are then passed into a network, and the outputs of the layer before the prediction layer are recorded, giving N feature vectors. These are averaged and then normalised to produce a single feature. Linear-SVM Classifiers are learnt using the training features per class in a one-vs-the-rest manner for assorted regularisation parameters (C). The C that produces the highest AP for each class when the corresponding classifier is applied to the validation set is recorded. The training and validation data are then combined to train classifiers using these C parameters. These classifiers are then applied to the test features, which are ranked by classifier score. Finally, these ranked lists are used to compute APs.

5 Classification by Detection

In this section we classify images by using a detector which is capable of locating small objects. For this we use the VGG-16 Faster R-CNN network of Ren *et al.* [33]. Detection proceeds in two stages: first, a Region Proposal Network (RPN) with an architecture resembling VGG-VD-16 [37] takes in an image and produces up to 300 rectangular regions at a wide variety of scales and aspect ratios each with an "objectness" score. These regions are then used in a Fast R-CNN [21] network that identifies and regresses the bounding box of regions likely to contain PASCAL VOC classes. To obtain a ranked list for a given class, the entire VGG-16 Faster R-CNN network (both the RPN and the pre-trained Fast R-CNN) is applied to each painting in the test set, and the images are ranked according to the score of the highest confidence detection window.

Results and Discussion. Some example detections are given in Fig. 5. The AP and Pre@k per class is reported in Table 4. The pointwise average mAP and Pre@k curves are given in Fig. 4, and compared with those of the image-level classifiers.

Very interestingly, the mAP resulting from this detection network is higher than that of the image-level classifiers trained on natural images, marginally outperforming even the most powerful ResNet classifiers (62.7 % vs. 62.3 %).

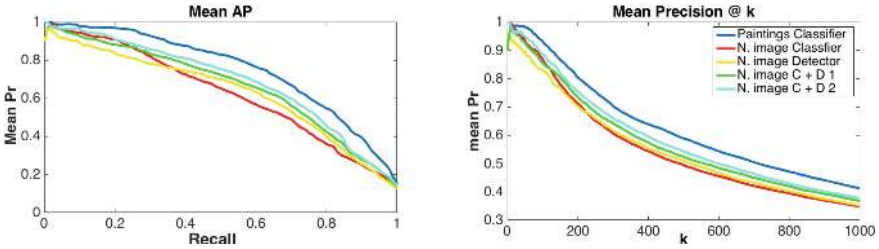

Fig. 4. Left: A point-wise average of mAP across Recall. Right: The average of class precision at rank k for $k < 1000$. Plots are given for image-level classifiers learnt on paintings (blue), and on natural images (red). The Faster-RCNN detector of Sect. 5 is given (yellow), as well as the combinations with image-level classfiiers (green: ranked list combination, cyan: score combination). Notice the significant gap in the performance of natural image and painting-trained classifiers and how the classifier-detection combination ameliorates this. (Color figure online)

The most notable success is on the sheep class, (70.6 % vs. 48.7 %). This is probably because sheep in natural images are typically quite large and near the foreground, whereas in paintings they are often tiny and dotted across an idyllic Welsh hillside. A similar, although smaller such discrepancy can be observed for dogs which are depicted in paintings not only as beloved pets, but also in hunting scenes where they are often small. However, Pre@k for small k is on average lower for the detector than the classifier. This is probably due to the detector fixating on shapes, and not seeing enough context. For example, the 'aeroplane' detector incorrectly fires with confidence on a dragonfly as its wingspan resembles that of a plane. The dragonfly is hovering above a table covered in fruit, clearly not a setting for an aeroplane. This mistake would not be made if images (with context) rather than regions were used for training.

In spite of this, we observe from the right-hand plot of Fig. 4 that when $k > 220$, the mean Pre@k of the detector overtakes that of the classifier. As we suspect, the detector is simply able to locate small objects the image-level classifier is not. This is confirmed by the plots of Fig. 6, which compares the image classifier score for each object label in a test image, to the size of the detection window given by the Faster R-CNN network. The tall bins/light colours in the lower-left corners confirms that typically, classifying the entire painting is poor when the regions found successfully by the detector are smaller.

