Feature Extraction and Learning Using Context Cue and Rényi Entropy Based Mutual Information

Hong Pan[1,2(✉)], Søren Ingvor Olsen[1], and Yaping Zhu[1]

[1] Department of Computer Science, University of Copenhagen, 2100 København Ø, Denmark
{hong.pan, ingvor, y.zhu}@di.ku.dk
[2] School of Automation, Southeast University, Nanjing 210096, China

Abstract. Feature extraction and learning play a critical role for visual perception tasks. We focus on improving the robustness of the kernel descriptors (KDES) by embedding context cues and further learning a compact and discriminative feature codebook for feature reduction using Rényi entropy based mutual information. In particular, for feature extraction, we develop a new set of kernel descriptors−Context Kernel Descriptors (CKD), which enhance the original KDES by embedding the spatial context into the descriptors. Context cues contained in the context kernel enforce some degree of spatial consistency, thus improving the robustness of CKD. For feature learning and reduction, we propose a novel codebook learning method, based on a Rényi quadratic entropy based mutual information measure called Cauchy-Schwarz Quadratic Mutual Information (CSQMI), to learn a compact and discriminative CKD codebook. Projecting the original full-dimensional CKD onto the codebook, we reduce the dimensionality of CKD while preserving its discriminability. Moreover, the latent connection between Rényi quadratic entropy and the mapping data in kernel feature space further facilitates us to capture the geometric structure as well as the information about the underlying labels of the CKD using CSQMI. Thus the resulting codebook and reduced CKD are discriminative. We verify the effectiveness of our method on several public image benchmark datasets such as YaleB, Caltech-101 and CIFAR-10, as well as a challenging chicken feet dataset of our own. Experimental results show that our method has promising potential for visual object recognition and detection applications.

Keywords: Context Kernel Descriptors · Cauchy-Schwarz Quadratic Mutual Information · Feature extraction and learning · Object classification and detection

1 Introduction

Recognition and detection of real-world objects are challenging, because it is difficult to model objects with significant variations in color, shape and texture. In addition, the backgrounds in which the objects exist are often complex and cluttered, and we have to account for changes of illumination, pose, size, and number of objects in the most

A. Fred et al. (Eds.): ICPRAM 2015, LNCS 9493, pp. 69–88, 2015.
DOI: 10.1007/978-3-319-27677-9_5

contrived situations. Currently, local based image representations [1–13] prevail in the state-of-the-art object recognition and detection algorithms. These local based image representations follow the bag-of-features framework [5, 6]. It first extracts low-level patch descriptors over a dense grid or salient points, then encodes them into mid-level features in a unsupervised way using mix of Gaussian, K-means or sparse coding, and finally derives the image-level representation using spatial pooling schemes [5–7]. Usually, carefully designed descriptors such as SIFT [8], SURF [9], LBP [10] and HOG [11] are used as low-level descriptors to gather statistics of pixel attributes within local patches. However, design of hand-crafted descriptors is non-trivial as sufficient prior knowledge is required and well-tuned parameters are necessary to achieve a good performance. Besides, we still lack a deep understanding on the design rules behind them. Recently, Bo et al. [1, 2] tried to answer how SIFT and HOG measure the similarity between image patches and interpret the design philosophy behind them from a kernel's view. They showed that the inner product of orientation histogram applied in SIFT and HOG is a particular match kernel over image patches. This insight provides a general way to turn pixel-level attributes into patch-level features with match kernels comparing similarities between image patches. Based on that, they designed a set of low-level descriptors called kernel descriptors (KDES) and kernel principal component analysis (KPCA) [14, 15] was used to reduce the dimensionality of KDES. However, KPCA only captures second-order statistics of KDES and cannot preserve its high-order statistics. It inevitably degrades the distinctiveness of KDES for nonlinear clustering and recognition where high-order statistics are needed. Wang et al. [4] merged the image label into the design of patch-level KDES and derived a variant KDES called supervised kernel descriptors (SKDES). Guiding KDES under a supervised framework with the large margin nearest neighbor criterion and low-rank regularization, SKDES reported an improved performance on object recognition.

In this work, we focus on improving the original KDES by embedding context cues into the descriptors and further learning a compact and discriminative Context Kernel Descriptors (CKD) codebook for object recognition and detection using information theoretic learning techniques. In particular, for feature extraction, we develop a set of CKD that enhance the KDES with embedded spatial context. Context cues enforce some degree of spatial consistency which improves the robustness of the resulting descriptors. For feature learning, we adopt the Rényi entropy based Cauchy-Schwarz Quadratic Mutual Information (CSQMI) [28], as an information theoretic measure, to learn a compact and discriminative CKD codebook from a rich and redundant CKD dictionary. In our method, codebook learning involves two steps including the codebook selection and refinement. In the first step, a group of compact and discriminative basis vectors are selected out of all available basis vectors to construct the codebook. By maximizing the CSQMI between the selected basis vectors in the codebook and the remaining basis vectors in the dictionary, we obtain a compact CKD codebook. By maximizing the CSQMI between the low-dimensional CKD generated from the codebook and their class labels, we also boost the discriminability of the learned CKD codebook. In the second step, we further refine the codebook for improved discriminability and low approximation error with a gradient ascent method that maximizes the CSQMI between the low-dimensional CKD and their class labels, given the constraint on a sufficient approximation accuracy. Projecting the full-dimensional CKD onto the

learned CKD codebook, we derive the final low-dimensional discriminative CKD for feature representation. Evaluation results on standard recognition benchmarks, and a challenging chicken feet dataset show that our proposed CKD model outperforms the original KDES as well as carefully tuned descriptors like SIFT and some sophisticated deep learning methods.

The low-level patch features used in our work is built upon the KDES. Conceptually, it is related to [1], but our work departs from it in two distinct ways that improve the robustness and discriminability of our feature representation. First, we propose an enhanced match kernel called context match kernel (CMK). CMK strengthens the spatial consistency of the original match kernel by embedding the extra neighboring information into it. Spatial occurrence constraints implicit in the CMK significantly improve the robustness of similarity matching between feature sets, even for ambiguous or impaired features generated from partially occluded objects. Second, rather than using KPCA for reduction of the feature dimensionality, we perform the feature dimensionality reduction by projecting the original high-dimensional CKD onto a compact and discriminative CKD codebook. The CKD codebook is learned from a novel information theoretic feature selection algorithm based on the CSQMI. Because CSQMI is derived from the Rényi quadratic entropy, we can efficiently approximate it using a Parzen window [28]. In addition, considering the geometric interpretation of the CSQMI [28], it allows us to learn a discriminative CKD codebook that captures the cluster structure of input samples as well as the information about their underlying labels. Hence, the low-dimensional CKD derived from our model is more discriminative than the original KDES derived from KPCA.

2 Feature Extraction Using CKD

We enhance the original match kernel in [1] by embedding neighborhood constraints into it. As neighborhood defines an adjacent set of pixels surrounding the center pixel, neighborhood information can be regarded as the spatial context of the center pixel. So we refer to this enhanced match kernel as Context Match Kernel and the resulting descriptors as Context Kernel Descriptors. Intuition behind CMK is that pixels with similar attributes from two patches should have a high probability to have neighboring pixels whose attributes are also similar. Considering the spatial co-occurrence constraint, our CMK significantly improve the matching accuracy. CMK can be easily applied to develop a set of local descriptors using any pixel attributes, such as gradient, color, texture, and shape, etc. Next we derive the CMK, then we introduce several specific CMKs used in this work.

