In Search of Art

Elliot J. Crowley(✉) and Andrew Zisserman

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

Abstract. The objective of this work is to find objects in paintings by learning object-category classifiers from available sources of natural images. Finding such objects is of much benefit to the art history community as well as being a challenging problem in large-scale retrieval and domain adaptation.

We make the following contributions: (i) we show that object classifiers, learnt using Convolutional Neural Networks (CNNs) features computed from various natural image sources, can retrieve paintings containing these objects with great success; (ii) we develop a system that can learn object classifiers on-the-fly from Google images and use these to find a large variety of previously unfound objects in a dataset of 210,000 paintings; (iii) we combine object classifiers and detectors to align objects to allow for direct comparison; for example to illustrate how they have varied over time.

Keywords: Domain Adaptation · Object Classification · Computer Vision in Art

1 Introduction

"I do not search, I find." – Pablo Picasso.

Natural images (i.e. everyday photos taken with a camera) annotated with objects are everywhere – large numbers of annotated photos are readily available in curated datasets [20,23]; and, simply typing the name of an object into Google Image search will produce high quality images of that object. Unfortunately the same cannot be said of paintings; these are largely lacking in annotation. Art historians are often interested in determining when an object first appeared in a painting or how the portrayal of an object has evolved over time, to achieve this they have the unenviable task of finding paintings for study manually [11,29,46]. If they are instead provided with paintings annotated with objects they can conduct these studies far more easily.

In this paper, we provide this object annotation by using readily available natural images to learn object category classifiers able to find objects across hundreds of thousands of paintings. This is not a straightforward task; natural images and paintings can differ substantially in their low level statistics, and

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part I, LNCS 8925, pp. 54–70, 2015.
DOI: 10.1007/978-3-319-16178-5_4

paintings can exist in a number of depictive styles (e.g. impressionism, surrealism) where the very objects themselves can be warped like the clocks in Dali's 'Persistence of memory'. In addition to this, the objects themselves may have changed with time – photographs of planes will typically be of modern commercial jetliners whereas those in paintings can be more akin to Wright Flyers or Spitfires.

One of our contributions is to show that features generated using Convolutional Neural Networks (CNN) [33,34] are able to overcome much of this adversity. These networks have been shown to be effective for a variety of tasks [21,28,37,42], and we show that classifiers learnt with CNN features on natural images are very successful at retrieving objects in paintings, overcoming the problem of domain adaptation [19,32,43] and greatly outperforming classifiers learnt using Fisher Vectors [40,41] (section 3). We also compare the performance of curated datasets vs. images crawled from the internet to assess the suitability of the net as a training source.

We develop an on-the-fly system [8,15,39] (section 4) that learns classifiers for object categories in real-time by crawling Google images. These are then applied to a dataset of 210,000 oil paintings to retrieve paintings containing these object categories with high precision over many and disparate classes. The entirety of this process for a given query takes a matter of seconds.

Finally, inspired by the work of Lee *et al.* [35] we conduct longitudinal studies to examine how the portrayal of particular objects have varied over time by combining classifiers with Deformable-Parts based models (DPMs) (section 5) to produce mosaics of aligned objects; this benefits cultural historians as well as sating curiosity.

2 Datasets

In this section, the datasets of natural images and paintings used for evaluation (section 3) and large-scale object retrieval (section 4) are introduced.

2.1 Paintings

The publicly available 'Your Paintings' dataset [1] is utilized for this work. This dataset consists of over 210,000 oil paintings of medium resolution (the width is usually around 500 pixels). 10,000 of these have been annotated as part of the 'Tagger' project [7] whereby members of the public tag the paintings with the objects that they contain.

To quantitatively assess classifier performance a dataset of paintings with complete annotation in the PASCAL VOC [23] sense – that each painting has been annotated for the categories under consideration – is required for use as a test set. To satisfy this requirement we construct the `Paintings Dataset` as a subset of 'Your Paintings'. This subset is obtained by searching 'Your Paintings' for annotations and painting titles corresponding to the classes of VOC. With tags and

Table 1. The statistics for the datasets used for evaluation: the number of images containing an instance of a particular class are given, as well as the total number of training and validation images (where applicable). The `Paintings Dataset` is only ever used as a test set, whereas other datasets are used for training and validation.

Dataset	Split	Aero	Bird	Boat	Chair	Cow	Din	Dog	Horse	Sheep	Train	Total
`Paintings`	Total	200	805	2143	1202	625	1201	1145	1493	751	329	8629
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
	Total	670	765	508	1119	303	538	1286	482	325	544	6078
VOC12+	Train	769	1007	613	1428	419	659	1471	798	364	793	7812
	Val	343	370	248	553	152	269	654	245	154	271	3028
	Total	1112	1377	861	1981	571	928	2125	1043	518	1064	10840
`Net`	Train	252	264	254	267	254	278	269	272	258	260	2628
`Noisy`	Val	84	88	85	89	85	93	90	91	87	87	879
	Total	336	352	339	356	339	371	359	363	345	347	3507
`Net`	Train	203	192	173	197	149	254	220	192	216	222	2018
`Curated`	Val	68	64	58	66	50	85	74	64	72	74	675
	Total	271	256	231	263	199	339	294	256	288	296	2693

titles complete annotation is assumed as long as 'people' are ignored, as this particular class has a tendency of appearing frequently without being acknowledged. Thus, the 'person' class is not considered, and also we do not include classes that lack a sufficient number of tags (cat, bicycle, bus, car motorbike, bottle, potted plant, sofa, tv/monitor). Paintings are included for the remaining classes – aeroplane, bird, boat, chair, cow, dining-table, dog, horse, sheep, train. The statistics are given in table 1, and example class images are shown in figure 1. The URLs for the paintings in this dataset are provided at [6].