Combining Detection and Image-Level Classification. We consider two methods of combining the ranked lists produced by the image-level classifiers (learnt on natural images), with those produced by the detector. Other methods are discussed in [4]. The first method is a simple rank merge that combines the two ordered lists (but does not require the scores). This obtains an mAP of 66.0 % (Table 4), closing the mAP gap to 9 %. The second method uses a linear combination of the scores: $\alpha A + (1 - \alpha)B$, and orders on these, where A is the classifier score and B is the detector score. This gives an even higher mAP of

Table 4. Retrieval performance comparison for **image-level classifiers** trained using ResNet features where the images have been augmented using 'Stretch' vs. the Faster-RCNN detector used for **classification-by-detection** on the test set of the `Paintings Dataset`. Note that everything has been trained using natural images. C+D 1 refers to the combination of the classifier and detector ranked lists, and C+D 2 is the combination of their scores.

Method	Metric	Aero	Bird	Boat	Chair	Cow	Din	Dog	Horse	Sheep	Train	Avg
Classifier	AP	69.4	42.0	88.7	57.3	62.4	48.4	50.5	73.5	48.7	81.9	62.3
	Pre@k=50	94.0	94.0	100.0	72.0	84.0	92.0	100.0	100.0	98.0	100.0	93.4
	Pre@k=100	61.0	82.0	99.0	72.0	89.0	84.0	98.0	100.0	86.0	98.0	86.9
Detector	AP	67.4	36.2	88.8	32.8	65.1	48.7	57.6	79.6	70.6	80.0	62.7
	Pre@k=50	86.0	92.0	100.0	66.0	80.0	88.0	92.0	98.0	94.0	100.0	89.6
	Pre@k=100	58.0	71.0	99.0	58.0	84.0	80.0	91.0	98.0	92.0	100.0	83.1
C+D 1	AP	72.7	42.8	90.9	48.1	67.0	52.4	58.4	79.6	65.3	83.1	66.0
	Pre@k=50	90.0	92.0	100.0	74.0	86.0	96.0	96.0	98.0	94.0	100.0	92.6
	Pre@k=100	62.0	80.0	99.0	66.0	84.0	83.0	94.0	98.0	93.0	99.0	85.8
C+D 2	AP	75.2	45.0	92.3	54.8	69.1	53.3	60.4	80.8	70.5	83.7	68.5
	Pre@k=50	94.0	96.0	100.0	76.0	84.0	98.0	100.0	100.0	100.0	100.0	94.8
	Pre@k=100	64.0	77.0	99.0	76.0	90.0	89.0	99.0	100.0	94.0	100.0	88.8

Fig. 5. Example detection windows obtained using the Faster R-CNN network. From left to right: aeroplane, bird, chair, cow. Only, the highest ranked window is shown in each image, even though multiple successful detection windows may have been found. Notice that very small objects are captured, such objects are often missed by an image-level classifier.

68.5 % for $\alpha = 0.3$. The pointwise average mAP and Pre@k curves for these two combinations are given in Fig. 4. This high performance is probably because the image-level classifier and detector are able to complement each other: the classifier is able to utilise the context of a painting and the detector is able to reach small objects otherwise unnoticed.

6 Detecting Objects in Paintings On-the-Fly

It is evident from Sect. 5 that by using the network of [33] it is possible to retrieve objects in paintings through detection that are not retrieved using image-level classification. However, these objects are limited to those of PASCAL VOC, which isn't very useful if an art historian is interested in search for depictions of fruit or elephants. To accommodate for this, we provide a live system, inspired by [7,9,12] where a user may supply a query, and paintings are retrieved that

contain the object with its bounding box provided. This improves on our image-level painting retrieval system [12] in two ways: Firstly, it retrieves small objects that cannot be located at image-level. Secondly, as the region containing the object is provided it is much easily to locate. The method is demonstrated over the entire 210,000 paintings of the 'Art UK' dataset [1].

