Sub-Category Optimization through Cluster Performance Analysis for Multi-View Multi-Pose Object Detection

Dipankar DAS†, Yoshinori KOBAYASHI†, Nonmembers, and Yoshinori KUNO†(a), Member

SUMMARY The detection of object categories with large variations in appearance is a fundamental problem in computer vision. The appearance of object categories can change due to intra-class variations, background clutter, and changes in viewpoint and illumination. For object categories with large appearance changes, some kind of sub-categorization based approach is necessary. This paper proposes a sub-category optimization approach that automatically divides an object category into an appropriate number of sub-categories based on appearance variations. Instead of using predefined intra-category sub-categorization based on domain knowledge or validation datasets, we divide the sample space by unsupervised clustering using discriminative image features. We then use a cluster performance analysis (CPA) algorithm to verify the performance of the unsupervised approach. The CPA algorithm uses two performance metrics to determine the optimal number of sub-categories per object category. Furthermore, we employ the optimal sub-category representation as the basis and a supervised multi-category detection system with $\chi^2$ merging kernel function to efficiently detect and localize object categories within an image. Extensive experimental results are shown using a standard and the authors’ own databases. The comparison results reveal that our approach outperforms the state-of-the-art methods.

key words: sub-category optimization, cluster performance analysis (CPA), object detection and localization, unsupervised clustering, chi-square merging kernel

1. Introduction

Object categorization schemes should be applicable to a wide range of objects and scalable to handle a large number of object classes with appearance changes. However, the task of classifying, detecting, and clustering general object categories with large changes in appearance is still extremely challenging for computer vision to handle. This is because the image appearance of object categories changes due to many factors, such as large intra-category variations (e.g. different textures and colors), illumination, viewpoints, and poses. For object categories with a small intra-category variation, the cascade structure classifier proposed by Viola and Jones[1] has proven to be an efficient solution. However, for more diverse patterns with large appearance changes, such as multi-view cars, cows, dogs, etcetera, a more powerful classifier model is needed. One of the underlying philosophies for such classifiers is the divide-and-conquer method used in [2], [3]. In their approach, when the category cannot be modeled as a whole, it is divided into several sub-categories, and the system learns a model for each individually. However, when the appearance changes due to multiple factors, it is hard to find a dominating property to divide the samples. Manually assigning the sub-category labels for the training samples could be difficult and time consuming. Moreover, the domain knowledge based sub-categorization may not be optimal for the classification task. Thus, the real challenge and our goal here, is to find an optimal number of sub-category divisions of an object category.

This paper defines cluster performance analysis (CPA) criteria that can be used to automatically determine the optimal number of sub-categories for an object category. In particular, we present two criteria. In the first, instead of using a predefined intra-class sub-categorization based on domain knowledge, we divide the sample space by analyzing the performance metric of an unsupervised algorithm based on discriminative image features. For this purpose, we use the probabilistic latent semantic analysis (pLSA) model[4] to determine the object specific cluster probabilities. These probabilities are used to determine the sum of the cluster probabilities of an object category for a given number of topics. Then the metric is determined based on the cluster scatter probability values. To determine an optimal number of sub-categories, we first compute the metric for each possible value of clusters, and then select the minimum. Although this is a reasonably good approach for determining the optimal number of sub-categories using an unsupervised clustering approach, it considers only the overall cluster scatter probabilities. Thus, some categories may be underestimated in comparison to others, and is not always stable for complex object categories. To determine a more stable criterion we consider both the between and within sub-category scatter properties of the object categories.

In the second criterion, the number of clusters is served as the number of sub-categories and we use both between and within sub-category scatter matrices to analyze the cluster performance. Finally, both the first and second criteria are combined to determine an optimal number of sub-categories. Since our method locates an optimal number of sub-categories by analyzing cluster performance determined by an unsupervised clustering algorithm, we will refer to this method as Cluster Performance Analysis (CPA).

During an extensive evaluation involving a different standard and the authors’ own databases, we examine several aspects of our CPA. First, we analyze the classification performance of our CPA on a simple database. Second, we use the CPA to determine the most probable locations within

---

Manuscript received July 9, 2010.
Manuscript revised December 20, 2010.
†The authors are with the Department of Information and Computer Sciences, Saitama University, Saitama-shi, 338–8570 Japan.
a) E-mail: kuno@cv.ics.saitama-u.ac.jp
DOI: 10.1587/transinf.E94.D.1467

Copyright © 2011 The Institute of Electronics, Information and Communication Engineers
a more complex image with background clutter, and each hypothesis location is verified using SVM classifier with our $\chi^2$ merging kernel function. Moreover, we compare the performance of our approach to other related approaches [3], [5], [6].

The remainder of the paper is organized as follows. After reviewing related work in the next section, we describe the cluster performance analysis technique to determine the optimal number of sub-categories in Sect. 3. In Sect. 4, we describe our object detection technique using the optimized model. A description of the databases and a detailed description of the experimental procedure are given in Sect. 5. Our experimental results and conclusions are given in Sect. 6 and Sect. 7, respectively.

2. Related Work

In an unsupervised learning approach, non-negative matrix factorization (NMF) [7], [8] finds meaningful and physically interpretable latent variable decompositions. Although NMF is successful in learning object parts and semantic topics, this success does not imply that the method can learn parts from any database, such as images of objects viewed from different viewpoints or articulated objects. The pLSA is a statistical latent class model (or aspect model) that has shown excellent results in several information retrieval tasks [4] as well as in image categorization [9]. The pLSA parameters may be interpreted as probabilities, whereas the NMF factors are a set of values. Moreover, in the vocabulary of mixture models, the NMF factors are not identifiable, whereas pLSA model is. In the probabilistic approach to object categorization, the most important thing is to determine what kind of probability model (distribution) to employ as a representation of an object category. In statistical text analysis, the finite mixture model (FMM) proposed by Li et al. [10] is particularly well-studied for document analysis when documents concentrate on a single topic. Recently, the FMM has been proposed for image segmentation [11]. However, the performance of their segmentation task decreases with the complexity of the image to be segmented.

