Image and Vision Computing 52 (2016) 206–217

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Discovering object aspects from video ☆

Anestis Papazoglou*, Luca Del Pero, Vittorio Ferrari

University of Edinburgh, UK

ARTICLE INFO

Article history:
Received 13 September 2015
Received in revised form 19 March 2016
Accepted 21 April 2016
Available online 4 May 2016

Keywords:
Visual aspects
Object aspect discovery

ABSTRACT

We investigate the problem of automatically discovering the visual aspects of an object class. Existing methods discover aspects from still images under strong supervision, as they require time-consuming manual annotation of the objects' location (*e.g.* bounding boxes). Instead, we explore using video, which enables automatic localisation by motion segmentation. We introduce a new video dataset containing over 10,000 frames annotated with aspect labels for two classes: cars and tigers. We evaluate several strategies for aspect discovery using state-of-the-art descriptors (*e.g.* CNN), and assess the benefits of using automatic video segmentation. For this, we introduce a new protocol to evaluate aspect discovery directly, in contrast to the general trend of evaluating it indirectly (*e.g.* its impact on a recognition pipeline). Our results consistently show that leveraging the nature of video to discover visual aspects yields significantly more accuracy. Finally, we discuss two new applications to showcase the potential of aspect discovery: image retrieval of aspects, and learning aspect transitions from video.

1. Introduction

Traditionally, visual aspects have been defined as distinct viewpoints of rigid 3-D objects [1,2,3,4]. However, viewpoint alone cannot capture the appearance variations of complex, articulated objects in natural images. For example, tigers seen from a similar viewpoint can look very different due to articulated pose (*e.g.* a tiger lying and a tiger standing, Fig. 1). We use a broader notion of aspect that considers four factors of variation: viewpoint, articulated pose, occlusions and cropping by the image border. We explore the problem of automatically discovering such aspects from natural images of an object class. This task requires finding different object instances showing the same aspect (*e.g.* tigers running to the right, face close-ups, Fig. 1).

While some recent methods discover aspects from *still images* [5,6,7,8,9,10], they all require manual annotations of the object's location (*e.g.* bounding boxes). Location annotations allow focussing on the appearance of the object rather than the background, but they are expensive and time-consuming to create. In this paper instead we discover aspects from *video*, where we can segment the foreground objects from the background automatically, by exploiting motion [11,12,13]. Hence, it is possible to discover aspects under weak supervision, *i.e.* only one label per video shot is required. As an additional advantage, we can easily obtain video data for a large number of classes from several sources (*e.g.* DVDs, YouTube).

We present an extensive exploration of weakly-supervised aspect discovery in video, which we pose as an image clustering problem (Section 5). We measure the quality of the discovered aspects in terms of the compactness and diversity of the clustering (Section 6.1). We experiment with several modern appearance descriptors (SIFT [14], shape contexts [15], CNN features [16]), and various levels of spatial support (*e.g.* whole image, foreground segmentation). This enables to carefully evaluate the benefits of automatically segmenting objects (Section 6).

Our exploration relies on a new protocol for evaluating aspect discovery directly. In contrast, previous works evaluate aspect discovery indirectly, typically by measuring its impact on object detection performance [5,6,7,8]. For this, we collected a large dataset sourced from videos of two different classes, car and tiger (for a total of 2664 video shots, Section 4). The choice of the car and tiger classes allows us to explore two very different scenarios. Cars are rigid objects, and the major factors of aspect variations are different viewpoint, occlusions and croppings. Tigers display a broader range of different poses due to their complex articulation (Fig. 1). As an additional difference, cars exhibit higher intra-class variability in color and shape than tigers (*e.g.* different makes).

We annotated a few frames per shot with ground-truth aspect labels using an efficient labelling scheme (totalling over 10,000 frames, Section 3). This scheme captures the four factors of aspect variation by labelling simple, discrete properties of the object's physical parts. For example, we can distinguish between the top two

☆ This paper has been recommended for acceptance by Sinisa Todorovic.

* Corresponding author.
E-mail addresses: a.papazoglou@sms.ed.ac.uk (A. Papazoglou), ldelper@staffmail.ed.ac.uk (L. Pero), vferrari@staffmail.ed.ac.uk (V. Ferrari).

http://dx.doi.org/10.1016/j.imavis.2016.04.014
0262-8856/

Fig. 1. Aspects discovered by our method (one per row). Despite showing tigers from the same viewpoint, the top two aspects look very different due to articulated pose and cropping. Our notion of aspect considers all these factors (Section 3).

aspects in Fig. 1 by considering that the hind legs are not visible in the second. We plan to release this dataset and the aspect labels.

Our experimental exploration demonstrates the great potential of using video for weakly supervised discovery (Section 6). In particular, the accuracy of the discovered aspects improves significantly if we use motion segmentation to get an estimate of the object location. After evaluating aspect discovery directly, we also show that it is useful for other applications. First, we use the aspects discovered by our system to enable a new kind of image retrieval based on aspects (Section 7.1). Second, we exploit the temporal nature of video to learn models of aspect transitions (*e.g.* from lying to standing, Section 7.2).

The rest of the paper is organized as follows. We start by discussing the two main components of our evaluation protocol: the labelling scheme (Section 3) and the dataset (Section 4). We then present several strategies for aspect discovery (from both videos and still images, Section 5) and present the results of our extensive exploration (Section 6). We conclude by introducing two applications that benefit from aspect discovery (Section 7).

2. Related work

2.1. Early work on aspects

Early work considered simple objects for which all possible aspects could be exhaustively enumerated [1,2,3]. More recently, Cyr and Kimia [4] tried to learn a manageable collection of representative views of an object instance. All these methods are limited to synthetic views of a single object instance.

2.2. Aspect discovery

Several methods [5,6,7,8,9,10,17,18,19] discover aspects implicitly, in order to train specialised classifiers for each of them (components of a mixture model). Some of these works [5,6,7,8] cluster HOG descriptors extracted from bounding boxes in the training images (manually annotated). Others [9,10] use exemplar SVMs [20] as a similarity measure between bounding boxes to drive the clustering. A few methods require additional time-consuming annotations, such as the location of object parts [17] or keypoints [18,19]. None of the methods above is weakly supervised. Moreover, while aspect discovery is a crucial intermediate step in their pipeline, it is evaluated only indirectly by measuring the performance improvement of the overall system.

