From Tiger to Panda: Animal Head Detection

Weiwei Zhang, Jian Sun, and Xiaoou Tang, *Fellow, IEEE*

***Abstract*—Robust object detection has many important applications in real-world online photo processing. For example, both Google image search and MSN live image search have integrated human face detector to retrieve face or portrait photos. Inspired by the success of such face filtering approach, in this paper, we focus on another popular online photo category—animal, which is one of the top five categories in the MSN live image search query log. As a first attempt, we focus on the problem of animal head detection of a set of relatively large land animals that are popular on the internet, such as cat, tiger, panda, fox, and cheetah. First, we proposed a new set of gradient oriented feature, Haar of Oriented Gradients (HOOG), to effectively capture the shape and texture features on animal head. Then, we proposed two detection algorithms, namely Bruteforce detection and Deformable detection, to effectively exploit the shape feature and texture feature simultaneously. Experimental results on 14 379 well labeled animals images validate the superiority of the proposed approach. Additionally, we apply the animal head detector to improve the image search result through text based online photo search result filtering.**

***Index Terms*—Feature, fusion, object detection.**

I. Introduction

AUTOMATIC detection of all generic objects in a general scene is a long term goal in image understanding and remains to be an extremely challenging problem due to large intra-class variation, varying pose, illumination change, partial occlusion, and cluttered background. However, researchers have recently made significant progresses on a particularly interesting subset of object detection problems, face [21], [26], [28] and human detection [3], achieving near 90% detection rate on the frontal face in real-time [26] using a boosting based approach. Meanwhile, with the recent advance on robust object detection, some major image search engines start to use high level image features to filter text based image search results [1]. For example, Google and MSN image search engines already integrated human face detection as a high level filter. However, designing high level filter for objects other than human face is still a challenging problem.

Inspired by the success of face detection and the real world online photo search challenge, we are interested in investigating whether the success of face detection can be extended to a broader set of object detection applications for online photo processing. Obviously, it is difficult to use the face detection approach on generic object detection such as tree, mountain, building, and sky detection, since they do not have a relatively fixed intra-class structure like human faces. To go one step at a time, we need to limit the objects to the ones that share somewhat similar properties as human face. If we can succeed on such objects, we can then consider to go further.

According to MSN image search statistics, the top five image search categories on the internet are Adult, Celebrity, News, Travel, and Animal. Human face filter is clearly aimed at celebrity search filtering. For the other categories, there are already existing researches on adult image detection [8], [12], news image categorization [11], scene image classification [25]. Unfortunately, since news and travel are not limited to a set of well defined scenes or subjects, it is difficult to develop a practical filter. On the other hand, the "animal" category seems to be less challenging with clearly defined objects in the image. If we can detect some animals in the image, the detector can then be used as an animal filter. In addition to online image filtering, animal detection can also be useful for photo tagging in online photo sharing and off-line photo album management [2]. Therefore, in this paper, we focus on this popular image category—animal.

Due to the large diversity of animal types, it is not possible to develop detectors for all animals at once. As a first attempt, we focus on relatively large land animals that are popular on the internet, such as cat, tiger, panda, fox, and cheetah. We observe that most of these animals have distinctive ears and frontal eyes. We select ten representative ones to study in this paper which includes: cat, tiger, panda, puma, leopard, wolf, fox, cheetah, raccoon, and red panda. Sample images of these animals are shown in Fig. 1. Moreover, it is not easy to detect the entire animal body because of large variation. In this paper, we focus on detecting animal head of the selected animals. Given the great difficulty of the topic, we are not trying to cover a large number of animals. Instead, as a first attempt, we hope this work will inspire more future researches on much larger number of animal types.

A natural approach to start with is to look into human face detection algorithm [26] since human face and animal head do share some similar structures. Unfortunately, directly applying the existing face detection approaches to detect the animal heads has apparent difficulties. First, the animal faces have larger appearance variations compared with the human face, as shown in Fig. 1. The textures on the animal faces are more complicated than those on the human face. Second, the animal heads have a globally similar, but locally variant shape or silhouette. How to effectively utilize both texture and shape information to train a robust animal head detector is a challenging new issue.

To deal with the above difficulties, we propose a joint learning approach to jointly capture the shape and texture features on animal head. Our basic idea is to decompose the animal head

Manuscript received April 20, 2010; revised September 21, 2010; accepted November 24, 2010. Date of publication December 13, 2010; date of current version May 18, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kenneth K. M. Lam.

W. Zhang and X. Tang are with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong, China (e-mail: wwzhang2002@gmail.com).

J. Sun is with the Visual Computing Group, Microsoft Research Asia, Beijing 100080, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2099126

Fig. 1. .Head images of popular animals on the internet: tiger, panda, leopard, puma, wolf, cat, fox, cheetah, raccoon, red panda.

into shape feature and texture feature according to animal ears and animal eyes in the first step, and then jointly capture those shape and texture features in the second step. The novelty of our approach is the discovery that we need to separate the shape and texture features first for feature extraction then combine them through joint detector for detection.

First, to effectively capture both shape and texture features, we proposed a new set of oriented gradient feature: Haar of Oriented Gradients (HOOG) to handle the shape and texture variation on animal head. After that, we proposed two joint detection algorithms: 1) Bruteforce detection, combine the shape and texture features in a straightforward approach. 2) Deformable detection, combine the shape and texture features with misalignment punish cost. Later experiment clearly validate the effectiveness of the new proposed HOOG feature and the two joint detection algorithms. In addition, we extend our Deformable detection algorithm to multiple animal categories, i.e., train a single binary classifier for the selected 10 animal categories. Again, our Deformable detection shows much better performance than either individual face detector or individual head detector. Finally, we demonstrate the applications of the animal head detector for online image search in terms of search result filtering.

This paper is organized as follows. We review the related works in Section II. The new proposed oriented gradient feature set HOOG is introduced in Section III. Two joint detection algorithms are introduced in IV. Section V shows experimental results. Finally, we conclude the proposed approaches and discuss future works in Section VI.

II. Related Works

Since a comprehensive review of the related works on object detection is beyond the scope of the paper, we only review the most related works here.

Low level features play a crucial role in object detection. Those low level features can be grouped into two main categories: image features and gradient features. The image features are directly extracted from the image, such as intensity values [21], image patch [13], PCA coefficients [18], and wavelet coefficients [19], [23], [26]. Henry *et al.* [21] trained a neural network for human face detection using the image intensities in a 20×20 sub-window. Haar wavelet features have become very popular since Viola and Jones [26] presented their real-time face detection system. The image features are suitable for small window and usually require a good photometric normalization. Contrarily, the gradient features are more robust to illumination changes. The gradient features are extracted from the edge map [9], [7] or oriented gradients, which mainly include SIFT [15], EOH [14], HOG [3], covariance matrix [24], shapelet [22], HOOG [29], and edgelet [27]. Tuzel *et al.* [24] demonstrated very good results on human detection using the covariance matrix of pixel's 1st and 2nd derivatives and pixel position as features. We will give a detailed comparison of our proposed features with HOG and EOH features in Section III.A.

