Object Recognition by Discriminative Combinations of Line Segments and Ellipses

Alex Yong-Sang Chia[1,2] Susanto Rahardja[1] Deepu Rajan[2] Maylor Karhang Leung[2]
[1]Institute for Infocomm Research, Singapore [2]School of Computer Engineering, Nanyang Technological University, Singapore
alex_chia@scholars.a-star.edu.sg rsusanto@i2r.a-star.edu.sg {asdrajan,asmkleung}@ntu.edu.sg

Abstract

We present a contour based approach to object recognition in real-world images. Contours are represented by generic shape primitives of line segments and ellipses. These primitives offer substantial flexibility to model complex shapes. We pair connected primitives as shape tokens, and learn category specific combinations of shape tokens. We do not restrict combinations to have a fixed number of tokens, but allow each combination to flexibly evolve to best represent a category. This, coupled with the generic nature of primitives, enables a variety of discriminative shape structures of a category to be learned. We compare our approach with related methods and state-of-the-art contour based approaches on two demanding datasets across 17 categories. Highly competitive results are obtained. In particular, on the challenging Weizmann horse dataset, we attain improved image classification and object detection results over the best contour based results published so far.

1. Introduction

This paper addresses two goals of recognition: image classification and category level object detection. The task of image classification is to determine if an object category is present in an image, while object detection localizes all instances of that category from an image. Fueled by emerging consensus that shapes are often the more discriminative features shared between instances of a category as compared to image patches [12, 14], contour based recognition techniques have recently attracted strong interest in the research community [10, 15, 16, 19].

In this work, we also use contours for recognition. However, unlike other methods, our novelty lies in the representation of contours by very simple and generic shape primitives of line segments and ellipses, coupled with an efficient approach to learn category specific primitive combinations. Each combination is a two-layer abstraction of primitives: connected pairs of primitives (termed *shape tokens*) at the first layer, and a learned number of tokens at the second layer. We do not impose combinations to have a fixed number of tokens, but allow them to automatically adapt to a category. This number influences a combination's ability to represent shapes (and structures) where simple shapes favor fewer tokens than complex ones. Consequently, discriminative combinations of varying complexity can be used to represent a category. We learn these combinations by harnessing distinguishing geometric, structural and appearance constraints of a category in a unified framework. Geometric constraints describe the spatial layout (configurations) of tokens, while structural constraints enforce possible poses/structures of an object (*e.g.* XOR relationships of tokens). Appearance constraints describe the visual aspect of tokens, which we represent by line segments and ellipses.

Line segments and ellipses are complementary in nature; the former models straight contours while the latter curved contours. Unlike edge based local descriptors [12, 14], they support abstract reasoning like parallelism and adjacency. While one can also use contour fragments [15, 19] to represent shapes, the proposed primitives offer unique advantages. Firstly, matching between primitives can be easily computed (by their geometric properties), unlike contour fragments which require comparison between individual edge pixels. More importantly, as geometric properties can be easily scale normalized, they simplify matching across scales. In contrast, contour fragments are not scale invariant and one is forced either to rescale contour fragments which introduces aliasing effects (*e.g.* edge pixels are squeezed together), or to resize image before extracting fragments which degrades image resolution.

In recent independent studies [6, 16], it was shown that the generic nature of line segments and ellipses afford them an innate ability to represent complex shapes and structures. While individually less informative, by combining a learned number of these primitives, we empower a combination to be sufficiently discriminative. In this aspect, we attempt to strike a winning tradeoff: exploit generic primitives to achieve flexibility in describing local object shape structures at the lower level, and combine these primitives to allow a combination to be sufficiently complex to capture discriminative information at the higher level.

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

1.1. Related Work

Ferrari *et al.* [10] used connected straight contour fragments as features. This is similar to our work where line segments (and ellipses) are combined. However, their method requires the powerful Berkeley boundary detector [13] to find meaningful object boundaries and to filter noisy background contours before training or testing. In contrast, our method extracts contours by the traditional Canny detector and is robust to detected noisy edge pixels. More importantly, they represent shapes by features extracted from category neutral images. We depart from this framework and instead explicitly tailor primitive combinations to a specific object category. Our choice for constructing category specific features is motivated by the substantial success in Shotton *et al.* [19] and Opelt *et al.* [15]. A difference between our work and theirs is that they represent local shapes by contour fragments while we employ generic shape primitives. Consequently, they suffer the shortcomings presented earlier in Sect. 1. More importantly, to keep the learning of discriminative features tractable, they limit each feature to contain a fixed number of fragments (single fragment in [19] and two fragments in [15]). Our approach imposes no such restriction and instead learns category specific features that have a variable number of shape primitives.

Shape primitives have been used previously for object recognition. Line segments were used in [16] to detect objects in cluttered scenes. Their method does not model curved object boundaries, which inhibits their ability to learn complex class models. Jurie and Schmid [12] detected circular arcs in edge image and described the spatial distribution of edge pixels in a thin neighborhood of the circle. As one weakness, circle represents a limited class of curved shapes. Ellipses were defined on the second moment matrix of image regions in [4, 18]. This differs from our proposed framework, in which ellipses are extracted directly from edge images and model curved image contours. Rothwell *et al.* [17] computed projective invariant values from lines and ellipses for object representation, where they focused on identifying specific planar objects, rather than recognizing object classes which we addressed here.