2.3. Aspects in multi-view models

The works above use the discovered components in isolation. In contrast, other methods take the relationships between different aspects into account to build multi-view models [21,22,23,24,25]. They either require expensive bounding-box and viewpoint annotations for each training image [21,22,23] or very detailed 3-D CAD models [24,25]. Only the work of [26] uses video for this task. Their method is trained on a single short cellphone video per class, taken by walking around the object. While this procedure captures viewpoints well, it might fail to record other factors of variation, such as articulated pose. Moreover, it is not easily applicable for certain classes, such as wild animals. In practice, [26] only considers common rigid objects *i.e.* cars, motorbikes, wheelchairs, *etc.*

2.4. Modelling pose variations with parts

In the context of object detection and segmentation, some works [18,27,28] model variations in pose and articulation using poselets, *i.e.* parts that are tightly clustered in both appearance and configuration space (*e.g.* crossed hands, frontal face). This is somewhat related to our definition of aspects in terms of part properties (Section 3). However, learning poselets requires manual annotation

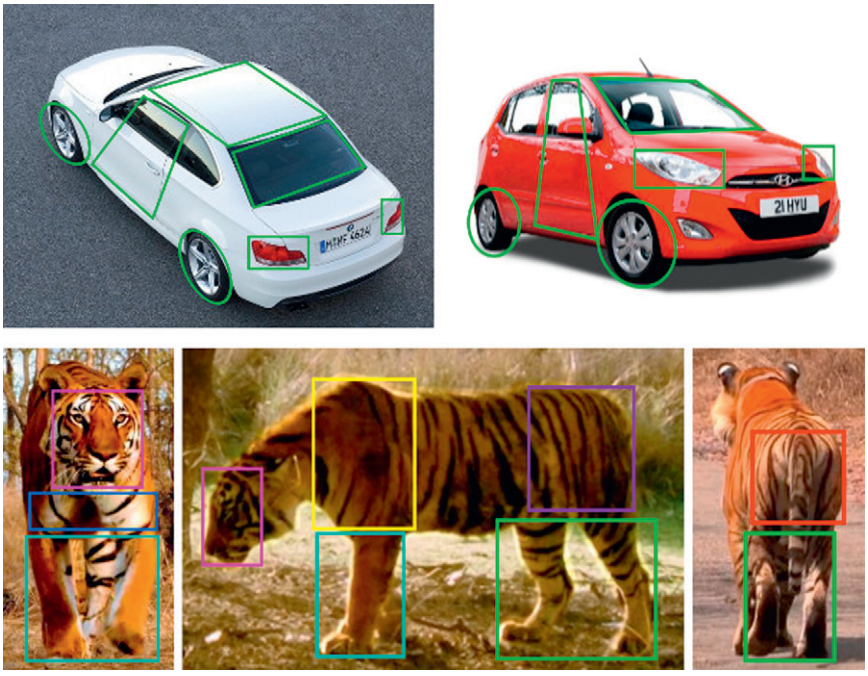

Fig. 2. Part visibility labels. (*Top*) We annotate 13 physical parts of cars with visibility tags (Section 3). (*Bottom*) We annotate 9 physical body parts of tigers with visibility tags. Note that our annotation is weak, we do not mark the parts with bounding boxes.

of keypoints [18,27] and 3-D joint configurations [18], so they are not suitable for weakly supervised aspect discovery.

3. Aspect labels

Our labels accurately capture the four factors of aspect variation (viewpoint, articulated pose, occlusions, cropping), by considering simple properties of the object's physical parts (*e.g.* head, legs, Figs. 2 and 3). We uniquely identify the viewpoint, occlusions and cropping by considering which parts of the objects are visible in the image (*e.g.* when a tiger is seen from the back, the face is not visible, Fig. 2). We capture pose variations using additional configuration labels for the articulated parts (*e.g.* standing, lying for legs).

This scheme provides a compact yet fine-grained description of the object's aspect. As an additional advantage, it is easy to annotate accurately and unambiguously. Moreover, it naturally allows us to define a distance between aspects, which we will use for evaluation. Note, that we use these aspects labels only to evaluate the quality of the aspect clusters discovered by our method (Section 6.1) by evaluating the label similarity between frames in a cluster. During aspect discovery, we do not try to estimate the labels themselves (*e.g.* we do not try to localise object parts).

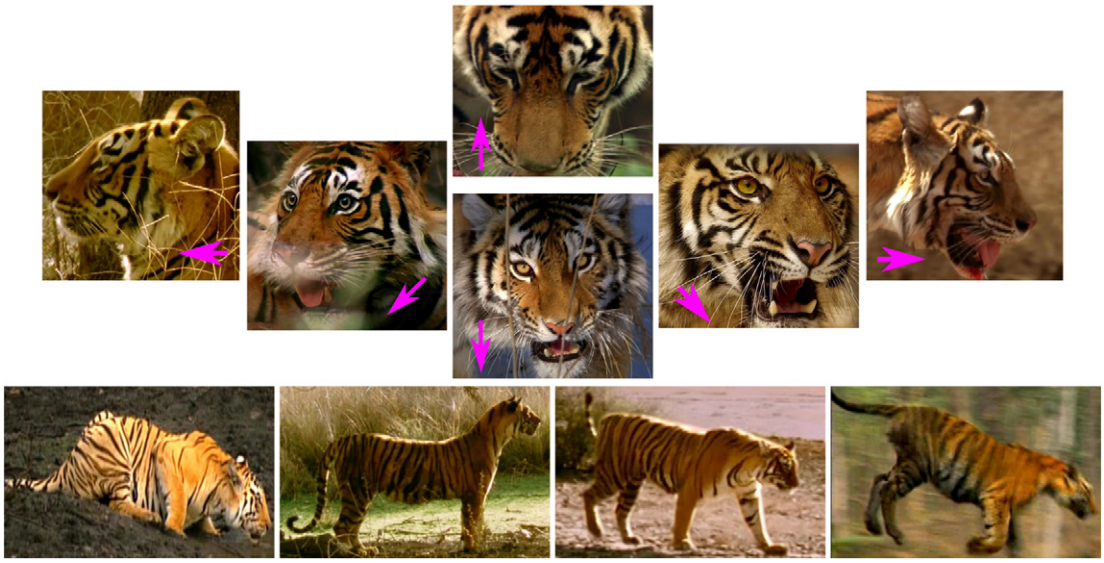

Fig. 3. Part configuration labels (for tigers only). (*Top*) We annotate 6 different face orientations (Section 3). (*Bottom*) We annotate 4 different leg configurations.

3.1. Part visibility labels

For cars we use 13 parts: windscreen, wheels, lights, frontal doors and roof (Fig. 2 top). For tigers we use 9: face, sternum, left and right shoulders, left and right thighs, front and hind legs, and buttocks (Fig. 2 bottom). We annotate a part as visible if more than 50% of the area of that part is visible.

3.2. Part configuration labels

For tigers, we choose the orientation of the face from six possible orientations (when visible, Fig. 3 top). This allows to distinguish across different face close-ups, which are very frequent in animal videos. We also choose the leg configuration from: lying, standing, walking and running (Fig. 3). This property is indicative of both pose and appearance (due to motion blur). It is significantly easier and less time-consuming for humans to annotate than, say, specifying the angles of the joints of the leg.

3.3. Distance between aspects

We now define a distance to measure the similarity between two aspects. For instance, walking to the right should be closer to running to the right than a face close-up. Standing facing right should be closer to laying facing right than to laying facing towards the camera. Our distance captures such transitions in aspect space smoothly by using the part labels. We argue that this is much more expressive than considering aspects as mutually exclusive categories, which would require complex hand-defined rules to determine the distance between every pair of aspect categories. Instead, our distance measures similarity by simply considering how many parts are common between the two aspects.

Let A_i and A_j be two aspects. We define:

$$D(A_i, A_j) = 1 - \sum_p \; d_p(A_i, A_j) \; |V(A_i) \cup V(A_j)| \tag{1}$$

where d_p is the distance with respect to part p, and $V(A)$ the set of visible parts in A; $d_p(A_i, A_j) = 1$ if p is visible in both aspects, 0 otherwise. For face and legs, d_p further depends smoothly on the difference in orientation/action (Fig. 4).