2.1 Formulation of CMK

An image patch can be modelled as a set of pixels $\boldsymbol{X} = \{x_i\}_{i=1}^{n}$, where x_i is the coordinate of the ith pixel. Let a_i be attribute vector of the ith pixel x_i. The k-neighborhood N_k^i of pixel x_i in $\boldsymbol{X}$ is defined as a group of pixels (including itself) that are closest to it. Mathematically, $N_k^i = \{x_j \in \boldsymbol{X} | \ \|x_i - x_j\| \leq k;\ k \geq 1\}$. To eliminate the image

noise, we smooth the image using a Haar wavelet filter and compute the local gradient in the k-neighborhood. For the k-neighborhood centered at x_p, we first normalize the neighborhood's attribute by voting the pixel's attribute in N_k^p with its gradient magnitude weighted by a Gaussian function centered at x_p. The width of Gaussian function, which normalizes the attributes contributed from off-center pixels, is controlled by the neighborhood size k. Similarly, we also normalize the attribute in the k-neighborhood centered at x_q. With the normalized attributes in N_k^p and N_k^q, we then define the context kernel of attributes a between x_p and x_q as

$$\boldsymbol{\kappa_{con}}[(x_p, a_p), (x_q, a_q)] = \boldsymbol{\kappa_a}(\bar{a}_p, \bar{a}_q)$$
$$\bar{a}_p = \frac{1}{\left|N_k^p\right|} \sum_{x_u \in N_k^p} a_u m_u \exp\left(-\frac{8\left\|x_u - x_p\right\|^2}{k^2}\right), \bar{a}_q = \frac{1}{\left|N_k^q\right|} \sum_{x_v \in N_k^q} a_v m_v \exp\left(-\frac{8\left\|x_v - x_q\right\|^2}{k^2}\right) \tag{1}$$

where m_u and m_v are the gradient magnitudes at pixels x_u and x_v, respectively; $\bar{a}_p$ and $\bar{a}_q$ are the normalized image attributes in k-neighborhood centered at x_p and x_q, respectively; $\boldsymbol{\kappa_a}(\bar{a}_p, \bar{a}_q) = exp(-\gamma_a||\bar{a}_p - \bar{a}_q||^2) = \varphi_a(\bar{a}_p)^{\mathrm{T}} \varphi_a(\bar{a}_q)$ is the Gaussian kernel measuring the similarity of normalized attributes $\bar{a}_p$ and $\bar{a}_q$. The context kernel $\boldsymbol{\kappa_{con}}$ provides a normalized measure of the attribute similarity between two k-neighborhoods centered at pixels x_p and x_q. Merging $\boldsymbol{\kappa_{con}}$ into match kernels [1] and replacing the attribute a in Eq. (1) with specific attributes, we can derive a set of ad hoc attribute based CMKs.

For example, let θ'_p and m'_p be normalized orientation and normalized magnitude of the image gradient at pixel x_p, such that θ'_p = (sinθ_p, cosθ_p) and $m'_p = m_p \Big/ \sqrt{\sum_{p \in P} m_p^2 + \tau}$, with τ being a small positive number. To compare the similarity of gradients between patches P and Q from two different images, the gradient CMK $\boldsymbol{K_{gck}}$ can be defined as

$$\boldsymbol{K_{gck}}(P, Q) = \sum_{p \in P} \sum_{q \in Q} m'_p m'_q \boldsymbol{\kappa_o}(\theta'_p, \theta'_q) \boldsymbol{\kappa_s}(x_p, x_q) \boldsymbol{\kappa_{con}}[(x_p, \theta'_p), (x_q, \theta'_q)] \tag{2}$$

where $\boldsymbol{\kappa_o}(\theta'_p, \theta'_q) = \exp(-\gamma_o \|\theta'_p - \theta'_q\|^2) = \varphi_o(\theta'_p)^{\mathrm{T}} \varphi_o(\theta'_q)$ is the orientation kernel measuring the similarity of normalized orientations at two pixels x_p and x_q; $\boldsymbol{\kappa_s}(x_p, x_q) = \exp(-\gamma_s \|x_p - x_q\|^2) = \varphi_s(x_p)^{\mathrm{T}} \varphi_s(x_q)$ is the spatial kernel measuring how close two pixels are spatially; and $\boldsymbol{\kappa_{con}}[(x_p, \theta'_p), (x_q, \theta'_q)]$ is given by Eq. (1).

Similarly, to measure the similarity of color attributes between P and Q, the color CMK $\boldsymbol{K_{cck}}$ can be defined as

$$\boldsymbol{K_{cck}}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \boldsymbol{\kappa_c}(c_p, c_q) \boldsymbol{\kappa_s}(x_p, x_q) \boldsymbol{\kappa_{con}}[(x_p, c_p), (x_q, c_q)] \tag{3}$$

where $\boldsymbol{\kappa_c}(c_p, c_q) = \exp(-\gamma_c \parallel c_p - c_q \parallel^2) = \varphi_c(c_p)^{\mathrm{T}} \varphi_c(c_q)$ is the color kernel measuring the similarity of color values c_p and c_q. For color images, we use normalized *rgb* vector as color value, whereas intensity value is used for grayscale images.

For the texture attribute, we derive the texture CMK, $\boldsymbol{K_{lbpck}}$, based on Local Binary Patterns (*lbp*) [10]

$$\boldsymbol{K_{lbpck}}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \sigma'_p \sigma'_q \boldsymbol{\kappa_{lbp}}(lbp_p, lbp_q) \boldsymbol{\kappa_s}(x_p, x_q) \boldsymbol{\kappa_{con}}[(x_p, lbp_p), (x_q, lbp_q)] \quad (4)$$

where $\sigma'_p = \sigma_p \Big/ \sqrt{\sum_{p \in N_3} \sigma_p^2 + \tau}$ is the normalized standard deviation of pixel values within a 3 × 3 window around x_p; $\boldsymbol{\kappa_{lbp}}(lbp_p, lbp_q) = \exp(-\gamma_{lbp} \| lbp_p - lbp_q \|^2)$ is a Gaussian match kernel for *lbp* operator.

As shown in Eqs. (2)-(4), each attribute based CMK consists of four terms: (1) normalized linear kernel, e.g. $m'_p\, m'_q$ for $\boldsymbol{K_{gck}}$; 1 for $\boldsymbol{K_{cck}}$ and $\sigma'_p\, \sigma'_q$ for $\boldsymbol{K_{lbpck}}$, weighting the contribution of each pixel to the final attribute based CMK; (2) attribute kernel evaluating the similarity of pixel attributes; (3) spatial kernel $\boldsymbol{\kappa_s}$ measuring the relative distance between two pixels; (4) context kernel $\boldsymbol{\kappa_{con}}$ comparing the spatial co-occurrence of pixel attributes. In this sense, we formulate these attribute CMKs, defined in Eqs. (2)-(4), in a unified way as

$$\boldsymbol{K}(P, Q) = \sum_{p \in P} \sum_{q \in Q} w_p w_q \boldsymbol{\kappa_a}(a_p, a_q) \boldsymbol{\kappa_s}(x_p, x_q) \boldsymbol{\kappa_{con}}[(x_p, a_p), (x_q, a_q)] \quad (5)$$

where $w_p w_q$ and $\boldsymbol{\kappa_a}$ correspond to normalized linear weighting kernel and attribute kernel, respectively.

2.2 Approximation of CMK

Using the inner product representation, we rewrite the match kernel matrix $\boldsymbol{K}$ as

$$\boldsymbol{K}(P, Q) = \langle \boldsymbol{\psi}(Q), \boldsymbol{\psi}(P) \rangle = \boldsymbol{\psi}(P)^{\mathrm{T}} \boldsymbol{\psi}(Q)$$
$$\boldsymbol{\psi}(P) = \sum_{p \in P} w_p \varphi_a(a_p) \otimes \varphi_s(x_p) \otimes \varphi_{con}(x_p, a_p),\ \boldsymbol{\psi}(Q) = \sum_{q \in Q} w_q \varphi_a(a_q) \otimes \varphi_s(x_q) \otimes \varphi_{con}(x_q, a_q) \quad (6)$$

where $\otimes$ is the tensor product and $\boldsymbol{\psi}(\cdot)$ gives the mapping features in kernel space, namely the CKD. Note that the dimensions of φ_a, φ_s and φ_{con} are all infinite, since Gaussian kernel is used. To obtain an accurate approximation of $\boldsymbol{K}$, we have to uniformly sample φ_a, φ_s and φ_{con} using a dense grid along sufficient basis vectors. In particular, for φ_a and φ_{con}, we discretize a into G bins and approximate them with their projections onto the subspaces spanned by G basis vectors $\{\varphi_a(a^g)\}$ ($g = 1 \ldots G$). Similarly, for space vector x, we discretize spatial basis vectors into L bins and sample along L basis vectors spatially. Finally, we can approximate $\boldsymbol{\psi}(\cdot)$ by its projections onto

the $G \times L \times G$ joint basis vectors: $\{\phi_l\}_{l=1}^{G\times L\times G} = \{\varphi_a(a^1) \otimes \varphi_s(x^1) \otimes \varphi_{con}(a^1), \ldots, \varphi_a(a^G) \otimes \varphi_s(x^L) \otimes \varphi_{con}(a^G)\}$.

$$\boldsymbol{\psi}(\cdot) \simeq \sum_{l=1}^{G\times L\times G} f_l \phi_l \tag{7}$$

where f_l is the projection coefficient onto the lth joint basis vector ϕ_l. Thus, dimensionality of the resulting CKD $\boldsymbol{\psi}$ is $G \times L \times G$. Uniform sampling provides a set of representative joint basis vectors, but does not guarantee their compactness. Projecting onto these basis vectors usually yield a group of redundant CKD. Next, we show how to learn a CKD codebook by selecting and refining a subset of compact and discriminative joint basis vectors using a CSQMI based information theoretic feature learning scheme. Projecting the original CKD $\boldsymbol{\psi}$ onto the codebook reduces the redundancy of $\boldsymbol{\psi}$ and gives a low-dimensional discriminative CKD representation.