2.2 Natural Images

When learning classifiers from natural images there are two important factors to explore. The first of these is the number of images used for training; this factor will be explored by comparing two datasets: the first is the training and validation data of PASCAL VOC 2012 [24] for the 10 classes of the `Paintings Dataset` (hereby referred to as VOC12), the second is a larger dataset consisting of VOC12 plus all of the training, validation and test data of PASCAL VOC 2007 for the relevant classes (we refer to this as VOC12+).

The second factor to consider is the source of the training images, particularly the suitability of images obtained from the internet that may have label noise (as not all of the results from an online image search for an object will actually contain that object). For each of the classes in the `Paintings Dataset` the top 200 Google Image Search [5] results and top 200 Bing Image Search [2] results are collated for a search query of the class name to form the `Net Noisy` dataset (split randomly into training and validation); note that there are less than 400 images per class, this is because some links did not return an image. `Net Noisy`

Fig. 1. Example class images from the `Paintings Dataset`. From top to bottom row: dog, horse, train. Notice that the dataset is challenging: objects have a variety of sizes, poses and depictive styles, and can be partially occluded or truncated.

is then manually filtered to remove erroneous instances forming `Net Curated`. The statistics for all datasets are given in table 1.

3 Domain Transfer Experiments

In this section we evaluate the performance of object classifiers that have been learnt on natural images when they are applied to paintings. In all cases, the classifiers are applied to the `Paintings Dataset` which is used as a test set. Classifiers are learnt from four datasets of natural images as described in section 2 – VOC12, VOC12+, `Net Noisy` and `Net Curated` – to compare two factors: (i) the effect of increasing the number of training examples, (ii) the difference between learning from curated and extemporary datasets. Two different features are compared: (i) the Improved Fisher Vector (FV) [41], and (ii) features extracted from Convolutional Neural Networks (CNNs) over a number of training and testing augmentation strategies. The evaluation of these classifiers using Average Precision (AP) for each class and the mean of these (mAP) is given in section 3.1, and implementation details are given in section 3.2.

3.1 Evaluation

Average Precision (AP) is calculated for each class as well as the mean of these values (mAP), these measures are given in table 2.

In all instances mAP improves substantially when switching to CNNs from Fisher Vectors. For CNNs, using augmentation schemes causes mAP to increase

Table 2. Average Precision for Classification Performance on the `Paintings Dataset` for different training methods and sources. In the case of the CNN features, the type of training augmentation and testing pooling is indicated for each set.

Method	Dim	Train	Aero	Bird	Boat	Chair	Cow	Din	Dog	Horse	Sheep	Train	mAP
FV	84K	VOC12	32.3	18.7	74.2	35.5	21.7	34.1	23.9	42.8	19.6	64.0	36.7
(x,y)		VOC12+	32.3	20.9	73.1	33.6	19.8	33.5	25.0	46.7	26.8	69.1	38.1
		Net Noisy	28.2	15.6	68.9	24.4	11.1	23.9	22.8	38.2	21.2	51.8	30.6
		Net Cur	29.9	13.9	68.7	18.2	12.0	24.2	22.8	38.7	20.5	47.5	29.6
CNN	2K	VOC12	58.5	33.9	84.1	45.7	44.9	40.1	40.2	60.5	40.4	72.5	52.1
no aug		VOC12+	57.9	33.4	84.3	45.0	44.5	39.1	39.1	58.5	41.4	72.4	51.6
no pool		Net Noisy	49.4	35.8	82.1	34.5	23.2	38.2	32.3	59.3	31.4	71.2	45.7
		Net Cur	52.8	36.6	82.6	38.5	29.0	37.0	35.5	61.1	36.3	71.2	48.1
CNN	2K	VOC12	59.2	36.9	83.9	40.7	44.5	45.7	41.4	61.0	46.0	75.2	53.5
aug		VOC12+	59.9	37.1	85.6	41.6	44.7	43.3	40.3	61.2	47.3	76.4	53.7
max pool		Net Noisy	50.3	36.2	81.8	35.6	25.4	35.5	30.6	57.4	34.7	74.3	46.2
		Net Cur	52.0	37.5	82.3	38.2	31.5	33.8	36.0	61.8	35.2	71.1	47.9
CNN	2K	VOC12	59.4	37.2	84.6	42.0	44.9	46.1	41.5	61.2	48.0	75.9	54.1
aug		VOC12+	60.1	36.5	85.9	43.4	44.6	43.7	39.9	62.2	49.2	77.7	54.3
sum pool		Net Noisy	51.0	35.7	82.9	37.1	27.2	35.4	31.3	58.9	36.2	74.8	47.1
		Net Cur	52.6	37.9	83.0	40.0	33.3	34.0	36.7	62.8	36.2	72.0	48.8
CNN	2K	VOC12	59.5	35.0	84.7	45.6	46.9	40.2	42.5	61.6	42.7	74.5	53.3
no aug		VOC12+	59.9	35.0	86.0	45.7	45.9	40.5	41.3	59.4	43.8	75.1	53.3
sum pool		Net Noisy	52.7	37.8	84.0	37.2	23.7	39.0	32.7	61.2	34.2	74.4	47.7
		Net Cur	53.6	39.2	83.3	41.1	27.1	39.9	36.9	63.0	35.6	67.8	48.8
CNN	128	VOC12	57.8	33.2	85.4	48.8	41.9	44.5	39.3	60.2	45.3	75.6	53.2
aug		VOC12+	56.4	33.8	86.1	49.3	40.8	41.4	38.6	57.4	44.7	75.3	52.4
sum pool		Net Noisy	52.1	33.2	79.5	29.8	24.0	32.3	33.2	55.7	34.9	76.8	45.1
		Net Cur	53.2	37.7	81.9	35.9	32.2	31.4	34.6	58.4	34.1	73.9	47.3
CNN	1K	VOC12	60.2	38.9	85.5	40.4	45.5	46.6	41.4	61.5	48.3	75.7	54.4
aug		VOC12+	60.5	38.2	87.1	43.3	45.6	47.3	40.4	59.8	49.2	76.9	54.8
sum pool		Net Noisy	50.7	35.2	83.1	36.5	32.0	37.5	30.5	60.7	37.6	75.4	47.9
		Net Cur	53.0	37.2	83.4	36.3	37.3	39.2	35.6	64.3	36.8	72.1	49.5
CNN	4K	VOC12	54.5	35.4	84.2	40.7	42.4	50.2	39.4	56.2	43.3	73.8	52.0
aug		VOC12+	52.0	35.2	84.0	41.5	43.3	49.9	37.1	60.4	41.0	76.5	52.1
sum pool		Net Noisy	45.8	32.7	79.2	31.8	32.6	39.7	29.6	55.9	36.1	73.5	45.7
		Net Cur	47.5	33.7	77.5	42.0	31.9	38.6	35.1	55.2	35.6	68.1	46.5

