The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition

Nuno Vasconcelos

ECE Department

University of California, San Diego

Purdy Ho Pedro Moreno

HP Labs

Cambridge Research Laboratory

Classification architectures for vision

- modern learning theory favors discriminant over generative architectures for classification
- for vision, a fundamental difference is the set of constraints imposed on the representation
  - discriminant classifiers favor holistic representations (image as a point in high-dimensional space)
  - generative classifiers favor localized representations (images as bags of local features)
- localized representations have various advantages
  - more invariant
  - more robust to occlusion
  - lower dimensionality

Classification architectures for vision

- also, despite weaker guarantees, generative architectures have great practical appeal
  - better scalability in number of classes
  - encoding of prior knowledge in the form of statistical models
  - modular solutions, using Bayesian inference
- Q: can all this be combined with discriminant guarantees?
- we consider SVMs, and the Kullback-Leibler kernel
- investigate its ability to seamlessly combine discriminant recognition with generative models based on localized representations

Support vector machines

- **SVM:** given training $(\mathbf{x}_i, y_i)$, linear SVM is $(\mathbf{w}, b)$

  $$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \text{ s.t } \sum_i \alpha_i y_i = 0, \ \alpha_i \geq 0$$

  where $\{\alpha_i\}$ is a set of Lagrange multipliers, and

  $$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad b = \left\langle y_i - \sum_j y_j \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \right\rangle_{i|\alpha_i > 0} .$$

- extension to non-linear boundaries via a feature transformation

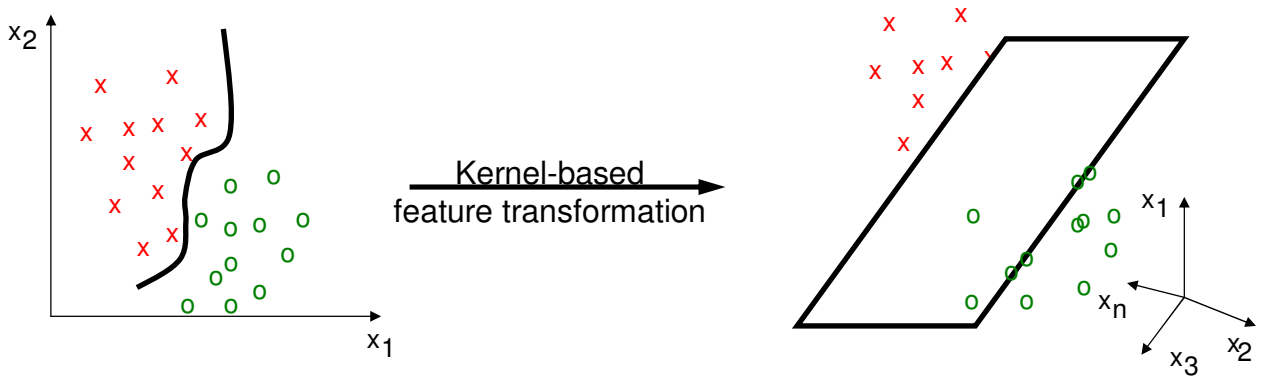
  $$\mathbf{\Phi} : \mathcal{X} \rightarrow \mathcal{Z}$$

Kernels

- exclusive dependence on dot-products allows implementation via a **kernel** function

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$$

- standard dot product implies Euclidean metric.
- Kernel extends to non-Euclidean measures of similarity: $\mathcal{K}(\mathbf{x}_i \cdot \mathbf{x}_j)$ is similarity between $\mathbf{x}_i, \mathbf{x}_j$.



Constraints on representation

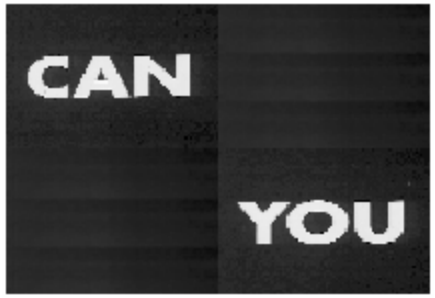
- two possible image representations
  - holistic: $\mathcal{X}$ the space of all images, each image is a point in $\mathcal{X}$
  - localized: image broken into local windows, $\mathcal{X}$ the space of such windows
- SVM training is $O(N^2)$, N = number of examples
- holistic: N = I, dim($\mathcal{X}$) large (e.g. 5,000)
- localized: $N = k \times I$, I = # images
  k = windows/image, k large (e.g. 5,000)
  dim($\mathcal{X}$) small (e.g. 8x8)
- complexity of localized ~ k^2 times that of holistic (e.g. 2.5 x10^7 increase)
- no way to capture grouping of windows into images