3 Feature Learning Using CSQMI

Shannon entropy and its related measures, such as mutual information and Kullback-Leibler divergence (KLD) are widely used in feature learning [16–26]. However, Shannon entropy based feature learning methods share the common weakness of high evaluation complexity involved in the estimation of probability density function (*pdf*) in Shannon entropy [16]. Recently, Rényi entropy [27, 28] has attracted more attentions in information theoretic learning. The most impressive advantage of Rényi entropy is its moderate computational complexity because the estimate of Rényi entropy can be efficiently implemented by the kernel density estimation [29] (e.g. the Parzen windowing). Several novel information theoretic metrics derived from Rényi entropy are introduced in feature learning [30–33].

3.1 Rényi Entropy and CSQMI

Let $\boldsymbol{S} \in \mathscr{R}^d$ be a discrete random variable which has a *pdf* of $p(\boldsymbol{s})$, then its Rényi entropy is defined as [27]

$$H_\alpha(\boldsymbol{S}) = \frac{1}{1-\alpha} \log_2 \sum_{\boldsymbol{s}\in\boldsymbol{S}} p^\alpha(\boldsymbol{s}) \tag{8}$$

Rényi entropy defines a family of functions that quantify the diversity in a data distribution. Standard Shannon entropy can be treated as a special case of Rényi entropy as $\alpha \to 1$. Rényi entropy of order $\alpha = 2$, given in Eq. (9), is called Rényi quadratic entropy $H_2(\boldsymbol{S})$.

$$H_2(\boldsymbol{S}) = -\log_2 \sum_{\boldsymbol{s}\in\boldsymbol{S}} p^2(\boldsymbol{s}) \tag{9}$$

Similar to KLD defined using Shannon entropy, Cauchy-Schwarz divergence (CSD) based on Rényi quadratic entropy also defines a measure of divergence between different *pdfs*. Given two discrete random variables $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$, with $\boldsymbol{S_1}$ having a *pdf* of $p_1(\boldsymbol{s_1})$ and $\boldsymbol{S_2}$ having a *pdf* of $p_2(\boldsymbol{s_2})$, the CSD [28, 31] of p_1 and p_2 is given by

$$CSD(p_1;p_2) = -\log_2 \frac{\left(\sum\limits_{\boldsymbol{s_1}\in\boldsymbol{S_1},\boldsymbol{s_2}\in\boldsymbol{S_2}} p_1(\boldsymbol{s_1})p_2(\boldsymbol{s_2})\right)^2}{\sum\limits_{\boldsymbol{s_1}\in\boldsymbol{S_1}} p_1^2(\boldsymbol{s_1}) \sum\limits_{\boldsymbol{s_2}\in\boldsymbol{S_2}} p_2^2(\boldsymbol{s_2})} = 2H_2(\boldsymbol{S_1}, \boldsymbol{S_2}) - H_2(\boldsymbol{S_1}) - H_2(\boldsymbol{S_2}) \tag{10}$$

where $H_2(\boldsymbol{S_1}, \boldsymbol{S_2}) = -\log_2 \sum\limits_{\boldsymbol{s_1}\in\boldsymbol{S_1},\boldsymbol{s_2}\in\boldsymbol{S_2}} p_1(\boldsymbol{s_1})p_2(\boldsymbol{s_2})$ measures the similarity (distance) between the two *pdfs* and can be considered as the Rényi quadratic cross entropy. We can interpret $H_2(\boldsymbol{S_1}, \boldsymbol{S_2})$ as the information gain from observing p_2 with respect to the "true" density p_1, and vice versa. Hence, the CSD derived from Rényi quadratic entropy is semantically similar to Shannon's mutual information. Note that $CSD(p_1; p_2) \geq 0$ is a symmetric measure that equals zero if and only if $p_1(\boldsymbol{s}) = p_2(\boldsymbol{s})$, and increases towards positive infinity as the two *pdf*s are apart further and further. Based on $CSD(p_1; p_2)$, the Cauchy-Schwarz Quadratic Mutual Information between two discrete random variables $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ is defined as [28].

$$\begin{aligned} I_{CSD}(\boldsymbol{S_1};\boldsymbol{S_2}) &= CSD(p_{12}(\boldsymbol{s_1},\boldsymbol{s_2}); p_1(\boldsymbol{s_1})p_2(\boldsymbol{s_2})) \\ &= log_2 \sum_{\boldsymbol{s_1}\in\boldsymbol{S_1}\boldsymbol{s_2}\in\boldsymbol{S_2}} p_{12}^2(\boldsymbol{s_1},\boldsymbol{s_2}) + log_2 \sum_{\boldsymbol{s_1}\in\boldsymbol{S_1}\boldsymbol{s_2}\in\boldsymbol{S_2}} p_1^2(\boldsymbol{s_1})p_2^2(\boldsymbol{s_2}) - 2log_2 \sum_{\boldsymbol{s_1}\in\boldsymbol{S_1}\boldsymbol{s_2}\in\boldsymbol{S_2}} p_{12}(\boldsymbol{s_1},\boldsymbol{s_2})p_1(\boldsymbol{s_1})p_2(\boldsymbol{s_2}) \end{aligned} \tag{11}$$

where $p_{12}(\boldsymbol{s_1}, \boldsymbol{s_2})$ is the joint *pdf* of $(\boldsymbol{S_1}, \boldsymbol{S_2})$, and $p_1(\boldsymbol{s_1})$ and $p_2(\boldsymbol{s_2})$ are marginal *pdf* of $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$, respectively. $I_{CSD}(\boldsymbol{S_1}; \boldsymbol{S_2}) \geq 0$ meets the equality if and only if $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ are independent. So $I_{CSD}(\boldsymbol{S_1}; \boldsymbol{S_2})$ is a measure of independence that reflects the information shared between $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$. In other words, it measures how much knowing $\boldsymbol{S_1}$ reduces the uncertainty about $\boldsymbol{S_2}$, and vice versa.

To calculate CSD and I_{CSD}, we have to estimate marginal *pdf* $p(\bullet)$ and joint *pdf* $p_{12}(\bullet,\bullet)$. Fortunately, Principe [28] showed that, for Rényi quadratic entropy and its induced measures such as CSD and I_{CSD}, these marginal and joint *pdfs* can be efficiently estimated with a Parzen window density estimator [29], even in a high-dimensional feature space like CDK. Whereas, it is not possible for Shannon entropy [28]. This explains why we choose the Rényi quadratic entropy based I_{CSD}, instead of the Shannon entropy based mutual information, as information theoretic measure in our codebook learning algorithm.

In addition, recent findings from Jenssen et al. [30, 31] uncovered the latent connections between Rényi quadratic entropy and mapping features in the kernel space. It shows that, when applying a Gaussian Parzen window estimator, Rényi quadratic entropy estimator is equivalent to $\|\boldsymbol{m}\|^2$, where $\boldsymbol{m} = \frac{1}{M}\sum\limits_{\boldsymbol{s_t}\in\boldsymbol{S}} \varphi(\boldsymbol{s_t})$ is the mean vector of mapping data samples $\varphi(\boldsymbol{s_t})$ ($t = 1,\cdots,M$) in the kernel feature space. Meanwhile, the

CSD estimator is directly associated with the angle between the mean vectors $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ of the clusters of mapping data samples in the kernel feature space. These clusters correspond to the mapping data samples yielded from $p_1(\boldsymbol{s})$ and $p_2(\boldsymbol{s})$, respectively. Consequently, CSQMI, measuring the CSD between a joint *pdf* and the product of two marginal *pdf*s, also relates to the cluster structure in the kernel feature space. The relationships between Rényi quadratic entropy, CSD/CSQMI and the mean vector of mapping features in the kernel space provide us the geometric interpretation behind $H_2(\boldsymbol{S})$ and CSD/CSQMI. It means that the Rényi quadratic entropy based measures are very suitable for the analysis of nonlinear data (even in high-dimensional spaces) because they are able to capture the geometric structure of the data. In contrast, the Shannon entropy and the KLD do not have such good properties.