2. Shape Tokens

We extract line segments and ellipses from an edge image by the method in [7]. A shape token is constructed by pairing a reference primitive to its connected neighbor, where edge gaps are bridged in a similar way as [11]. This captures string-like shape structures. In our work, given two connected primitives of different types, an ellipse will always be the reference primitive. For connected primitives of the same type, we consider each primitive in turn as the reference primitive. This gives the following types of shape tokens: line-line, ellipse-line and ellipse-ellipse.

2.1. Describing shape tokens

A numerical descriptor comprising geometric attributes is used to describe the appearance of a shape token. Let θ denote the orientation of a primitive. For an ellipse whose eccentricity is more than λ_ε (fixed at 0.8), θ is assigned to be the orientation of its major axis; orientation of a circle or an ellipse whose eccentricity is less than or equal to λ_ε is fixed as π. We define $[\mathrm{v}^{\mathrm{x}}\ \mathrm{v}^{\mathrm{y}}]^T$ as the unit vector from the center of the reference primitive to the center of its neighbor, and h as the distance between their centers. The midpoint between their centers is defined to be the token centroid. We denote the length and width of a primitive as l and w respectively. For an ellipse, the length is given by its major axis. We fix the width of a line segment to be one pixel thick and define the width of an ellipse by its minor axis. Given these notations, the appearance descriptor of a token is

$$A = [\theta^r\ l^r\ w^r\ \theta^n\ l^n\ w^n\ h\ \mathrm{v}^{\mathrm{x}}\ \mathrm{v}^{\mathrm{y}}]^T\,, \tag{1}$$

where the superscripts r and n differentiate attributes of a reference primitive from its neighbor.

2.2. Matching tokens across multiple scales

As stated before, three different types of shape tokens are constructed. A token is compared only with similar typed tokens. We first present our approach to compare tokens at a single scale. For this purpose, we define a distance measure between two tokens with descriptors A_i and A_j as

$$\begin{aligned} D\left(A_i, A_j\right) = \sum_{p\in(r,n)} D_\theta\left(\theta_i^p, \theta_j^p\right) + \sum_{p\in(r,n)} D_l\left(l_i^p, l_j^p\right) \\ + \sum_{p\in(r,n)} D_l\left(w_i^p, w_j^p\right) + D_l\left(h_i, \mathrm{h}_j\right) + D_v\left(\mathrm{v}_i^{\mathrm{x}}, \mathrm{v}_i^{\mathrm{y}}, \mathrm{v}_j^{\mathrm{x}}, \mathrm{v}_j^{\mathrm{y}}\right), \end{aligned} \tag{2}$$

where

$$\begin{aligned} D_\theta\left(\theta_i, \theta_j\right) &= \frac{\min\left(\left|\theta_i - \theta_j\right|,\ \pi - \left|\theta_i - \theta_j\right|\right)}{\pi/2}, \\ D_l\left(l_i, l_j\right) &= \min\left(\left|\ln\left(l_i/l_j\right)\right|, 1\right), \\ D_v\left(\mathrm{v}_i^{\mathrm{x}}, \mathrm{v}_i^{\mathrm{y}}, \mathrm{v}_j^{\mathrm{x}}, \mathrm{v}_j^{\mathrm{y}}\right) &= \sqrt{\left(\mathrm{v}_i^{\mathrm{x}} - \mathrm{v}_j^{\mathrm{x}}\right)^2 + \left(\mathrm{v}_i^{\mathrm{y}} - \mathrm{v}_j^{\mathrm{y}}\right)^2}, \end{aligned}$$

and $D_\theta \in [0,1]$ measures the difference in orientations, $D_l \in [0,1]$ the difference in lengths, and $D_v \in [0,2]$ the difference in relative primitive positions. Thus, eq. (2) combines the difference in geometric attributes of the tokens into a single useful dissimilarity measure, where the distance range for comparing line-line, ellipse-line and ellipse-ellipse tokens are $[0,7]$, $[0,8]$ and $[0,9]$ respectively.

It is easy to extend the above matching to multiple scales. Specifically, the descriptor of a token can be normalized against an object scale b_s as $\overline{A} = f(A,\ \frac{1}{b_s})$, where

$$f\left(A,\ \frac{1}{b_s}\right) = \left[\theta^r\ \frac{l^r}{b_s}\ \frac{w^r}{b_s}\ \theta^n\ \frac{l^n}{b_s}\ \frac{w^n}{b_s}\ \frac{h}{b_s}\ \mathrm{v}^{\mathrm{x}}\ \mathrm{v}^{\mathrm{y}}\right]^T \tag{3}$$

Matching at scale s between a scale normalized $\overline{A}_i$ and an unscaled A_j is then readily computed as $D(f(\overline{A}_i, s), A_j)$.

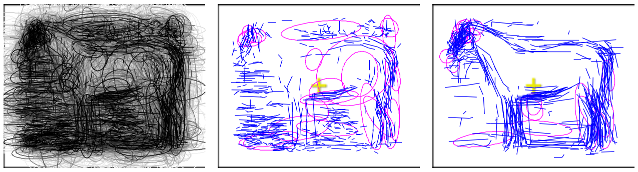
3. Codebook of shape tokens

We build a codebook of representative shape tokens of the target category, before selecting codewords into a discriminative primitive combination. Here, explicit effort is made to learn codewords which not only have coherent appearances and positions with a large number of tokens, but also cover substantial spatial extent of the object model. To learn this codebook, we extract an initial set of tokens from within the bounding box bb of every training object. We normalize the appearance descriptor of each token by the object scale b_s (diagonal length of bb). Each token is also parameterized by a vector $\overline{\mathrm{x}}$ that is directed from the object centroid (center of bb) to the token centroid and normalized by b_s.