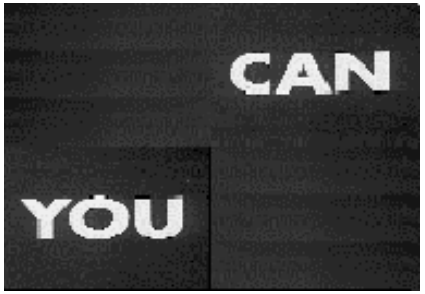
Constraints on representation

- localized not suited for traditional SVM
- holistic has been successful, but has limitations
- resolution:
  - images too high-dimensional, drastically down-sampled (e.g. from 240x360 to 20x20)
  - discarded information important for fine classification (near boundary)
- invariance:
  - images as points span quite convoluted manifolds in X, when subject to transformations
- occlusion:
  - a few occluded pixels can lead to a very large jump in X

Localized representations

- resolution is no issue (simply more points per image)
- greater robustness to invariance, e.g.





- greater robustness to occlusion
  - X% occluded pixels, means that x% of the probability mass changes
  - the remaining (1-x)% should still be enough to obtain a good match
  - when x% of a vector components change, matching is hard

Probabilistic kernels

- since
  - kernel captures similarities between examples
  - bags of localized examples best described by their prob. density
- natural to make the kernel function a measure of distance between probability density functions
- various kernels proposed in the literature
  - Fischer Kernel (Jaakkola et al, 1999),
  - TOP kernel (Tsuda et al, 2002),
  - diffusion kernels (Lafferty and Lebanon, 2002) ,
  - generalized correlation kernel (Kondor, Jebara, 2003),
  - KL-kernel (Moreno et al, 2003)

Probabilistic kernels

- three main advantages over holistic kernels
  - enable representations of variable length
  - enable a compact representation of a large sequence of vectors (through pdf)
  - can exploit prior knowledge about the classification problem (selection of suitable probability models)
- interpretation of the standard Gaussian kernel as

$$K(x, y) = e^{-\alpha d(x,y)}$$

where *d* is the Euclidean distance $d(x, y) = ||x - y||^2$

suggests a natural extension based on pdf distances

- this leads to the KL kernel

The KL kernel

- relies on the (symmetric) Kullback-Leibler divergence as the measure of pdf distance
- **Definition:** given densities $p(\mathbf{x})$ and $q(\mathbf{x})$, the KL-kernel is

$$KLK = e^{-a\mathcal{J}[p(\mathbf{x}),q(\mathbf{x})]+b}, \qquad (1)$$

where $\mathcal{J}(p(\mathbf{x}), q(\mathbf{x})) = KL(p(\mathbf{x}), q(\mathbf{x}))+KL(q(\mathbf{x}), p(\mathbf{x}))$ is the symmetric KL divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$,

$$KL(p(\mathbf{x}), q(\mathbf{x})) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \qquad (2)$$

the KL divergence between the two densities, and a and b constants.

A kernel taxonomy

- probabilistic kernels allow a great deal of flexibility over traditional counterparts
- KL kernel can be tuned to the problem in terms of:
  1. performance: choice of probability models that match the statistics of the data
  2. computation: using approximations to the KL that have been shown to work well in certain domains
  3. joint design of features and kernel
- here we focus on 1 and 2, stay tuned for 3
- it is possible to develop a taxonomy of kernels that implement various trade-offs between performance and computation

Parametric densities

- are good models or approximations for various problems
- the kernel can be tailored to the particular pdf family
- **Property:** For densities in *exponential family*

$$p(\mathbf{x}|\theta) = \alpha(\mathbf{x}) \exp\left[a(\theta) + \mathbf{b}(\theta)\mathbf{c}(\mathbf{x})\right],$$

(Gaussian, Poisson, Binomial, Beta, etc),

$$KL(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) =$$

$$= \quad a(\theta_i) - a(\theta_j) + \left[\mathbf{b}(\theta_i) - \mathbf{b}(\theta_j)\right]^T E_{\theta_i}[\mathbf{c}(\mathbf{x})]$$

where E_{θ_i} is expectation with respect to $p(\mathbf{x}|\theta_i)$.