To detect all possible instances of an object in the image, two different searching strategies have been developed. The sliding window detection [21], [19], [26], [3], [24], [22], [28] sequentially scans all possible sub-windows in the image and makes a binary classification on each sub-window. Viola and Jones [26] presented the first highly accurate real-time frontal face detector. A cascade classifier is trained by AdaBoost algorithm on a set of Haar wavelet features. Dalal and Triggs [3] described an excellent human detection system through training a SVM classifier using histogram of gradient (HOG) features. On the contrary, the parts based detection [10], [20], [16], [13], [7] detects multiple parts of the object and assembles the parts according to geometric constrains. For example, the human can be modeled

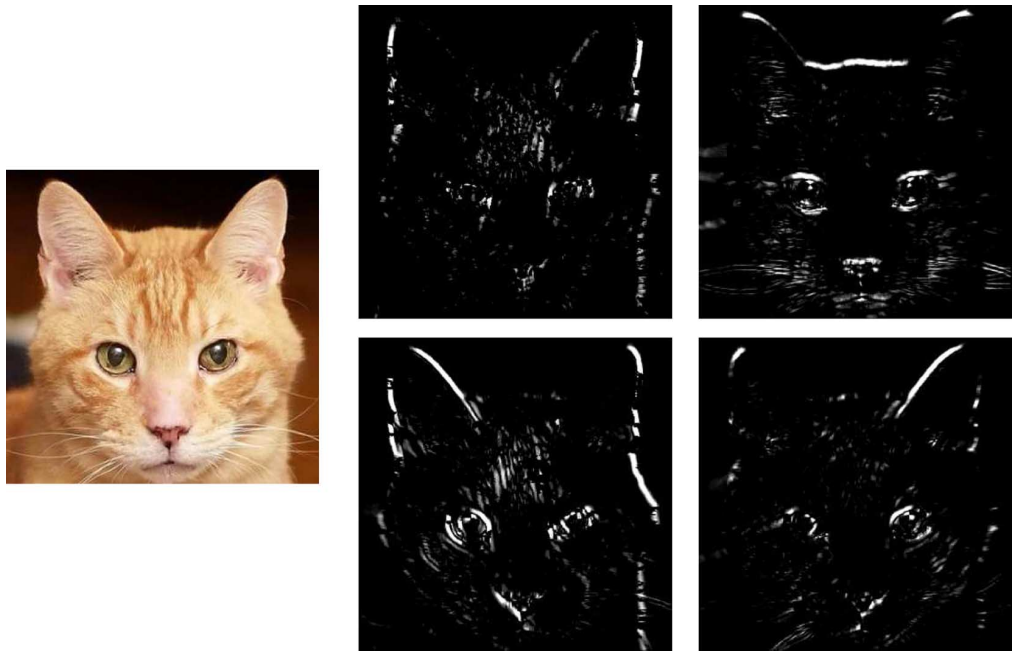


Fig. 2. Oriented gradients channels in four directions.

as assemblies of parts [16], [17] and the face can be detected using component detection [10].

In this paper, we adopt the sliding window detection approach due to its excellent real time performance. Meanwhile, we propose a set of low level features and two joint detection algorithms to effectively capture both texture and shape information on animal head.

III. Haar of Oriented Gradient

To effectively capture both shape and texture features on animal head, we propose a set of new features based on oriented gradients.

A. Oriented Gradients Features

Given an image I, the image gradient $\overrightarrow{g}(x) = \{g_h, g_v\}$ for the pixel x is computed as

$$g_h(x) = G_h \otimes I(x), \quad g_v(x) = G_v \otimes I(x) \tag{1}$$

where G_h and G_v are horizontal and vertical filters, and $\otimes$ is convolution operator. A bank of oriented gradients $\{g_o^k\}_{k=1}^K$ are constructed by quantifying the gradient $\overrightarrow{g}(x)$ on a number of K orientation bins

$$g_o^k(x) = \begin{cases} |\overrightarrow{g}(x)|, & \theta(x) \in \mathrm{B}_k \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\theta(x)$ is the orientation of the gradient $\overrightarrow{g}(x)$ and B_k is gradient orientation histogram bin k. We call the image g_o^k *oriented gradients channel*. Fig. 2 shows the oriented gradients on a cat head image. In this example, we quantify the orientation into four directions. We also denote the sum of oriented gradients of a given rectangular region R as

$$S^k(R) = \sum_{x \in R} g_o^k(x). \tag{3}$$

It can be very efficiently computed in a constant time using integral image technique [26].

Since the gradient information at an individual pixel is limited and sensitive to noise, most of previous works aggregate the gradient information in a rectangular region to form more informative, mid-level features. Here, we review two most successful features: HOG and EOH.

HOG-cell. The basis unit in the HOG descriptor is the weighted orientation histogram of a "cell" which is a small spatial region, e.g., 8×8 pixels. It can be represented as

$$\mathrm{HOG\text{–}cell}(R) = [S^1(R), \ldots, S^k(R), \ldots, S^K(R)]. \tag{4}$$

The overlapped cells (e.g., 4×4) are grouped and normalized to form a larger spatial region called "block." The concatenated histograms form the HOG descriptor.

In Dalal and Triggs's human detection system [3], a linear SVM is used to classify a 64×128 detection window consisting of multiple overlapped 16×16 blocks. To achieve near real-time performance, Zhu *et al.* [30] used HOGs of variable-size blocks in the boosting framework.

EOH. Levi and Weiss [14] proposed three kinds of features on the oriented gradients

$$\mathrm{EOH}_1(R, k1, k2) = (S^{k_1}(R) + \epsilon)/(S^{k_2}(R) + \epsilon)$$

$$\mathrm{EOH}_2(R, k) = (S^k(R) + \epsilon) \Big/ (\sum_j (S^j(R) + \epsilon))$$

$$\mathrm{EOH}_3(R, \bar{R}, k) = (S^k(R) - S^k(\bar{R}))/\mathrm{sizeof}(R)$$

where $\bar{R}$ is the symmetric region of R with respect to the vertical center of the detection window, and ϵ is a small value for smoothing. The first two features capture whether one direction is dominative or not, and the last feature is used to find symmetry or the absence of symmetry. Note that using EOH features only

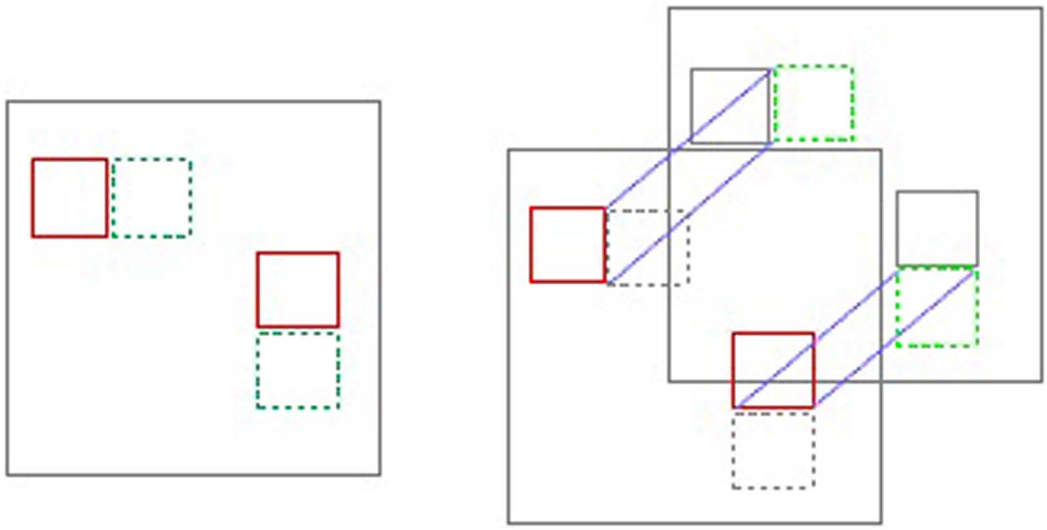


Fig. 3. HOOG. Left: in-channel features. Right: orthogonal features.

may be insufficient. In [14], good results are achieved by combining EOH features with Haar features on image intensity.

B. Our Features—Haar of Oriented Gradients

In face detection, the Haar features demonstrated their great ability to discover local patterns—intensity difference between two subregions. But it is difficult to find discriminative local patterns on the animal head which has more complex and subtle fine scale textures. On the contrary, the above oriented gradients features mainly consider the marginal statistics of gradients in a single region. It effectively captures fine scale texture orientation distribution by pixel level edge detection operator. However, it fails to capture local spatial patterns like the Haar feature. The relative gradient strength between neighboring regions is not captured either.