3.1. Finding candidate codewords

Candidate codewords are found by two-step clustering for the appearances and positions [19] of the initial set of tokens. In the first step, we adapt the computationally efficient bisecting k-medoid method to find clusters whose members have similar appearances. Specifically, we apply 2-medoid clustering to similar typed tokens where distances between tokens are computed using its scale normalized descriptors with eq. (2). For each cluster, we evaluate its intra-cluster appearance dissimilarity value by the average appearance distance between the medoid and its members. A cluster is repartitioned by 2-medoid clustering if its dissimilarity value is more than th (fixed at 20% of the maximum range of $D(\cdot)$ in eq. (2)). This method improves on k-medoid clustering used by [19] in that it avoids specifying the number of clusters as input, which is unknown apriori and vary between datasets.

In the second step, we apply the robust mean-shift algorithm to each cluster on vectors $\overline{\mathrm{x}}$ of its members. Tokens in each mean-shift sub-cluster therefore have similar appearances and are located at similar positions relative to the object centroid. We identify the medoid (based on the appearance distance measure) in each mean-shift sub-cluster as a candidate codeword φ and associate it with an appearance distance allowance τ. This allowance indicates the range of appearance the candidate represents and is computed as the mean appearance distance between the candidate and its sub-cluster members plus one standard deviation. Each candidate is also parameterized with a scale normalized circular window specifying where it is expected to be found relative to an object centroid. We compute the relative center of this window, $\overline{c}$, as the mean of vectors $\overline{\mathrm{x}}$ of the sub-cluster members, and its radius, $\overline{r}$, as the mean Euclidean distance between $\overline{c}$ to $\overline{\mathrm{x}}$ of each member plus one standard deviation.



(a) (b) (c)

Figure 1. Candidate codewords from (a) all mean-shift clusters, (b) 350 most populated clusters, and (c) 350 clusters selected by our method for Weizmann horse dataset. Line segments are shown in blue and ellipses in magenta. Yellow + denotes object centroid.

3.2. Selecting candidates into codebook

Fig. 1(a) shows all candidate codewords that are obtained from an initial set of 330K tokens for the Weizmann horse dataset [3], in which a candidate is shown darker if it is from a more populated mean-shift sub-cluster. A simple heuristic to select candidates based on cluster size can inadvertently pick non-salient candidates as shown in Fig. 1(b). Instead, we score each candidate as a product of i) its intra-cluster appearance similarity value, ii) the number of unique training bounding boxes its members are extracted from, and iii) its value of $1/\overline{r}$. The first two terms seek candidates that have distinctive appearance and are flexible enough to accommodate intra-class variations, while the last term seeks candidates that estimate a stable and precise location for its members.

To ensure selected candidates represent a large spatial extent of the object model, as opposed to a localized portion, we select high scoring candidates by a radial ranking scheme. We emanate a pair of rays from the object centroid to delineate a sector. Candidates within each sector are identified and we collect the top scoring t candidates of each sector. We illustrate this in Fig. 2, where the top row shows three sectors with centroid positions of the t candidates in each sector represented as small green $\times$, and the bottom row visualizes the t candidates. Observe that spatial layout of each sector has selected salient shape structures corresponding to head, rear-end and leg of a horse. For all experiments, we divide the object into 30 non overlapping sectors and fix t to be 20. Finally, instead of using all collected candidates as codewords, we retain the 350 highest scoring candidates as codewords. This provides robustness against collecting "poor" candidates when a sector has substantial overlap with the background.

Fig. 1(c) shows the 350 codewords selected by the described procedure. Compared to Fig. 1(b), there is substantial reduction of candidates from the background. In addition, a large spatial extent of a horse comprising salient structures *e.g.* head, neck, legs and belly are accounted for, with typically several codewords representing different poses of the same object part. This provides tolerance towards intra-class variations, small pose changes and par-

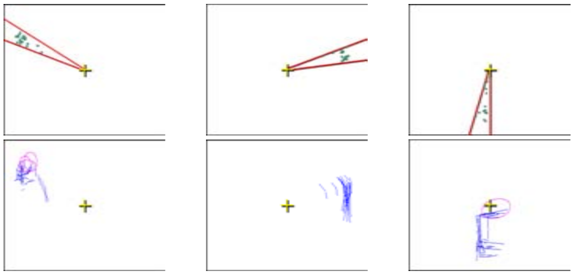


Figure 2. Radial method for selecting candidate codewords.

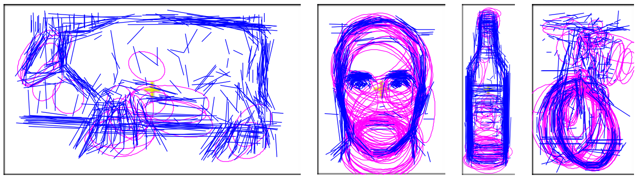


Figure 3. Codewords selected for four Graz-17 object categories. From left to right: Cow-side, face, bottle and bike-front.

tial occlusion. Fig. 3 shows codewords selected for four categories of the Graz-17 dataset [15]. As observed, most salient shape structures, *e.g.* bike wheels, are represented.

