

Object Classification and Detection with Context Kernel Descriptors

Hong Pan^{1,2}, Søren Ingvor Olsen¹, and Yaping Zhu¹

¹ Department of Computer Science, University of Copenhagen, 1017 Copenhagen K, Denmark

² School of Automation, Southeast University, Nanjing, 210096, China

Abstract. Context information is important in object representation. By embedding context cue of image attributes into kernel descriptors, we propose a set of novel kernel descriptors called Context Kernel Descriptors (CKD) for object classification and detection. The motivation of CKD is to use spatial consistency of image attributes or features defined within a neighboring region to improve the robustness of descriptor matching in kernel space. For feature selection, Kernel Entropy Component Analysis (KECA) is exploited to learn a subset of discriminative CKD. Different from Kernel Principal Component Analysis (KPCA) that only keeps features contributing mostly to image reconstruction, KECA selects the CKD that contribute mostly to the Rényi entropy of the image. These CKD are discriminative as they relate to the density distribution of the histogram of image attributes. We report superior performance of CKD for object classification on the CIFAR-10 dataset, and for detection on a challenging chicken feet dataset.

Keywords: Object classification and detection, Feature selection, Kernel descriptors, Kernel entropy component analysis.

1 Introduction

Classification and detection of objects are challenging, as real-world objects are difficult to model with significant variations in appearance and pose. Recently, local descriptor-based methods dominate the state-of-the-art object classification and detection algorithms. Many local descriptors [1-7] are designed to achieve a stable and robust representation for objects. In particular, orientation histogram descriptors like SIFT [1] and HOG [2] build a histogram of gradient orientations weighted by the gradient magnitudes within the feature point neighborhood. Histograms extracted from small image patches are normalized across larger areas and concatenated to form the final descriptor. Meanwhile, the impressive recognition performance offered by kernel methods, especially the SVM, further inspired people to develop local kernel-based descriptors. Wallraven et al. [8] first bridged local feature representation with SVM classifier by introducing a set of local kernels that satisfy the Mercer condition. With these local kernels, the feature matching step can be formulated as a part of the kernel itself, thus used as the input to SVM classifier. Based on that, more match kernels [9-11] over local feature sets are addressed. In particular, Bo et al.[6] linked

the design philosophy behind SIFT with the match kernel over image patches and developed kernel descriptors (KDES) to achieve the state-of-the-art performance on classification benchmarks.

Integrating the strength of local descriptor representation and kernel method, we propose a novel and systematic framework for object classification and detection from kernel’s perspective. In this framework, each module from feature extraction to feature selection, as well as classification is completely treated in a ‘kernelized’ way. More specifically, for feature extraction, we develop a set of Context Kernel Descriptors (CKD) which enhance the original KDES [6] by embedding the spatial context into KDES. Context cue enforces some degree of spatial consistency that improves the robustness of the resulting descriptors. Different from KDES that applied KPCA to reduce feature dimensionality, in our framework, Kernel Entropy Component Analysis (KECA) is exploited in the kernel feature space to learn and select a set of low-dimensional discriminative CKD. Finally, these reduced discriminative CKD are fed into a SVM classifier to perform the classification/detection. Fig.1 illustrates the framework of our kernel-based object classification and detection model. Evaluation results on the standard classification benchmark and a challenging chicken feet dataset show that our CKD outperforms the original KDES as well as carefully tuned SIFT descriptor and sophisticated deep learning-based methods.

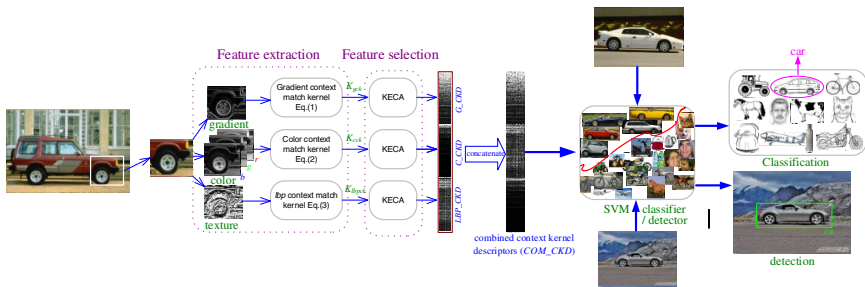


Fig. 1. Framework of our kernel-based object classification and detection model

2 Feature Extraction with CKD

Recently, [6] showed that the similarity measure of orientation histograms applied in HOG and SIFT-based feature matching is equivalent to a particular match kernel over image patches. This insight provides a general way to turn pixel-level attributes into patch-level features with match kernel measuring the similarity between image patches. Since only the pixel’s attribute and position are considered in the match kernel, it may cause unstable feature matching between different image patches with similar attributes.