4. Dataset

We assembled a dataset containing several hundreds video shots for two different classes (car and tiger). We annotated frames in each shot with the *aspect labels* (Section 3), which allows direct evaluation of aspect discovery (Section 5). Finally, we exploit the nature of video to provide automatic object localisation for each frame using foreground segmentation through motion.

We collected the shots from 188 car ads (~1–2 min each) and 14 nature documentaries about tigers (~40 min), amounting to roughly 14 h of video. We automatically partitioned these raw videos into shorter shots [29], and kept only those showing at least one instance of the class. This produced 806 shots for the car and 1880 for the tiger class, typically 1–100 sin length.

We annotated aspect labels as follows. First, we randomly chose five frames per shot, and annotated each of them with the number of objects shown. We then gave aspect labels only to frames showing exactly one object (to avoid ambiguities). This produces a total of 6610 frames with aspect label for tigers, and 3485 for cars.

Last, we used [12] to automatically segment the foreground in each shot. For the frames with aspect labels we also marked whether the segmentation is *accurate*. The segmentations and the aspect labels will be available on our website.

4.1. Statistics

For the aspect labels, we observed 643 unique combinations for the tigers, and 293 for cars. Some are more frequent, for example there are 221 frontal face close-ups.

In order to measure the accuracy of the segmentation algorithm, we have manually annotated one frame for each shot with a bounding box on the object. We measure accuracy using the CorLoc performance measure of [30], *i.e.* the percentage of bounding boxes which are correctly localised up to the PASCAL VOC [31] criterion (intersection-over-union $\geq$0.5). For the purposes of this evaluation, we automatically fit a bounding box around the largest connected component of the segmentation output. The segmentation algorithm achieves get 55% CorLoc, which is in line with the results reported in [12] on another dataset (YouTube-Objects [30]).

5. Automatic aspect discovery from video

We treat aspect discovery as a frame clustering problem. We explore two families of descriptors: bag-of-visual-words (BoVW, Section 5.1) and Convolutional Neural Networks (CNN, Section 5.2). We consider various spatial support over which to compute descriptors, including the whole frame or the foreground segmentation produced automatically by [12].

5.1. Bag-of-Visual-Words descriptors

The Bag-of-Visual-Words (BoVW) approach models an image as an orderless collection of *visual words* (*i.e.* quantized local features).

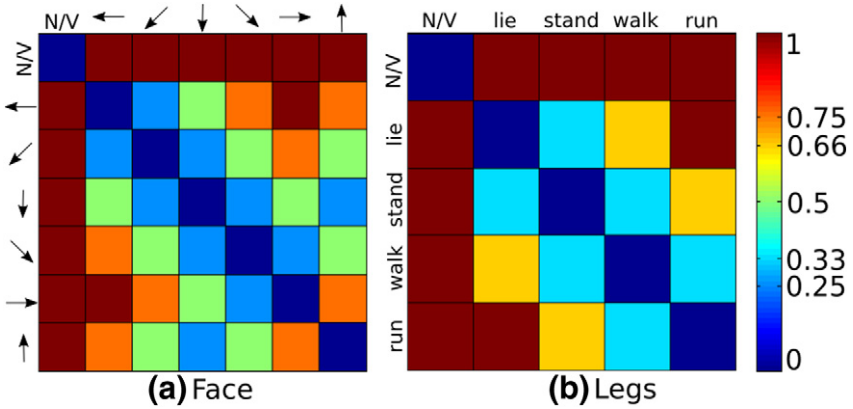

Fig. 4. (a) Distance matrix for the "face" part (d_p, Section 3). The entries show the distance between the different face orientations, denoted by the arrows (Fig. 3 top). *N/V* denotes that the part is not visible. (b) Distance matrix for the "legs" part (d_p, Section 3). The entries show the distance between the different leg configurations (Fig. 3 bottom).

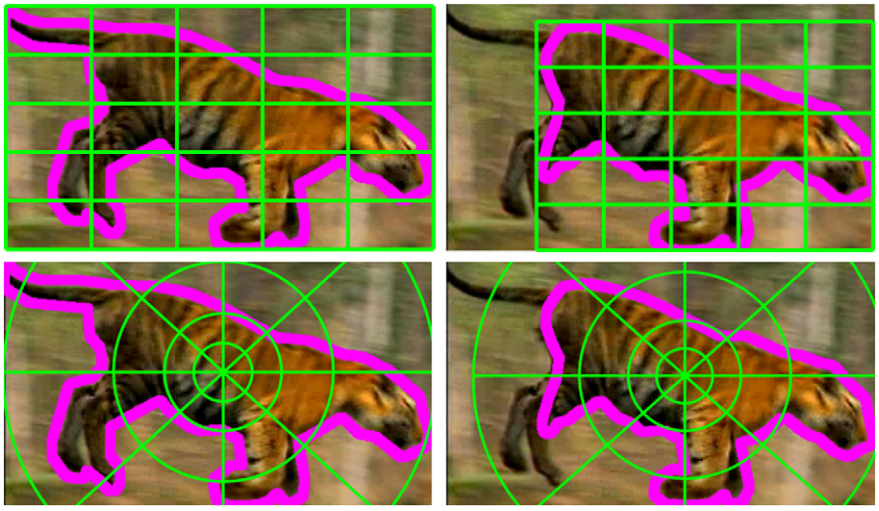

Fig. 5. Spatial binning for BoVW descriptors. (*Top*) A rectangular grid fit over the segmentation [12]. Even small segmentation errors (right) lead to a very different configuration of the spatial bins. (*Bottom*) A log-polar grid placed on the centre of mass of the segmentation. Both the centre of mass and the radius of the log-polar grid are robust to small segmentation errors (Section 5.1).

The BoVW descriptor is a histogram recording the frequencies of the visual words over a *spatial support* of interest (*e.g.* an entire image or an image region). While this disregards information about the spatial layout of the image, adding geometric information using *spatial binnings* (Fig. 5) can help image classification [32] and object detection [33] performance.

We consider various combinations of visual words (SIFT [14] and shape-contexts [34]), spatial supports (*e.g.* foreground segmentation [12]), and spatial binnings (*e.g.* spatial pyramids [32]). Each combination produces a different BoVW descriptor.

5.1.1. Visual words

First, we consider dense *SIFT* [14] computed on 4×4 pixel patches at every pixel. Second, we use [34] to extract shape-context features from the contour of the segmentation. We convert these features into visual words using a vocabulary of 1000 visual words for *SIFT* and 100 for shape-contexts.

5.1.2. Spatial support

We consider three types of spatial supports to determine the extent to which each feature contributes to the BoVW: *whole frames*, *segmentation* [12], and *motion saliency* [12]. We use a general, uniform treatment for all supports, by assigning a weight $w_i \in [0, 1]$ to each pixel i in the frame. The feature at i contributes by w_i to the BoVW.

For whole frames, we give equal weight to all pixels (*i.e.* $w_i = 1 \forall i$). For the segmentation we set $w_i = 1$ if i is part of the foreground, otherwise $w_i = 0$. Motion saliency uses motion to compute the probability p_i that pixel i is part of foreground (we simply set $w_i = p_i$). This can be seen as a soft version of the segmentation. Typically, it produces a roughly correct localisation even when the segmentation is very inaccurate (Fig. 6).