4. Discriminative primitive combinations

This section presents our method, which capitalizes on distinguishing appearance, geometric and structural constraints of a category, for learning discriminative primitive combinations. Each combination has a variable number of *x* codewords, and is termed in the following as a *x codeword combination* or *xCC*.

4.1. Matching *xCC* in an image

We first describe the matching of a *xCC* in an image, before explaining how discriminative *xCC* are learned. A *xCC* is matched at scale $\hat{\mathrm{s}}$ in an image if i) appearance distance at scale $\hat{\mathrm{s}}$ between each codeword in that combination and a token of the image is within the appearance distance allowance τ of the codeword (appearance constraint), and ii) centroid predictions by all codewords in the combination concur (geometric constraint). A codeword which satisfies the appearance constraint with a token that is located at position x in the image will predict an object centroid by a circular window with center $\hat{\mathrm{x}} = \mathrm{x} - \hat{\mathrm{s}} \times \bar{c}$, and radius $\hat{\mathrm{s}} \times \bar{r}$. These centroid predictions concur if there is a common region among their windows.

Fig. 4 exemplifies the appearance and geometric constraints for matching a *xCC*, where codewords, matched tokens and centroid predictions are shown color coded. A combination comprising four codewords is depicted in Fig. 4(a). Using a same scale $\hat{\mathrm{s}}$, tokens from a positive and a negative image which satisfy the appearance constraints with the codewords are shown in Fig. 4(b), where colored

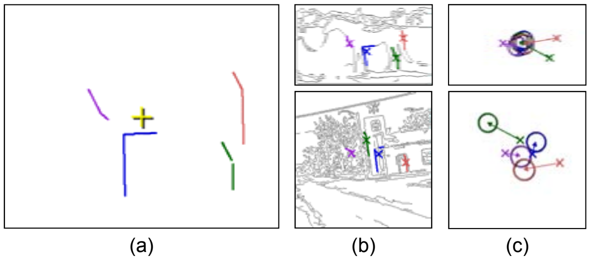


Figure 4. Matching a combination of four codewords in a positive image and a negative image. Best viewed in color.

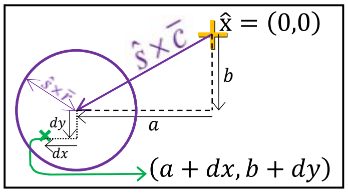


Figure 5. Illustration for finding matched *xCC*. See text for details.

'×' denote token centroids. Centroid windows predicted by these codewords are shown in Fig 4(c), and colored arrows denote scaled vectors -$\hat{\mathrm{s}} \times \bar{c}$. These windows share a common region in the positive image but not the negative image, and hence this *xCC* is matched only in the positive image.

4.2. Finding all *xCC* matched in training images

Discriminative *xCC* are learned by harnessing the set of *xCC* that are matched in training images. Here, we propose an efficient method (linear in codebook size) to find the exhaustive set of matched *xCC* for unconstrained *x* values. The following theorem states the basic idea of this method.

Theorem. *For a scale $\hat{\mathrm{s}}$ and location $\hat{\mathrm{x}}$ in image I, all codewords which satisfy the appearance matching constraint with at least one token located within its estimated window of center $\hat{\mathrm{x}} + \hat{\mathrm{s}} \times \bar{c}$ and radius $\hat{s} \times \bar{r}$ in I also fulfill the geometric matching constraint.*

Proof. *Consider Fig. 5. Let $\hat{\mathrm{x}} = (0, 0)$ be the origin in image I. For scale $\hat{\mathrm{s}}$ and location $\hat{\mathrm{x}}$ in I, let the scaled vector $\hat{\mathrm{s}} \times \bar{c}$ of a codeword be $[a\ b]^T$. Thus, this codeword estimates a window center (a, b) and radius $\hat{s} \times \bar{r}$. Suppose a token in I satisfies the appearance matching constraint with this codeword, and is located at position $(a + dx, b + dy)$ in this window. Then, it follows that*

$$\sqrt{dx^2 + dy^2} \;\leq\; \hat{\mathrm{s}} \times \bar{r}. \tag{4}$$

From Sect. 4.1, this codeword also predicts an object centroid in I by a window of center $(a+dx, b+dy) - (a, b) = (dx, dy)$ and radius $\hat{\mathrm{s}} \times \bar{r}$. From eq. (4), the point $\hat{\mathrm{x}} = (0, 0)$ is within this window. Thus, at a same scale $\hat{\mathrm{s}}$ and location $\hat{\mathrm{x}}$, all codewords which satisfy the appearance matching constraints with a token located in its estimated window will predict centroids that contain a common point $\hat{\mathrm{x}}$, and also satisfy the geometric matching constraint.

For a scale $\hat{\mathrm{s}}$ and location $\hat{\mathrm{x}}$ in *I*, we use a numerical value $\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ to indicate if a codeword γ_i finds matching tokens that satisfy the appearance matching constraint and that are also located within its estimated window. Let A' be the appearance descriptor of a token t' located at position x' in *I*. The token which best matches codeword γ_i is defined as one whose appearance is most similar to γ_i (and within appearance distance allowance τ_i of the codeword), and whose position is closest to its expected position (and within the estimated window of the codeword),

$$t^* = \arg\min_{t'} \big(d_{app}(\gamma_i, t') + d_{geo}(\gamma_i, t')\big), \tag{5}$$

where

$$d_{app}(\gamma_i, t') = \begin{cases} \frac{D(\hat{\mathrm{s}}A_i, A')}{\tau_i} & \text{if } D(\hat{\mathrm{s}}A_i, A') \leq \tau_i \\ \infty & \text{otherwise} \end{cases},$$