In this work, we enhance the original match kernel in KDES by embedding it with a neighborhood constraint. As neighborhood defines a set of adjacent pixels surrounding the center pixel, this 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 (CMK) and the resulting descriptors as context kernel descriptors. Intuition behind CMK is that pixels with similar attributes from two image patches should have a high probability to have neighboring pixels with similar attributes. Considering the spatial co-occurrence constraint, our CMK significantly improve the matching accuracy.

CMK can be conveniently applied to develop a set of local descriptors using any kind of image attributes, such as gradient, color, texture, and shape. Next we derive the CMK, then introduce several specific CMKs used in our work.

2.1 Formulation of CMK

An image patch can be modeled as a set of pixels $X = \{x_i\}$ ($i = 1 \dots n$), where x_i is coordinate of the i th pixel. Let a_i be attribute vector at the i th pixel x_i . The k -neighborhood N_k^i of pixel x_i in X is defined as a group of pixels (including itself) that are closest to it. Mathematically, $N_k^i = \{x_j \in X \mid \|x_i - x_j\| \leq k; k \geq 1\}$. So the context kernel of attributes a between two pixels x_p and x_q is defined as $\kappa_{con}[(x_p, a_p), (x_q, a_q)] = 1/|N_k^p \cap N_k^q| \sum_{x_u \in N_k^p} \sum_{x_v \in N_k^q} \kappa_a(a_u, a_v)$, where $\kappa_a(a_u, a_v) = \exp(-\gamma_a \|a_u - a_v\|^2) = \varphi_a(a_u)^T \varphi_a(a_v)$ is a Gaussian kernel measuring the similarity of attributes between two pixels x_u and x_v . Merging κ_{con} into match kernels [6] and replacing attribute a in κ_{con} with specific attributes, we can derive a set of ad hoc attribute-based CMKs.

Let θ_p and m_p be orientation and magnitude of the image gradient at pixel x_p . For the convenience of matching, we use normalized orientation $\theta'_p = (\sin\theta_p, \cos\theta_p)$ and normalized magnitude $m'_p = m_p / \sqrt{\sum_{p \in P} m_p^2 + \varepsilon}$ (ε is a small positive number). To compare the similarity of gradients between patches P and Q from two different images, the gradient context match kernel \mathbf{K}_{gck} can be defined as

$$\mathbf{K}_{gck}(P, Q) = \sum_{p \in P} \sum_{q \in Q} m'_p m'_q \kappa_o(\theta'_p, \theta'_q) \kappa_s(x_p, x_q) \kappa_{con}[(x_p, \theta'_p), (x_q, \theta'_q)] \quad (1)$$

where $\kappa_s(x_p, x_q) = \exp(-\gamma_s \|x_p - x_q\|^2) = \varphi_s(x_p)^T \varphi_s(x_q)$ is a spatial kernel measuring how close two pixels are spatially; $\kappa_o(\theta'_p, \theta'_q) = \exp(-\gamma_o \|\theta'_p - \theta'_q\|^2) = \varphi_o(\theta'_p)^T \varphi_o(\theta'_q)$ is the orientation kernel measuring the similarity of normalized orientations at two pixels x_p and x_q . Similarly, to measure the proximity of color attributes between P and Q , the color context match kernel \mathbf{K}_{cck} can be defined as

$$\mathbf{K}_{cck}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \kappa_c(c_p, c_q) \kappa_s(x_p, x_q) \kappa_{con}[(x_p, c_p), (x_q, c_q)] \quad (2)$$

where $\kappa_c(c_p, c_q) = \exp(-\gamma_c \|c_p - c_q\|^2) = \varphi_c(c_p)^T \varphi_c(c_q)$ 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 define our texture context match kernel, \mathbf{K}_{lbpcck} , using Local Binary Patterns (*lbp*) [3]

$$\mathbf{K}_{lbpcck}(P, Q) = \sum_{p \in P} \sum_{q \in Q} \sigma'_p \sigma'_q \kappa_{lbp}(lbp_p, lbp_q) \kappa_s(x_p, x_q) \kappa_{con}[(x_p, lbp_p), (x_q, lbp_q)] \quad (3)$$

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

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

$$K_a(P, Q) = \sum_{p \in P} \sum_{q \in Q} w_p w_q \kappa_a(a_p, a_q) \kappa_s(x_p, x_q) \kappa_{con}[(x_p, a_p), (x_q, a_q)] \tag{4}$$

where $w_p w_q$ and κ_a are linear weighting kernel and attribute kernel, respectively.