Since shape-contexts are defined on object contours, we only use them with the segmentation (we try all supports for SIFT). Last, note how segmentation and motion saliency enable to measure appearance purely on the object, excluding the background. They are made possible by exploiting the temporal nature of video.

5.1.3. Spatial binning

The basic idea of spatial binning is to partition the spatial support into a fixed set of spatial bins, and compute a separate histogram for each. Here, we consider two different variants.

First, we use 3-level spatial pyramids over a *rectangular grid* [32]. Second, we propose a *log-polar radial* binning inspired by [34]. The log-polar bins are placed on the centre of mass of a given spatial support (Fig. 5). We use 8 angular bins and 6 radial bins. To achieve scale invariance, the step of the radial bins is proportional to the scale of the spatial support, *i.e.* the average distance between each pixel i and the centre of mass weighted by w_i. This scheme is more robust to small errors in the segmentations than a rectangular grid (Fig. 5). Last, we consider orderless BoVWs (‘*no binning*’) as a baseline.

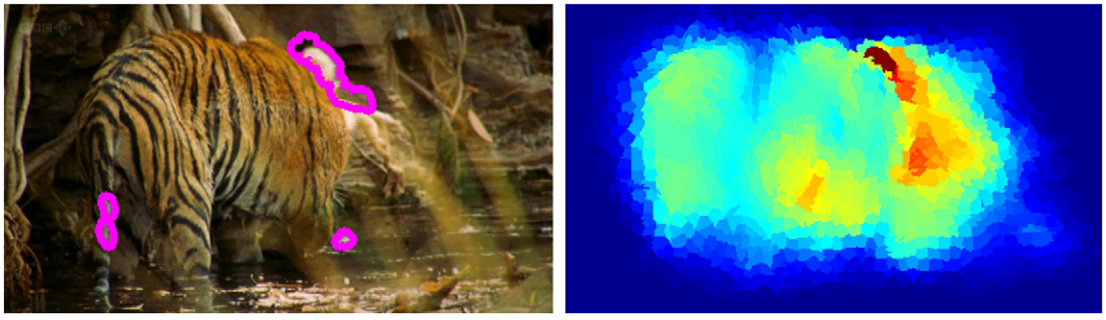

Fig. 6. Spatial support. Motion saliency (*right*) estimates the probability of being part of the foreground at each pixel (Section 5.1). It often provides a good rough localisation even when the segmentation fails (*left*), where pixels are instead hard assigned to foreground/background.

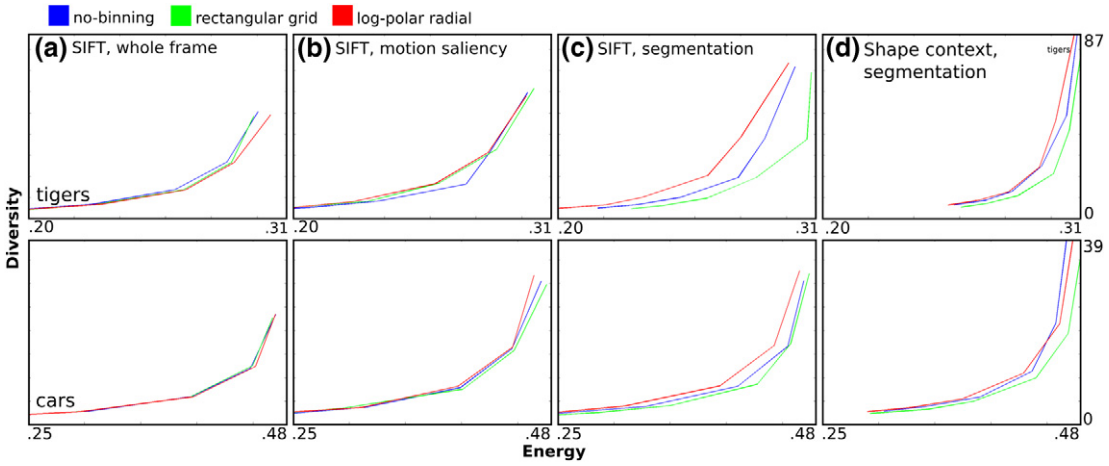

Fig. 7. Comparison of different spatial binnings for bags-of-visual-words descriptors (Section 6.2). The first and second rows correspond to tigers and cars, respectively.

5.2. CNN descriptors

CNN descriptors achieve state-of-the-art performance on various tasks (*e.g.* classification [35], detection [36]). Like BoVWs, CNN descriptors can be computed on different spatial supports. Note that the concept of binning does not apply here.

5.2.1. Spatial support

First, we extract 4096-dimensional CNN descriptors from the whole frame using CAFFE [16] (we use the AlexNet network architecture [35]). This CNN model was trained for whole image classification on Imagenet [37]. Second, we extract 4096-dimensional CNN descriptors from the bounding box of the segmentation. Here, we use a model fine-tuned for object localisation on class-agnostic object proposals [36]. We found this to be more suitable for the segmentation support. We do not consider motion saliency, since incorporating individual pixel weights into the CNN framework is not straightforward.

5.3. Clustering

We cluster frame descriptors using k-medoids, which is suitable for any distance function. We compute distances between BoVW descriptors using histogram intersection. For CNN descriptors we use Euclidean distance. For efficiency, we precompute the distance matrix between all frames before clustering. We cluster 1000 times and keep the clustering with the lowest energy to reduce the effects of random initialisation.

6. Evaluation of aspect discovery

6.1. Protocol

For evaluation, we use two different criteria: clustering energy and diversity. The combination of these two carefully designed measures provides a complete picture of the quality of the clustering.

6.1.1. Clustering energy

This measures the compactness of the clusters, *i.e.* it penalises assigning dissimilar aspects to the same cluster. Let A_k be the medoid of cluster k, *i.e.* the aspect in k minimising the sum of distances to all other aspects in k. We define the energy as: $\frac{1}{N}\sum_k\sum_{j\in k} D(A_k, A_j)$, where N is the total number of points being clustered. This is a generalisation of the standard purity evaluation measure [38] for a continuous label space, *i.e.* using a smooth D penalises putting items with different labels in the same cluster proportionally to their distance.