$$d_{geo}(\gamma_i, t') = \begin{cases} \frac{\|\hat{\mathrm{x}}+\hat{\mathrm{s}}\times\overline{c_i}-\mathrm{x}'\|_2}{\hat{\mathrm{s}}\times\overline{r_i}} & \text{if } \|\hat{\mathrm{x}}+\hat{\mathrm{s}}\times\overline{c_i}-\mathrm{x}'\|_2 \leq \hat{\mathrm{s}}\times\overline{r_i} \\ \infty & \text{otherwise} \end{cases}$$

and $\|\cdot\|$ is the L_2 norm. $\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ for codeword γ_i at scale $\hat{\mathrm{s}}$ and location $\hat{\mathrm{x}}$ in an image is then computed as

$$\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}}) = d_{app}(\gamma_i, t^*) + d_{geo}(\gamma_i, t^*). \tag{6}$$

It is easily verified that a codeword which finds a token that satisfies its appearance matching constraint and that is in its estimated window has $\Re(\cdot)$ value in the range $[0,\ 2]$, where a lower value indicates better matching. In contrast, a non-matching codeword has infinity $\Re(\cdot)$ value. Then, from the above theorem, at $\hat{\mathrm{s}}$ and $\hat{\mathrm{x}}$ in image *I*, every combination of codewords whose $\Re(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ values are less than infinity are matched in *I*. By iterating through all scales and locations across all training images, the exhaustive set of matched *xCC* can thus be found. The computational complexity of this search is $O(l\sigma N)$, where N is the codebook size, and l and σ are respectively the number of locations and scales being searched. For greater efficiency, we sample locations at every 15 pixels in each direction, and use a number of scales that covers the range of object scales in training images. This reduces computation overheads, and is similar in concept to the efficient sliding window technique.

4.3. Learning an ensemble of discriminative *xCC*

We seek a *xCC* which models discriminative appearance, geometric and structural constraints of a category to reliably predict object locations. The formulation for $\Re(\cdot)$ in eq. (6) provides a mathematically convenient method to find such a *xCC*. We take as input $\Re(\cdot)$ values of all codewords at every sampled scale and location $(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ for all training images. Each $(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ represents an object hypothesis, and we pair it with a binary label to indicate if it localizes an object.

Consider first a *xCC* which comprises two codewords γ_i and γ_j. From Sect. 4.2, the matching of this *xCC* at $(\hat{\mathrm{s}}, \hat{\mathrm{x}})$ in image *I* can be mathematically represented as

$$\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}}) \leq \theta_i \text{ and } \Re_j(\hat{\mathrm{s}}, \hat{\mathrm{x}}) \leq \theta_j\ ,$$

where θ is a threshold in the range $[0,\ 2]$. Note that since this *xCC* is matched at the hypothesis, it therefore implicitly models the appearance and geometric configurations of shape tokens at this hypothesis. This representation can be further generalized to the following form,

$$p_i\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}}) \leq p_i\theta_i \text{ and } p_j\Re_j(\hat{\mathrm{s}}, \hat{\mathrm{x}}) \leq p_j\theta_j\ ,$$

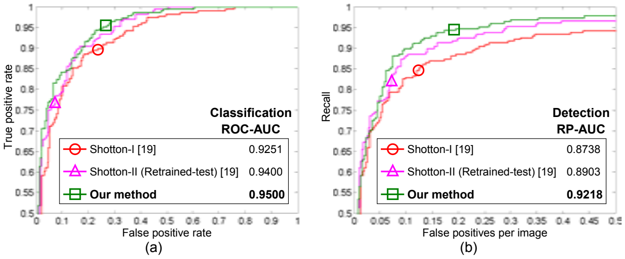
where p has value +1 or -1 to indicate the direction of the inequality, and θ is relaxed to take on any real number value. With this representation, structural configurations *e.g.* XOR relationships between codewords at this hypothesis (matching of γ_i implies γ_j is weakly matched or unmatched) can be integrated with the appearance and geometric configurations of shape tokens at the same hypothesis. This representation can be modeled by a *xCC*, with each of its codeword having a *p* and θ values. Additionally, by allowing a *xCC* to have a variable number of codewords, we can impart greater flexibility to this *xCC* to model shapes and structures of varying complexity. Our aim here is to learn such discriminative *xCC* (*i.e.* codewords in the *xCC*, and its p and θ values) which can reliably predict the presence/absence of an object instance at an object hypothesis $(\hat{\mathrm{s}}, \hat{\mathrm{x}})$.

We exploit decision tree to learn such a *xCC*. It is easily shown that a path from a root node to any leaf node in a binary tree encounters between 1 to *k* non-leaf nodes, where each non-leaf node is a predicate of the form $p_i\Re_i(\hat{\mathrm{s}}, \hat{\mathrm{x}}) \leq p_i\theta_i$, and *k* is the number of splits in the tree. Each path (*i.e.* the number of non-leaf nodes and their predicates, and the predicted label at the leaf node) automatically adapts to an object category, and is discriminatively learned to predict the presence/absence of an object at an object hypothesis. Hence, by learning a binary tree with *k* splits, discriminative *xCC* can be found by simple path transversal from a root node to each leaf node, where each *xCC* has a variable number of between 1 to *k* codewords, and a hypothesis is assigned a predicted label by exactly one *xCC* of the tree.