Recent success of the unsupervised topic model in object categorization [6], [9], [12] has led to a significant interest in the related fronts of topic optimization and subcategorization. Sivic et al. [9] investigated the use of both Latent Dirichlet Allocation (LDA) [13] and pLSA for clustering objects in image collections. In their approach, each model consistently gave similar results. However, the number of topics in [9] was chosen by hand to be equal (or very close) to the number of object categories, so that images are seen as mixtures of one ‘background’ topic with one ‘object’ topic. Since both LDA and pLSA give the same results, in our approach we choose the simpler pLSA model for initial sub-categorization. Contrarily to [9], Fei-Fei and Perona [12] proposed to model a scene category as a mixture of topics, and each topic is defined by a multinomial distribution over the quantized local descriptors. This is achieved by the introduction of an observed class node in their models [12] which explicitly requires each image example to be labeled during the learning process. Fritz and Schiele [6] proposed a representation to decompose, discover, and detect visual object categories using the topic model. In their approach, the total number of sub-categories is fixed during the learning process. Thus, their number is not optimized on the training dataset. In our approach, we model object categories by using pLSA and quantized local descriptors, but without assuming a one to one correspondence between categories and topics as in [9], and without learning a single distribution over topics per scene category as in [12].

The mixture models, like finite mixtures, require a priori choice of the number of topics or aspects. To address this issue, Teh et al. [14], [15] developed the Hierarchical Dirichlet Process (HDP), a Bayesian non-parametric mixed membership model. The HDP provides more flexible mixture model, avoiding costly model comparisons in order to determine an appropriate number of topics. However, the principle challenge in developing hierarchical models for images is to specify tractable, scalable, methods for handling uncertainty in the number of objects. To overcome this problem Sudderth et al. [16] proposed a Transformed Dirichlet Process (TDP) to allow more flexible sharing of mixture components between images. The TDP is naturally suited to their scene understanding application, as well as many other domains where ‘style and content’ are combined to produce the observed data. However, to model complex object categories that share a large number of visual similarity (such as horses and cows), only the generative model is not sufficient enough to determine an appropriate number of sub-categories of an object category. Thus we use the simpler pLSA model to determine the initial subcategorization and then use a discriminant analysis to finalize the appropriate number of sub-categories. Zhu and Martinez [17] proposed a discriminant analysis approach to optimize the classification performance on cross validation datasets. In their approach, they use an equal number of sub-categories for each object category, so the number of sub-categories cannot be optimized. Another disadvantage is that unnecessary sub-category division for an object category increases both the confusion rate and the training time of the system. When the appearance changes due to object viewpoints, some sub-categorization technique improves the detection performance [2], [18]. However, when the appearance changes due to multiple factors, it is hard to find one dominating property to divide an object category into an appropriate number of sub-categories. Huang et al. [2] make use of domain knowledge based upon a top-down sub-categorization technique to predefine five view categories based on the left-right out-of-plane rotation angle of the faces. The top-down sub-categorization is usually done manually because the automatic extraction of high level knowledge could be an even harder problem. Some researchers attempted to use an exemplar-based method to model complicated object categories. Shan et al. [18] proposed a boosted exemplar-based weak classifier to learn a pedestrian detector. In their approach, each weak classifier
is built based on one representative exemplar. However their algorithm is sensitive to the initial candidate set of exemplars. Moreover, how a candidate set is selected remains an open problem.

It has been recently shown that combining the power of generative modeling with discriminative classifier allows us to obtain good image classification [19], [20]. However, the proposed hybrid approach in [19] was mainly used for scene classification and did not provide any location information of objects. On the other hand, Fritz et al. [20] did not provide any sub-categorization and used single kernel in their discriminative stage. However, successful proliferation of a variety of kernels has recently motivated several researchers to combine multiple kernels [21], [22]. In [21], the main emphasis is to advocate the transition from support vector machine to support kernel machine-convex and sparse learners that can be used to train multiple kernels by jointly optimizing both the coefficient of a conic combination of kernel matrices and the coefficient of a discriminative classifier. However, we use two feature specific exponential $\chi^2$ kernel functions for SVM classifier. It has been shown in [23] that $\chi^2$ kernel is effective for histogram comparison.

3. Cluster Performance Analysis (CPA)

In this section we describe our CPA approach that combines cluster and sub-category performance metrics. Starting from the image, we first present our object representation technique. Then we describe how we apply the topic model to this representation and generate clusters for each object category. Finally, we analyze the sub-category performance using the generated clusters as sub-categories and mixing coefficient as discriminant features.

3.1 Data Representation

To construct a visual vocabulary for the pLSA model, we detect and describe keypoints from all training images. A promising recent research direction in computer vision is to use local features and their descriptions [24], [25]. The combination of keypoint detectors and invariant local descriptors has shown interesting capabilities of describing objects within an image [26]. In this research, local keypoints are detected in two phases. In the first, corner points are detected in the image and all of the corner points are selected as the one set of keypoints as indicated by the small circles in Fig. 1 (c). We use the corner detector algorithm proposed by He and Yung [27]. The algorithm reliably detects corner points from an edge map based on the local and global curvature properties of edges. In the second phase, the rest of the total keypoints are selected by taking samples on the object edges (indicated by the small rectangles in Fig. 1 (c)). We use the corner detector algorithm proposed by He and Yung [27]. The algorithm reliably detects corner points from an edge map based on the local and global curvature properties of edges. In the second phase, the rest of the total keypoints are selected by taking samples on the object edges (indicated by the small rectangles in Fig. 1 (c)). We use the corner detector algorithm proposed by He and Yung [27].