by a small amount (typically ∼1–2%) relative to not using augmentation; the highest performance is obtained using augmented training data with sum-pooling of the augmented test data. Note that it takes around 0.3s to compute the features for a single frame of an image compared to 2.4s for augmented features; if time is of the essence the small improvement in performance is likely not worth the additional computation. The mAP is highest for 1024-D CNN features, although this performance is still very similar to that of other dimensions.

The benefits of augmentation differ by class; using 2K CNN features, the AP for bird using VOC12 with no augmentation is 33.9 whereas with training augmentation and sum pooling it rises to 37.2, sheep sees an even sharper rise from 40.4 to 48.0. This is likely because such objects can appear quite small

(a bird from the distance, or a lone sheep on a hill) and can be missed if only a single central frame is extracted. Some objects (boat, cow) remain unaffected by augmentation. Sum pooling generally outperforms max pooling; sum pooling allows for a richer contextual description of each test painting; an exception to this is for chair, paintings for which can contain a lot of distracting clutter.

There is very little difference between the performance of VOC12 and VOC12+ despite VOC12+ having almost twice as many train-val images. This indicates that only a few hundred CNN training examples are required for a classifier to learn the important features for these classes. There is also minimal difference between the performance of `Net Noisy` and `Net Curated`; the classifier learning is robust to outliers being present in the training data. The important implication of this is that there is no real need to pre-filter images obtained from the internet.

Although there is a substantial mAP difference between training on VOC12 (or VOC12+) and `Net` images, some of the individual class APs are quite similar – notably bird and train.

In general, the most successful classifiers are those for boats, horses and trains. This is very likely because these objects are typically depicted similarly in both paintings and natural images. Conversely, furniture varies a lot between natural images and paintings (chairs and tables are of very different shapes and styles) leading to lower performance, though such classes are hard to classify even in the case of natural images [24].

3.2 Implementation Details

Fisher Vector Representation. For generating Fisher Vector features the pipeline of [13] is used with the implementation available from the website [4]: RootSIFT [9] features are extracted at multiple scales from each image. These are decorrelated and reduced using PCA to 80-D and augmented with the (x,y) co-ordinates of the extraction location. These features are used to learn a 512 component Gaussian Mixture Model (GMM). For each image, the mean and covariance of the distances between features and each GMM centre are recorded and stacked resulting in a $82 \times 2 \times 512 = 83{,}968$-D Fisher Vector.

CNN Representation. A deep CNN network similar to that of [47] is used [14] using the implementation available from the website [3]. It consists of 5 convolutional layers and 3 fully-connected layers. It is trained solely using ILSVRC-2012 using stochastic gradient descent as in [31].

To obtain a feature vector, a given image frame is passed through the network and the output of the penultimate layer is recorded. It has been shown previously that this output can be used as a powerful descriptor, readily applicable to other datasets [22]. Multiple networks are trained for various sizes of this layer to produce output vectors of different sizes (128-D, 1024-D, 2048-D and 4096-D).