We learn an ensemble of discriminative *xCC* by AdaBoost, where each boosting round outputs a CART decision tree with *k* splits. For all experiments, we use 300 boosting rounds. Rather than fixing a same *k* value for all categories, we learn a *k* value for each category by 3-fold validation over values 1 to 10 on the training hypotheses. This approach to optimize parameter values for each category was used in [19], and further customizes the ensemble for the category. After finding the optimal *k* value, we learn the ensemble of *xCC* from the entire training hypotheses.

5. Object Detection

We detect objects in a test image by a multi-scale sliding window approach. The sliding and scale steps for each object category are the same as that used during training in Sect. 4.2, and the aspect ratio of a window is equal to



(a)

(b)

Figure 6. Image classification and object detection performance on the Weizmann horse dataset, with comparison to [19].

that of the average training bounding box. Each window is an object hypothesis $(\hat{s}, \hat{x})$ and we evaluate the detection confidence of each window with the boosted ensemble of *xCC*. We consider local maxima as candidate detections and apply a post-filtering step to remove candidate detections whose windows overlap with a stronger candidate. The retained detections yield the final set of detections.

6. Experimental evaluations

We evaluate our technique on two challenging datasets which cover 17 categories, and compare against the best (to our knowledge) contour based recognition results and other published results. Both object detection and image classification results are reported, and we adhere to the evaluation criteria of other methods. As closely as possible, we use the same training and testing object images as other methods for comparing performances.

An object is correctly detected if overlap of the ground truth and detected bounding boxes is above 50%, and multiple detections of a same object count as false positives. We compare detection performance by two scores of a recall-precision (RP) curve: equal error rate (RP-EER) and area under curve (RP-AUC). RP-EER reports recall at a single precision value, while RP-AUC measures detection performance across all precision levels and so give a more representative score for comparison purposes. For image classification, we use the strongest detection in each test image. An image is correctly classified if it contains the object category. We compare classification performance by two scores of a ROC-curve: the ROC-EER and the ROC-AUC scores.

6.1. Evaluating on Weizmann horse dataset [3]

This challenging dataset contains near-side views of horses under varying scales and illuminations, strongly varying poses and substantially cluttered backgrounds. We use the first 100 horse images for training and the remaining 228 horse images for testing, and pair the object images in the training and testing sets with an equal number of Caltech-256 background images. Following [19], we downsample all images to a maximum 320 pixels width or height.

We report quantitative results in Fig. 6. The best (as far as we know) results obtained by contour based approach on this dataset are also included in Fig. 6, and we denote these methods as *Shotton-I* and *Shotton-II* [19]. While *Shotton-I* used the same training object images as us, *Shotton-II* augmented a set of object hypotheses from the test images to the training data, and retrained their system on the augmented data. In this aspect, *Shotton-II* learned from a larger and more diverse set of training images. Nevertheless, we achieve better classification and detection performances as evident from the higher ROC-AUC and RP-AUC scores (given in legends). The plot of recall against false positives per image (fppi) in Fig. 6(b) further reveals the better detection performance of our method in which 93% of test objects are correctly detected at a low fppi of 0.15 (around 1 false positive every 7 images). In contrast, *Shotton-I* and *Shotton-II* managed a recall of 87% and 90% respectively.

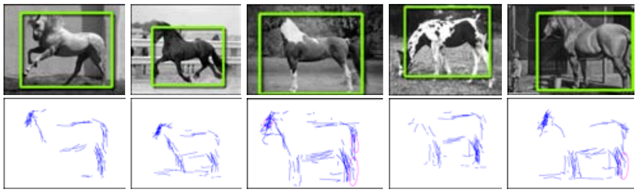


Figure 7. Example detections on Weizmann horse test images. The top row shows the bounding box of the detection, and the bottom row the codewords that contributed positively to the detection. Codewords are visualized at their detected positions and scales.

Our detection result is also superior to that achieved very recently by the contour based method of Bai *et al.* [1], which obtained 0.8032 RP-AUC. While their method used a smaller training subset, this is due to their necessity of obtaining figure-ground segmented training objects. In contrast, since our method needs only bounding boxes, we fully exploit all training images, which likely account for some part of the improvement. We show example detections by our method in Fig. 7, where codewords which contribute positively to detections are back-projected at their detected positions and scales and shown at the bottom of test images.

6.2. Evaluating on Graz-17 dataset [15]

This demanding dataset features 17 diverse object categories. Object instances for each category are presented at varying scales, and considerable intra-class variations are evident. Images contain one or more object instances, and many images have substantial background clutter. We use the same experimental setup as [15,19]. Except for the number of scales (which we optimized against the training data), parameter values remain unchanged as the previous Weizmann horse experiment. We show example detections in Fig. 8. Table 1 reports quantitative results, with comparison to recent state-of-the-art contour based recognition methods of Opel *et al.* [15] and Shotton *et al.* [19]. Classification and detection results are averaged over 17 categories and shown

Table 1. Performance comparison with state-of-the-art contour based methods [15, 19] on Graz-17 dataset.