In the sampling process, the edge strengths are used as the weight of the samples. For this purpose, first we divide the strength of each edge pixel by the sum of all edge pixel strengths to determine sample density. Then we calculate the cumulative distribution function (CDF) of the sample density and draw samples (the number of samples is equal to the rest of the total keypoints) from the uniform distribution. Finally, we compare each sample of the uniform distribution to the CDF of sample density and select the sample that exceeds the sample value of the uniform distribution. This sampling process produces dense sampling for stronger edges and coarse sampling for weaker edges. Both corner points and samples on the edges make the model more shape informative, which is important in obtaining an overall estimate of the object boundary. Thus, during hypothesis generation, the model can give an estimate of the possible object shapes in addition to the probable object locations. Since the SIFT descriptors [26] are insensitive to change in scale, illumination, and minor viewpoint, each generated keypoint is described using the 128-dimensional SIFT descriptors. The SIFT descriptors are computed over the circular patch with radius $r = 10$. Keypoints are then grouped into a large number of clusters with those with similar descriptors assigned into the same cluster. By treating each cluster as a visual word that represents the specific local pattern shared by the keypoints in that cluster, we have a visual vocabulary (codebook) describing all kinds of such local image patterns. This visual vocabulary is formed by vector quantizing the SIFT descriptors using a $k$-mean clustering algorithm. We use the visual vocabulary of size (codebook size) 600. In the $k$-mean, first we randomly assign the cluster center to the codebook size. Then we iterate maximum 10 times for clustering. With its keypoints mapped into visual words, an object can be represented as a bag-of-visual-words (BOVW), or specifically, as a vector containing the count of each visual word in that object.

3.2 Cluster Performance Metric

In statistical text analysis, the pLSA model is used to discover topics in a document using bag-of-words documents representation. In this research we have images as documents and we discover topics as object categories (coffee mug, coffee jar etc.), so that an image containing instances of several objects is modelled as a mixture of topics (clusters). The model is applied to images by using a visual analogue of a word, formed by vector quantizing SIFT descriptors. Thus, using SIFT descriptors with pLSA model, objects with the same appearances are grouped under a single cluster. During the learning stage, the pLSA model associates each observation of a visual word $w$ within an object.
3.3 Sub-Category Performance Metric

Once the data distribution of each cluster has been determined using a mixture of Gaussian, it is easy to use the following generalize eigenvalue decomposition equation to find those discriminant vectors that best classify the data,

$$ S_w^{-1}S_B V = VA, $$

(4)

where $S_w$ is the within sub-category scatter matrix, and $S_B$ is the between sub-category scatter matrix (both matrices are symmetric, positive semi-definite); $V$ is a matrix whose column corresponds to the discriminant vectors, and $A$ is a diagonal matrix of corresponding eigenvalues. In the following section we explain how the between and within sub-category scatter matrices can be defined for our sub-category performance metric. Let $M_{ijk} = (\mu_{ij} - \mu_k)$, then the between sub-category scatter matrix is defined as,

$$ S_B = \sum_{i=1}^{C} \sum_{j=1}^{K_i} \sum_{k=1}^{K} \sum_{l=1}^{C} p_{ij}p_{il}M_{ijk}M_{ijl}^T, $$

(5)

where $p_{ij}$ and $\mu_{ij}$ are the prior and mean of the $j$-th sub-category in category $i$, and $K_i$ is the number of sub-categories in category $i$ with $\sum_{i=1}^{C} K_i = K$ being the total number of sub-category divisions and equal to the total number of clusters generated by the pLSA model. We can define the within sub-category scatter matrix as,

$$ S_w = \frac{1}{n} \sum_{i=1}^{C} \sum_{j=1}^{K_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^T, $$

(6)

where $x_{ij}$ is the $j$-th sample (mixing coefficient) of category $i$, and $\mu_i$ the sample mean of category $i$.

For sub-category discriminant analysis, we want to simultaneously maximize the measure given by the scatter matrix $S_B$ and minimize that computed by $S_w$. However, there may be some cases when this cannot be accomplished in parallel. In such cases, Eq. (4) may lead to an incorrect classification. Fortunately, this problem can be easily detected because when this happens the angle between the eigenvectors of $S_B$ and $S_w$ is small. This can be formally computed as $P = \sum_{i=1}^{C} \sum_{j=1}^{K_i} \langle u_j^Tw_i \rangle^2$, where $u_i$ and $w_i$ are the eigenvectors of $S_w$ and $S_B$ associated to the $i$-th largest eigenvalues, and the value of $m$ is less than the rank of $(S_B)^T$ [28]. We want the value of $P$ defined above to be as small as possible. We can compute the normalized value of this equation for each possible value of $K$ as,

$$ P_K = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{i} \langle u_j^Tw_{K,j} \rangle^2. $$

(7)

Based on the above two metrics our CPA algorithm can be summarized as follows:

1. Generate the training set $X = \{x_1, x_2, \ldots, x_n\}$ for $n$ training samples.
2. Repeat the following steps 3 to 9 for $K = C$ to $(C + r)$, where the value of $r$ can be specified by the user.
3. Learn the pLSA model for $K$-topic.
4. For all training objects compute the probabilities $P(z_k|d_j)$ using Eq. (1).
5. Assign an object category $i$ to the cluster with the highest $P(z_k|D_i)$. If the cluster with the highest $P(z_k|D_i)$ has already been assigned to another category, then try to find the cluster with the second highest $P(z_k|D_i)$ and so on. During this assignment process, if a cluster is shared by more than one category then we select the cluster for the category with the maximum value of $P(z_k|D_i)$.
6. Calculate the cluster performance metric $E_K$ using Eq. (3).
7. Calculate the matrices $S_B$ and $S_W$ using mixing coefficient given by Eq. (1) and taking $L_t$ as the number of sub-categories.
8. Calculate the sub-category performance metric $P_K$ using Eq. (7).
9. Compute $Q_K = \frac{1}{2}(E_K + P_K)$.
10. Select the minimum of $Q_K$ over $K$ as the optimal number of topic $K_0$, i.e., $K_0 = \arg \min_K Q_K$.