To capture both the fine scale texture and the local patterns, we develop a set of new features combining the advantage of both Haar and gradient features. Taking a close look at Fig. 2, we notice many local patterns in each oriented gradients channel which is sparser and clearer than the original image. We may consider that the gradient filter separates different orientation textures and pattern edges into several channels thus greatly simplified the pattern structure in each channel. Therefore, it is possible to extract Haar features from each channel to capture the local patterns. For example, in the horizontal gradient map in Fig. 2, we see that the vertical textures between the two eyes are effectively filtered out so we can easily capture the two eye pattern using Haar features. Of course, in addition to capturing local patterns within a channel, we can also capture more local patterns across two different channels using Haar like operation. In this paper, we propose two kinds of features as follows:

In-channel features:

$$\mathrm{HOOG}_1(R_1, R_2, k) = \frac{S^k(R_1) - S^k(R_2)}{S^k(R_1) + S^k(R_2)}. \tag{5}$$

These features measure the relative gradient strength between two regions R_1 and R_2 in the same orientation channel. The denominator plays a normalization role since we do not normalize $S^k(R)$.

Orthogonal-channel features:

$$\mathrm{HOOG}_2(R1, R2, k, k^*) = \frac{S^k(R_1) - S^{k^*}(R_2)}{S^k(R_1) + S^{k^*}(R_2)} \tag{6}$$

where k^* is the orthogonal orientation with respect to k, i.e., $k^* = k + K/2$. These features are similar to the in-channel features but operate on two orthogonal channels. In theory, we can define these features on any two orientations. But we decide to only compute the orthogonal-channel features based on two considerations: 1) orthogonal channels usually contain most complementary information. The information in two channels with similar orientations is mostly redundant; 2) we want to keep the size of feature pool small. The AdaBoost is a sequential, "greedy" algorithm for the feature selection. If the feature pool contains too many uninformative features, the overall performance may be hurt. In practice, all features have to be loaded into the main memory for efficient training. We must be careful about the size of the features.

Considering all combinations of R_1 and R_2 will be intractable. Based on the success of Haar features, we use Haar patterns for R_1 and R_2, as shown in Fig. 3. We call the features defined in (5) and (6), Haar of oriented gradients (HOOG).

IV. Joint Detection

It is known that the animal head has similar facial structure with the human face. Meanwhile, the animal head has a globally similar, but locally variant shape or silhouette. How to effectively make use of both texture and shape features to further improve the detection performance is a challenging new issue. In this paper, we proposed two joint detection approaches to address this issue.

A. Shape and Texture on Animal Head

It is known that the accuracy of a detector can be dramatically improved by first transforming the object into a canonical

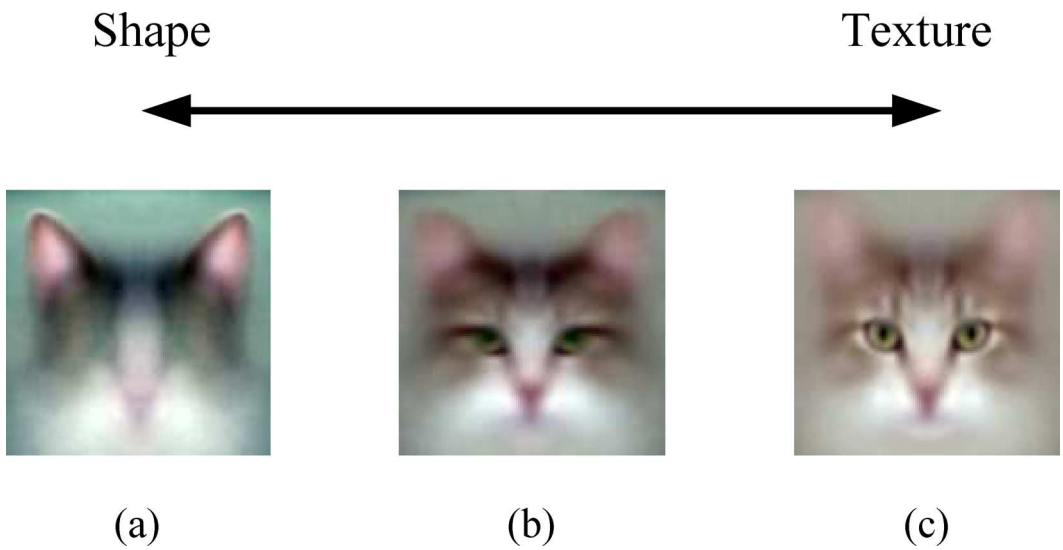


Fig. 4. Mean cat head images on all training data. (a) Aligned by ears. More shape information is kept. (b) Aligned by both eyes and ears using an optimal rotation + scale transformation. (c) Aligned by eyes. More texture information is kept.

pose to reduce the variability. For example, in human face detection all training samples are normalized by a rotation & scale transformation. The face is detected by scanning all sub-windows with different orientations and scales. Unfortunately, unlike the human face, the animal head cannot be well normalized by a rotation & scale transformation due to the large intra-class variation.

In Fig. 4, we show three mean cat head images over 5000 training images by three normalization methods. In Fig. 4(a), we rotate and scale the cat head so that both ears appear on a horizontal line and the distance between two ears is 36 pixels. As we can see, the shape or silhouette of the ears is visually distinct but the textures in the face region are blurred. In a similar way, we compute the mean image aligned by eyes, as shown in Fig. 4(c). The textures in the face region are visible but the shape of head is blurred. In Fig. 4(b), we take a compromised approach to compute an optimal rotation + scale transformation for both ears and eyes over the training data, in a least square sense. As expected, both ears and eyes are somewhat blurred.

Intuitively, using the optimal rotation+scale transformation may produce the best result because the image normalized by this method contains two kinds of information. However, the detector trained in this way does not show superior performance in our experiments. Both shape and texture information are lost to a certain degree. The discriminative power of shape features or texture features is hurt by this kind of compromised normalization. Apparently, using single normalization method cannot fully utilize both shape and texture information on the animal head. Meanwhile, extending the traditional detection approach with two normalization methods jointly is nontrivial. To this end, we proposed two joint shape and texture detection approaches. The first approach is to train two detectors, one for shape and the other for texture, a fusion classifier is trained based on the output of those two detectors, we named this approach as Bruteforce Detection. To further considering the misalignment cost between the two detectors, the second approach is proposed to consider both misalignment cost and the shape/texture detector's output, we named this approach as Deformable Detection. We give the details of those two joint detection approaches in the following subsection. Please note that we denote the detector trained with ears tips aligned images as "shape detector," to emphasize the silhouette of animal head. On the contrary, we denote the detector trained with eyes centers aligned images as "texture detector" to emphasize the texture on the animal face.

B. Bruteforce Detection

First of all, we proposed the Bruteforce Detection approach to jointly capture the shape and texture features. There are two phases in this algorithm, training and detection.

1) Training: In the *training phase*, we train two individual detectors and a fusion classifier:

1) Train a shape detector, using the aligned training images by mainly keeping the shape information, as shown in Fig. 4(a); train a texture detector, using the aligned training image by mainly preserving the texture information, as shown in Fig. 4(c). Thus, each detector can capture most discriminative shape or texture features respectively.
2) Train a joint shape and texture fusion classifier to fuse the output of the shape and texture detectors.

To train the fusion classifier, animal head images in the validation set are used as the positive samples. The key is the construction of the negative samples which consist of all incorrectly detected samples by either the shape detector or the texture detector in the non-animal images. The learned fusion classifier is able to effectively reject many false alarms by using both shape and texture information. We use support vector machine (SVM) as our fusion classifier and HOG descriptors as the representations of the features f_s and f_t.