The Gaussian

- is a particularly popular case
- $\mathcal{G}(\mathbf{x}, \mu, \Sigma) = \frac{1}{2\pi^{d/2}|\Sigma|} \exp\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\}$

  for which (2) becomes

  $$KL(\mathcal{G}(\mathbf{x}, \mu_i, \Sigma_i), \mathcal{G}(\mathbf{x}, \mu_j, \Sigma_j)) =$$
  $$\frac{1}{2}\log\frac{|\Sigma_\mathbf{j}|}{|\Sigma_i|} - \frac{d}{2} + \frac{1}{2}tr(\Sigma_j^{-1}\Sigma_i) + (\mu_i - \mu_j)^T\Sigma_j^{-1}(\mu_i - \mu_j)$$

  where d is the dimensionality of the $\mathbf{x}$.
- in general, it is possible to derive the kernel function for the parametric cases

Non-parametric densities

- non-parametric density models can be a lot trickier
- some have closed-form KL kernels, e.g. the histogram
- **Histogram:** $\pi = \{\pi_1, \ldots, \pi_b\}$, where π probability mass on partition of $\mathcal{X}$ defined by $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_b\}$.
- KL-divergence between π^i and π^j is

$$KL(\pi^i, \pi^j) = \sum_{k=1}^{b} \pi_k^i \log \frac{\pi_k^i}{\pi_k^j}$$

- extensions available for histograms defined on different partitions (Vasconcelos, Trans. Info. Theory, 2004)

Approximations

- various are possible for kernels without closed form
- χ^2 **distance:** linearizing log, $\log(x) \approx x - 1$,

$$KL(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) = \int \frac{(p(\mathbf{x}|\theta_i) - p(\mathbf{x}|\theta_j))^2}{p(\mathbf{x}|\theta_j)} dx$$

  the KL-divergence becomes the χ^2 distance commonly used in histogram matching.
- in some cases, even this has no closed-form, e.g.
- **Gaussian mixtures:**

$$p(\mathbf{x}|\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^c) = \sum_{k=1}^{c} \pi_k \mathcal{G}(\mathbf{x}, \mu_k, \Sigma_k)$$

Approximations and sampling

- various specific approximations have been recently proposed for the Gauss mixture case
  - *log-sum bound* (Singer and Warmuth, NIPS 98)
  - *asymptotic likelihood approximation* (Vasconcelos, ICCV 2001, trans. IT, 2004)
  - *unscented transformation* (Goldberger et al, ICCV 2004)
- finally, one can always use Monte Carlo sampling

$$KL[p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)] \approx \frac{1}{s} \sum_{m=1}^{s} \log \frac{p(\mathbf{x}_m|\theta_i)}{p(\mathbf{x}_m|\theta_j)}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_s$ is a sample drawn according to $p(\mathbf{x}|\theta_i)$.

Experiments

- all on COIL-100, three resolutions: 32x32, 64x64, 128x128
- 4 different combinations of train/test:
  - I images of each object used as training set, *I in {4, 8,18,36}*
  - remaining used for test
  - dataset with *I = n* referred to as $\mathcal{D}_n$
- holistic representation:
  - each image one vector
- localized representation:
  - image as feature bag: extract 8x8 windows, compute DCT, keep 32 first features
  - mixture of 16 Gaussians fit to each image