4. Object Detection

A classical problem in computer vision is how to classify a set of objects into a group of known categories along with their locations and scales within an image. Our object detection approach uses the CPA algorithm to determine the optimized model. Then in the recognition stage, when a test image is given, we extract a visual word $w$ from the image and each visual word is classified under the topic with highest topic specific word probability using the optimized model.

For a single object per image without any background clutter, such as in the remainder of the databases of Fig. 2, visual words are generated in those other object with background clutter, such as in the ETH-80 database [29] (Fig. 2 (a)), we classify images based on the maximum number of visual words supported by the specific sub-categories of an object category. However, in the case of multiple objects per image or an object with background clutter, such as in the remainder of the databases of Fig. 2, visual words are generated in those other than the desired target object areas. Thus, we use CPA to generate promising hypothesis locations within an image using the technique we recently introduced in [30]. However, we significantly improve the hypothesis generation speed in our new approach. Our modified hypothesis generation algorithm consists of the following steps.

1. Let ROI (region of interest) be the smallest rectangular window within the image that contains all possible visual words for a particular object category. Then, repeat the following steps with their corresponding rectangular ROI (Fig. 3 (a)) for all object categories.
2. Compute the average aspect ratio $M_{wi}$ of the window for each object category $i$ as $M_{wi} = W/W_{ROI}$, where $W_{wi}$ and $W_{ROI}$ are the mean width and mean height of the object category $i$ computed during the training stage using ground truth bounding boxes.
3. For each object category, slide the window with size $M_{wi} \times H_{ROI}$ (Fig. 3 (b)), and count the number of visual
words, $N_{vw} = \sum_{i=1}^{L_i} n_{vw(i)}$, where $n_{vw(i)}$ is the number of visual words for object category $i$ and sub-cluster $L_i$ and $H_{ROI(i)}$ is the height of ROI for category $i$.

4. Determine the local maxima (Fig. 3(c)) based on the number of visual words $N_{vw}$ in each of the sliding windows.

5. For all local maxima regions within an image find and suppress the windows, if any, which overlap by 75% or more with the window containing the maximum number of visual words. This step is nearly identical to the non-maximum suppression technique (Fig. 3(d)).

6. After suppressing the non-maximum windows in each neighborhood, within the remaining windows, slide the object window with average aspect ratio $M_{avg}$ (Fig. 3(e)) to determine the window that contains the maximum number of visual words in each local region. Such windows are selected as the promising hypotheses for our optimized model.

Our modifications are in steps 3 to 6, which reduce the search space significantly. Thus, the new algorithm is much faster compared to [30]. Since the maximum number of visual words can be generated from the probable objects’ location, the effect of non-maximum suppression as described in step 5 has very little effect on the hypothesis generation performance. If the size of the rectangular ROI is $m \times n$ and the object window size is $p \times q$, then the algorithm proposed in [30] requires $(m-p+1) \times (n-q+1)$ times of sliding window search for hypothesis generation. On the other hand, if there are $r$ promising hypotheses ($r$ is much smaller than $(n-q+1)$, typically in the range 1 to 10 in our case), then our algorithm requires only $(m-p+1) \times r + (n-q+1)$ times of sliding window search. For ten object categories, our modified algorithm requires 0.084s to generate all hypotheses, whereas [30] requires 7.72s on a 2.40 GHz PC.

Each of the generated hypotheses is verified using the SVM classifier (Fig. 3(f)). We use feature specific $\chi^2$ merging kernel function with both shape and appearance features. Our appearance feature is represented by the bag-of-visual-words (BOVW) histogram constructed from the keypoints (corner and edge sample points) with SIFT descriptors as described in Sect. 3.1. To extract the shape feature we use the pyramid histogram of oriented gradient (PHOG) [31]. We represent the features specific $\chi^2$ merging kernel as:

$$K(X, Y) = \alpha K_{PHOG}(X, Y) + \beta K_{BOVW}(X, Y) \tag{8}$$

where,

$$K_{PHOG}(X, Y) = e^{-\sum_{i=0}^{H} \frac{(x_i^T y_i)^2}{2 \sigma^2}},$$

$$K_{BOVW}(X, Y) = e^{-\frac{(x_i^T y_i)^2}{\sigma}},$$

where $H$ is the number of pyramid level for PHOG descriptor, and $\alpha$ and $\beta$ are weights for the PHOG and BOVW kernels, respectively. To find the similarities between the feature sets $X$ and $Y$, our kernel functions use $\chi^2$ distance on the normalized PHOG and BOVW feature histograms, as they are demonstrated to be good distance measures for histogram comparison [23] than linear, quadratic, and rbf kernels. We use the LIBSVM package [32] for our experiment in multi-class mode. For comparison with our kernel function, we also use the LIBSVM’s rbf kernel with the merging feature.

5. Databases and Evaluation Methodology

5.1 Databases

We present an extensive experimental evaluation involving both a standard and the authors’ databases. To compare our approach with others, we evaluate our classification algorithm on three standard databases: (i) ETH-80 multi-view database, (ii) four datasets collected from PASCAL database collection [33], and (iii) ETHZ-shape database [34]. We also evaluate our approach on our 10-category database related to our application (service robots). Figure 2 shows example images for each database, and the contents are summarized as follows:

**ETH-80 multi-view database.** Contains 3280 images of the following 8 categories: cars, cows, apples, cups, dogs, pears, tomatoes, and horses. Each category includes images of ten objects (e.g., ten different cows in the case of the cow category) photographed from a total of 41 viewpoints. This gives us a total of 410 images per category.