For each training image, with no augmentation (no aug) each image is downsized so that its smallest dimension is 224 pixels and then a 224×224 frame is extracted from the centre, this frame is passed into the CNN producing a feature vector. When augmentation (aug) is used 10 frames are extracted from an image: the images are resized so the smallest dimension is 256 and then a

224×224 frame is extracted from the centre and four corners of each image and the left-right flip of these frames, these are passed in to the CNN resulting in 10 feature vectors. Each of these vectors is considered an independent training sample and one-vs-the-rest classifiers are learnt.

At test time different pooling schemes are utilized: there can be no pooling (no pool) where classifiers are applied to a single feature vector extracted from each test image (as in no aug above). Alternatively each test image is augmented and the classifiers are either applied to the mean of the 10 vectors extracted using the above augmentation scheme (sum pool) or they are applied to the vectors separately and the highest response is recorded (max pool).

Classification. Linear-SVM Classifiers are learnt using the training data per class in a one-vs-the-rest manner for a range of regularization parameters (C). The C that produces the highest mAP when the corresponding classifiers are applied to the validation set is recorded. The training and validation data are then combined to train classifiers using this C parameter, which are finally applied to the test data. For the Fisher Vector representation, a single feature vector is obtained for each image using the entirety of that image. For CNNs, augmentation and pooling schemes are incorporated at training and testing.

4 Finding Objects in Paintings on-the-fly

It is clear from section 2 that classifiers learnt from CNN features extracted from natural images are surprizingly successful at retrieving object categories from paintings. A further advantage is the speed of this process: extracting a frame from an image and producing a CNN feature takes ∼0.3s, training and applying a classifier takes only a fraction of a second because the features are sparse and of low dimensionality. In particular, the performance of `Net Noisy` indicates that the images crawled from a Google search are a suitable training source for a variety of classes without any pre-filtering, the natural extension being to obtain classifiers that are able to retrieve paintings for classes other than those in VOC.

With this is mind, we develop a live system similar to VISOR [15] thats crawls Google images in real-time for a given object query, downloads the images and learns a CNN-based classifier. This classifier is then applied to the entirety of 'Your Paintings' [1] (210K paintings) to produce a ranked list of paintings. We test this system for 200 different object queries across many categories and record the precision of the highest-ranked paintings (see section 4.1). A diagram of this process is given in figure 2 and the steps of the implementation are described below:

Features. 1024-D CNN features are used as these produced classifiers with the highest performance on the `Paintings Dataset`. As discussed in section 3.1 augmenting either the training and/or testing data improves performance at the cost of computation time. As such, for pre-processed features stored offline, sum-pooled feature vectors are used, whereas online where time is very important, features for positive training examples are computed without any augmentation.

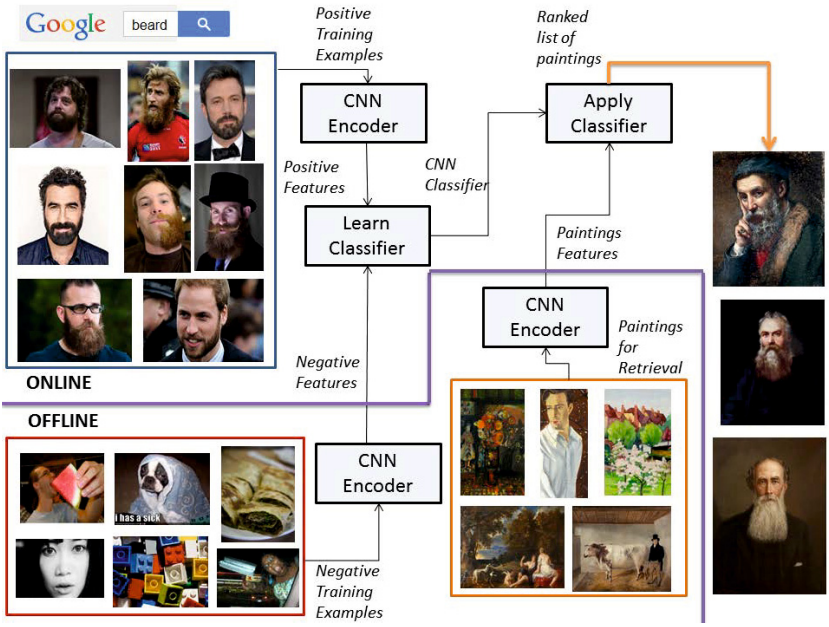

Fig. 2. A diagram of the on-the-fly system. The user types in a class query and positive training examples of that class are crawled from Google Images. These are then encoded using CNN features and used in conjunction with a pre-computed pool of negative features to learn a classifier in real time. This classifier then ranks hundreds of thousands of paintings for which the features are stored in memory before displaying the highest-ranked paintings. This entire process takes a matter of seconds.

Obtaining positive training images. To generate the positive training set for a given class, Google Image Search is queried with the class as a search term and the URLs for the top 200 images are recorded. These are then downloaded in parallel with a timeout of 1s to prevent the process from being too slow. CNN features are then computed in parallel over multiple cores from a single frame extracted from each image as in the 'no augmentation' scheme of section 3.2. For the Google Image search the photo filter is used; this may seem counter-intuitive as one would expect that without this filter paintings would appear which would benefit training, in actuality non-photo images tend to be clip-art that are even further in likeness from paintings than natural images are.

Negative training images. A fixed pool of negative training images is used to aid classification. This set consists of ∼1000 images from the Google searches 'things' and 'photos'. The augmented CNN features of these images are pre-computed and stored in memory for immediate access. This only amounts to 40MB of memory.