2) Detection: In the *detection phase*, we first run the shape and texture detectors independently. Then, we apply the joint shape and texture fusion classifier to make the final decision. Specifically, we denote $\{c_s, c_t\}$ as output scores or confidences of the two detectors, and $\{f_s, f_t\}$ as extracted features in two detected sub-windows. The fusion classifier is trained on the concatenated features $\{c_s, c_t, f_s, f_t\}$.

Using two detectors, there are three kinds of detection results: both detectors report positive at roughly the same location, rotation, and scale; only the shape detector reports positive; and only the texture detector reports positive. For the first case, we directly construct the features $\{c_s, c_t, f_s, f_t\}$ for the joint fusion classifier. In the second case, we do not have $\{c_t, f_t\}$. To handle this problem, we scan the surrounding locations to pick a sub-window with the highest scores by the texture detector. Specifically, we denote the sub-window reported by the detector as $[x, y, w, h, s, \theta]$, where (x, y) is window's center, w, h are width and height, and s, θ are scale and rotation level. We search sub-windows for the texture/shape detector in the range $[x \pm w/4] \times [y \pm h/4] \times [s \pm 1] \times [\theta \pm 1]$. Note that we use real value score of the texture detector and do not make 0–1 decision. The score and features of the picked sub-window are used for the features $\{c_t, f_t\}$. For the last case, we compute $\{c_s, f_s\}$ in a similar way.

C. Deformable Detection

One problem of the Bruteforce Detection is that they do not consider the spatial misalignment(deformation) cost between the two detectors, it is desirable to design a detection approach to considering both the spatial misalignment cost and appearance likelihood. Inspired by the remarkable work of [5], [6] on human detection, we introduce a distance transform and dynamic programming approach to handle the misalignment cost, and we named this approach as Deformable Detection. There are two steps within the Deformable Detection, training and detection. In training step, we train two detectors for shape and texture respectively as in the Bruteforce detection approach. The detection procedure is described as below.

1) Detection: It is desirable to deform one detector around the other detector to find the best match between the two detectors and punish the misalignment between them. Without loss generality, we fix the texture detector t and deform the shape detector s around the texture detector t in this paper. let $\{c_s, c_t\}$ be the response of the detector s and t respectively. Specifically, we denote $\{c_s(x, y)\}$ as the response of detector s at a given position(x, y), according to [5], [6], we can compute the response transform for the detector s as

$$c_{s,d}(x, y) = \arg\max_{dx,dy}\{c_s(x + dx, y + dy) - \phi(dx, dy)\} \tag{7}$$

where $\phi(dx, dy)$ is a punish term for the misalignment (dx, dy) between the two detectors, in our experiment, we set $\phi(dx, dy)$ as

$$\phi(dx, dy) = \beta * (dx * dx + dy * dy). \tag{8}$$

β is an normalization term to balance the detection score and the spatial misalignment, we set β as 0.05 in this paper. Then the final response of the two detectors are simply accumulation of $c_t(x, y)$ and $c_{s,d}(x, y)$.

$$R(x, y) = c_t(x, y) + c_{s,d}(x, y). \tag{9}$$

V. Experiment

A. Performance Evaluation

1) Data Set and Evaluation Methodology: Our evaluation data set includes two parts, the first part is our own data, which contains 10 animal categories and 13 700 images. The animal images are collected from Flickr.com and image search engine. Most of the animal images have near frontal view. Each animal head is manually labeled with nine points, two for eyes, one for mouth, and six for ears, as shown in Fig. 5. We randomly chose 50% of the images for training, 20% for validation and 30% for testing. Fig. 1 shows some sample images from our database. The second part is from the PASCAL 2007 cat data, which includes 679 cat images. We follow the PASCAL 2007 original separations of training, validation, and testing set on the cat data. Table I summarizes the animal categories and statistics. The data can be downloaded at http://mmlab.ie.cuhk.edu.hk.

We use the evaluation methodology similar to the PASCAL challenge for object detection. Suppose the ground truth rectangle and the detected rectangle are r_g and r_d, and the area of those rectangles are A_g and A_d. We say we correctly detect a animal head only when the overlap of r_g and r_d is larger than 50%

$$D(r_g, r_d) = \begin{cases} 1, & \text{if } \frac{(A_g \cap A_d)}{(A_g \cup A_d)} > 50\% \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $D(r_g, r_d)$ is a function used to calculate detection rate and false alarm rate.

2) Implementation Details: We discuss several implementation details in this subsection.

HOOG Features. We use six unsigned orientations to compute the HOOG features. We find the improvement is marginal when finer orientations are used. The horizontal and vertical filters are $[-1, 0, 1]$ and $[-1, 0, 1]^T$. No thresholding is applied on the computed gradients. For both shape and texture detector, we construct feature pools with 200 000 features by quantifying the size and location of the Haar templates. Meanwhile, we use the gray image to extract the HOOG feature.

Joint Detection. We investigate the performance of the two proposed joint detection algorithms. 1) For the Bruteforce detection, we trained two shape and texture detectors using HOOG features and boosting classifier, and the final classifier is trained based on the output of those two detectors as in Section IV.B. 2) For the Deformable detection, we train two detectors as in Bruteforce Detection and we fix the texture detector and deform the shape detector. For both approaches, the shift range of the head window is $[x \pm w/2] \times [y \pm h/2] \times [s \pm 1]$, where x, y is the center of the face window and w, h is the width and height of the head window. We use 20% of animal images as validation set to tune the parameters for the final classifier for both approaches. During detection, we use four pixels as the shift step size of the head window w_s. For deformable, we choose β with a straightforward approach based on our validation set. First, we uniformly choose 20 value from 0 to 0.2, then we compute the detection ROC performance. We found 0.05 is the best value on our validation set, therefore, we use this value in all experiment. The larger the β, the more punishment on the two detector's misalignment and vice versa.

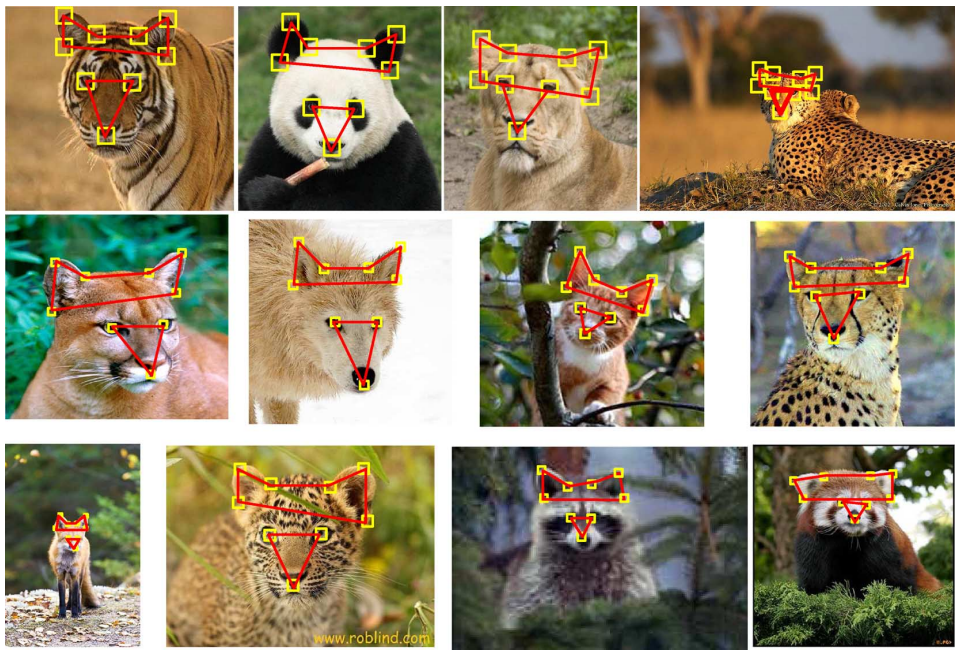


Fig. 5. Cat head image is manually labeled by nine points.