**Four categories from the PASCAL database.** Dataset is taken from the PASCAL database collection and contains a single object per image. It includes 1607 images of four categories: 328 horses from Weizmann dataset, 111 cows from TUD dataset, 798 motorbikes from Caltech dataset, and 370 cars from UIUC dataset.

**ETHZ shape database.** Contains a total of 255 images divided along apple logos (40), bottles (48), giraffes (87), mugs (48), and swans (32).

**Authors’ database.** Contains images of multiple objects per image and is created with ground truth bounding boxes. It consists of 10 categories of everyday objects related to our application (service robot) in different environments against cluttered, real-world background with occlusion, scale, and viewpoint changes. There are total of 809 images containing 2138 objects of 10 categories: coffee jars, coffee mugs, spoons, hand soaps, cup noodles, computer monitors, computer keyboards, computer mice, CDs, and books. Among them 630 objects (340 images) are used for training, and the remainder of the 1508 objects (469 images) are used for testing the system.

5.2 Evaluation Methodology

The classification task on the ETH-80 database is to assign each test image to one of the eight categories. For this purpose, we use the optimized model generated by our CPA algorithm to classify visual words and images. For other databases, given a completely unlabeled image of single or multiple object categories, our goal is to automatically
detect and localize all target objects within the image. In our approach, object presence detection means determining if one or more objects are present in an image and localization means finding the locations of objects in that image. The localization performance is measured by comparing the detected window area to the ground truth object window. Based on the object presence detection and localization, an object is counted as a true positive object if $\frac{\text{area}(B_{\text{pred}} \cap B_{\text{gt}})}{\text{area}(B_{\text{gt}})} \geq 0.5$, with $B_{\text{pred}}$ being the predicted bounding box (generated by the hypothesis generation algorithm and verified by the SVM classifier) and $B_{\text{gt}}$ the bounding box denoted by the ground truth (annotated bounding box for test image). Otherwise, the detected object is counted as false positive. In the experimental evaluation, the detection and localization rate (DLR) is defined as:

$$\text{DLR} = \frac{\# \text{ of true positive objects}}{\# \text{ of annotated objects}}. \quad (9)$$

The false positive per image (FPPI) is defined as:

$$\text{FPPI} = \frac{\# \text{ of false positive objects}}{\# \text{ of total test image}}. \quad (10)$$

In the experimental evaluation, we prefer DLR at the incidence of FPPI because FPPI has a clear interpretation for multiple objects per image and is independent of negative test images.

During the training stage, the optimal number of sub-categories is detected on the training datasets using our CPA algorithm without any validation datasets. The optimized model is used to generate promising hypotheses locations that are verified by our SVM classifier. In this paper, we investigate the benefit of feature combination at kernel level than single feature. In the $\chi^2$ merging kernel, we use a pyramid matching kernel with the number of pyramid level $H = 2$ for PHOG features (which is equal to the number of pyramid level for PHOG feature sets) and a simple $\chi^2$ kernel for BOVW histogram features. The different parameters of the SVM classifier are estimated as follows: the values of cost parameters $C_{\text{SVM}}$ for both rbf and custom kernels, and the value of kernel parameter, $\gamma$, for the rbf kernel are obtained using five-fold cross-validation datasets. For example, on the authors’ database the best cross-validation accuracy (98.00%) for rbf kernel is obtained with $C_{\text{SVM}} = 20$ and $\gamma = 0.5$. However, for our $\chi^2$ merging kernel the best cross-validation accuracy (99.33%) is obtained with $C_{\text{SVM}} = 20$ on the same database. While we attempted with different values of $\alpha$ and $\beta$ within the range of 0 to 1, in our experiments $\alpha = \beta = 1$ gives the best result.

6. Experimental Results

We present extensive experimental evaluations, involving several existing and our own datasets. In the training stage, all the labeled training datasets are presented to the system. Our CPA algorithm uses the BOVW feature (extracted from the object area using the technique as described in Sect. 3.1) to determine the optimal number of sub-categories. At the same time the SVM classifier is learned with merging kernel function that uses BOVW and PHOG features. During testing stage, when a new test image is given, the system generates a set of promising hypotheses with BOVW feature. Then in the verification stage, we extract PHOG and BOVW features from the hypothesis locations for the SVM classifier. To understand how the proposed CPA method performs, in the following subsections we investigate three areas: (1) optimization performance on the ETH-80 database, (2) comparisons with other methods, and (3) performance analysis based on the authors’ database using $\chi^2$ merging kernel with or without the sub-categorization approach.

6.1 Optimization Performance on ETH-80 Database

To measure the classification and optimization performance, the ETH-80 multi-view database is split into test and train datasets: 10 images of different views from 10 instances of a category are used as a test dataset and the rest of the images are used for the training dataset. Thus, for each phase we use 80 test images and 3200 training images from 8 object categories. Since there are 41 views for each instance, there are the same number (41) of train/test phases. In each training stage, we use the CPA algorithm to determine the optimal number of sub-categories of object categories. We have conducted experiments using the cluster performance metric to find the optimal number of sub-categories. The performance metric only considers the sum of the overall cluster scatter probabilities. Thus, some categories may be underestimated or overestimated in comparison to others and is not stable for this multi-view database. For this database our actual optimization occurs at 53 sub-categories (Fig. 4 (d)). However, for 53 sub-categories the cluster performance metric $E_{K}$ does not provide an optimal performance as indicated in Fig. 4 (a). Although the sub-category performance metric $P_{K}$ produces better results than the $E_{K}$, the metric is not always sufficient to find the optimal number of sub-categories (Fig. 4 (b)). However, the combination of both criteria is able to produce the optimal number of sub-categories (Fig. 4 (c)) for one of the training phases. The number of sub-categories for each object category is shown in Table 1. Figure 5 shows a few typical images of 8 optimal sub-categories for the car object category.