Classification. Classifiers are learnt on a single core using the positive and negative features with a Linear-SVM. The classifier is then applied to all of 'Your Paintings' in a single matrix operation; the sum-pooled CNN features

of 'Your Paintings' are pre-computed and stored in memory. This is the most memory intensive part of the process as all of 'Your Paintings' stored as 1024-D features with single precision amounts to 800MB.

Offline and Online Processing. In summary: the features for negative training images and all the paintings for retrieval are pre-computed offline. Online, i.e. at run time, the positive training images are downloaded and have their features computed; then the classifier is learnt, and finally the paintings are ranked on their classifier score.

Performance. Obtaining the 200 URLs for a search query typically takes 0.5s. The time taken to download the images at these URLs can vary by class but is often ∼2s. Across the 200 queries described below, the average number of images downloaded successfully is 177. Computing CNN features for the downloaded images takes ∼4.5s using 16 cores. Learning a classifier using the Liblinear [25] package and performing a matrix operation between the classifier and the dataset features only takes a fraction of a second.

In total, the entire process from typing in a search query to receiving the retrieved paintings takes roughly 7 seconds.

4.1 Evaluation

To evaluate how well the system works we test it for 200 different queries over a broad range of object categories. These include structures (arch, bridge, column, house), animals (bird, dog, fish), colours (red, blue, violet), vehicles (boat, car), items of clothing (cravat, gown, suit) as well as environments (forest, light, storm). The resulting classifier for each search term returns a ranked list of retrieved paintings, for each such list Precision-at-k (Prec@k) – the fraction of the top-k ranked results that are classified correctly – is recorded for the first 50 retrieved paintings. The highest ranked paintings for selected queries as well as the corresponding Prec@k curves are given in figure 3.

In general, the learnt classifiers are very successful and are able to retrieve paintings for a large variety of objects with high precision. The vast majority of the correctly retrieved paintings had not previously been tagged on 'Your Paintings', so these are new discoveries for those object classes.

In more detail, the classifiers that produce the highest precision are those for which objects in the training photos and paintings are portrayed in a similar manner. For example, for 'person' the vast majority of photos and paintings will be in portrait-style. The same can be said of animals such as 'horse' that are predominately captured from the side in a rigid pose. Conversely for certain smaller objects, particularly human body parts (arm, hand, eye) classifiers are not very successful. This is because of the drastic differences in depiction between the photos and paintings; in photos, the entire image will contain the object whereas in paintings the object is much smaller (in the case of eye, rarely more than a few pixels wide).

For objects with very simple shapes like circles (buttons, wheels) and rectangles (books, doors) results retrieved tend to be poor, simply consisting of paintings containing the shape rather than the object itself. Classifiers trained

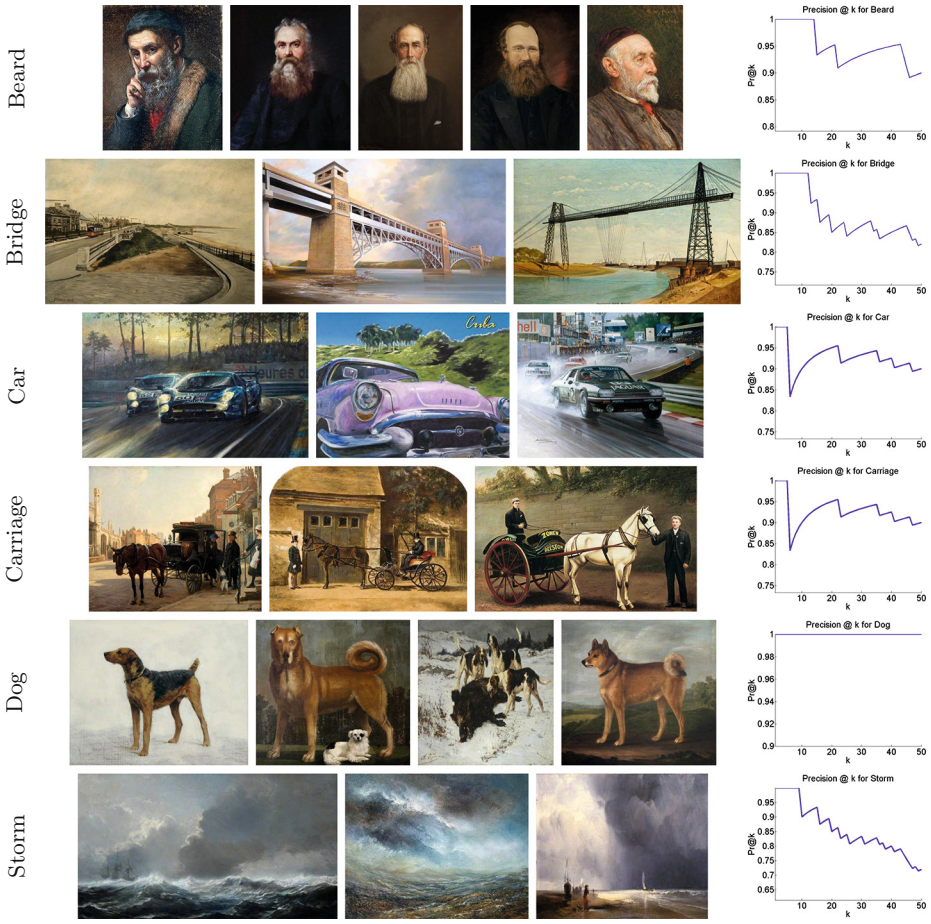

Fig. 3. Highest ranked paintings when classifiers are applied to 'Your Paintings' where the classifiers have been learnt from selected Google Image Search Queries as well as the Prec@k curve for the top 50 results.

on environments with no real fixed boundaries (winter, woodland) perform with great success, this is because paintings of these tend to be very realistic, mirroring nature. Also there is the added advantage that for environments the entire image is relevant rather than a smaller region; it is harder to inadvertently learn something else.