COIL-100

- 100 objects subject to 3D rotation
- one view every 5^o

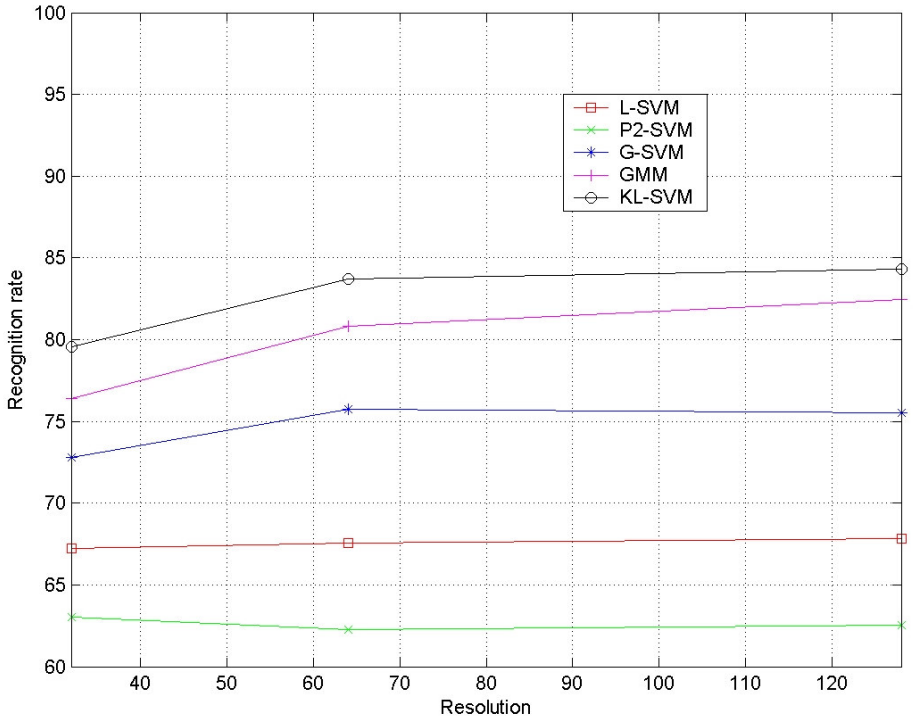
Results

- holistic: SVM with three different kernels
  - linear (L-SVM), polynomial order 2 (P2-SVM), Gaussian (G-SVM)
- localized:
  - standard maximum-likelihood Gauss mixture classifier
  - KL kernel with Gauss mixture models (KL-SVM)
- recognition rates (%)

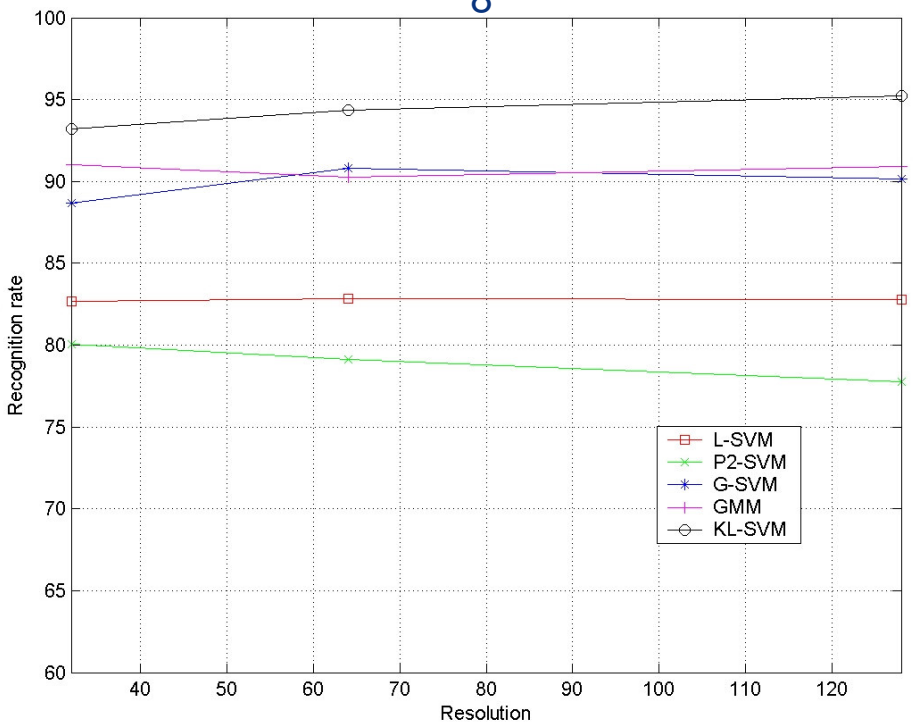
	Resolution 32×32				Resolution 64×64				Resolution 128×128			
	$\mathcal{D}_4$	$\mathcal{D}_8$	$\mathcal{D}_{18}$	$\mathcal{D}_{36}$	$\mathcal{D}_4$	$\mathcal{D}_8$	$\mathcal{D}_{18}$	$\mathcal{D}_{36}$	$\mathcal{D}_4$	$\mathcal{D}_8$	$\mathcal{D}_{18}$	$\mathcal{D}_{36}$
L-SVM	67.24	82.67	92.98	97.31	67.54	82.84	92.85	97.39	67.85	82.80	92.89	97.50
P2-SVM	63.02	80.03	93.09	98.11	62.27	79.11	92.30	97.89	62.53	77.78	92.85	97.58
G-SVM	72.79	88.67	96.85	**99.78**	75.75	90.80	97.78	**99.68**	75.54	90.13	97.04	**99.17**
GMM	76.41	91.05	96.30	97.83	80.82	90.27	94.89	95.31	82.48	90.89	94.72	94.89
KL-SVM	**79.56**	**93.20**	**97.32**	98.28	**83.69**	**94.36**	**98.89**	98.83	**84.32**	**95.22**	**98.65**	98.67