2.2 Approximation of CMK

Using the inner product, we rewrite K_a as $K_a(P, Q) = \langle \psi_a(Q), \psi_a(P) \rangle = \psi_a(P)^T \psi_a(Q)$, with $\psi_a(\cdot) = \sum w \varphi_a(a) \otimes \varphi_s(x) \otimes \varphi_{con}(x, a)$, where \otimes is the tensor product. ψ_a gives the mapping feature in kernel space, namely the CKD. Note that the dimensions of φ_a , φ_s and φ_{con} are all infinite, since Gaussian kernels are used. To obtain an accurate approximation of the CMK matrix K_a , 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 $\varphi_a(a)$ and $\varphi_{con}(x, a)$ by their projections onto the subspaces spanned by the G basis vectors $\{\varphi_a(a^g)\}_{g=1}^G$. For space vector x , we discretize spatial basis vectors into L bins and sample x along the L basis vectors. Finally, we approximate ψ_a by its projections onto the $G \times L \times G$ joint basis vectors: $\{\varphi_a(a^i) \otimes \varphi_s(x^j) \otimes \varphi_{con}(a^k), \dots, \varphi_a(a^G) \otimes \varphi_s(x^L) \otimes \varphi_{con}(a^G)\}$, and derive the $G \times L \times G$ -dimensional CKD, i.e.

$$\psi_a(\cdot) \approx \sum_{i=1}^G \sum_{j=1}^L \sum_{k=1}^G f_{ijk} \varphi_a(a^i) \otimes \varphi_s(x^j) \otimes \varphi_{con}(a^k) \tag{5}$$

where f_{ijk} is the projection coefficient onto the joint basis vector $\varphi_a(a^i) \otimes \varphi_s(x^j) \otimes \varphi_{con}(a^k)$. As aforementioned, we approximate the resulting CKD ψ_a with finite joint basis vectors. However, due to the tensor product, the full dimensionality of projected CKD onto the joint basis vectors is high. For example, for gradient CKD ψ_{gck} , if we quantize basis vectors of φ_o and φ_{con} into 25 bins respectively, and choose the basis vectors of spatial kernel φ_s on a 5×5 grid, the dimensionality of ψ_{gck} equals 25×5×5×25=15625. This dimensionality is too high for practical computation. Next, we show how to further reduce the dimensionality of CKD to a tractable scale.

3 Feature Selection with KECA

By uniform sampling along sufficient basis vectors over a dense spatial grid in image attribute space, we approximate the CKD ψ_a with its projection onto finite dimensional basis vectors. Uniform sampling usually yields redundant basis vectors. To reduce the redundancy and further lower the feature dimensionality, original KDES applied KPCA. However, KPCA can only capture second-order statistics of KDES and select features contributing mostly to image reconstruction. This inevitably degrades the distinctiveness of KDES for image classification. In contrast, we apply KECA that selects basis vectors relating to the probability distribution of the histogram of image attributes from available basis vectors to generate low-dimensional discriminative CKD features.

Similar to KPCA [12], KECA [13] is a novel spectral analysis method based on kernel similarity matrix. But it does not necessarily use the top eigenvalues and eigenvectors of the kernel matrix. Unlike KPCA which maximally preserves the variance of the data set, KECA, built on information theory, tries to preserve the maximum Rényi entropy of the input data set with the smallest number of extracted features. KECA provides a new information theoretic learning tool that enables to capture geometric structure of the input data [13]. Thus, it is extremely suitable for nonlinear data analysis.

The quadratic Rényi entropy [14] is defined as

$$H(p) = -\log \int p^2(s)ds = -\log[V(p)] \tag{6}$$

where $p(s)$ is the probability density function of a data set $S = \{s_i\} (i = 1, \dots, N)$. Since the logarithm is a monotonic function, we only focus on $V(p)$. Using Parzen window, $V(p)$ can be estimated as $\hat{V}(p) = 1/N^2 \sum_{s_i \in S} \sum_{s_j \in S} k(s_i, s_j | \sigma) = 1/N^2 \mathbf{1}^T \mathbf{K} \mathbf{1}$, where $k(s, s_j | \sigma)$ is the

Parzen window (kernel), centered at s_i with a width of σ ; \mathbf{K} is a $N \times N$ kernel matrix corresponding to the Parzen window; and $\mathbf{1}$ is an $N \times 1$ vector with each element being 1.