The test dataset of images is used to measure the classification performance for all of the generated models. The classification performance for the test data set is shown in Fig. 4 (d). As indicated by the circular marker on this graph, our optimized model produces the best classification performance (84.75%). We also evaluate the average classification performance over all 41 test phases. The average classification result for 8 object categories without sub-categorization is only 59.5%. However, for sub-category optimization, the best average classification accuracy is 84.5%, which is also obtained for 53 sub-categories. Our average classification result is comparable with Zhuowen Tu [3] (the average recognition rate is 76%). Our better result in comparison to
6.2 Comparison to Previous Results

We compare the performance of our sub-category optimization-based detection approach to other methods based on different standard databases.

Table 1  Sub-categories for each object category.

<table>
<thead>
<tr>
<th>Category</th>
<th>apple</th>
<th>car</th>
<th>cow</th>
<th>cup</th>
<th>dog</th>
<th>horse</th>
<th>pear</th>
<th>tomato</th>
</tr>
</thead>
<tbody>
<tr>
<td>#sub-categories</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Fig. 4  Sub-category optimization and its classification accuracy: (a) cluster performance metric vs. number of cluster, (b) sub-category performance vs. number of sub-categories, (c) sub-categories optimization based on confliction metric, and (d) number of sub-categories vs. classification performance.

Table 2  Comparison among different features.

<table>
<thead>
<tr>
<th>Category</th>
<th>SIFT Feature</th>
<th>MSER Feature</th>
<th>BOVW Feature</th>
<th>PHOG Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DLR at 0.74 FPP</td>
<td>DLR at 0.76 FPP</td>
<td>DLR at 0.75 FPP</td>
<td>DLR at 0.73 FPP</td>
</tr>
<tr>
<td>Coffee jar</td>
<td>0.74</td>
<td>0.76</td>
<td>0.75</td>
<td>0.73</td>
</tr>
<tr>
<td>Coffee mug</td>
<td>0.65</td>
<td>0.67</td>
<td>0.64</td>
<td>0.81</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.68</td>
<td>0.61</td>
<td>0.80</td>
<td>0.82</td>
</tr>
<tr>
<td>Hand soap</td>
<td>0.78</td>
<td>0.75</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>Cup noodle</td>
<td>0.80</td>
<td>0.78</td>
<td>0.78</td>
<td>0.76</td>
</tr>
<tr>
<td>C. monitor</td>
<td>0.44</td>
<td>0.72</td>
<td>0.76</td>
<td>0.84</td>
</tr>
<tr>
<td>C. keyboard</td>
<td>0.75</td>
<td>0.78</td>
<td>0.82</td>
<td>0.72</td>
</tr>
<tr>
<td>C. mouse</td>
<td>0.26</td>
<td>0.35</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>CD</td>
<td>0.61</td>
<td>0.62</td>
<td>0.60</td>
<td>0.78</td>
</tr>
<tr>
<td>Book</td>
<td>0.80</td>
<td>0.68</td>
<td>0.73</td>
<td>0.60</td>
</tr>
<tr>
<td>Average</td>
<td>0.660</td>
<td>0.675</td>
<td>0.735</td>
<td>0.761</td>
</tr>
</tbody>
</table>

6.2.1 Comparison among Different Features

We investigate four detectors and descriptors and compare their performance on our database with ten object categories. Some categories are best described by shape alone (coffee mug, spoon etc.), while others by appearance patches (computer keyboard, book etc.). Table 2 shows the performance of our system on different detectors and descriptors. From the experimental results of Table 2 and Table 6 (last column) it can be seen that the feature combination at the kernel level produces better results than the single feature alone.

SIFT Features. Interest points are detected using the SIFT keypoint detection algorithm [26]. Each of the interest point is described using the SIFT descriptors. The SIFT detector perform well for feature-rich objects (such as cup noodle, book, hand soap). However, the main limitation of SIFT like feature detectors are that they do not find proper matches for objects with little or no texture (such as spoon, computer mouse, computer monitor). The average detection and localization rate is only 66% for ten object categories.

MSER Feature. Interest points are detected using the maximally stable extremal regions detector [35]. We use the SIFT descriptors to describe each of the keypoints detected by MSER detector. The MSER produces good repeatability of interest point with large viewpoint changes and we obtain better result than SIFT detector with a detection rate of 67.5%. However, the method still performs lower detection rate for objects with little or no texture.

BOVW Feature. In our BOVW feature, interest points are detected using both corner points and edge sample points. The interest points are described with the SIFT descriptors. Since our detector detects interest points on texture less objects (spoon, computer monitor, etc.), we are able to improve the detection rate for these object categories than the SIFT detector. In this case, our average detection rate is 73.5%.

PHOG Feature. Here an object is represented by the local shape and spatial layout of the shape. The local shape is captured by the distribution over edge orientations within a region, and spatial layout by tiling the object area (deter-
Table 3  Comparison against other methods on ETHZ shape database. Average detection rate at 0.37 false positive per image (FPPI) averaged over five-fold cross-validation. Standard deviation is specified in parentheses.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple logos</td>
<td>83.2 (1.7)</td>
<td>89.9 (4.5)</td>
<td>89.0 (4.2)</td>
</tr>
<tr>
<td>Bottles</td>
<td>83.2 (7.5)</td>
<td>76.8 (6.1)</td>
<td>90.8 (5.4)</td>
</tr>
<tr>
<td>Giraffes</td>
<td>58.6 (14.6)</td>
<td>90.5 (5.4)</td>
<td>95.1 (1.9)</td>
</tr>
<tr>
<td>Mugs</td>
<td>83.6 (8.6)</td>
<td>82.7 (5.1)</td>
<td>83.3 (9.3)</td>
</tr>
<tr>
<td>Swans</td>
<td>75.4 (13.4)</td>
<td>84.0 (8.4)</td>
<td>87.5 (9.9)</td>
</tr>
<tr>
<td>Average</td>
<td>76.8</td>
<td>84.8</td>
<td>89.1</td>
</tr>
</tbody>
</table>

Table 4  Performance comparison of our algorithm with the other methods using their databases.