TABLE I
SUMMARY OF THE ANIMAL DATABASE

Animal Name	Number of Image
Cat	10,000
PASCAL Cat	679
Fox	800
Cheetah	800
Tiger	300
puma	300
Leopard	300
Panda	300
RedPanda	300
Racoon	300
Wolf	300

Training samples. We choose the best cropping size for each of the two joint detection algorithms. We align all animal head image with respect to the ears to train the shape detector. We rotate and scale the image so that two tips of the ears appear on a horizontal line and the distance between the two tips is 36 pixel. Then, we extract a 48×48 pixel region, centered 20 pixels below the two tips. For the texture detector, a 32×32 pixel region is extracted. The distance between the two eyes is 20 pixels. The region is centered six pixel below the two eyes.

3) Comparison of Features: First of all, we compare the proposed HOOG features with Haar, Haar + EOH, and HOG features on both the shape detector and the texture detector using our own cat data. The reason is that: 1) We have enough cat data to train a robust boosting detector. 2) Cat has bigger shape and texture variations than the other animals, which could test the feature's performance more extensively. For the Haar features, we use all four kinds of Haar templates. For the EOH features,

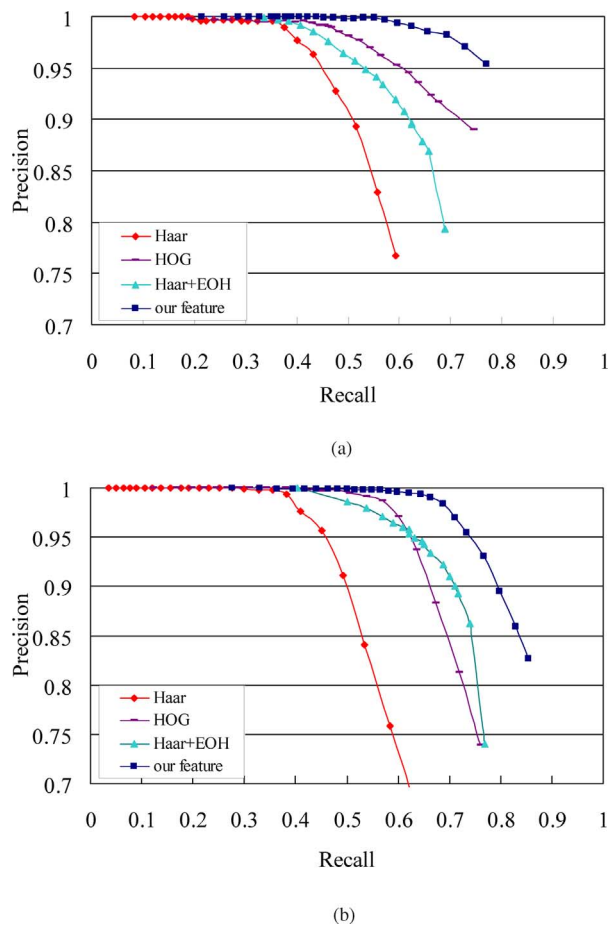


Fig. 6. Comparison of Haar, Haar+EOH, HOG, and our features on the cat data. (a) Shape detector. (b) texture detector.

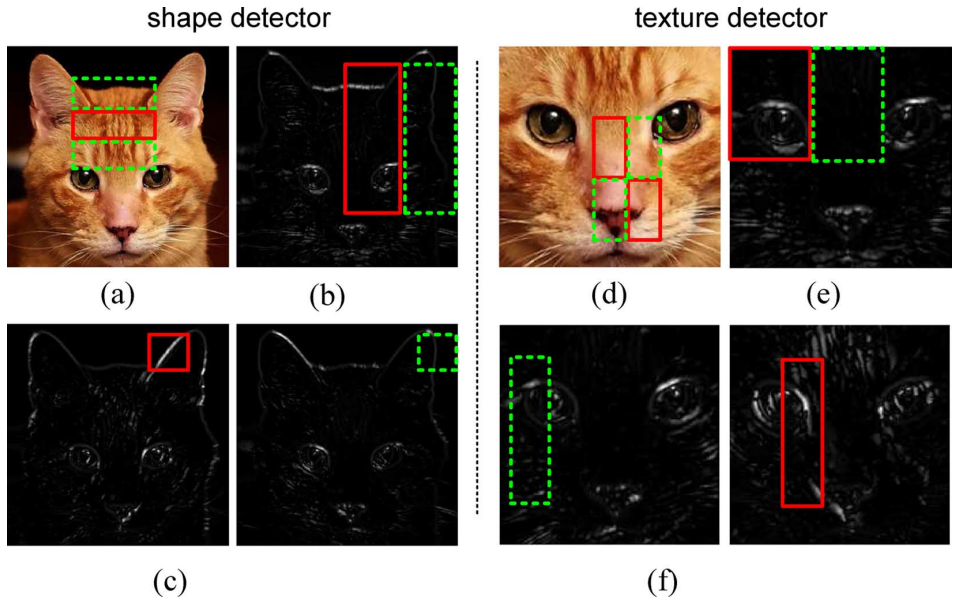


Fig. 7. Best features learned by the AdaBoost. Left (shape detector): (a) Best Haar feature on image intensity. (b) Best in-channel feature. (c) Best orthogonal feature on orientations 60° and 150°. Right (texture detector). (d) Best Haar feature on image intensity. (e) Best in-channel feature. (f) Best orthogonal-channel feature on orientations 30° and 120°.

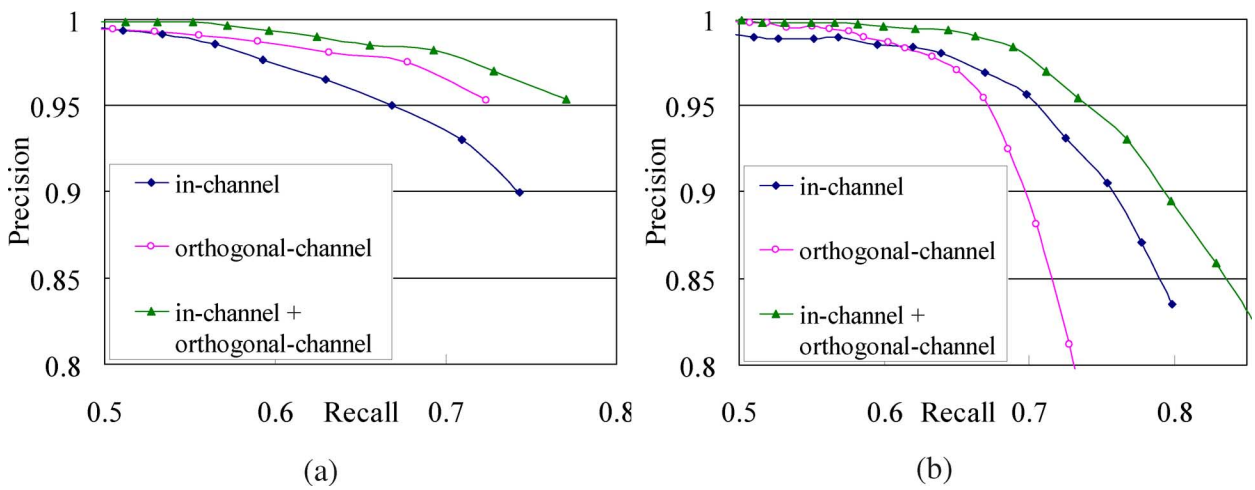


Fig. 8. Importance of in-channel features and orthogonal-channel features on the cat data. (a) Shape detector. (b) Texture detector.

we use default parameters suggested in [14]. For the HOG features, we use 4×4 cell size which produces the best results in our experiments.