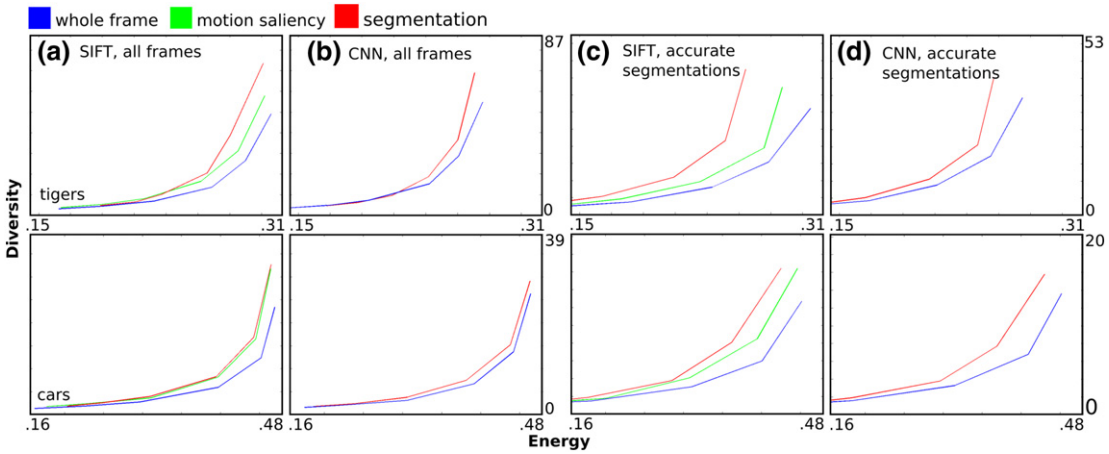

Fig. 8. Comparison of different spatial supports (Section 6.2). All SIFT plots (a, c) use log-polar binning. The first and second rowscorrespond to tigers and cars, respectively.

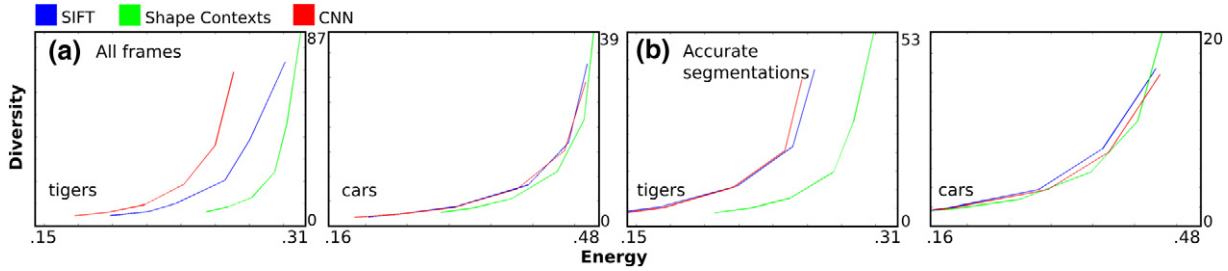

Fig. 9. When using segmentation as spatial support, CNN is better or comparable to the other descriptors (a). If we evaluate only on frames where the segmentation is accurate (b, Section 4), the gap between SIFT and CNN is significantly reduced, especially on tigers (Section 6).

6.1.2. Clustering diversity

In the video domain, energy can be trivially minimised by clustering together all frames in a shot, which on average contains only 1–2 aspects of the same object instance. Instead, applications using these aspect clusters need to see different object instances of the same aspect (*e.g.* learning a multi-view class model, or retrieving different instances of a query aspect, Section 7.1).

Hence, we also measure the diversity of a cluster, *i.e.* the average number of different shots per cluster: $\frac{1}{K}\sum_k |S_k|$, where K is the number of clusters and S_k is the set of shots present in cluster k. Diversity rewards clustering together occurrences of the same aspect from different shots (hence different object instances).

6.2. Results

We present here an extensive exploration of the various descriptors for aspect discovery (Section 5) on our dataset (Section 4). We evaluate each descriptor separately by computing clustering energy and diversity. Since the true number of aspect clusters is not known a priori we experiment with different numbers of clusters: 50, 100, 200, 400, 600 and 800. Last, we explore learning a better distance for clustering by combining them.

6.2.1. Spatial binning

We first evaluate spatial binnings for SIFT on the whole frame (Fig. 7a). Interestingly, both rectangular grid and log-polar radial are comparable to no binning, which is in contrast to the findings of [32] for image classification. This happens because most bins end up covering the background regardless of the choice of spatial binning, when applied to whole frames. On motion saliency (Fig. 7b), log-polar radial and rectangular grid perform similarly, both being slightly better than no binning. On segmentation, log-polar performs significantly better than rectangular grid for both SIFT (Fig. 7c) and shape contexts (Fig. 7d). No binning performs better than rectangular grids, showing that naive rectangular grids are not robust to small

Fig. 10. Example aspect clusters discovered for the tiger class. Each row corresponds to a different cluster. Here, we used CNN on segmentation as descriptors (Section 5.2).

errors in the automatic segmentations (Fig. 5). In all the following experiments we use log-polar binning, as it always performs equally or better than the alternatives.

6.2.2. Spatial support

Here we evaluate the different spatial supports. For the SIFT descriptors (Fig. 8a), both segmentation and motion saliency outperform whole frame, with segmentation offering the best performance. This is because it allows to focussing on the appearance of the foreground object. Instead, whole frame is confused by the background, which has little correlation with the object's aspect. When clustering only the frames with accurate segmentation (Section 4), the segmentation spatial support outperforms the others by an even larger margin (Fig. 8c).

Experiments on CNNs reveal the same trend, *i.e.* segmentation outperforms the whole frame (Fig. 8b), and the gap between them increases when using only accurate segmentations (Fig. 8d).

These experiments demonstrate that video offers an advantage over still images as it enables automatic object localisation. Using segmentation improves on the other supports even if it is accurate only half of the time (Section 4). When we focus on frames with accurate segmentations only, the gap increases substantially. This indicates that further advances in video segmentation can lead to even better aspect discovery. Figs. 10 and 11 show some aspect clusters found using CNN on segmentation.

6.2.3. Descriptors

Here we compare the different descriptors (SIFT BoVW, shape-contexts BoVW, CNN, Fig. 9). For each, we use the best combination of spatial support/binning based on the experiments above.

Shape-contexts is generally inferior to the others, especially on tigers (Fig. 9b), possibly because the automatic segmentations often miss the fine details of the contours (*e.g.* paws, tail).

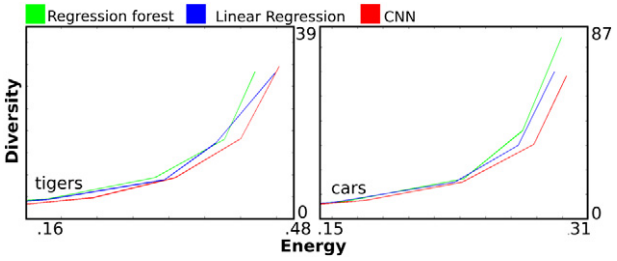

Fig. 12. Distance learning. We learn a distance function that combines all the individual descriptors tested for clustering (Section 6), using two alternative regression methods: linear (pink) and regression forest (cyan). Both outperform CNN on segmentation (red), which is the descriptor that individually performs best (Section 6).

CNN outperforms SIFT BoVW significantly on tigers, while they are comparable on cars. When clustering only frames with accurate segmentations, SIFT performs better than CNNs on cars, and is comparable on tigers. This goes against the general trend of CNN outperforming SIFT for various computer vision tasks [35,36,39,40]. This might be because CNN do not take full advantage of the detailed pixel-wise support that the segmentation provides, as they are extracted from its bounding box. Unfortunately, extracting CNNs from a pixel-wise support is still an open problem. Given the ongoing advancements in automatic video segmentation [11,12,13], this is a promising area to explore.