Colours are retrieved with high precision, something that is clearly not possible when using a handcrafted descriptor based around gradients (e.g. HOG [18] or SIFT [36]), CNNs are able to capture both gradient and colour information. This has a disadvantage for certain classes that are based around colours such as 'fire' and 'steam'; the paintings retrieved for these classes share colours

(red, orange, yellow for 'fire', grey and black for 'steam') but not the classes themselves.

Vehicles are retrieved successfully despite the temporal depictive differences between vehicles in photos and paintings; of particular interest is 'car' – as addressed in section 2 very few paintings in 'Your Paintings' were known to contain cars, making this retrieval particularly impressive.

Unsurprizingly, classifiers trained on words afflicted with polysemy (those that can have multiple meanings – for example bow can be a weapon, a gesture, or a part of a ship) rarely retrieve any correct paintings because the positive training data is inherently noisy, this phenomenon has been noted previously by Schroff *et al.* [44].

5 Longitudinal Studies

In section 4 we have shown that it is possible to retrieve many paintings containing a given object in very little time with high precision. Inspired by the work of Lee *et al.* [35] we use these retrieved paintings to observe how the depiction of objects has varied over time. This is possible as many paintings in 'Your Paintings' are accompanied with a date.

To make these observations it is ideal for instances of objects to be aligned. For some classes objects are inherently aligned, such as for 'moustache'; the retrieved paintings are almost entirely portraits so the moustaches are side by side and easy to compare. A mosaic of moustaches over time is presented in figure 4. For most classes this is not the case: it is known from the classifier that the object is present but not its location. If an art historian were to compare objects between these paintings it would not be ideal to have to manually pick out, scale and align each of several hundred objects.

To find, scale and align objects automatically we employ the Deformable Part Model (DPM) [26,27] object category detector to find object locations in high-ranked paintings. This has the added benefit of depicting left/right facing objects in the same way (from the appropriate component response – see details below). Consider figure 6: the top half of the figure contains paintings from 'Your Paintings' that have been ranked highly for 'train' by a classifier learnt as in section 4; the trains are at different positions and scales, making comparison difficult. By applying a DPM a mosaic can be formed as in the bottom half of figure 6, allowing for much easier comparison.

Implementation Details. Classifiers are learnt using Google images as in section 4 to produce a ranked list of paintings. A DPM is learnt either (i) using PASCAL VOC 2012 bounding boxes or (ii) using the same positive and negative training instances used to learn the classifier. Note that for (ii) no bounding-box regions of interest (ROI) are provided so the entire image is taken to be an ROI. DPMs have been trained previously using the entire image as the ROI for scene classification [38]. The DPM has 3 mirrored components (for a total of 6) each comprising 8 parts. These are applied to the highest classified paintings for the class and the highest scoring detection windows are recorded. By left-right

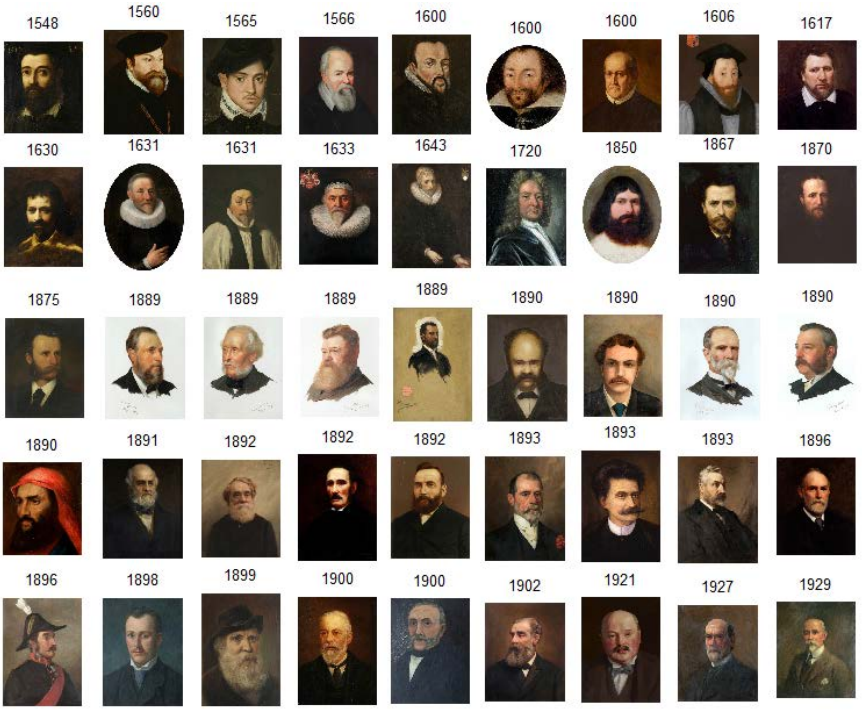

Fig. 4. Moustaches through the ages. The nature of the object means the moustaches are aligned without the need for an object detector.