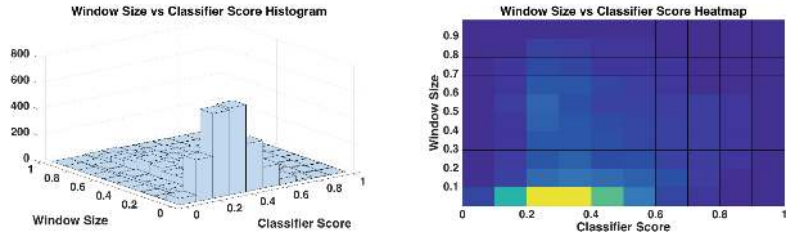

Fig. 6. Left: A 2-D histogram, showing the distribution of image classifier scores (computed from a single vector representing the entire image) against the window size of the highest scored detection window. Classifier scores are mapped between 0 and 1 and window sizes are relative to the size of the image (i.e. window area over image area). Note that the image classifier score is low when the window is small. Right: An overhead view of the histogram where tall peaks are represented by light colours, and short ones by dark colours. (Color figure online)

Overview. At run time, the user supplies an object query as text (e.g. "elephant"). Images are then downloaded for this query using Bing/Google Image Search, and object regions are extracted from them. The object regions are used to generate features, which are used with a pool of negative features to learn a classifier, which is then applied to the features of millions of object regions across the 'Art UK' dataset. The paintings containing the highest scoring object regions are retrieved with their object region annotated. A diagram of this system is provided in Fig. 7.

Feature Representation. Here, we describe how, given an image, features are produced for this system. The image is passed into the Region Proposal Network (RPN) of [33]. This produces up to 300 rectangular regions at a wide variety of scales and aspect ratios each with an "objectness" score. To allow for context, each region is expanded by 5 % in width and height. N of these regions are cropped from the image and resized to 224 by 224, then passed into the VGG-M-128 network of Chatfield *et al.* [8]. The 128-D output of fc7 (the fully connected layer before the prediction) is extracted and L2-normalised. This network is used primarily because the resulting small features minimise memory usage.

Off-line Processing. The features for object regions across the 'Art UK' dataset, and the features used as negative training examples for classification are computed offline. For each painting in 'Art UK', features are produced as

above with $N = 300$ resulting in around 60 million features which are stored in memory. This amounts to ∼32 GB. A fixed set of 16,000 negative features are computed for classification: Google and Bing image searches are performed for vague queries ('miscellanea', 'random stuff' and 'nothing in particular' to name a few) and the images are downloaded. For each image, the region from the RPN with the highest "objectness" score is used to produce a feature i.e. $N = 1$.

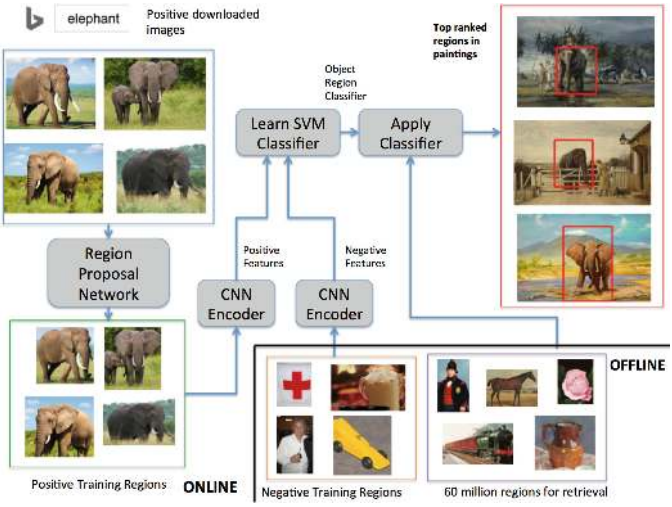

Fig. 7. A diagram of the on-the-fly system. The user types in a query, in this case elephant. Images of that object are downloaded from Bing/Google and passed into a region proposal network to localise the object. These localised regions are passed into a CNN to produce features, which are used in conjunction with pre-computed negative features to learn a region classifier. This region classifier is applied to 60 million object regions across 210,000 paintings and the highest scoring regions are retrieved.