6.2.4. Distance learning

Here, we explore combining all the descriptors mentioned above in order to improve the clustering. Intuitively, we want to drive the clustering with a distance that is as close as possible to the true distance between aspects (1). We pose this as a regression problem: we

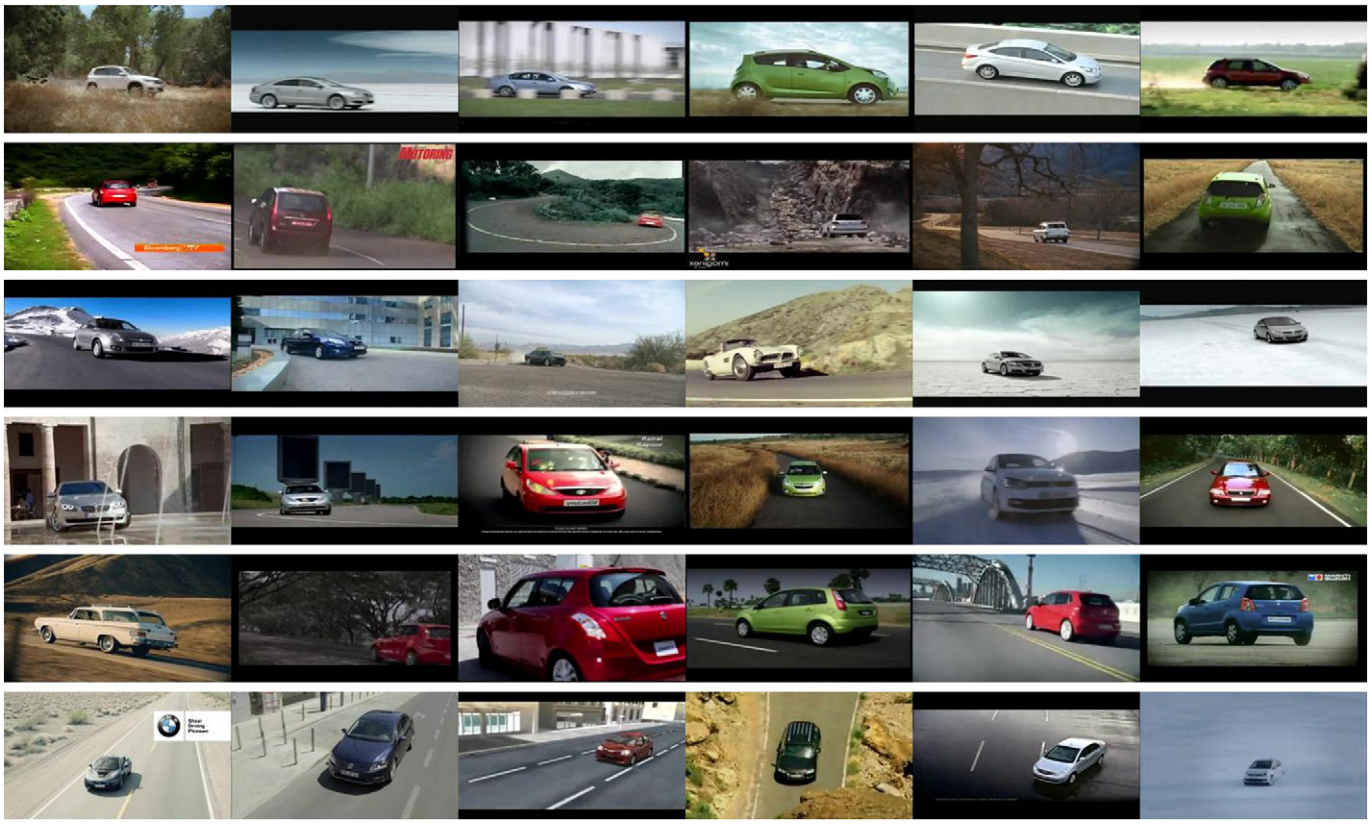

Fig. 11. Example aspect clusters discovered for the car class. Each row corresponds to a different cluster. Here, we used CNN on segmentation as descriptors (Section 5.2).

use the distances computed with respect to individual descriptors as predictors, and the distance (1) between ground-truth aspect labels as target.

We begin by splitting the dataset into two halves. We first train a regressor to predict the distance between ground-truth labels (1) from the distances of the individual descriptors in one half. Then we