3.2 Codebook Selection and Refinement Using CSQMI

As mentioned in Sect. 2.2, we approximate the original CKD $\boldsymbol{\psi}$ with a group of redundant joint basis vectors $\{\phi_l\}_{l=1}^{G\times L\times G}$.We define these joint basis vectors as dictionary, and represent it as $\boldsymbol{\Phi}$ ($\boldsymbol{\Phi}$ has a cardinality of $G \times L\times G$). Assuming that we are given CKD, $\boldsymbol{\psi}^1,\cdots, \boldsymbol{\psi}^M$, of M samples from C classes, for each class c ($c = 1,\cdots,C$), it has M_c samples and the corresponding CKD are denoted as $\boldsymbol{\Psi}_c = [\boldsymbol{\psi}_c^1,\cdots, \boldsymbol{\psi}_c^{Mc}]$. Then we formulate the CKD of all samples as $\boldsymbol{\Psi} = \{\boldsymbol{\Psi}_c\}_{c=1}^{C}$. Similarly, we denote $\boldsymbol{F} = \{\boldsymbol{F}_c\}_{c=1}^{C}$, where $\boldsymbol{F}_c = [\boldsymbol{F}_c^1,\cdots, \boldsymbol{F}_c^{Mc}] = \left[(f_{c1}^1,\cdots,f_{cG\times L\times G}^1)^{\mathrm{T}},\cdots,(f_{c1}^{M_c},\cdots,f_{cG\times L\times G}^{M_c})^{\mathrm{T}}\right]$. Then, Eq. (7) can be represented as $\boldsymbol{\Psi} = \boldsymbol{\Phi F}$, where $\boldsymbol{\Phi} = [\phi_1,\cdots, \phi_{G\times L\times G}]$ and $\boldsymbol{F} = \begin{bmatrix} f_{11}^1 & \cdots & f_{C1}^{M_C} \\ \vdots & & \vdots \\ f_{1G\times L\times G}^1 & \cdots & f_{CG\times L\times G}^{M_C} \end{bmatrix}$ is the projection coefficients matrix. Given a CKD $\boldsymbol{\psi}$ from a random sample, the uncertainty of its class label L in terms of the class prior probabilities can be measured by $H_2(L)$, given in Eq. (9). Whereas, the CSQMI $I_{CSD}(\boldsymbol{\psi}; L)$ defined in Eq. (11) measures the decrease in uncertainty of the pattern $\boldsymbol{\psi}$ due to the knowledge of the underlying class label L.

Given $\boldsymbol{\Psi}$ and an initial dictionary $\boldsymbol{\Phi}$, we aim to learn a compact and discriminative subset of joint basis vectors $\boldsymbol{\Phi}^*$ out of $\boldsymbol{\Phi}$, such that *cardinality* ($\boldsymbol{\Phi}^*$) < *cardinality* ($\boldsymbol{\Phi}$). We refer to $\boldsymbol{\Phi}^*$ as codebook. Projecting the original CKD $\boldsymbol{\Psi}$ onto the codebook $\boldsymbol{\Phi}^*$ gives a low-dimensional CKD, $\boldsymbol{\Psi}^* = \boldsymbol{\Phi}^* \boldsymbol{F}^*$. We expect $\boldsymbol{\Psi}^*$ should be compact and discriminative. To learn a compact codebook, we maximize the CSQMI between $\boldsymbol{\Phi}^*$ and the unselected basis vectors $\boldsymbol{\Phi} - \boldsymbol{\Phi}^*$ in $\boldsymbol{\Phi}$, i.e. $I_{CSD}(\boldsymbol{\Phi}^*; \boldsymbol{\Phi} - \boldsymbol{\Phi}^*)$. As $I_{CSD}(\boldsymbol{\Phi}^*; \boldsymbol{\Phi} - \boldsymbol{\Phi}^*)$ signifies how compact the codebook $\boldsymbol{\Phi}^*$is, a higher value of $I_{CSD}(\boldsymbol{\Phi}^*; \boldsymbol{\Phi} - \boldsymbol{\Phi}^*)$ means a more compact codebook. However, that codebook may not be discriminative, because it does not give any information regarding the new CKD $\boldsymbol{\Psi}^*$ from their class label L. Therefore, we also need to maximize the CSQMI between $\boldsymbol{\Psi}^*$ and L, i.e. $I_{CSD}(\boldsymbol{\Psi}^*; L)$, which provides the discriminability of the new CKD generated from the codebook $\boldsymbol{\Phi}^*$. To this end, the codebook learning problem can be mathematically formulated as

$$\arg\max_{\mathbf{\Phi}^*}[I_{CSD}(\mathbf{\Phi}^*;\mathbf{\Phi}-\mathbf{\Phi}^*)+\lambda I_{CSD}(\mathbf{\Psi}^*;L)] \tag{12}$$

where λ is the weight parameter to make a tradeoff between the compactness and discriminability terms. We use a two-step strategy to optimize the compactness and discriminability of the codebook simultaneously. In the first step (*Codebook Selection*), the codebook that maximizes Eq. (12) is selected from the initial dictionary in a greedy search manner. In the second step (*Codebook Refinement*), the selected codebook is refined via a gradient ascent method to further maximize the discriminability term $I_{CSD}(\mathbf{\Psi}^*;L)$ while keeping the approximation error as low as possible.

3.2.1 Codebook Selection

The first term in Eq. (12), i.e. $I_{CSD}(\mathbf{\Phi}^*;\mathbf{\Phi}-\mathbf{\Phi}^*)$, is a compactness term which measures the compactness of the codebook $\mathbf{\Phi}^*$. The second term, i.e. $I_{CSD}(\mathbf{\Psi}^*;L)$, measures the discriminability of the codebook $\mathbf{\Phi}^*$. Based on [34], the probability of Bayes classification error resulted from the final CKD $\mathbf{\Psi}^*$, i.e. $P(e^{\mathbf{\Psi}^*})$, has its upper bound given by $P(e^{\mathbf{\Psi}^*}) \le \frac{1}{2}(H_2(L)-I_{CSD}(\mathbf{\Psi}^*;L))$. Thus, the selected discriminative codebook $\mathbf{\Phi}^*$ corresponding to the minimal Bayes classification error bound should maximize the $I_{CSD}(\mathbf{\Psi}^*;L)$.

During the codebook selection, we start with an empty set of $\mathbf{\Phi}^*$ and iteratively select the next best basis vector ϕ^* out of the remaining set $\mathbf{\Phi}-\mathbf{\Phi}^*$, such that the mutual information gain between the new codebook $\mathbf{\Phi}^* \cup \phi^*$ and the remaining set, as well as the mutual information gain between the CKD derived from the new codebook and the class label, are maximized, i.e.

$$\arg\max_{\phi^*\in\mathbf{\Phi}-\mathbf{\Phi}^*}\left\{[I_{CSD}(\mathbf{\Phi}^*\cup\phi^*;\mathbf{\Phi}-(\mathbf{\Phi}^*\cup\phi^*))-I_{CSD}(\mathbf{\Phi}^*;\mathbf{\Phi}-\mathbf{\Phi}^*)]+\left[I_{CSD}(\mathbf{\Psi}^{\mathbf{\Phi}^*\cup\phi^*};L)-I_{CSD}(\mathbf{\Psi}^{\mathbf{\Phi}^*};L)\right]\right\} \tag{13}$$