Object category	Number of images		Image classification results				Object detection results			
	Training	Testing	ROC-AUC Our method	ROC-EER Our method	ROC-EER [15]	ROC-AUC [19]	RP-AUC Our method	RP-EER Our method	RP-EER [15]	RP-AUC [19]
Plane	100	400	0.9826	6.9%	2.6%	0.9953	0.9325	10.0%	7.4%	0.9310
Motorbike	100	400	0.9983	1.0%	3.2%	1.0000	0.9996	1.0%	4.4%	1.0000
Face	100	217	0.9974	2.6%	1.9%	0.9966	0.9895	2.7%	3.6%	0.9850
Car-rear	100	400	0.9883	6.9%	0.05%	0.9992	0.9797	4.0%	2.3%	0.9912
Car-2/3-rear	32	14	0.9643	14.2%	-	0.9000	0.6843	35.7%	12.5%	0.6925
Car-front	34	16	0.9688	6.2%	-	0.9727	0.8256	18.7%	10.0%	0.7233
Bike-rear	29	13	0.9468	7.6%	-	0.9172	0.6042	40.0%	25.0%	0.6398
Bike-front	19	12	0.9584	8.3%	-	0.9375	0.7421	33.3%	41.7%	0.6344
Bike-side	90	53	0.9445	9.4%	-	0.9366	0.8299	18.8%	28.0%	0.6959
Bottle	54	64	0.9773	7.8%	-	0.9802	0.9009	12.5%	9.0%	0.9468
Cow-front	34	16	0.9844	6.2%	-	0.9727	0.8335	18.7%	18.0%	0.8575
Cow-side	45	65	0.9944	6.1%	-	0.9992	0.9945	6.1%	0.0%	0.9975
Horse-front	44	22	0.9918	4.5%	-	0.9566	0.7368	27.2%	13.8%	0.7852
Horse-side	55	96	0.9756	7.2%	-	0.9816	0.9361	11.4%	8.2%	0.9680
Person	39	18	0.9352	16.6%	-	0.9321	0.5730	47.6%	47.4%	0.4271
Mug	30	20	0.9525	5.0%	-	0.9600	0.9619	5.0%	6.7%	0.9035
Cup	31	20	0.9800	10.0%	-	0.9825	0.8964	15.0%	18.8%	0.9158
Averaged across categories			**0.9730**	**7.4%**	**1.9%***	**0.9659**	**0.8483**	**18.1%**	**15.1%**	**0.8291**

*Average ROC-EER for [15] is calculated from the four categories whose classification results are reported by the authors.

Table 2. Comparison of ROC-EER classification scores on first four categories of Graz-17 dataset to other published results.

Object category	Our method	Sivic '05 [20]	Crandall '06 [8]	Fergus '07 [9]	Bar-Hillel '08 [2]	Chen '09 [5]	Zhu '09 [21]
Plane	6.9%	3.4%	9.3%	6.3%	6.7%	8.2%	8.2%
Motorbike	1.0%	15.4%	3.0%	3.3%	4.9%	5.4%	7.1%
Face	2.6%	5.3%	2.0%	8.3%	6.3%	2.0%	2.3%
Car-rear	6.9%	21.4%	6.5%	8.8%	0.6%	-	-
Averaged across categories	**4.4%**	**11.4%**	**5.2%**	**6.7%**	**4.6%**	**5.2%***	**5.9%***

* Average ROC-EER scores for [5, 21] are calculated from the three categories whose classification results are reported by the authors.

(in bold) in the last row of the table.

Although AUC scores are more representative, we compare against Opelt *et al.* [15] by the EER scores (since only EER scores are provided by the authors). For classification, we achieve an average ROC-EER (across first four categories) of 4.4% which is not as good as that of their state-of-the-art method. Nevertheless, it is still quite competitive and attains a better average ROC-EER score compared to other published results as shown in Table 2. It has to be mentioned that while we use stronger supervision than the methods in Table 2, our results are obtained with 100 positive and 100 negative training images, which is less than half of what was used by these methods during training.

For object detection, we perform better on some categories (*e.g.* motorbike and face), but considerably worse for categories like car-2/3-rear and bike-rear which have very few test images. The number of test images, m, can sharply affect a RP-EER score. Specifically, RP-EER reports detector performance at a single precision value (point at which number of false positives and false negatives are equal) and hence even one false positive or miss detection can have a large effect on RP-EER of up to $\frac{100}{m}\%$, as also pointed out in [19]. Consequently, much more significant are the detection results for categories with more test images. In particular, considering categories with more than 200 test images (*i.e.* first four categories), we attain an average RP-EER of 4.425%, exactly matching that obtained by [15] (Table 1).

We compare performance with Shotton *et al.* [19] by the more representative AUC scores (which evaluate performance across all precision values). Overall, we achieve better classification and detection results: [19] obtained an average ROC-AUC of 0.9659, which we improve upon with 0.9730, and an average RP-AUC of 0.8291, which is lower than 0.8483 attained by our method. Importantly, this improvement is achieved with a smaller training set, compared to [19] where hypotheses from the test data is augmented with the original training data for retraining (as in *Shotton-II* method of the previous Weizmann horse experiment).