For the approximated kernel matrix \mathbf{K}_a , defined by ψ_a in Eq.(5), assuming its dimensionality is $N \times N$, then it can be eigendecomposed as $\mathbf{K}_a = \mathbf{E} \mathbf{D} \mathbf{E}^T$, where \mathbf{D} is a diagonal matrix containing eigenvalues $\lambda_1, \dots, \lambda_N$ and \mathbf{E} is a matrix with the corresponding eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_N$ as columns. Taking \mathbf{K}_a into $\hat{V}(p)$, we have $\hat{V}(p) =$

$$1/N^2 \sum_{i=1}^N (\sqrt{\lambda_i} \mathbf{e}_i^T \mathbf{1})^2 = 1/N^2 \sum_{i=1}^N \phi_i$$

The term ϕ_i defines the contribution of each eigenvalue

and its corresponding eigenvector to the Rényi entropy estimate $\hat{V}(p)$.

Note that $\mathbf{K}_a = \psi_a^T \psi_a$, projecting ψ_a onto a single principal axis \mathbf{u}_i in the kernel space gives $\mathbf{u}_i^T \psi_a = \sqrt{\lambda_i} \mathbf{e}_i^T$. Projecting ψ_a onto all principal axes results $\mathbf{U}^T \psi_a = \mathbf{D}^{1/2} \mathbf{E}^T$, where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N]$ is the projection matrix. Different from KPCA that selects m ($m < N$) principal axes such that the projection data corresponding to the largest m eigenvalues $\lambda_1, \dots, \lambda_m$ ($\lambda_1 \geq \dots \geq \lambda_m \geq \dots \geq \lambda_N$) and their eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_m$, whereas, KECA selects

m principal axes U_m^{eca} , corresponding to the largest m $\{\phi_i\}_{i=1}^m$ ($\phi_1 \geq \dots \geq \phi_m \dots \geq \phi_N$), that contribute mostly to the Rényi entropy estimate of the data. Projecting ψ_a onto U_m^{eca} , we obtain reduced KECA feature set $\psi_a^{eca} = U_m^{ecaT} \psi_a = D_m^{1/2} E_m^T$. This feature set is the solution to $\min_{\lambda_1, \epsilon_1, \dots, \lambda_N, \epsilon_N} \hat{V}(p) - \hat{V}_m(p)$, subject to $\psi_a^{eca} = D_m^{1/2} E_m^T$, which means that the Rényi entropy estimate $\hat{V}(p)$ of the original data are maximally preserved in $\hat{V}_m(p)$, after KECA transform.

To sum up, for each type of CMK designed in section 2.1, we first approximate it with finite dimensional basis vectors, then KECA is applied to learn the reduced distinctive basis vectors. Projecting attribute, spatial position and context cue of a pixel onto learned basis vectors, we obtain its gradient (G_CKD), color (C_CKD), and texture (LBP_CKD) CKD. These heterogeneous CKDs represent an image from different aspects and complement each other. To fully exploit all cues, we concatenate G_CKD , C_CKD and LBP_CKD into a combined CKD (COM_CKD). Similar to [7], we aggregate patch-level CKDs hierarchically to build the final image-level feature representation.

4 Evaluation Results

We test CKD features on CIFAR-10[15] for object classification and on a chicken feet dataset for object detection. To make a fair comparison, we follow the parameter setting in [6], except for the final feature dimensionality. Namely, basis vectors for κ_o , κ_c , and κ_s are sampled over 25, $5 \times 5 \times 5$, and 5×5 uniform grids, respectively. For κ_{lbp} , we choose all 256 basis vectors. Basis vectors of κ_{con} for different CKDs are selected as the same vectors of their attribute kernels. The optimal dimensionality of CKD and neighborhood size k are decided by cross-validation. According to our observation, an optimal 200-dimensional feature learned from KECA is selected for each type of CKD, resulting a final 600-dimensional COM_CKD feature descriptor per image patch.