3.2.2 Codebook Refinement

Once the initial codebook $\mathbf{\Phi}^*$ is achieved, we refine $\mathbf{\Phi}^*$ to further enhance its discriminability by maximizing the discriminability term in Eq. (12), i.e. $\max_{\mathbf{\Phi}^*}\lambda I_{CSD}(\mathbf{\Psi}^*;L)$. To guarantee a compact codebook, we assume that *cardinality* $(\mathbf{\Phi}^*) \ll$ *cardinality* $(\mathbf{\Phi})$. Under such an assumption, the projection coefficient is solved by $\boldsymbol{F}^* = \mathbf{\Phi}^{\dagger}\mathbf{\Psi}$ which minimizes the approximation error $\boldsymbol{e} = \| \mathbf{\Psi}-\mathbf{\Phi}^*\boldsymbol{F}^* \|^2$, where $\mathbf{\Phi}^{\dagger} = pinv(\mathbf{\Phi}^*) = \left(\mathbf{\Phi}^{*\mathbf{T}}\mathbf{\Phi}^*\right)^{-1}\mathbf{\Phi}^{*\mathbf{T}}$ is the pseudo-inverse of $\mathbf{\Phi}^*$. Thus, the problem of refining $\mathbf{\Phi}^*$ for improving the discriminability of codebook while keeping its approximation accuracy is converted to the following constraint optimization problem.

$$\max_{\mathbf{\Phi}^*} I_{CSD}(\mathbf{\Psi}^*;L), \text{ subject to } \boldsymbol{F}^* = \mathbf{\Phi}^{\dagger}\mathbf{\Psi} \tag{14}$$

Since $I_{CSD}(\bullet;\bullet)$ is a quadratic symmetric measure, the objective function $I_{CSD}(\mathbf{\Psi}^*;L)$ is differentiable. We use the gradient ascend method to iteratively refine $\mathbf{\Phi}^*$ such that $I_{CSD}(\mathbf{\Psi}^*;L)$ is maximized. In each iteration, $\mathbf{\Phi}^*$ is updated with a step size υ. After k-th iteration, $\mathbf{\Phi}_k^*$ becomes

$$\begin{aligned}\boldsymbol{\Phi}_k^* &= \boldsymbol{\Phi}_{k-1}^* + \upsilon \frac{\partial I_{CSD}(\boldsymbol{\Psi}^*; L)}{\partial \boldsymbol{\Phi}^*}\Big|_{\boldsymbol{\Phi}^*=\boldsymbol{\Phi}_{k-1}^*} \\ \frac{\partial I_{CSD}(\boldsymbol{\Psi}^*; L)}{\partial \boldsymbol{\Phi}^*} &= \sum_{c=1}^{C}\sum_{i=1}^{M_c} \frac{\partial I_{CSD}(\boldsymbol{\Psi}^*; L)}{\partial \boldsymbol{\psi}_c^{*i}} \frac{\partial \boldsymbol{\psi}_c^{*i}}{\partial \boldsymbol{\Phi}^*} = \sum_{c=1}^{C}\sum_{i=1}^{M_c} \left(F_c^i\right)^T \frac{\partial I_{CSD}(\boldsymbol{\Psi}^*; L)}{\partial \boldsymbol{\psi}_c^{*i}}\end{aligned} \tag{15}$$

Once $\boldsymbol{\Phi}^*$ is refined, we update the projection coefficients $\boldsymbol{F}^*$ and the low-dimensional discriminative CKD $\boldsymbol{\Psi}^*$ according to $\boldsymbol{F}^* = \boldsymbol{\Phi}^{\dagger}\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^* = \boldsymbol{\Phi}^*\boldsymbol{F}^*$, respectively. The bound of $I_{CSD}(\boldsymbol{\Psi}^*; L)$ guarantees the convergence of codebook refinement.

4 Experiments

To verify the effectiveness of our method in the context of object recognition and detection, we first investigate the performance of CSQMI based codebook learning on the extended YaleB face dataset [35], then we test our model on Caltech-101 [36] and CIFAR-10 [37] for recognition and on our own chicken feet dataset for detection. We also compare our results with other state-of-the-art works, including the original KDES [1], supervised kernel descriptors [4], handcrafted dense SIFT features [7, 8], and the popular deep feature learning approaches [44, 51–53].

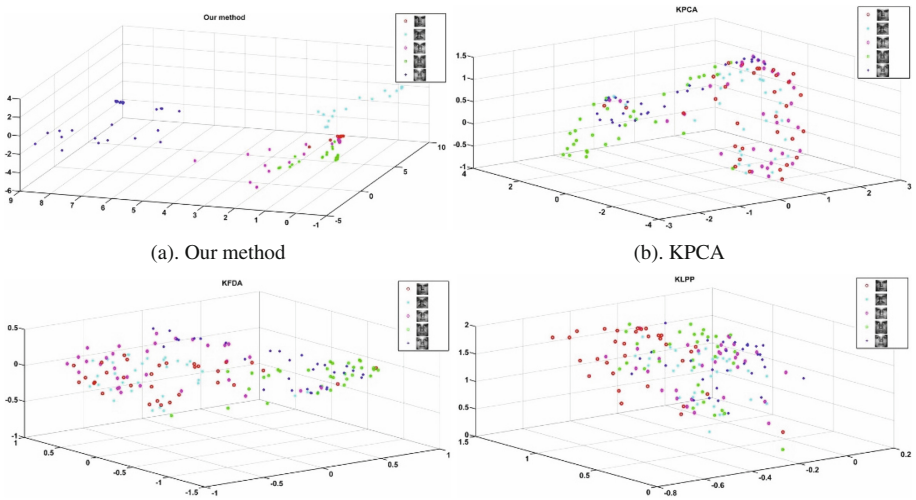
4.1 Parameter Configuration

We adopt the code provided from www.cs.washington.edu/robotics/projects/kdes/ to implement the original KDES. To make a fair comparison, in all experiments, except for the final feature dimensionality, we follow the setting of [1] for common parameters used in our method. Namely, basis vectors for $\boldsymbol{\kappa_o}$, $\boldsymbol{\kappa_c}$, and $\boldsymbol{\kappa_s}$ are sampled using 25, 5 × 5 × 5, and 5 × 5 uniform grids, respectively. For $\boldsymbol{\kappa_{lbp}}$, we choose all 256 basis vectors. For all CKD, $\boldsymbol{\kappa_{con}}$ shares the same basis vectors with their attribute kernels $\boldsymbol{\kappa_a}$. We use a 3-level spatial pyramid for pooling CKD at different levels. The pyramid level is set as 1 × 1, 2 × 2 and 4 × 4. Gaussian Parzen windows are used to estimate the CSQMI, and the width parameter σ is tuned using a grid search in the range $[0.01\sigma_d, 100\sigma_d]$, where σ_d is the median distance of all training samples. The best window width is selected by cross-validation. The optimal neighborhood distance parameter, k, is decided using a grid search between 1 and 8. Linear SVM classifiers used in all experiments are implemented with the LIBlinear, downloaded from www.csie.ntu.edu.tw/∼cjlin/liblinear/.

4.2 Evaluation of Codebook Learning

We first evaluate the discriminability of our CSQMI based codebook learning method by comparing it with other popular kernel based dimensionality reduction methods on the extended YaleB face dataset [35] that contains 16128 face images from 28 individuals. This dataset is challenging due to varying illumination conditions and expressions.

For each individual, half of the frontal face images are used to train the relevant codebook and feature subset. The remaining frontal face samples are used to test the distinctiveness of the learned codebook. *LBP_CKD* is applied to extract the face features. KPCA [14, 15], Kernel Fisher Discriminant Analysis (KFDA) [38], and Kernel Locality Preserving Projections (KLPP) [39] are compared with our codebook learning method. For each method, as suggested in [1], a reduced 200-dimensional feature subset is learned. To visualize the results, we randomly select five subjects and plot the distributions of projected samples onto the leading three most significant feature subsets yielded from each method in Fig. 1. As shown in Fig. 1, the clusters of the face samples resulted from our codebook represents a significant improvement on the class separation over that obtained from the alternative kernel based dimensionality reduction methods. This is because that the feature subset derived from CSQMI captures the angular pattern of the cluster distribution of the analyzing face patterns. Consequently, it is more discriminative than the feature subset selected from principal component vectors based only on magnitude of eigenvalues, such as KPCA.



(a). Our method (b). KPCA

(c). KFDA (d). KLPP

Fig. 1. Visualization of the leading 3-dimensional *LBP_CKD* features from different methods.