On-line Processing. Computing positive training features and learning, then applying a classifier occur online. Positive features are obtained as follows: a Bing/Google Image Search is carried out using the query as a search term. The URLs for the first 100 images are recorded and downloaded in parallel across 12 CPU cores. Each of these images is passed into the RPN and the highest "objectness" region is used to produce a feature ($N = 1$), operating on the presumption that in these "Flickr style" images (the object is in focus, large and is often against a plain background) the region with the highest "objectness" score corresponds to the object in question. Instances of such windows can be seen in Fig. 8 where it is evident that this is often the case.

The positive and negative features are used in a Linear-SVM to produce a classifier. This can be done on a single core and takes a fraction of a second. The classifier is applied to 60 million painting features in a single matrix operation.

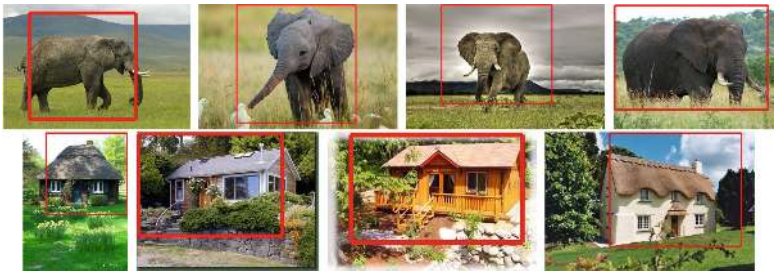

Fig. 8. Highest scoring "objectness" regions (in red) when images downloaded from Bing/Google are passed into an RPN. Top row: 'elephant', Bottom row: 'cottage'. Notice that the regions manage to contain the object, with quite a tight bound. (Color figure online)

6.1 Evaluation

The system is assessed for 250 different object queries over a variety of subjects. This include vehicles (boats, cars), animals (elephants, dogs), clothes (uniform, gown), structures (cottage, church), parts of structures (spires, roof) among others. Performance is evaluated quantitatively as a classification-by-detection problem as in Sect. 5: we rank each of the 210,000 paintings according to the score corresponding to its highest scoring object region and by eye, compute Pre@k – Precision at k, the fraction of the top-k retrieved paintings that contain the object – for the 50 top retrieved paintings. Some examples detections and Pre@k curves are provided in Fig. 9.

This system is crucially able to overcome one of the difficulties experienced by our image-level classification system [12]: a notable difference in performance occurs when an object is large in natural images and small in paintings. A good examples of this is 'wheel'. Bing/Google images of wheels mainly comprise of a single wheel, viewed head-on against a plain background. Conversely, wheels in paintings are often attached to carriages (or to a lesser extent, cars) and are a small part of the image. An image-level classifier succeeds if the natural images resemble the paintings in their entirety so cannot cope with this discrepancy, whereas a region-level classifier can cope with only a small part of a painting resembling the natural image. However, a drawback of the system relative to image-level classification occurs when the context of an object is lost. A similar observation was made in Sect. 5. A good example of this is for the query 'tie'. Some of the paintings retrieved are indeed of people wearing ties, but others are abstract **V** shapes. Several natural images for 'tie' are of a person's torso wearing a tie but by isolating the object, this context has been lost. The bounding boxes of the objects in paintings are often quite loose. Although not ideal, this isn't too important as the objects are sufficiently localised for human use.