4.1 Object Classification

CIFAR-10: This dataset contains 80 million tiny images with a size of 32×32 pixels. It has 10 classes, with 5000 training images and 1000 test images per class. We use this dataset for a comprehensive comparison on the classification performance between original KDES and our CKD. We calculate CKD with 3-neighborhood around 8×8 image patches on a dense grid with a spacing of 2 pixels. The whole training set is split into 10,000/40,000 training/validation set, and the validation set is used to optimize the kernel parameters using the grid search. Table 1 lists the classification accuracy of different kernel descriptors options. Some recent results are also provided in table 2.

Table 1. Comparison of classification accuracy (%) between KDES and CKD on CIFAR-10

Features	KDES[6]	CKD
gradient	66.3	68.5
color	53.9	62.2
texture(<i>lbp</i>)	68.2	72.0
combination	76.0	85.4

Table 2. Classification accuracy (%) of other methods on CIFAR-10

Methods	Accuracy
HKDES [7]	80.00
SPN [16]	83.96
Zeiler&Fergus [17]	84.88
MCDNN [18]	88.79

From table 1, we observe that the proposed context-based kernel descriptors consistently outperform original counterparts, for both individual and combination version. Except for *G_CKD*, both *C_CKD* and *LBP_CKD* are significantly better than the original ones. For combination version, the accuracy of *COM_CKD* is 85.4%, which is 9.4% higher than the combined KDES. We argue that performance improvement of CKD comes from two facts: 1) compared with KDES, additional spatial co-occurrence constraint defined in CKD further improves its robustness to the semantic ambiguity due to the missing of features in case of partial occlusion; 2) feature selection in our method is guided by KECA. Different from KPCA used in [10] that only selects KDES contributing mostly to image reconstruction, KECA selects CKD that contribute mostly to the Rényi entropy of the original image. These CKD are more discriminative than KDES in that they relate to the density distribution of the histogram of image attributes. Compared with some sophisticated deep learning methods [16-18], our model achieves a comparable classification accuracy with a simpler and efficient feature learning and classification structure. For a 32×32 sized image, our model takes 84.63ms to calculate the full dimensional 3-neighborhood *COM_CKD* and 20.21ms to conduct the feature selection using KECA on a platform with Intel Core i7 2.7GHz CPU and 16G RAM.

4.2 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 detect locations of all candidate objects using a sliding window technique. We test our detector on a chicken feet dataset collected in a chicken slaughter house. The aim of our detector is to localize chicken feet. This chicken feet dataset is very challenging, considering the following facts: chicken feet are very small; 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 under different poses.

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 feet patches never occurred in the training set are used as test set. Since chicken feet are very tiny, we also calculate 3-neighborhood CKD around 8×8 image patches on a dense grid with a spacing of 2 pixels. The parameters of CKD and SVM are tuned by 10-fold cross-validation on the training set. We compare the detection performance of our model with that of the hierarchical

kernel descriptors (HKDES) [7] and a 3-level spatial pyramid dense SIFT (SPM_SIFT) [10] in terms of equal error rate (EER). Fig.2 plots the precision-recall (PR) curves for all methods. As we see, among all test models, *COM_CKD* achieves the best overall performance (EER=76.77%), followed by HKDES (EER=75.61%) that combines gradient, color and shape information. This further confirms that merging different visual cues into object representation can significantly boost the performance of the classifier. One interesting observation is that, except for *C_CKD*, results from our single CKD models are better than the sophisticated SIFT method. In particular, EERs of *LBP_CKD* and *G_CKD* model are 70% and 68.35%, respectively, whereas EER of SPM_SIFT is only 59.41%. Considering the individual CKD, *C_CKD* gives the worst result with EER=42.10%. Both *LBP_CKD* and *G_CKD* perform well, with *LBP_CKD* achieving a slightly better average accuracy. This is not surprising, because color difference between chicken feet and other parts (feather and chest) is marginal (refer to Fig.3). 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 classification based on color cue alone. In contrast, feet show different texture structures from feature and chest. Hence, texture based *LBP_CKD* outperforms other single feature for this dataset. Fig.3 shows some detection examples resulted from the best *COM_CKD* features.