<table>
<thead>
<tr>
<th>Category and dataset</th>
<th>[36]</th>
<th>[20]</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car (UIUC)</td>
<td>87.6%</td>
<td>88.6%</td>
<td>95.9%</td>
</tr>
<tr>
<td>Cow (TUD)</td>
<td>92.5%</td>
<td>93.2%</td>
<td>95.8%</td>
</tr>
<tr>
<td>Horse (Weizmann)</td>
<td>88.5%</td>
<td>88.5%</td>
<td>99.4%</td>
</tr>
<tr>
<td>Motorbike (Caltech)</td>
<td>80.0%</td>
<td>84.0%</td>
<td>98.8%</td>
</tr>
</tbody>
</table>

Table 5  Comparison of cross-category confusion to [20], [36] using their databases.

<table>
<thead>
<tr>
<th>Cat.</th>
<th>Car</th>
<th>Cow</th>
<th>Horse</th>
<th>Motorbike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>1.00</td>
<td>0.49</td>
<td>0.24</td>
<td>0.07</td>
</tr>
<tr>
<td>Cow</td>
<td>0.71</td>
<td>0.16</td>
<td>0.26</td>
<td>0.53</td>
</tr>
<tr>
<td>Hor.</td>
<td>1.07</td>
<td>0.08</td>
<td>0.08</td>
<td>0.29</td>
</tr>
</tbody>
</table>

In this section we compare the performance of our approach to the topic model based decomposition approach of Fritz et al. [6], and the deformable shape model of Ferrari et al. [5], using the ETHZ shape database. It is a challenging database as objects appear in a wide range of scales and exhibit considerable intra-class variations, and many images are extensively cluttered. Experiments are conducted using the same evaluation criteria with the same settings as described in [5]. In this experiment, our optimization algorithm decomposes 5 object categories into 11 sub-categories. Using the five-fold cross-validation as proposed in [5], we obtained the results presented in Table 3. Averaged over all five categories we obtain a 89.1% detection rate. We have also done experiment without any categorization technique and obtained an average detection rate of 83.5%.

Here we report the object detection performance as the detection rate at 0.37 FPPI, averaged over five trials. For all object categories except apple logos and mugs our approach performs better than the other two approaches. Although the detection performance of our approach on apple logos category is slightly less than [6], the performance is better than [5]. For the mugs category, our method performs better than [6], however it performs slightly less than [5]. Averaged over all categories we improve the detection and localization performance by 12.3% compared to Ferrari et al. [5] and 4.3% compared to Fritz et al. [6]. This better performance (compared to [5], [6]) could be due to the use of an optimized topic model with sub-categorization technique and the use of shape and appearance features with the $\chi^2$ merging kernel. Some detection results on the ETHZ shape database are shown in Fig. 6.

### 6.2.2 Performance Comparison on ETHZ Database

In this section we compare the performance of our approach to the topic model based decomposition approach of Fritz et al. [6], and the deformable shape model of Ferrari et al. [5], using the ETHZ shape database. It is a challenging database as objects appear in a wide range of scales and exhibit considerable intra-class variations, and many images are extensively cluttered. Experiments are conducted using the same evaluation criteria with the same settings as described in [5]. In this experiment, our optimization algorithm decomposes 5 object categories into 11 sub-categories. Using the five-fold cross-validation as proposed in [5], we obtained the results presented in Table 3. Averaged over all five categories we obtain a 89.1% detection rate. We have also done experiment without any categorization technique and obtained an average detection rate of 83.5%.

Here we report the object detection performance as the detection rate at 0.37 FPPI, averaged over five trials. For all object categories except apple logos and mugs our approach performs better than the other two approaches. Although the detection performance of our approach on apple logos category is slightly less than [6], the performance is better than [5]. For the mugs category, our method performs better than [6], however it performs slightly less than [5]. Averaged over all categories we improve the detection and localization performance by 12.3% compared to Ferrari et al. [5] and 4.3% compared to Fritz et al. [6]. This better performance (compared to [5], [6]) could be due to the use of an optimized topic model with sub-categorization technique and the use of shape and appearance features with the $\chi^2$ merging kernel. Some detection results on the ETHZ shape database are shown in Fig. 6.

### 6.2.3 Performance Comparison on Four Categories of Different Databases

In this section we compare the performance of our system on four object categories that are tested in [20], [36]. The comparison is performed on the categories using the multi-category discrimination results presented in [20]. The four object categories are fitted on 9 optimal sub-categories: 2 sub-categories each for cars, cows and motorbikes, and 3 sub-categories for horses. Table 4 summarizes the detection and localization performance of our method with other methods. We obtain the detection rate in Table 4 against the cross-category confusion rate as given in Table 5. In the case of the horse dataset, the performance of [20], [36] is calculated from the recall-precision curve that is presented in [20]. The comparison is valid because for the same detection results (as indicated in the recall-precision curve) Fritz et al. [20] provided cross-category confusion rate. Our result is comparable against cross-category confusion rate. Table 5 shows the comparison of cross-category confusion (false positive per image) on the same datasets. It can be noted that in all cases our method achieved better detection and localization results compared with the other two methods. This improved performance might be due to the use of better features and sub-categorization technique for hypothesis generation. Although the recognition task is different from our multiple object detection and localization, we perform this experiment to compare the basic performance of our method with others.
6.3 Multiple Object Detection Performance on Authors’ Database

While most of the images in the previous sections of our experiments contained a single object per image and relatively less cluttered background, we also created a database that contains multiple everyday objects per image with cluttered, real-world background. Thus our sub-categorization algorithm considers background information as a separate category. In the training period, the background features were taken from the areas other than the target objects in the images. However, since the background features may vary significantly, we need to restrict the number of sub-categories for the background up to a certain value. In this experiment, it was fixed to 2 sub-categories. The reason for adding background category is to give the system the opportunity of discovering background features generated from the areas other than the target object areas within the test images. It reduces the confusion between objects features and background features in the hypothesis generation stage. With the background category, the system increases the average detection rate by approximately 5% with a reduction of false positive of 2% than without the background category.