Fig. 6 shows the performances of the four kinds of features on the cat data. The Haar feature on intensity gives the poorest performance because of large shape and texture variations of the cat head. This in a way shows that the traditional human face detection algorithm is not suitable for the cat head detection. With the help of the oriented gradient features, $\mathrm{Haar} + \mathrm{EOH}$ improves the performance. As expected, the HOG features perform better on the shape detector than on the texture detector. Using both in-channel and orthogonal-channel information, the detectors based on our features produce the best results.

In Fig. 7, we show the best in-channel features in (b) and (e), and the best orthogonal-channel features in (c) and (f), learned by the two detectors. We also show the best Haar features on image intensity in Fig. 7(a) and (d). In both detectors, the best in-channel features capture the strength differences between a region with strongest horizontal gradients and its neighboring region. The best orthogonal-channel features capture the strength differences in two orthogonal orientations.

In the next experiment we investigate the role of in-channel features and orthogonal-channel features. Fig. 8 shows the performances of the detector using in-channel features only, orthogonal-channel features only, and both kinds of features. Not surprisingly, both features are important and complementary.

4) Joint Detection: In this sub-section, we evaluate the performance of the proposed two joint detection algorithms on three data sets: cat, fox, and cheetah. Fig. 9 shows seven precision-recall curves on cat data set: a boost shape detector, a SVM shape detector, a boost texture detector, a SVM texture detector, a head detector using optimal transformation, a Bruteforce detector, and a Deformable detector. The optimal transformation detector is trained using training samples aligned by an optimal rotation+scale transformation. From Fig. 9, several important observations can be made: 1) The performance of joint detector is substantially boosted! For a given precision 0.95, the recall is improved from 0.74/0.75/0.78(boost shape/boost

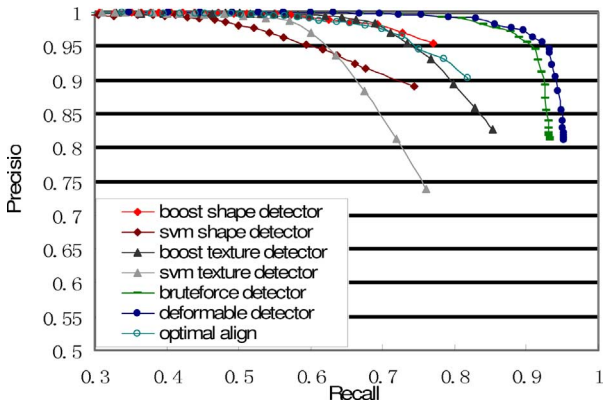


Fig. 9. Joint detection on the cat data.

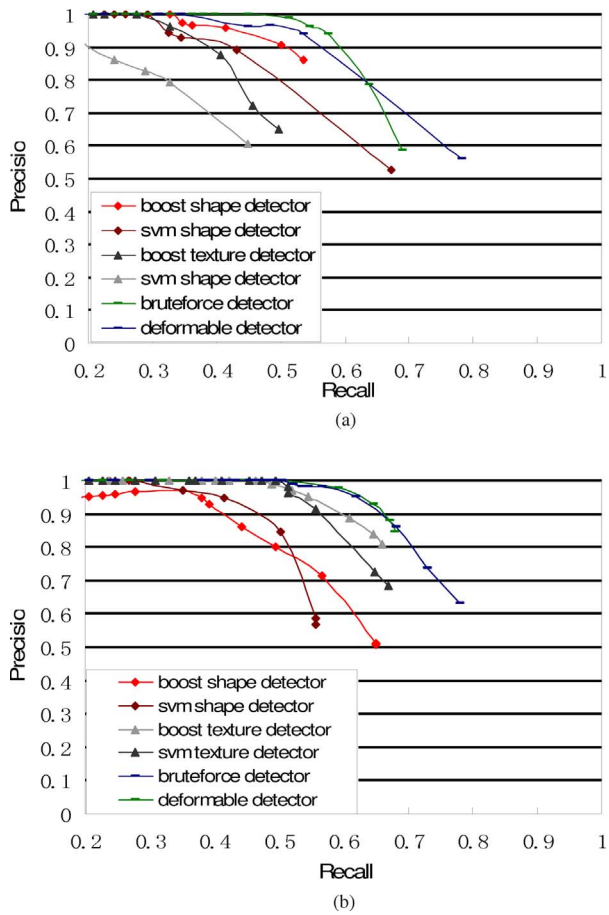


Fig. 10. Joint detection on the fox and Cheetah data. (a) Fox. (b) Cheetah.

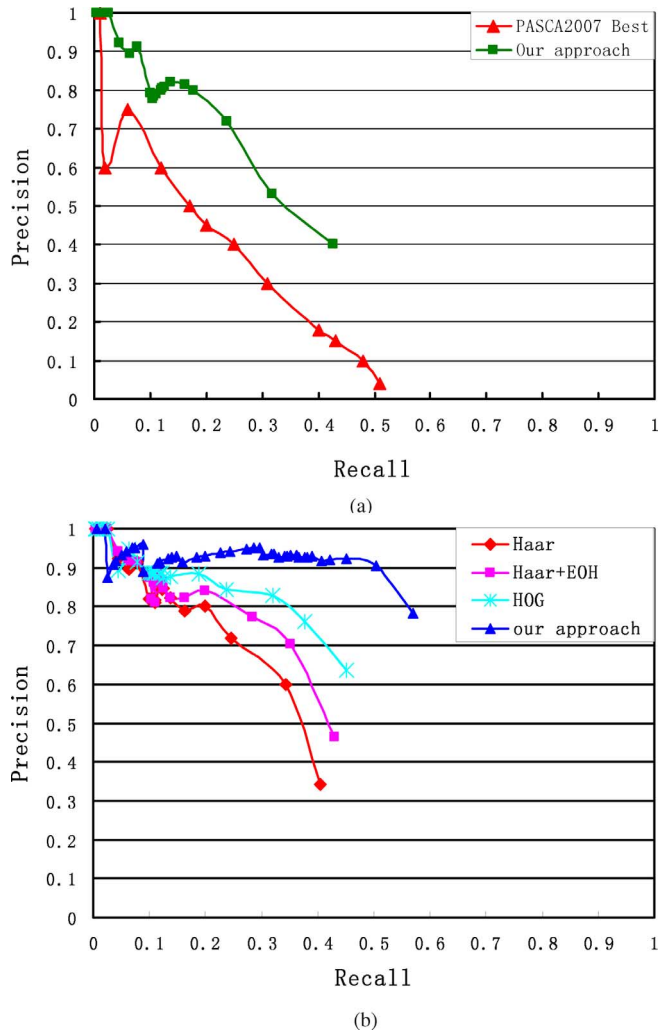


Fig. 11. Experiments on the PASCAL 2007 cat data. (a) Our approach and the best reported method on Competition 3 (specified training data). (b) Four detectors on Competition 4 (arbitrary training data).

texture/boost optimal align) to 0.92/0.925 through Bruteforce detector/Deformable detector. Or the precision is improved from 0.92/0.94/0.955(boost shape/boost texture/boost optimal align) to 0.995/0.998, for a fixed recall 0.76 by Bruteforce detector/Deformable detector. In image retrieval and search applications, it is a very nice property since high precision is preferred; 2) The head detector using optimal transformation does not show superior performance. The discriminative abilities of both shape and texture features are decreased by the optimal transformation; 3) The maximal recall value of the Bruteforce detector/Deformable detector (0.92/0.935) is larger than the maximal recall values of the three individual boost detectors(0.77/0.82/0.85). This shows the complementary abilities of the two detectors—one detector can find many animal heads which is difficult to the other detector; 4) Note that the curve of fusion detector is very steep in the high recall region, which means the fusion detector can effectively reject many false alarms while maintaining a very high recall. 5) The Deformable detector has slightly better detection performance than the Bruteforce detector. In some case, it has noticeable improvement. For example, the recall is improved from 0.92 to 0.935 for a given precision 0.85.