Results

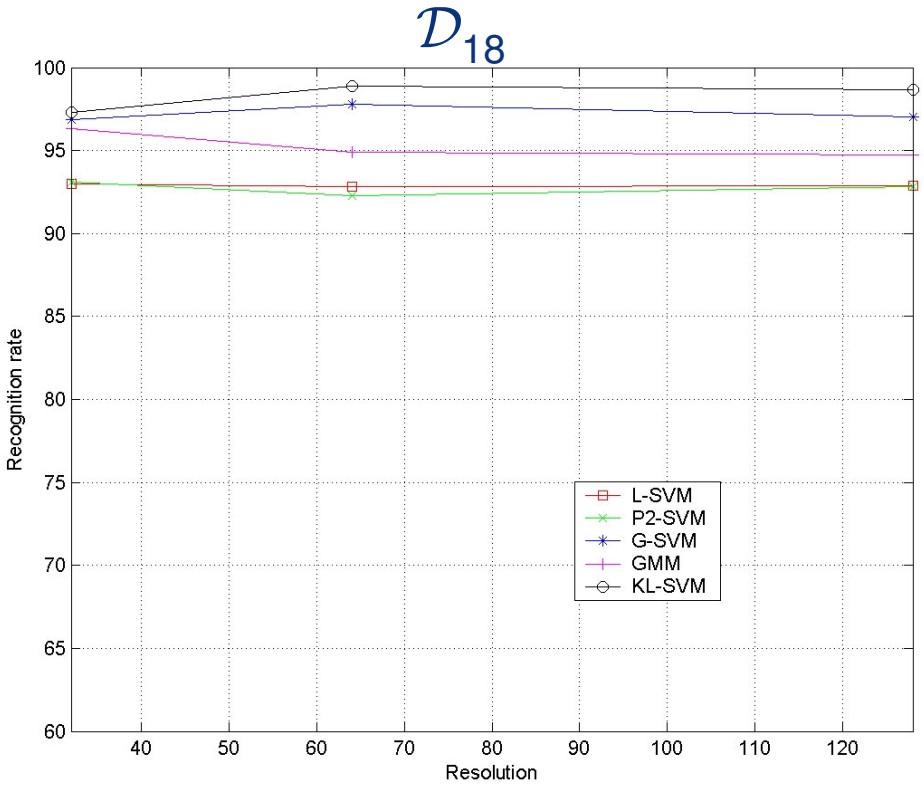
$\mathcal{D}_4$

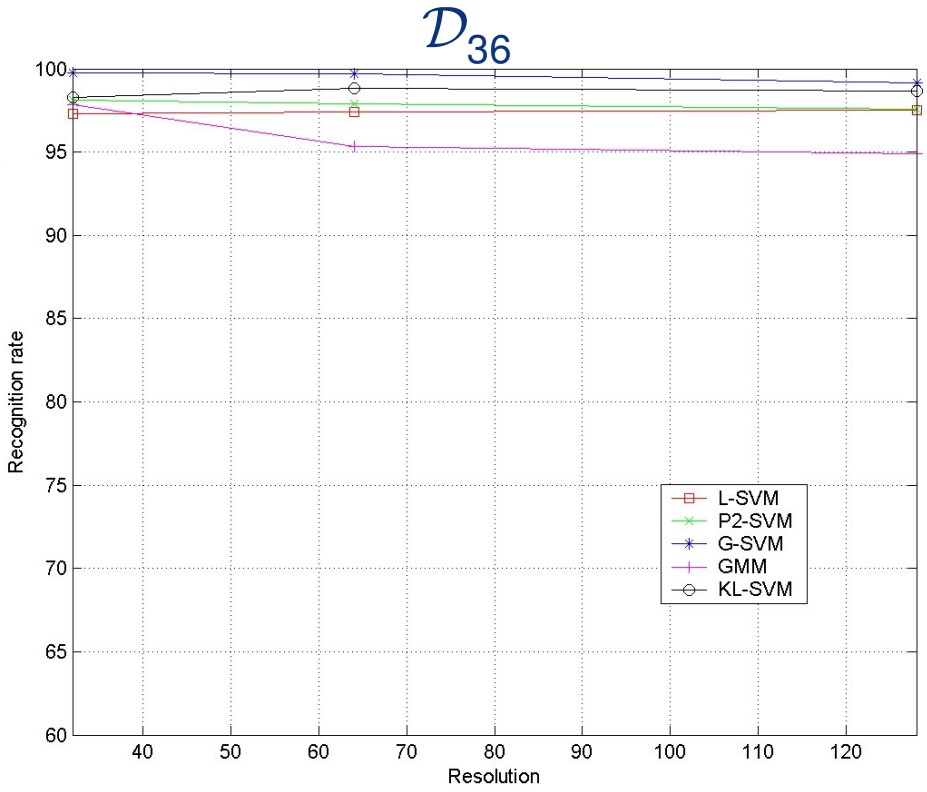


$\mathcal{D}_8$



Results


$\mathcal{D}_{18}$
Recognition rate
Resolution
L-SVM
P2-SVM
G-SVM
GMM
KL-SVM


$\mathcal{D}_{36}$
Recognition rate
Resolution
L-SVM
P2-SVM
G-SVM
GMM
KL-SVM

Observations

- holistic kernels: G-SVM clearly better
  - excellent when *n* is large, but drops quickly
  - for small n weaker than GMM!
- overall:
  - localized + discriminant (KL-SVM) is best
  - differences between KL-SVM and G-SVM as high as 10%
  - localized + weak (GMM) learner better than holistic + strong (G-SVM)
- conclusions:
  - localized is more invariant, leads to easier classification problem: weaker classifier (GMM) has better generalization!
  - resolution (higher dimensionality vs more image info):
    - losses of about 5% at lower resolution
    - KL-SVM much more robust than GMM

Flexibility

- discriminant attributes for recognition depend on task (e.g. shape better for digits, texture better for landscapes)
- KL kernel supports multiple representations
- comparison of representations based on
  - support: point-wise vs local appearance vs global appearance
  - color: grayscale vs color
- all experiments on $\mathcal{D}_4$, 128x128, compared
  - point-wise: KL-kernel ($\mathcal{X}^2$)+ histogram (16 bins/channel)
    histogram intersection (*Laplacian kernel*, Chapelle et al, trans. Neural Nets, 1999)
  - local: KL-kernel with GMM (8x8 windows)
  - global: G-SVM

Results

- color important cue for recognition on COIL
- the less localized the better: point-wise > local >> global
- localization/invariance trade-off:
  - color so discriminant that even invariance loss of 8x8 is too much
  - loss of holistic is so large that it performs quite poorly
  - on grayscale (less discriminant) localized does best
- conclusion: different representations perform best on different tasks, flexibility of KL-kernel is a great asset

	histogram-based	local appearance	global appearance
grayscale	χ^2 kernel: 71.72 Laplacian kernel: 69.90	KL-SVM: 84.32	G-SVM: 75.54
color	χ^2 kernel: 98.12 Laplacian kernel: 97.81	KL-SVM: 96.74	G-SVM: 84.90