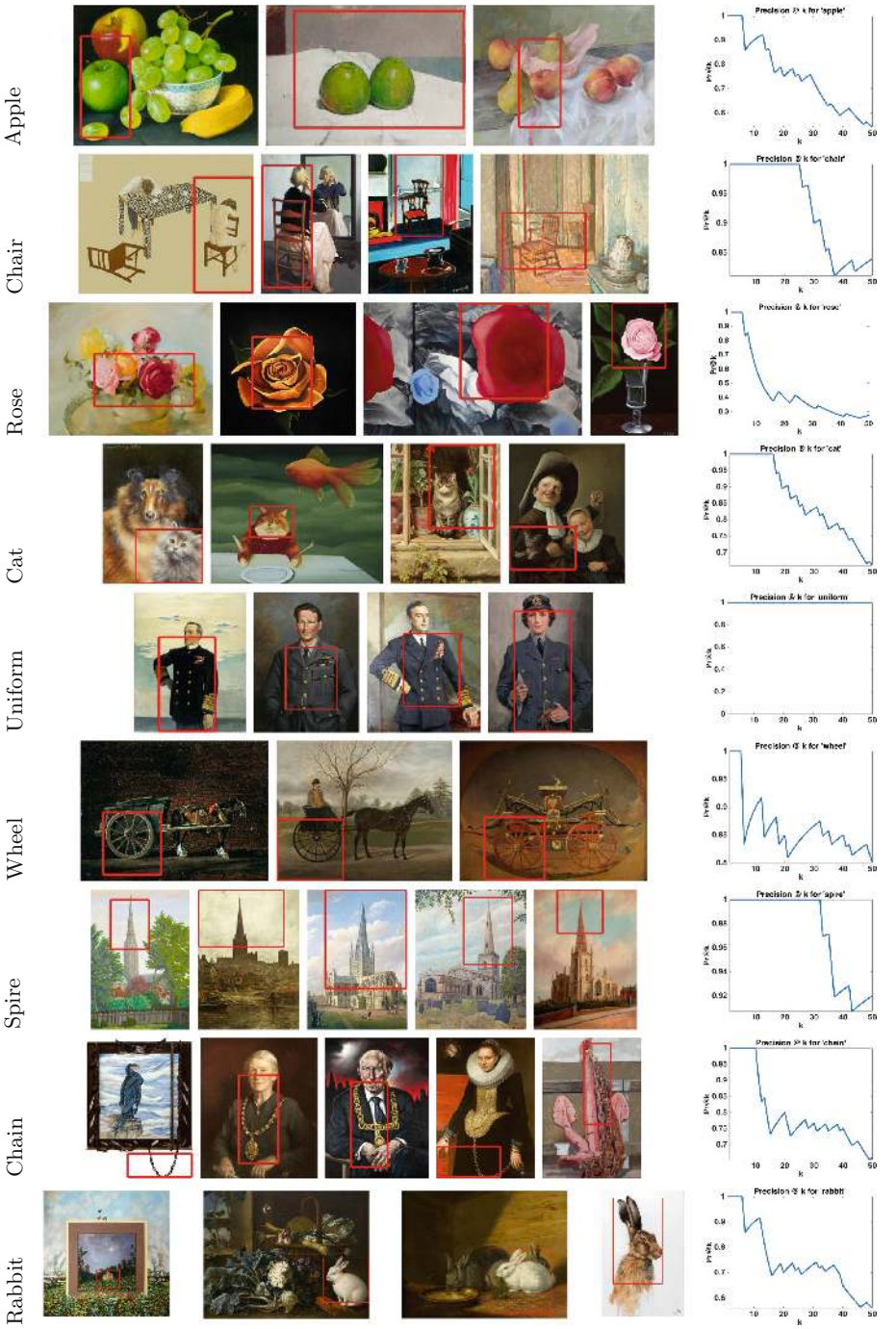

Fig. 9. Example detections for our on-the-fly system for assorted queries as well as the Pre@k curve for the top 50 results.

7 Conclusion and Future Work

In this paper, we have explored the domain shift problem of applying natural image-trained classifiers to paintings. We have further shown that detectors are able to find many objects in paintings that are otherwise missed, and based on this observation, have created an on-the-fly system that finds such objects across hundreds of different classes. Future work could consist of utilising the method of Cinbis *et al.* [10] to refine the locations of objects in natural images and paintings further. By doing this, the painting regions would be well suited for the query expansion method of [6].

Acknowledgements. Funding for this research is provided by EPSRC Programme Grant Seebibyte EP/M013774/1.