Fig. 10 shows the precision-recall curve on fox and cheetah data set respectively. From those two figures we have the following observations: 1) Both joint detectors have better performance than individual shape detector and head detector on the two data sets. For example, in Fig. 10, the best recall is improved from 0.5/0.6 to 0.6/0.66 at a fixed precision 0.9. 2) The Bruteforce detector and the Deformable detector have comparable performance on the two data sets. The recall is 0.58/0.60 on the fox data and 0.66/0.66 on the cheetah data at fixed precision 0.9. 3) Shape and texture have different performance on different animal categories. For example, shape detector has better performance than texture detector on the fox data, and texture detector has better performance than shape detector on the cheetah data. This is easy to understand, since the shape on fox head is more

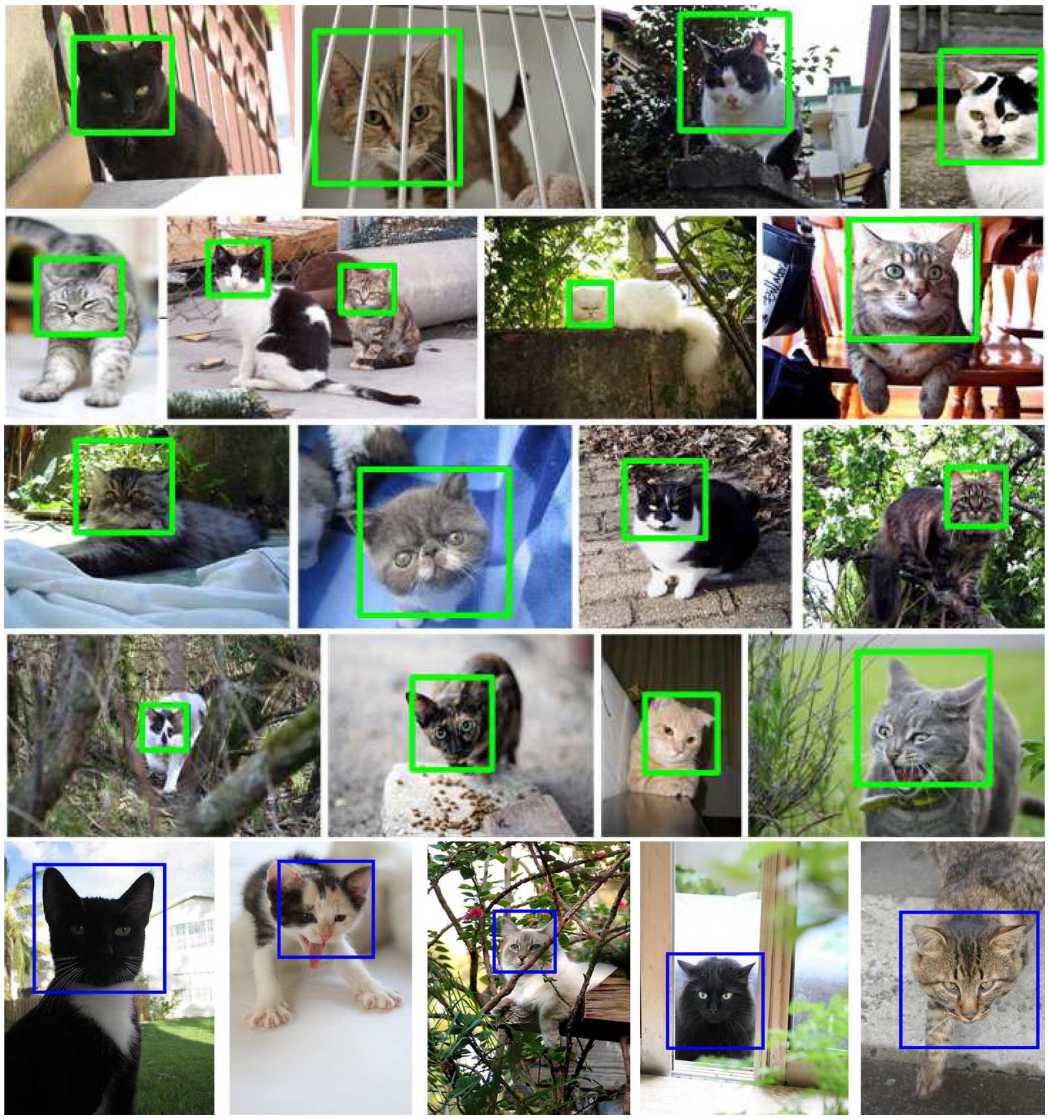


Fig. 12. Detection results on the cat data set. The bottom row shows some detected cats in the PASCAL 2007 data.

discriminative than texture on fox head because: a) The ear tips on the fox head are very sharp and big. b) The pose of the fox head tends to affect the fox face more than the fox head shape because of large depth on the fox face. On the contrary, the cheetah texture detector has better performance than cheetah shape detector because: a) Cheetah ear tips are round and small. b) The pose of the cheetah head has smaller effect on the cheetah face because of the smaller depth on the cheetah face. However, through combining the two different features, we get much better performance than either individual detector. These observations validate our separation of shape and texture again.

Another interesting observation from Figs. 9 and 10 is that the Bruteforce detection algorithm and Deformable detection algorithm have overall comparable performance. However, the Deformable detection is much fast to compute and more easy to train. In practice, we suggest to use the Deformable detection as the first choice.

5) Experiment on the Pascal 2007 Cat Data: We also evaluate the proposed Joint Detection algorithm I on the PASCAL 2007 cat data [4]. There are two kinds of competitions for the detection task: 1) Competition 3—using both training and testing data from PASCAL 2007; 2) Competition 4—using arbitrary training data. Fig. 11(a) shows the precision-recall curves of our approach and the best reported method [4] on Competition 3. We compute the average precision (AP) as in [4] for a convenient

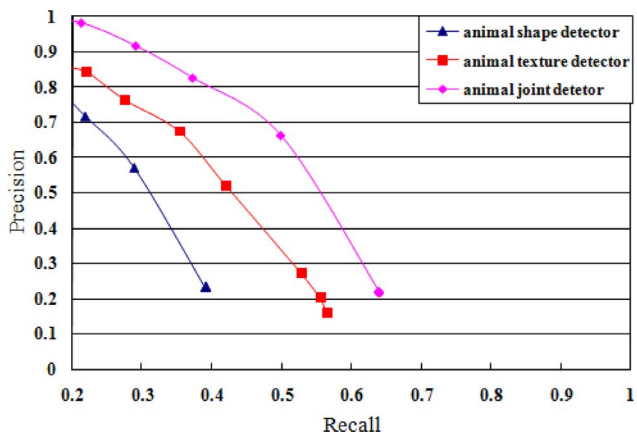


Fig. 13. Comparison of the joint detector with basic detectors.

Fig. 14. Failure cases.

comparison. The APs of our approach and the best reported method is 0.364 and 0.24, respectively. Fig. 11(b) shows the precision-recall curves on Competition 4. Since there is no reported result on Competition 4, we compare our approach with the detectors using Haar, EOH, and HoG respectively. All detectors are trained on the same training data. The APs of the four detectors (ours, HOG, Haar+EOH, Harr) are 0.632, 0.427, 0.401, and 0.357. Using larger training data, the detection performance is significantly improved. For example, the precision is improved from 0.40 to 0.91 for a fixed recall 0.4. Note that the PASCAL 2007 cat data treat the whole cat body as the object and only small fraction of the data contain near frontal cat faces. However, our approach still achieves good results ($\mathrm{AP} = 0.632$) on this very challenging data (the best reported method's $\mathrm{AP} = 0.24$).