Fig. 5. Horses through the ages. These horses have been aligned using a DPM trained from Google Images.

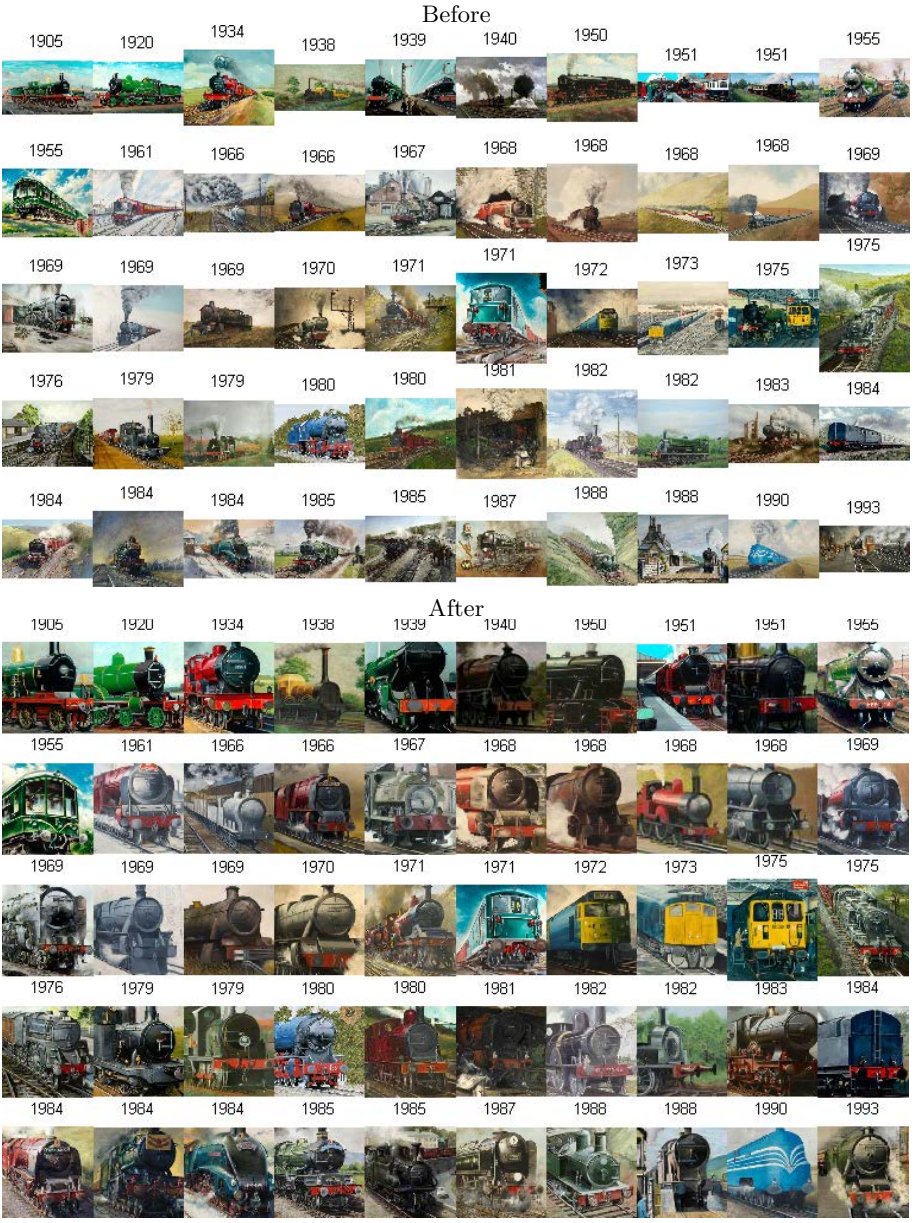

Fig. 6. Train Alignment. The top half of this figure shows paintings that have been retrieved using a train classifier learnt using CNN features. Although all the images contain trains these are at different scales, positions and viewpoints. By utilizing a DPM, it is possible to obtain the location and orientation of each train as in the bottom half of the figure; this mosaic of aligned trains allows for much easier comparison.

flipping regions found by a mirrored component it is possible to display objects facing the same way as in figure 6.

DPM Discussion. In figure 6 the DPM used has been learnt using (i) (above), the results displayed are for the component corresponding to a train face. The aligned horses in figure 5 have been learnt using (ii). It is clear that alignment is better with correct ROIs but rough alignment is still achieved from a DPM learnt from entire images, allowing the objects to appear facing the same way.

Observations. The mosaics give us some insight into the nature of the objects throughout time. It is rather remarkable that the pencil moustache, typically associated with 20$^{\text{th}}$ Century actors like Errol Flynn appears in a portrait from 1565 (figure 4: top row). One can notice styles of particular times; several men around the late 19$^{\text{th}}$ Century have combined their moustaches with sideburns.

Consider the horses in figure 5, it can be seen that in later years there is a more prominent portrayal of muscles. The context is rather interesting; horses are accompanied largely by jockeys but there is an instance of a horse mounted by a solider in 1902. Later paintings tend to have the horse against a plain background rather than in the wild.