References

1. Art UK. http://artuk.org/
2. The Paintings Dataset. http://www.robots.ox.ac.uk/~vgg/data/paintings/
3. Aljundi, R., Tuytelaars, T.: Lightweight unsupervised domain adaptation by convolutional filter reconstruction. arXiv preprint arXiv:1603.07234 (2016)
4. Aslam, J., Montague, M.: Models for metasearch. In: Proceedings of the SIGIR, pp. 276–284. ACM, New York (2001)
5. Aubry, M., Russell, B., Sivic, J.: Painting-to-3D model alignment via discriminative visual elements. ACM Trans. Graph. **33**(2), 14 (2013)
6. Cai, H., Wu, Q., Hall, P.: Beyond photo-domain object recognition: benchmarks for the cross-depiction problem. In: Workshop on Transferring and Adapting Source Knowledge in Computer Vision, ICCV (2015)
7. Chatfield, K., Arandjelović, R., Parkhi, O.M., Zisserman, A.: On-the-fly learning for visual search of large-scale image and video datasets. Int. J. Multimedia Inf. Retr. **4**(2), 75–93 (2015)
8. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. In: Proceedings of the BMVC (2014)
9. Chatfield, K., Zisserman, A.: VISOR: towards on-the-fly large-scale object category retrieval. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7725, pp. 432–446. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37444-9_34
10. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. (2016)
11. Crowley, E.J., Parkhi, O.M., Zisserman, A.: Face painting: querying art with photos. In: Proceedings of the BMVC (2015)
12. Crowley, E.J., Zisserman, A.: In search of art. In: Workshop on Computer Vision for Art Analysis, ECCV (2014)
13. Crowley, E.J., Zisserman, A.: The state of the art: object retrieval in paintings using discriminative regions. In: Proceedings of the BMVC (2014)
14. Daumé III., H.: Frustratingly easy domain adaptation arXiv preprint arXiv:0907.1815 (2009)

15. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: a deep convolutional activation feature for generic visual recognition (2013). CoRR abs/1310.1531
16. Everingham, M., Eslami, S.M.A., Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. IJCV **111**(1), 98–136 (2015). doi:10.1007/s11263-014-0733-5
17. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2011) (2012). http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
18. Felzenszwalb, P.F., Grishick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE PAMI **32**(9), 1627–1645 (2010)
19. Fernando, B., Tuytelaars, T.: Mining multiple queries for image retrieval: on-the-fly learning of an object-specific mid-level representation. In: Proceedings of the ICCV (2013)
20. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the ICLR (2015)
21. Girshick, R.B.: Fast R-CNN. In: Proceedings of the ICCV (2015)
22. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the CVPR (2014)
23. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: an unsupervised approach. In: Proceedings of the ICCV (2011)
24. Hall, P., Cai, H., Wu, Q., Corradi, T.: Cross-depiction problem: recognition and synthesis of photographs and artwork. Comput. Vis. Media **1**(2), 91–103 (2015)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the ICCV (2015)
26. Hoffman, J., Darrell, T., Saenko, K.: Continuous manifold based adaptation for evolving visual domains. In: Proceedings of the CVPR (2014)
27. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in neural information processing systems, pp. 601–608 (2006)
28. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond. In: Proceedings of the ICCV (2011)
29. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the CVPR (2014)
30. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2011)
31. Patterson, G., Hays, J.: Sun attribute database: discovering, annotating, and recognizing scene attributes. In: Proceedings of the CVPR, pp. 2751–2758 (2012)
32. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition (2014). CoRR abs/1403.6382
33. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2016)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, S., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Li, F.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
35. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)

36. Shrivastava, A., Malisiewicz, T., Gupta, A., Efros, A.: Data-driven visual similarity for cross-domain image matching. ACM Trans. Graph. **30**(6), 154 (2011)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
38. Sun, B., Saenko, K.: Subspace distribution alignment for unsupervised domain adaptation. In: Proceedings of the BMVC (2015)
39. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: Proceedings of the ICCV (2015)
40. Vedaldi, A., Lenc, K.: Matconvnet: convolutional neural networks for matlab. In: ACM International Conference on Multimedia (2015)
41. Wu, Q., Cai, H., Hall, P.: Learning graphs to model visual objects across different depictive styles. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 313–328. Springer, Heidelberg (2014)
42. Wu, Q., Hall, P.: Modelling visual objects invariant to depictive style. In: Proceedings of the BMVC (2013)
43. Yu, Q., Liu, F., Song, Y., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: Proceedings of the CVPR (2016)