In the following section we investigate the performance of our approach for the detection task of multiple objects. In the training period, the CPA algorithm generates an optimized model for our 10 object categories. The optimization occurs at 21 sub-categories for 10 object categories with two background sub-categories (Fig. 7 (a)). Figure 7 (b) shows the accuracy of correctly generated hypotheses against the number of sub-categories. Some typical best hypotheses locations and their accuracy (measured against ground truth bounding boxes) are shown in Figs. 7 (c) & (d). To compare the performance of our $\chi^2$ merging kernel we also evaluated the performance on the same datasets using LIBSVM’s standard $\text{rbf}$ kernel with the merging features (3rd column

<table>
<thead>
<tr>
<th>Category</th>
<th>$\chi^2$ merging kernel without sub-cat.</th>
<th>$\text{rbf}$ kernel with opt. sub-cat.</th>
<th>$\chi^2$ merging kernel with opt. sub-cat.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DLR at 0.68 FPPI</td>
<td>DLR at 0.64 FPPI</td>
<td>DLR at 0.61 FPPI</td>
</tr>
<tr>
<td>Coffee jar</td>
<td>0.76</td>
<td>0.87</td>
<td>0.88</td>
</tr>
<tr>
<td>Coffee mug</td>
<td>0.34</td>
<td>0.77</td>
<td>0.82</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.79</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>Hand soap</td>
<td>0.64</td>
<td>0.73</td>
<td>0.76</td>
</tr>
<tr>
<td>Cup noodle</td>
<td>0.62</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>C. monitor</td>
<td>0.80</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>C. keyboard</td>
<td>0.85</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>C. mouse</td>
<td>0.53</td>
<td>0.73</td>
<td>0.78</td>
</tr>
<tr>
<td>CD</td>
<td>0.22</td>
<td>0.78</td>
<td>0.81</td>
</tr>
<tr>
<td>Book</td>
<td>0.39</td>
<td>0.78</td>
<td>0.82</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.610</strong></td>
<td><strong>0.833</strong></td>
<td><strong>0.854</strong></td>
</tr>
</tbody>
</table>

Fig. 7  Hypothesis generation with sub-category optimization: (a) sub-categories optimization based on confliction metric, (b) number of sub-categories vs. hypothesis generation performance, and (c) & (d) example generated hypotheses and their accuracy.

Fig. 8  Detected target objects with their locations and scales.
As shown in Table 6, without the sub-categorization our system produces an average DLR of 61% at 0.68 FPPI. On the other hand, with sub-categorization, the system increases the average DLR to 83.3% with a reduction of FPPI from 0.68 to 0.63. The best performance is obtained by using our \( \chi^2 \) merging kernel function. In this case, the average DLR for 10 object categories is 85.4% at 0.61 FPPI. Figure 8 shows some detection results of our approach on our database under cluttered background, partial occlusion, significant scale, and viewpoint changes.

7. Conclusion

Over the last few years, unsupervised clustering algorithms have been proposed, most specify the number of clusters as either being equal to the number of actual categories to be learned or optimized them on validation datasets. Since generating a proper validation dataset is difficult and time consuming, in this research we have proposed a new approach of automatically optimizing object categories on a training dataset using unsupervised clustering and a sub-cluster performance analysis algorithm. We have also demonstrated how to generate hypotheses quickly and efficiently from the optimized model without using an exhaustive search of a quality function over all rectangular regions of interest. Thus, our hypothesis generation approach reduces the search space and speeds up our detection process. The system has shown the ability to discriminate among diverse object categories using a feature specific \( \chi^2 \) merging kernel with both shape and appearance features. Our experimental results have demonstrated that the sub-category optimization technique significantly improves the accuracy of the generated hypothesis, which in turn increases the detection and localization results for all object categories.

Acknowledgments

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology under the Grant-in-Aid for Scientific Research (KAKENHI 19300055).

References


[34] V. Ferrari, “Ethz shape classes,” http://www.vision.ee.ethz.ch/ datasets/


Dipankar Das received his B.Sc. and M.Sc. degrees in Computer Science and Technology from the University of Rajshahi, Rajshahi, Bangladesh in 1996 and 1997, respectively. He is an assistant professor of the Department of Information and Communication Engineering of the same university. Currently, he is also a Ph.D. student in the Graduate School of Science and Engineering, Saitama University. His research interests include Object Recognition and Human Computer Interaction.

Yoshinori Kobayashi completed the M.E. degree from the Department of Information Management Science at Graduate School of Information Systems, University of Electro-Communications in 2000, and joined the Design Systems Engineering Center of Mitsubishi Electric Corporation. He was in the doctoral program in 2004–2007 in Information and Communication Engineering at the Graduate School of Information Science and Technology, the University of Tokyo, and then he joined the Department of Information and Computer Sciences, Saitama University, as an assistant professor. He is interested in computer vision for human sensing and its application for human computer interaction.

Yoshinori Kuno received B.S., M.S. and Ph.D. degrees in 1977, 1979 and 1982, respectively, all in Electrical and Electronics Engineering from the University of Tokyo. In 1982, he joined Toshiba Corporation. From 1987 to 1988, he was a Visiting Scientist at Carnegie Mellon University. In 1993, he moved to Osaka University as an associate professor in the Department of Computer Controlled Mechanical Systems. Since 2000, he has been a professor in the Department of Information and Computer Sciences, Saitama University.