We can infer from the bottom half of figure 6 that trains first started to appear in paintings in the early 1900s. Seemingly artists prefer painting steam engines rather than their diesel or electric equivalents as these appear with the greatest frequency. Most of the trains have round faces; rectangular faced trains are most prevalent in 80s paintings.

6 Conclusions

In this paper we have demonstrated the benefit of using object classifiers learnt using CNN features from natural images to retrieve paintings containing an object for a large variety of objects. We have further shown that this process can be carried out in just seconds using our on-the-fly system.

Although this system works for many objects some prove elusive, particularly when there are large differences between the portrayal of the object in natural images and paintings. Several of these elusive objects are human body parts (eye, hand etc.); future work could use pose estimators to isolate areas in images (both natural and not) containing these objects, an area that has been partially investigated in the interesting study by Carneiro *et al.* [12].

Another difficulty is aligning objects in paintings using DPMs without image ROIs. This could be approached by utilizing discriminative regions [10,30,45] to isolate the object as in [17], allowing for better alignment. Query expansion [16] could also be explored: using retrieved paintings in conjunction with the initial training data to learn new classifiers that are able to find objects in paintings that previous classifiers have missed.

Acknowledgments. Funding for this research is provided by the EPSRC and ERC grant VisRec no. 228180. We are very grateful to Rachel Collings and Andy Ellis at the Public Catalogue Foundation and to Rob Cooper at BBC Research. We are

thankful to Ken Chatfield and Karen Simonyan for providing their CNN and VISOR implementations and for help in their use.

References

1. BBC - Your Paintings. http://www.bbc.co.uk/arts/yourpaintings/
2. Bing image search. http://www.bing.com/images
3. Deepeval encoder. http://www.robots.ox.ac.uk/~vgg/software/deep_eval/
4. Encoding methods evaluation toolkit. http://www.robots.ox.ac.uk/~vgg/software/enceval_toolkit/
5. Google image search. http://www.google.com/images
6. The Paintings Dataset. http://www.robots.ox.ac.uk/~vgg/data/paintings/
7. Your Paintings tagger. http://tagger.thepcf.org.uk/
8. Arandjelović, R., Zisserman, A.: Multiple queries for large scale specific object retrieval. In: Proc. BMVC (2012)
9. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proc. CVPR (2012)
10. Aubry, M., Russell, B., Sivic, J.: Painting-to-3D model alignment via discriminative visual elements. ACM Transactions of Graphics (2013)
11. Burke, J.: Nakedness and other peoples: Rethinking the italian renaissance nude. Art History **36**(4), 714–739 (2013)
12. Carneiro, G., da Silva, N.P., Del Bue, A., Costeira, J.P.: Artistic image classification: An analysis on the PRINTART database. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 143–157. Springer, Heidelberg (2012)
13. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: An evaluation of recent feature encoding methods. In: Proc. BMVC (2011)
14. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: Proc. BMVC (2014)
15. 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, Part II. LNCS, vol. 7725, pp. 432–446. Springer, Heidelberg (2013)
16. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. ICCV (2007)
17. Crowley, E.J., Zisserman, A.: The state of the art: Object retrieval in paintings using discriminative regions. In: Proc. BMVC (2014)
18. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: Proc. CVPR, vol. 2, pp. 886–893 (2005)
19. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. J. Artif. Intell. Res. (JAIR) **26**, 101–126 (2006)
20. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. CVPR (2009)
21. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. CoRR abs/1310.1531 (2013)
22. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visualrecognition. arXiv preprint arXiv:1310.1531 (2013)

23. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. IJCV **88**(2), 303–338 (2010)
24. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC 2012) (2012). http://www.pascal-network.org/challenges/VOC/voc2012/
25. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR **9**, 1871–1874 (2008)
26. Felzenszwalb, P.F., Grishick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE PAMI (2010)
27. Felzenszwalb, P.F., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proc. CVPR (2008)
28. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. CVPR (2014)
29. Juan, R.: The turn of the skull: Andreas Vesalius and the early modern memento mori. Art History **35**(5), 958–975 (2012)
30. Juneja, M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: Proc. CVPR (2013)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)
32. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011)
33. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989)
34. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
35. Lee, Y., Efros, A., Hebert, M.: Style-aware mid-level representation for discovering visual connections in space and time. In: ICCV (2013)
36. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV, pp. 1150–1157 (September 1999)
37. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proc. CVPR (2014)
38. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: Proc. ICCV (2011)
39. Parkhi, O.M., Vedaldi, A., Zisserman, A.: On-the-fly specific person retrieval. In: International Workshop on Image Analysis for Multimedia Interactive Services. IEEE (2012)
40. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: Proc. CVPR (2010)
41. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
42. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN Features off-the-shelf: An Astounding Baseline for Recognition. CoRR abs/1403.6382 (2014)
43. 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)
44. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting Image Databases from the Web. IEEE PAMI **33**(4), 754–766 (2011)

45. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 73–86. Springer, Heidelberg (2012)
46. Woodall, J.: Laying the table: The procedures of still life. Art History **35**(5), 976–1003 (2012)
47. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional neuralnetworks. arXiv preprint arXiv:1311.2901 (2013)