4.3 Evaluation of Object Recognition

Caltech-101: This dataset is one of the most popular benchmarks for multiclass image recognition. It collects 9144 images from 101 object categories and a background category. Each category has 31 to 800 images with significant color, pose and lighting variations. We use this dataset for a comprehensive comparison on the recognition performance of the original KDES, supervised kernel descriptor (SKDES) [4] and our CKD. A 4-neighborhood which achieves the best performance is used to evaluate the context information for CKD. For each category, following the experimental setup of

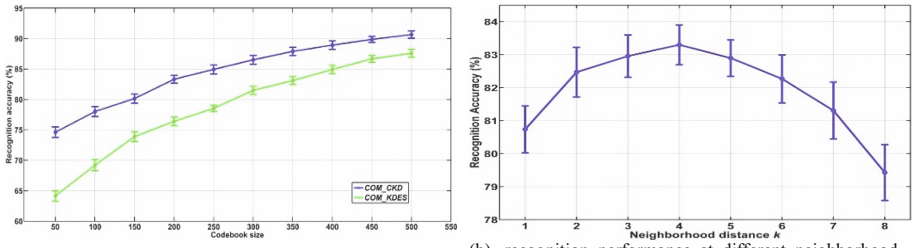
Table 1. Comparison of mean recognition accuracy (%) and standard deviation of KDES, SKDES and CKD on Caltech-101.

Features	KDES[1]	SKDES [4]	CKD
gradient	75.2±0.4	77.3±0.7	**77.8**±0.6
color	42.4±0.5	68.4±1.4	**69.1**±0.9
texture(*lbp*)	70.3±0.6	71.6±1.3	**74.3**±0.8
combination	76.4±0.7	79.2±0.6	**83.3**±0.6
Method	Accuracy	Method	Accuracy
Jia et al. [40]	75.3±0.7	Feng et al. [45]	82.60
SLC [47]	81±0.2	SDL [41]	75.3±0.4
Adaptive deconvolutional net [44]	71.0±1.0	SSC [46]	80.02±0.36
Boureau et al. [42]	77.3±0.6	M-HMP [48]	82.5±0.5
LSAQ [43]	74.21±0.8	SPM_SIFT [7]	64.6±0.8
Pyramid SIFT (P-SIFT) [49]	80.13	PHOW [50]	81.3±0.8

original KDES [1], we train one-vs-all linear SVM classifiers on 30 images and test on no more than 80 images for KDES and our method. We run five rounds of testing for a confident evaluation. Results of SKDES are quoted from the original papers. Table 1 lists the average recognition accuracy and standard deviation of different options of kernel descriptors. Some recently reported results are also provided for comparison.

From Table 1, we observe that our CKD consistently outperforms KDES and SKDES, for both individual and combined version. Except for the gradient CKD (*G_CKD*), both color CKD (*C_CKD*) and texture CKD (*LBP_CKD*) are significantly better than their original KDES. In particular, compared with the original color and texture KDES, the recognition accuracy of *C_CKD* and *LBP_CKD* is increased by 62.97 % and 5.69 %, respectively. For the combined version, the accuracy of combined *CKD* is 83.3 %, which is 6.9 % higher than the original KDES combination and 4.1 % higher than the SKDES combination. We also notice the smaller standard deviation of recognition accuracy in our results compared with that of the SKDES. It means CKD is more robust than SKDES, thanks to the spatial co-occurrence constraints embedded in the CKD. We argue that the performance improvement of CKD comes from two facts: (1) compared with KDES and SKDES, the additional spatial co-occurrence constraint defined in CKD further improves its robustness to the semantic ambiguity, caused by the lack of features in case of partial occlusion; (2) KDES applies KPCA to reduce feature dimensionality, whereas we use CSQMI to learn low-dimensional CKD. KPCA only keeps KDES components that contribute most significantly to image reconstruction. In contrast, our CSQMI criterion selects the CKD that minimize the information redundancy and approximation error while maximize the mutual information between the CKD and its class label in terms of the 'angle distance'. Therefore, the resulting low-dimensional CKD are more discriminative than KDES in that they reveal the cluster structure of density distribution of pixel attributes and relate to the angular manifold of the object category.

To investigate the impact of codebook size on the recognition performance, we train classifiers using different codebook sizes and compare the recognition accuracy of



(a). recognition performance at different codebook sizes

(b). recognition performance at different neighborhood distances

Fig. 2. Performance comparison at different codebook sizes and neighborhood distances on Caltech-101.

the combined CKD (*COM_CKD*) with that of the combined KDES (*COM_KDES*) in Fig. 2(a). As expected, *COM_CKD* outperforms *COM_KDES* consistently over all codebook sizes. We also note a relative small performance drop (14 %) of *COM_CKD* when codebook size decreases from 500 to 50, whereas for *COM_KDES* the accuracy drop is 26 %. This verifies the effectiveness of our codebook learning model, which can select discriminative CKD codebook even in low-dimensional situations. We also compare the recognition performance of CKD yielded under different neighborhood distances. As shown in Fig. 2(b), neighborhoods with moderate distances perform better than neighborhoods with small distances, and recognition accuracy tends to decrease for neighborhoods with large distances. This can be understood by the fact that the discriminability of CKD tends to be smoothed, as more noises and outlier data may be included when the neighborhood distance becomes larger.

CIFAR-10: This dataset consists of 60000 tiny images with a size of 32 × 32 pixels. It has 10 categories, with 5000 training images and 1000 test images per category. We choose this dataset to test the performance of our method on recognition of tiny objects. Similar to [1], we calculate CKD around 8 × 8 image patches on a dense grid with a spacing of 2 pixels. A 3-neighborhood which gives the best performance is applied to calculate CKD. The whole training images are split into 10,000/40,000 training/validation set, and the validation set is used to optimize the kernel parameters of γ_s, γ_o, γ_c, and γ_{lbp} using a grid search. Finally, a linear SVM classifier is trained on the whole training set using the optimized kernel parameters.

We compare the performance of *COM_CKD* with several recent feature learning approaches using deep learning (stochastic pooling based Deep Convolutional Neural Network − spDCNN [52], tiled Convolutional Neural Networks − tCNN [53], Multi-column Deep Neural Networks − MDNN [51]), sparse coding (improved local Coordinate Coding − iLCC [54], spike-and-slab Sparse Coding − ssSC [55]), hierarchical kernel descriptor (HKDES) [2] and spatial pyramid dense SIFT (SPM_SIFT) [7]. For SPM_SIFT, we use a 3-layer spatial pyramid structure and calculate dense SIFT feature in an 8 × 8 patch over a regular grid with a spacing of 2 pixels. Table 2 reports the recognition accuracy of various methods. As we see, *COM_CKD* and MDNN defeat other methods by a large margin. Compared with MDNN, *COM_CKD* achieves a comparable performance with only a 0.37 % deficit in classification rate. However,

Table 2. Comparison of recognition accuracy (%) of various methods on CIFAR-10.

Method	Accuracy	Method	Accuracy
spDCNN [52]	84.88	SPM_SIFT [7]	65.60
tCNN [53]	73.10	HKDES [2]	80.00
iLCC [54]	74.50	MDNN [51]	**88.79**
ssSC [55]	78.80	*COM_CKD*	88.42

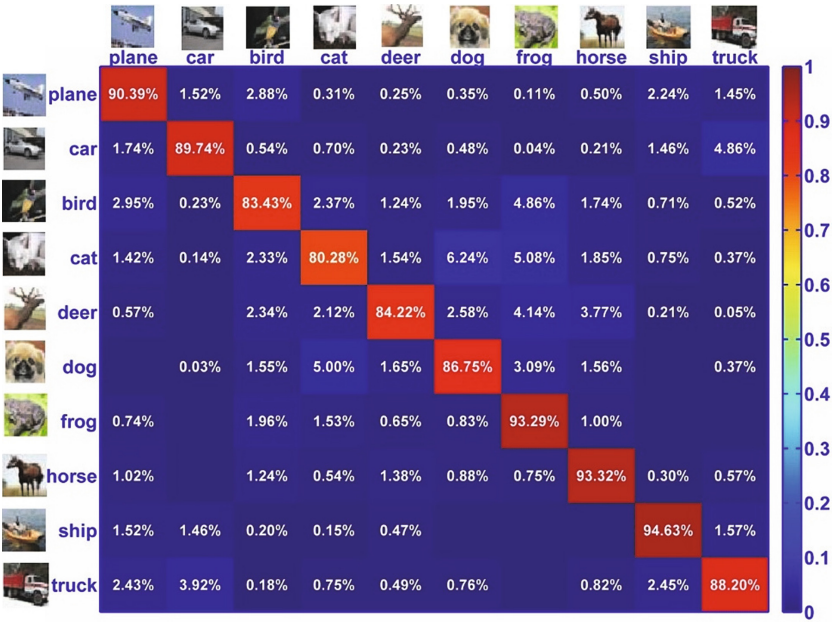


Fig. 3. Confusion matrix for CIFAR-10 using *COM_CKD*. Vertical axis shows the ground truth labels and the predicted labels go along the horizontal axis.

our method is much more simple and efficient than MDNN model. For example, for a 32 × 32 pixel image, our method takes 224.63 ms to calculate the full-dimensional 3-neighborhood *COM_CKD* and 320.21 s to learn a 200-dimensional discriminative codebook using CSQMI on average on a platform with Intel Core i7 2.7 GHz CPU and 16G RAM. Merging different pixel attributes in the kernel space, CKD tune low-level complementary cues into image-level discriminative descriptors. Even coupled with simple linear SVM classifier, our method still achieves superior performance compared with other sophisticated models.