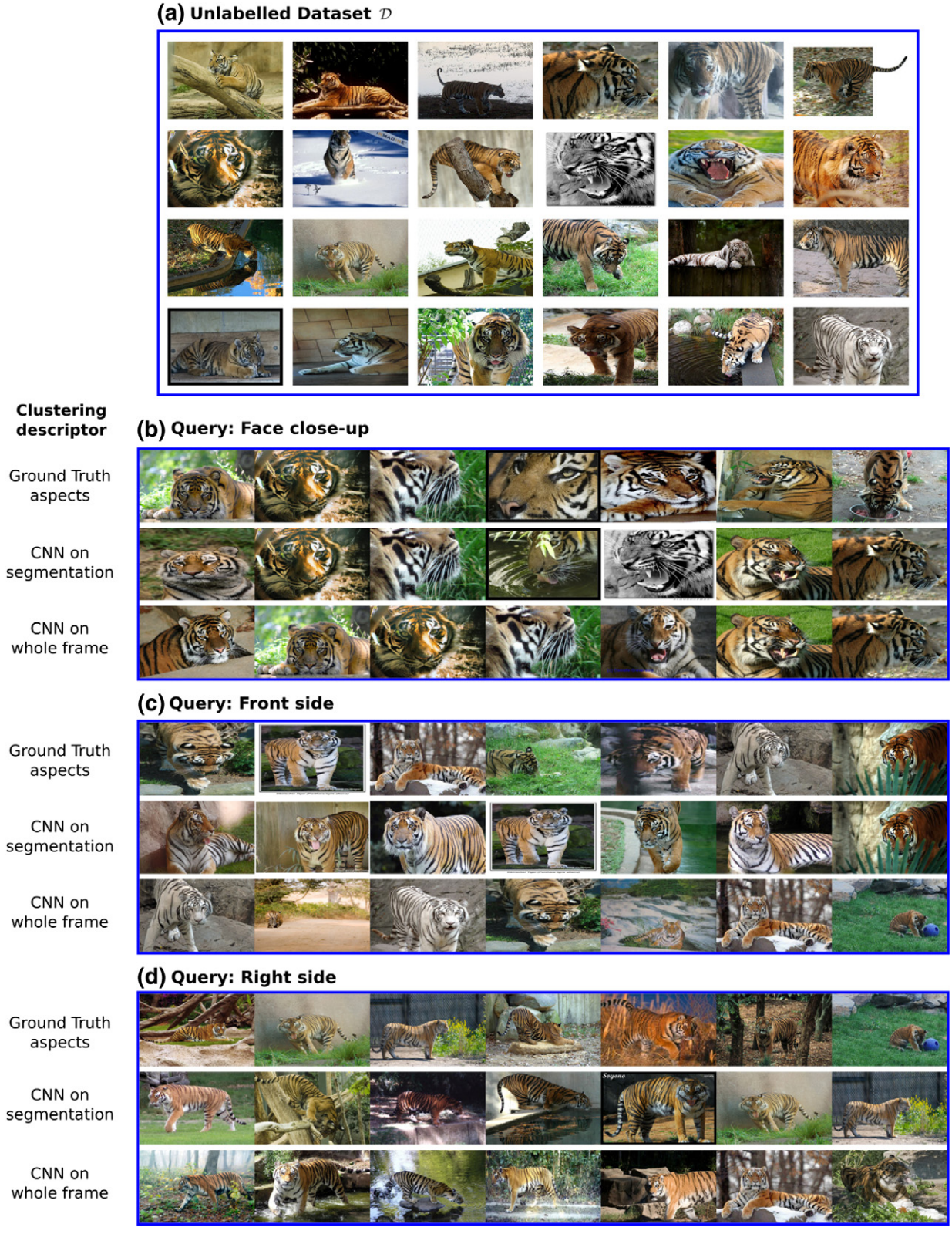

Fig. 13. Our aspect retrieval system (Section 7.1) supports searching a database $\mathcal{D}$ of unlabelled images using an aspect semantic label as query (*e.g.*"Right side" or "Face close-up"). Here, we show a subset of $\mathcal{D}$ (a), and some examples of the images retrieved by our system (b–d). We illustrate the 7 highest scoring images for: face close-up (b), front size (c) and right side (d). Each panel (b–d) shows the output of the retrieval system for three different strategies for aspect discovery: using ground-truth aspects, CNN on segmentation, and CNN on whole frame.

use this regressor to predict distances between frames in the other half, and use them for clustering.

We experiment with two alternative regression models, linear regression [41] and regression forests [42]. For the regression forest we used 250 trees with a depth of 100. Both regressors bring a moderate improvement to using individual descriptors (Fig. 12). While regression forests provide a better approximation of the ground-truth distance, both methods perform equally well for clustering. Note, however, that this experiment requires aspect labels for the training subset, whereas all the experiments before are unsupervised.

6.2.5. Summary of results

The log-polar binning scheme performs best under all circumstances. Segmentation is the best performing spatial support, and in general CNN performs better than SIFT. However, when we focus on videos with accurate segmentation only, the gap between CNN and SIFT disappears. We posit that this happens because CNNs operate on bounding boxes and cannot fully exploit the pixel-level support provided by the segmentation. Experimenting with accurate segmentation only also indicates that advances in video segmentation will lead to better aspect discovery.

7. Applications

We now introduce two possible applications of our aspect discovery system. We discuss an image retrieval system for aspects (Section 7.1), and how to learn transitions in aspect space (Section 7.2).

7.1. Aspect image retrieval

We now discuss an image retrieval application that exploits the aspect clusters discovered by our method. Specifically, we build an "aspect retrieval"system, where a user enters a textual query specifying an aspect with a natural semantic label (*e.g.* frontal tiger, face close-up), and the system automatically retrieves suitable images (Fig. 13b–d) from a large unlabelled database $\mathcal{D}$ (Fig. 13a).

To achieve this, the retrieval system needs to learn about the appearance of each semantic label. The traditional way to do it would require labelling a large number of training images per label. Instead, we use as training data a set $\mathcal{V}$ of videos of the class with no semantic labels. First, we let our system discover clusters of aspects in $\mathcal{V}$. The annotator then assigns *one* semantic label to each cluster, which significantly reduces the annotation effort (33× in the experiments below).

7.1.1. Protocol

For this experiment, we define five semantic aspect labels: face close-up, left side, right side, front side and back side. For training, we use the 6610 frames of the tiger class (Section 4) as $\mathcal{V}$. Instead of manually labelling each individual frame, we cluster them automatically (Section 5.3) using CNN on segmentation as descriptor (Section 5.2). We set the number of clusters to 200. We then label each cluster with the most frequent semantic label in it, choosing from the five options above (the label gets assigned to each image in the cluster). This effectively reduces the number of items to manually annotate from 6610 to 200, reducing the human effort by a factor 33.

For testing, we use a database $\mathcal{D}$ consisting of 200 images of tigers sourced from ImageNet [37] (Fig. 13a). Given a query semantic label, we score each image $I \in \mathcal{D}$ as follows. We find its k nearest neighbours in $\mathcal{V}$ according to the distance with respect to the CNN descriptor. We set the score of I to the number of neighbours with the same semantic label as the query. Finally, we rank the images in $\mathcal{D}$ according to their score and return them to the user.

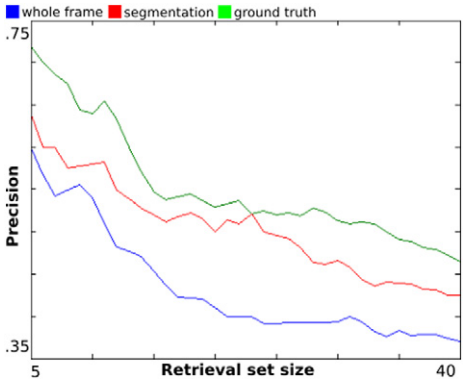

Fig. 14. Results on image retrieval (Section 7.1). The horizontal axis indicates the size of the retrieval set returned to the user. The vertical axis is precision. The curves differ only in the distance used for finding aspect clusters in the video database.

To evaluate the system, we manually annotate ground-truth semantic labels on $\mathcal{D}$, and compute the average precision of the five possible queries (Fig. 14). As baseline, we compare against a system equivalent to the one described above, except that we replace the spatial support used for finding the aspect clusters: CNN on whole frames, rather than on segmentation (Section 5.2). We also compare to an upper bound where we find the aspect clusters using the distance (1) between our ground-truth aspect label annotations (Section 3).

7.1.2. Results

CNN on segmentation (Fig. 14, red curve) clearly outperforms CNN on whole frames (blue curve). The only thing changing between the two curves is the method used for aspect discovery: exploiting video to get a segmentation leads to better aspect discovery, which in turn leads to better image retrieval performance. As expected, discovering aspects using the ground-truth annotations provides an upper bound for these automatic methods, showing that further improvements in aspect discovery would be beneficial to tasks like image retrieval (pink curve).

Fig. 13 shows a few qualitative examples. Consider the query "Right side"(d): when we use ground-truth labels and CNN on segmentation for clustering, five of the seven highest scoring images match the query, *i.e* the tiger in the retrieved images is actually facing right. This degrades to two when we use CNN on whole frame, showing that in general the aspect discovery system benefits from using the segmentation in this case. Instead, the performance of segmentation and whole frame are very similar on face close-up; in this case the tiger occupies most of the image, which allows CNN on whole frame to match the performance of CNN on segmentation.

7.2. Modelling aspect transitions

Another advantage of video over still images is that it allows reasoning about transitions across aspects, for example from frontal head to head facing right, or from lying to standing (Fig. 15). This can be useful in a variety of tasks, such as tracking object instances in new video, aspect-based video retrieval, or as a starting point for learning grammars of aspects.

We consider here learning a probabilistic model of aspect transitions from the ground-truth aspect labels in our video dataset

Fig. 15. Aspect transitions. We learn transitions between aspects from the labels in our dataset, and use them to generate interesting random walks in aspect space (Section 7.2). *Top row*: Tiger turning its head. *Middle row*: Tiger entering the frame and leaving. *Last row*: Tiger lying down, standing up and walking into tall grass.