7. Discussion

We have described a contour based approach that exploits very simple and generic shape primitives of line segments and ellipses for image classification and object detection. Primitive combinations which reliably predict the locations of object instances are learned by harnessing discriminative appearance, geometrical and structural constraints of a category in a unified framework. Our method

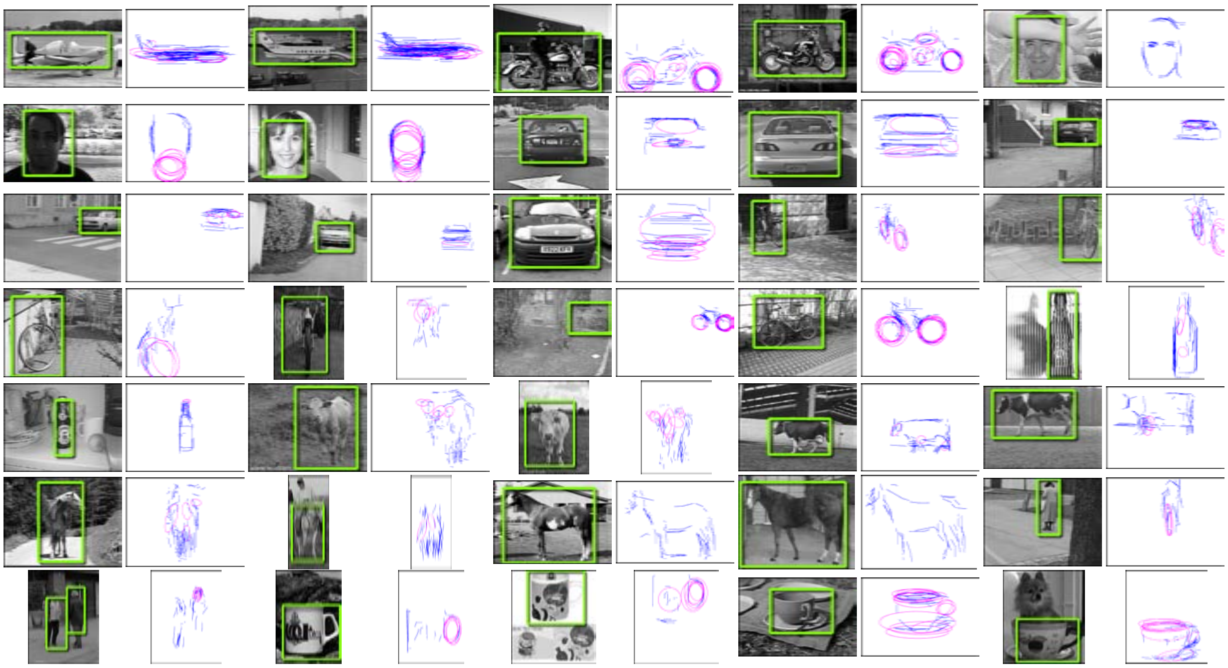


Figure 8. Example detections and codewords visualization for all categories of Graz-17 test set. Note accurate localization despite intra-class variations, poses and scale changes, illumination variations, partial occlusion, and background clutter.

does not restrict combinations to have a fixed number of primitives, but allow them to automatically adapt to an object category. This, coupled with generic nature of the primitives, empowers a combination with much flexibility to represent discriminative shape structures. We have evaluated our approach on the challenging Weizmann horse and Graz-17 datasets. Image classification and object detection results on both datasets demonstrate the effectiveness of our approach, in which for the Weizmann horse dataset, we achieved improvement over the best contour based results published so far for the dataset.

Acknowledgement. This work was supported by the Agency for Science, Technology and Research (A*STAR). M.K. Leung acknowledged funding from the Academic Research Fund RG17/08M52020088.

References

[1] X. Bai, X. Wang, L. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. *ICCV*, 2009.

[2] A. Bar-Hillel and D. Weinshall. Efficient learning of relational object class models. *IJCV*, pages 175–198, 2008.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *ECCV*, pages 639–641, 2002.

[4] L. Bretzner and T. Lindeberg. Qualitative multiscale feature hierarchies for object tracking. *J. Vis. Commun. Image Representation*, pages 115–129, 2000.

[5] Y. Chen, L. Zhu, A. Yuille, and H. Zhang. Unsupervised learning of probabilistic object models (poms) for object classification, segmentation and recognition using knowledge propagation. *TPAMI*, 2009.

[6] A. Chia, S. Rahardja, D. Rajan, and M. Leung. Structural descriptors for category level object detection. *TMM*, pages 1407–1421, 2009.

[7] A. Chia, D. Rajan, M. Leung, and S. Rahardja. A split and merge based ellipse detector. *ICIP*, pages 3212–3215, 2008.

[8] D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. *ECCV*, pages 16–29, 2006.

[9] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, pages 273–303, 2007.

[10] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *TPAMI*, pages 36–51, 2008.

[11] V. Ferrari, T. Tuytelaars, and L. Gool. Object detection by contour segment networks. *ECCV*, pages 14–28, 2006.

[12] F. Jurie and C. Schmid. Scale invariant shape features for recognition of object categories. *CVPR*, pages 90–96, 2004.

[13] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *TPAMI*, pages 530–549, 2004.

[14] K. Mikolajczyka, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. *BMVC*, pages 779–788, 2003.

[15] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, pages 16–44, 2008.

[16] X. Ren. Learning and matching line aspects for articulated objects. *CVPR*, pages 1–8, 2007.

[17] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *IJCV*, pages 57–99, 1995.

[18] A. Shokoufandeh, L. Bretzner, D. Macrini, M. Demirci, C. Jonsson, and S. Dickinson. The representation and matching of categorical shape. *CVIU*, pages 139–154, 2006.

[19] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *TPAMI*, pages 1270–1281, 2008.

[20] J. Sivic, C. Russell, A. Elfros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. *ICCV*, pages 370–377, 2005.

[21] L. Zhu, Y. Chen, and A. Yuille. Unsupervised learning of probabilistic grammar-markov models for object categories. *TPAMI*, 2009.