To further analyze the classification performance of our method, we visualize the confusion matrix in Fig. 3. The confusion matrix shows that our *COM_CKD* is able to clearly distinguish animals from rigid artifacts, except for planes and birds. It is understandable because flying birds look very similar to planes (as shown in Fig. 4), especially in low-resolution images. Due to the non-rigid and deformable property of articulated objects, we also observe many confusions between different animals.

(a) Images from plane wrongly classified as bird

(b) Images from bird wrongly classified as plane

Fig. 4. Some wrongly classified samples between plane and bird.

Among all animal classes, the frog class obtains the highest false positive rate of 18.07 % from other animal classes, but it has with very few false negatives. As expected, car and truck are the most confusing artifact classes, which collectively cause a classification error rate of 8.78 %. Whereas, cat and dog are the most confusing animal classes, which collectively cause a classification error rate of 11.24 %.

4.4 Evaluation of Object Detection

To adapt our method for object detection, we train a two-class linear SVM classifier as the detector using *COM_CKD* features. For an instance image, we decompose it into several scales and detect possible locations of all candidate objects using a sliding window at each scale. Finally, we merge detection results at different scales and remove the duplicate detections at the same location. We test our detector on a chicken feet dataset collected in a chicken slaughter house. The aim of our detector is to find and localize chicken feet. As illustrated in Fig. 6, this chicken feet dataset is very challenging due to the following facts: chicken feet themselves are very small compared with other parts of the body, usually more than forty chickens are squeezed in a box, multiple chicken feet may appear in one image, in many cases feet are severely occluded (most part of feet are hidden under feather), the appearance of feet changes drastically due to different poses, and finally the color of the feet is very similar to feather and chest.

We crop a total of 717 image patches containing chicken feet as positive training examples, and 2000 patches without chicken feet as negative training examples. Another set of 318 images containing chicken feet patches never occurred in the training set are used as test set. Since chicken feet are also tiny, we use the same patch size and sampling grid for the CIFAR-10 dataset to evaluate CKD. The parameters of CKD and SVM are tuned by a 10-fold cross-validation on the training set. To judge the correctness of detections, we adopt standards of the PASCAL Challenge criterion [56], i.e. a detection is considered as correct only if the predicted bounding box overlaps at least 50 % area with the ground-truth bounding box. All other detections of the same object are counted as false positives. We compare the detection performance of our model with that of the HKDES model [2] and a 3-level SPM_SIFT [7] in terms of the Equal Error Rate (EER) on the Precision-Recall (PR) curves, i.e. PR-EER. PR-EER defines the point on the PR curve, where the recall rate equals the precision rate.

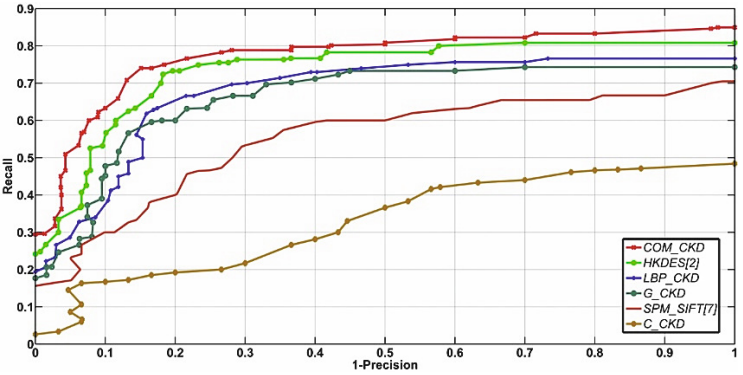


Fig. 5. Precision-Recall curves of all methods tested on chicken feet dataset.

Figure 5 plots the Precision-Recall curves for all methods. As we see, among all tested models, *COM_CKD* achieves the best overall performance (EER = 78.53 %), followed by the HKDES model (EER = 75.61 %) that combines gradient, color and shape cues into KDES. This further confirms that merging different visual cues into object representation can significantly boost the performance of the classifier. One interesting observation is that, expect for *C_CKD*, results from our single CKD models are better than the sophisticated SIFT method. In particular, EERs of *LBP_CKD* and

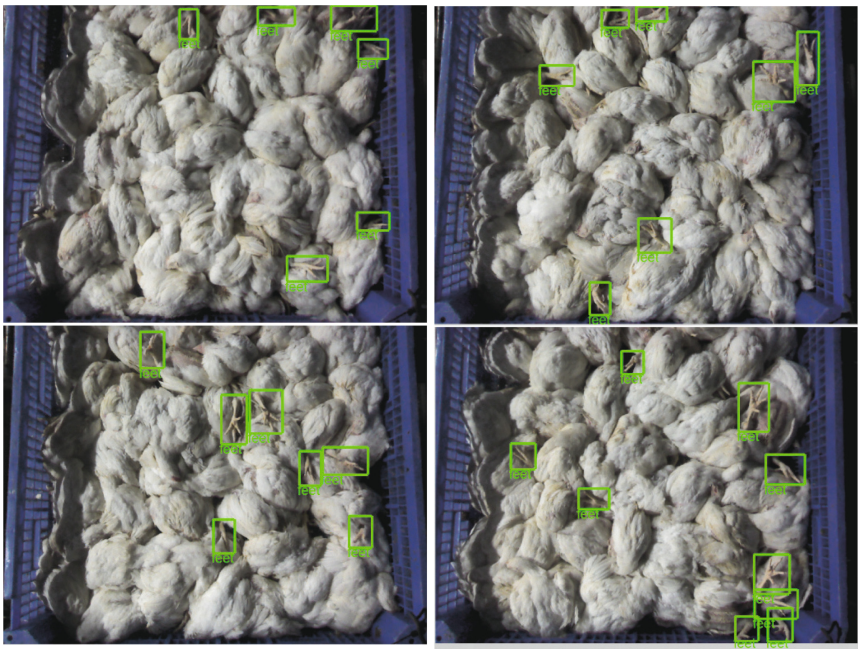


Fig. 6. Detection examples resulting from *COM_CKD* feature.

G_CKD model are 71.23 % and 69.55 %, respectively, whereas EER of SPM_SIFT is only 59.41 %. Considering individual CKD, *C_CKD* gives the worst result with EER = 44.10 %. Both *LBP_CKD* and *G_CKD* perform well, with *LBP_CKD* achieving a slightly better average accuracy. This is not surprising. Color difference between chicken feet and other parts (feather and chest) is marginal (refer to Fig. 6). Color distributions of chicken feet and other parts overlap quite much. In particular, the color distribution of feet and chest can hardly allow an acceptable separation based on color cue alone. In contrast, feet show a moderate difference in texture structures from feature and chest. Hence, texture based *LBP_CKD* outperforms other single feature for this dataset. Figure 6 shows some detection examples resulting from the best *COM_CKD* feature. Due to the influence of shadow caused by the box boundary and severe occlusions, some small chicken feet under the box shadow (in left images) or hidden by the feather (in right images) are missed by the detector, which give the false negative detections. But for these images no false positive detections appear.