(Section 3). Let $\mathcal{A}$ be the set of all unique aspects in the dataset (for a total of 643, Section 4). We construct a transition matrix T where each entry

$$T(K,L) = \left(1 - \frac{1}{1+N_k}\right) \cdot P(K,L) + \frac{1}{1+N_k}\Pi(K,L)\,, \tag{2}$$

is the probability of transitioning from aspect K to L. It is computed as the weighted sum of a transition probability P we learn from the ground-truth labels, and a smoothness prior Π (N_k is the number of occurrences of aspect K in the dataset).

We compute $P(K,L)$ from the ground-truth aspect labels as follows. Let $(f_i,f_j)^s_{KL}$ be any two frames in a shot s such that f_i contains an instance of aspect K and f_j an instance of aspect L. Each such pair contributes to $P(K,L)$ by $w(i,j) = e^{(1-|j-i|)}$, *i.e* the probability of transitioning from K to L is greater if f_i and f_j are close in time. This gives

$$P(K,L) = \frac{\sum_s \sum_{(f_i,f_j)^s_{KL}} w(i,j)}{Z}\quad, \tag{3}$$

where $Z = \sum_{A\in\mathcal{A}} P(K,A)$ is the normalisation constant.

To model the transitions between rare aspects more effectively, we include a smoothness prior Π

$$\Pi(K,L) = \frac{1 - D(K,L)}{\sum_{A\in\mathcal{A}} 1 - D(K,A)}, \tag{4}$$

where D is the distance (1) between aspects, which is smooth by construction (Section 3).

We demonstrate the expressiveness of the learnt transitions qualitatively, by using T to produce random walks in aspect space (Fig. 15). We choose the starting aspect A_0 by uniformly sampling from $\mathcal{A}$. At every step t we sample the next aspect A_{t+1} from the transition probability $T(A_{t+1},A_t)$. To visualize the random walk, for each $A_t = K$ we choose one instance of aspect K from those available in the dataset. This approach discovers several interesting aspect transitions, such as standing up (Fig. 15, third row): note how the four tigers illustrating this transition all come from different shots.

8. Conclusions

In this paper, we conducted an extensive exploration of weakly-supervised aspect discovery from video. Our exploration was evaluated on a novel, direct protocol. We experimented with several modern appearance descriptors (SIFT, shape contexts, CNN features), and various levels of spatial support (*e.g.* whole image, segmentation). We demonstrated that exploiting the nature of video through the use of automatic foreground segmentation leads to consistently better aspect discovery in all cases. Finally, we showed that aspect discovery can enable new applications, such as semantic-aspect image retrieval, and modelling transitions between aspects.

References

[1] J.J. Koenderink, A.J. van Doorn, The internal representation of solid shape with respect to vision, Biol. Cybern. (1979).
[2] W.H. Plantinga, C.R. Dyer, An algorithm for constructing the aspect graph, FOCS, 1986.
[3] K. Bowyer, J. Stewman, L. Stark, D. Eggert, ERRORS-2: a 3D object recognition system using aspect graphs, Proc. ICPR, 1988.
[4] C.M. Cyr, B.B. Kimia, 3D object recognition using shape similiarity-based aspect graph, ICCV, 2001.
[5] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Trans. PAMI 32 (9).
[6] C. Gu, X. Ren, Discriminative mixture-of-templates for viewpoint classification, ECCV, 2010.
[7] S. Divvala, A. Efros, M. Hebert, How important are 'Deformable Parts' in the deformable parts model?, ECCV, 2012.
[8] B. Drayer, T. Brox, Training deformable object models for human detection based on alignment and clustering, ECCV, 2014.
[9] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, S. Yan, Subcategory-aware object classification, CVPR, 2013.
[10] O. Aghazadeh, H. Azizpour, J. Sullivan, S. Carlsson, Mixture component identification and learning for visual recognition, ECCV, 2012.
[11] Y.J. Lee, J. Kim, K. Grauman, Key-segments for video object segmentation, ICCV, 2011.
[12] A. Papazoglou, V. Ferrari, Fast object segmentation in unconstrained video, ICCV, 2013.
[13] A. Faktor, M. Irani, Video object segmentation by non-local consensus voting, BMVC, 2014.
[14] D. Lowe, Local feature view clustering for 3D object recognition, CVPR, Springer 2001, pp. 682–688.
[15] S. Belongie, J. Malik, Shape matching and object recognition using shape contexts, IEEE Trans. PAMI 24 (24).
[16] Y. Jia, Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding, 2013, http://caffe.berkeleyvision.org/.
[17] H. Azizpour, I. Laptev, Object detection using strongly-supervised deformable part models, ECCV, 2012.
[18] L. Bourdev, S. Maji, T. Brox, J. Malik, Detecting people using mutually consistent poselet activations, ECCV, 2010.

[19] C. Gu, P. Arbeláez, Y. Lin, K. Yu, J. Malik, Multi-component models for object detection, ECCV, 2012.
[20] T. Malisiewicz, Ensemble of Exemplar-SVMs, implementation, 2011, https://github.com/quantombone/exemplarsvm.
[21] S. Savarese, L. Fei-Fei, View synthesis for recognizing unseen poses of object classes, ECCV, 2008.
[22] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, L. Van Gool, Towards multi-view object class detection, CVPR, 2006.
[23] L. Mei, J. Liu, A. Hero, S. Savarese, Robust object pose estimation via statistical manifold modeling, ICCV, 2011.
[24] J. Liebelt, C. Schmid, K. Schertler, Viewpoint-independent object class detection using 3D feature maps, CVPR, 2008.
[25] J. Liebelt, C. Schmid, Multi-view object class detection with a 3D geometric model, CVPR, 2010.
[26] H. Su, M. Sun, L. Fei-Fei, S. Savarese, Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories, ICCV, 2009.
[27] L. Bourdev, J. Malik, Poselets: body part detectors trained using 3D human pose annotations, ICCV, 2009.
[28] T. Brox, L. Bourdev, S. Maji, J. Malik, Object segmentation by alignment of poselet activations to image contours, CVPR, 2011. pp. 2225–2232.
[29] W.-H. Kim, J.-N. Kim, An adaptive Shot Change Detection algorithm using an average of absolute difference histogram within extension sliding window, ISCE, 2009.
[30] A. Prest, C. Leistner, J. Civera, C. Schmid, V. Ferrari, Learning object class detectors from weakly annotated video, CVPR, 2012.
[31] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012, http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
[32] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, CVPR, 2006.
[33] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, Selective search for object recognition, Int. J. Comput. Vis.
[34] S. Belongie, J. Malik, J. Puzicha, Matching with shape contexts, IEEE Trans. PAMI 24 (4) (2002) 509–522.
[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, NIPS, 2012.
[36] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR, 2014.
[37] ImageNet Large Scale Visual Recognition Challenge (ILSVRC), 2012, http://www.image-net.org/challenges/LSVRC/2012/index.
[38] C.D. Manning, P. Raghavan, H. Schtze, Introduction to Information Retrieval, Cambridge University Press 2008.
[39] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, DeepVision workshop at CVPR, 2014.
[40] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, arXiv preprint arXiv:1310.1531.
[41] C. Bishop, Pattern Recognition and Machine Learning, Springer 2006.
[42] A. Criminisi, J. Shotton, E. Konukoglu, Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning, Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114.