Moreover, the runtime is very important for online image processing. Actually, it is one of advantages of our system based on integral image and cascade structure. For example, to compute a 320×240 image without extra rotation processing, we only need 0.25 s on average with our unoptimized c++ code on a Intel (Due Core) 2.66 GHz PC, which is acceptable for online image processing. To process rotation, we rotate the images for eight times, from $-\mathrm{PI}/2$ to PI/2, and merge all the detection results to get the final detection results.

B. Extension to Multiple Animal Categories

It is interesting that our joint detection algorithm can be easily extended to detect multiple animal categories, i.e., training a single binary classifier for multiple animal categories. This is quite useful when running multiple animal detectors is not affordable. To this end, we construct an animal database which include 2000 cat images and all images of other animals. As before, we randomly choose 50% of the images as training set and 20% as validation set, and the rest 30% as testing set. We crop the animal face and animal head as in Section IV.C. We train a binary Deformable detector and denote this detector as animal joint detector (please note that we can train a Bruteforce detector in a similar way, but omit it here to save space). We also trained two individual binary detectors for animal face and animal head using HOOG feature and boosting classifiers. We denote the individual detectors as animal shape detector and animal texture detector. Fig. 13 report the precision-recall curves of the three detectors. As we can see from Fig. 13, our joint detector has much better performance compared with the two individual detectors. Specifically, we improve the recall rate from less than 25% to around 40% over the individual detector at a fixed precision 0.8.

C. Animal Photo Search Filtering

As discussed at the beginning, our animal head detector can be used as a high level filter to help on filtering the text based animal photo search results. We download the first 200 images of the selected 10 animals from Google image search engine and run our animal head detector on those images. Table II shows the animal filtering result. In Table II, the first column is the total image number of the ten animal categories used in this experiments. The second and third column are the number of human labeled true animal image and noise animal image respectively, and the forth column and fifth column are the number of animal image and noise image after filtering by our animal head detector. As we can see from the table, most of the noise images are filtered out.

TABLE II
ANIMAL PHOTO SEARCH FILTERING RESULT

Animal name	Total animal image	True animal image	Noise animal image	True image after filtering	Noise image after filtering
Cat	200	178	22	122	4
Tiger	200	153	47	131	8
Lion	200	132	68	68	22
Leopard	200	92	108	45	15
Panda	200	77	123	62	17
RedPanda	200	145	55	135	6
Racoon	200	174	26	148	11
Wolf	200	166	34	119	5
Fox	200	72	128	46	21
Cheetah	200	68	132	49	15

D. Failure Case

In this subsection, we discuss the failure case of the proposed algorithms. There are several cases which may cause the proposed algorithm to fail. 1) large pose variation. 2) low contrast. 3) extreme facial expression. 4) partial occlusion. Those factors will distort both shape and cause failure case. We will address these issues in our future work. Fig. 14 shows some failure images.

VI. CONCLUSION

In this paper, we have presented an animal head detection system. We achieved much improved results by decomposing texture and shape features firstly and improve the detection results through joint detection based on the shape and texture features. Then the texture and shape detectors are also improved by a set of new oriented gradient features. Experiments on 14 379 well labeled animal image database validate the effectiveness of our joint learning approach. Finally, we demonstrate the applications of the animal head detection for online image search. In the future, we plan to extend the proposed animal detection in two directions. First, we plan to cover more animal types and further improve the detection performance, e.g., explore more information such as texture on animal body, design more discriminative features. Second, we hope to extend the animal head detection to more applications, such as animal image categorization based on the detection results.

REFERENCES

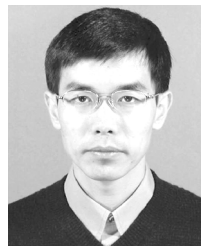
[1] J. Cui, F. Wen, and X. Tang, "Real time google and live image search re-ranking," *ACM Multimedia*, pp. 729–732, 2008.
[2] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang, "Easyalbum: An interactive photo annotation system based on face clustering and re-ranking," in *Proc. SIGCHI*, 2007, pp. 367–376.
[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, vol. 1, pp. 886–893.
[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[5] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," Cornell Univ., Ithaca, NY, Cornell Univ. Tech Rep., 2004, vol. 1, pp. 1963–2004.
[6] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 153–190, 2005.
[7] P. F. Felzenszwalb, "Learning models for object recognition.," in *Proc. CVPR*, 2001, vol. 1, pp. 1056–1062.
[8] M. Fleck, D. A. Forsyth, and C. Bregler, "Finding naked people.," in *Proc. Eur. Conf. Comput. Vis.*, 1996, pp. 593–602.
[9] D. M. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *Proc. CVPR*, 1999, p. 87.
[10] B. Heisele, T. Serre, M. Pontil, and T. Poggio, "Component-based face detection," in *Proc. CVPR*, 2001, vol. 1, pp. 657–662.
[11] J. Hu and A. Bagga, "Functionality based web image categorization," in *Proc. Int. World Wide Web Conf.*, 2003, p. 118.
[12] M. J. Jones and J. M. Rehg, "Statistical color models with applications to skin detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1999, pp. 81–96.
[13] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. CVPR*, 2005, vol. 1, pp. 878–885.
[14] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: The importance of good features," in *Proc. CVPR*, 2004, vol. 2, pp. 53–60.
[15] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, 1999, vol. 2, pp. 1150–1157.
[16] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Proc. ECCV*, 2004, vol. 1, pp. 69–82.
[17] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, Apr. 2001.
[18] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1863–1868, Nov. 2006.
[19] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, 2000.
[20] R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people," in *Proc. ECCV*, 2004, vol. 4, pp. 700–714.
[21] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.
[22] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Proc. CVPR*, 2007, pp. 1–8.
[23] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," in *Proc. CVPR*, 2000, vol. 1, pp. 746–751.
[24] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on Riemannian manifolds," in *Proc. CVPR*, 2007, pp. 1–8.
[25] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders, "Robust scene categorization by learning image statistics in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2006, pp. 105–108.
[26] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[27] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors," in *Proc. ICCV*, 2005, vol. 1, pp. 90–97.
[28] R. Xiao, H. Zhu, H. Sun, and X. Tang, "Dynamic cascades for face detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[29] W. Zhang, J. Sun, and X. Tang, "Cat head detection—How to effectively exploit shape and texture features," in *Proc. ECCV*, 2008, pp. 802–806.
[30] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. CVPR*, 2006, vol. 2, pp. 1491–1498.

Weiwei Zhang received the B.S. degree from the North China University of Technology, Beijing, in 1997, the M.S. degree from the Beihang University, Beijing, in 2000, and the Ph.D. degree from Institute of Automation, the Chinese Academy of Sciences, Beijing, in 2003.

He is currently a Research Associate in the Department of Information Engineering, the Chinese University of Hong Kong. He worked as Associate Researcher at Microsoft Research Asia from 2003 to 2010. His research interests include object detection, visual tracking, object segmentation, and recognition.

Jian Sun received the B.S., M.S., and Ph.D. degrees from Xian Jiaotong University, Shaanxi, China, in 1997, 2000, and 2003, respectively.

He joined Microsoft Research Asia in 2003. His research is in the fields of computer vision and computer graphics, with particular interests in Interactive Compute Vision (user interface + vision), and Internet Compute Vision (large image collection + vision). He is also interested in stereo matching, computational photography, and face recognition.



Xiaoou Tang (S'93–M'96–SM'02–F'09) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996.

He is a Professor in the Department of Information Engineering and Associate Dean (Research) of the Faculty of Engineering of the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at the Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing.

He received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI) and *International Journal of Computer Vision* (IJCV).