5 Conclusion

Based on the context cue and Rényi quadratic entropy based CSQMI, we propose a set of novel kernel descriptors called context kernel descriptors and an information theoretic measure to select a compact and discriminative codebook for object representation in kernel feature space. We evaluate the performance of our algorithm in applications of object recognition and detection. The highlights of our work lie in: (1) the new CKD enhances the original KDES by adding extra spatial co-occurrence constraints to reduce the mismatch of image attributes (features) in kernel space; (2) instead of applying the traditional KPCA for feature dimensionality reduction, CSQMI criterion is employed in our method to learn a subset of low-dimensional discriminative CKD that correspond to the cluster structure of the density distribution of CKD. Evaluation results on both popular benchmark and our own datasets show the effectiveness of our method for generic (especially tiny) object recognition and detection.

Acknowledgements. This work is supported by The Danish Agency for Science, Technology and Innovation, project “Real-time controlled robots for the meat industry”, and partly supported by Jiangsu Natural Science Foundation (JSNSF) under Grant BK20131296, and National Nature Science Foundation of China (NSFC) under Grant 61101165. The authors thank Lantmännen Danpo A/S for providing the chicken images.

References

1. Bo, L., Ren, X., Fox, D.: Kernel descriptors for visual recognition. In: NIPS, pp. 244–252 (2010)
2. Bo, L., Lai, K., Ren, X., Fox, D.: Object recognition with hierarchical kernel descriptors. In: CVPR, vol. 1, pp. 1729–1736 (2011)
3. Bo, L., Sminchisescu, C.: Efficient match kernel between sets of features for visual recognition. In: NIPS, vol. 1, pp. 135–143 (2009)

4. Wang, P., et al.: Supervised kernel descriptor for visual recognition. In: CVPR, vol. 1, pp. 2858–2865 (2013)
5. Jégou, H., Douze, M., Schmid, C.: Packing bag-of-features. In: ICCV, vol. 1, pp. 2357–2364 (2009)
6. Cao, Y. et al.: Spatial-bag-of-features. In: CVPR, vol. 1, pp. 3352–3359 (2010)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, vol. 1, pp. 2169–2178 (2006)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
9. Bay, H., Ess, A., Tuytelaars, T., Gool, L.: Van.: SURF: speeded up robust features. Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
10. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. PAMI **24**(7), 971–987 (2002)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
12. Pedersen, K., Smidt, K., Ziem, A., Igel, C.: Shape index descriptors applied to texture-based galaxy analysis. In: ICCV, vol. 1, pp. 2240–2447 (2013)
13. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: KAZE features. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 214–227. Springer, Heidelberg (2012)
14. Scholkopf, B., Smola, A., Mulle, K.: Kernel principal component analysis. In: ICANN, vol. 1327, pp. 583–588 (1997)
15. Scholkopf, B., Smola, A., Mulle, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
16. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Trans. Neural Netw. **5**(4), 537–550 (1994)
17. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. PAMI **27**(8), 1226–1238 (2005)
18. Yang, H., Moody, J.: Feature selection based on joint mutual information. Int. ICSC Symp. Adv. Intell. Data Anal. vol. 1, pp. 22–25 (1999)
19. Kwak, N., Choi, C.: Input feature selection by mutual information based on parzen window. IEEE Trans. PAMI **24**(12), 1667–1671 (2002)
20. Zhang, Z., Hancock, E.R.: A graph-based approach to feature selection. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 205–214. Springer, Heidelberg (2011)
21. Liu, C., Shum, H.: Kullback-Leibler boosting. In: CVPR, vol. 1, pp. 587–594 (2003)
22. Qiu, Q., Patel, V., Chellappa, R.: Information-theoretic dictionary learning for image classification. IEEE Trans. PAMI **36**(11), 2173–2184 (2014)
23. Brown, G., Pocock, A., Zhao, M., Luján, M.: Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J. Mach. Learn. Res. **13**(1), 27–66 (2012)
24. Leiva, J., Artes, A.: Information-theoretic linear feature extraction based on kernel density estimators: a review. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42**(6), 1180–1189 (2012)
25. Hild II, K., Erdogmus, D., Principe, J.: An analysis of entropy estimators for blind source separation. Sign. Proces. **86**(1), 182–194 (2006)
26. Hild II, K., Erdogmus, D., Torkkola, K., Principe, J.: Feature extraction using information-theoretic learning. IEEE Trans. PAMI **28**(9), 1385–1392 (2006)

27. Rényi, A.: On measures of entropy and information. In: Fourth Berkeley Symposium on Mathematical Statistics and Probability, pp. 547–561 (1961)
28. Principe, J.: Information theoretic learning: Renyi's entropy and kernel perspectives. Springer, Heidelberg (2010)
29. Parzen, E.: On the estimation of a probability density function and the mode. Ann. Math. Statist. **33**(3), 1065–1076 (1962)
30. Jenssen, R.: Kernel entropy component analysis. IEEE Trans. PAMI **32**(5), 847–860 (2010)
31. Jenssen, R., Eltoft, T.: A new information theoretic analysis of sum-of-squared-error kernel clustering. Neurocomputing **72**(1–3), 23–31 (2008)
32. Gómez, L., Jenssen, R., Camps-Valls, G.: Kernel entropy component analysis for remote sensing image clustering. IEEE Geosci. Remote Sens. Lett. **9**(2), 312–316 (2012)
33. Zhong, Z., Hancock, E.: Kernel entropy-based unsupervised spectral feature selection. Int. J. Pattern Recogn. Artif. Intell. **26**(5), 126002-1-18 (2012)
34. Hellman, M., Raviv, J.: Probability of error, equivocation, and the Chernoff bound. IEEE Trans. Inf. Theor. **16**(4), 368–372 (1979)
35. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Ilumination cone models for face recognition under variable lighting and pose. IEEE Trans. PAMI **23**, 643–660 (2001)
36. Li, F., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. PAMI **28** (4), 594–611 (2006)
37. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans. PAMI **30**(11), 1958–1970 (2008)
38. Mika, S., et al.: Fisher discriminant analysis with kernels. In: IEEE Neural Networks for Signal Processing Workshop, pp. 41–48 (1999)
39. He, X., et al.: Face recognition using laplacianfaces. IEEE Trans. PAMI **27**(3), 328–340 (2005)
40. Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: Receptive field learning for pooled image features. In: CVPR, vol. 1, pp. 3370–3377 (2012)
41. Jiang, Z., Zhang, G., Davis, L.: Submodular dictionary learning for sparse coding. In: CVPR, vol. 1, pp. 3418–3425 (2012)
42. Boureau, Y., et al.: Ask the locals: Multi-way local pooling for image recognition. In: ICCV, vol. 1, pp. 2651–2658 (2011)
43. Liu, L., et al.: In defense of soft-assignment coding. In: ICCV, pp. 2486–2493 (2011)
44. Zeiler, M., Taylor, W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: ICCV, vol. 1, pp. 2018–2025 (2011)
45. Feng, J., Ni, B., Tian, Q., Yan, S.: Geometric p-norm feature pooling for image classification. In: CVPR, vol. 1, pp. 2697–2704 (2011)
46. Oliveira, G., Nascimento, E., Vieira, A.: Sparse spatial coding: a novel approach for efficient and accurate object recognition. In: ICRA, pp. 2592–2598 (2012)
47. McCann, S., Lowe, D.G.: Spatially local coding for object recognition. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 204–217. Springer, Heidelberg (2013)
48. Bo, L., Ren, X., Fox, D.: Multipath sparse coding using hierarchical matching pursuit. In: CVPR, vol. 1, pp. 660–667 (2013)
49. Seidenari, L., Serra, G., Bagdanov, A., Del Bimbo, A.: Local pyramidal descriptors for image recognition. IEEE Trans. PAMI **36**(5), 1033–1040 (2014)
50. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV, vol. 1, pp. 1–8 (2007)
51. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: CVPR, pp. 3642–3649 (2012)

52. Zeiler, M., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. In: ICLR (2013)
53. Le, Q., et al.: Tiled convolutional neural networks. In: NIPS, vol. 1, pp. 1279–1287 (2010)
54. Yu, K., Zhang, T.: Improved local coordinate coding using local tangents. In: ICML, vol. 1, pp. 1215–1222 (2010)
55. Goodfellow, I., Courville, A., Bengio, Y.: Spike-and-slab sparse coding for unsupervised feature discovery. In: NIPS Workshop on Challenges in Learning Hierarchical Models (2011)
56. Everingham, M., et al.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)