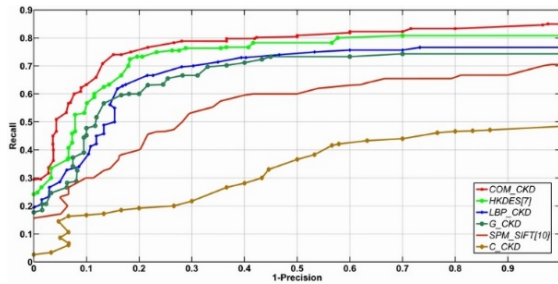


Fig. 2. PR curves of all methods tested on a chicken feet dataset

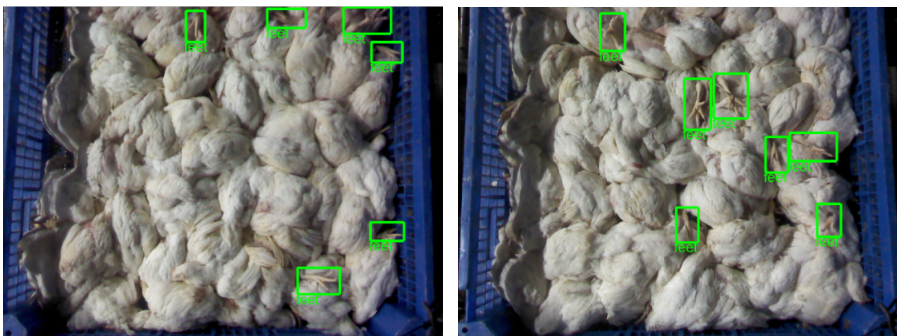


Fig. 3. Detection examples resulted from *COM_CKD* features

5 Conclusion

Based on the context cue and KECA, we propose a set of novel kernel descriptors called context kernel descriptors. The contributions of our work lie in 1) the new CDK enhances

KDES by adding extra spatial co-occurrence constraints to reduce the mismatch of image attributes (features) in the kernel space; 2) instead of applying traditional KPCA for feature selection, KECA is performed in our method to learn a subset of discriminative CKDs that correspond to density distribution of the histogram of image attributes. Evaluation results on both popular benchmark and our own datasets show the effectiveness of our method for generic (especially tiny) object classification 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 NSF of Jiangsu Province, China under Grant BK20131296, and NSFC under Grant 61005051. The authors thank Lantmännen Danpo A/S for providing the chicken images.

References

- [1] Lowe, D.: Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
- [2] Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: *Proc. of CVPR*, vol. 1, pp. 886–893 (2005)
- [3] Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE PAMI* 24(7), 971–987 (2002)
- [4] 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)
- [5] Pedersen, K., Smidt, K., Ziem, A., Igel, C.: Shape Index Descriptors Applied to Texture-based Galaxy Analysis. In: *Proc. of ICCV*, vol. 1, pp. 2240–2447 (2013)
- [6] Bo, L., Ren, X., Fox, D.: Kernel Descriptors for Visual Recognition. In: *Proc. of NIPS*, pp. 244–252 (2010)
- [7] Bo, L., Lai, K., Ren, X., Fox, D.: Object Recognition with Hierarchical Kernel Descriptors. In: *Proc. of CVPR*, pp. 1729–1736 (2011)
- [8] Wallraven, C., Caputo, B., Graf, A.: Recognition with Local Features: the Kernel Recipe. In: *Proc. of ICCV*, pp. 257–264 (2003)
- [9] Bo, L., Sminchisescu, C.: Efficient Match Kernel between Sets of Features for Visual Recognition. In: *Proc. of NIPS*, pp. 135–143 (2009)
- [10] Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: *Proc. of CVPR*, pp. 2169–2178 (2006)
- [11] Grauman, K., Darrell, T.: The Pyramid Match Kernel: Efficient Learning with Sets of Features. *JMLR* 8, 725–760 (2007)
- [12] Scholkopf, B., Smola, A., Mülle, K.: Kernel Principal Component Analysis. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) *ICANN 1997*. LNCS, vol. 1327, pp. 583–588. Springer, Heidelberg (1997)
- [13] Jenssen, R.: Kernel Entropy Component Analysis. *IEEE Trans. PAMI* 32(5), 847–860 (2010)
- [14] Rényi, A.: On Measures of Entropy and Information. *Sel. Papers Alfréd Rényi* 2, 565–580 (1976)
- [15] Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Technical report (2009)
- [16] Gens, R., Domingos, P.: Discriminative Learning of Sum-Product Networks. In: *Proc. of NIPS* (2012)
- [17] Zeiler, M., Fergus, R.: Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. In: *Proc. of ICLR* (2013)
- [18] Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification. In: *Proc. of CVPR*, pp. 3642–3649 (2012)