Mixture Component Identification and Learning for Visual Recognition

Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson

Computer Vision and Active Perception laboratory (CVAP), KTH, Sweden

Abstract. The non-linear decision boundary between object and background classes - due to large intra-class variations - needs to be modelled by any classifier wishing to achieve good results. While a mixture of linear classifiers is capable of modelling this non-linearity, learning this mixture from weakly annotated data is non-trivial and is the paper's focus. Our approach is to identify the modes in the distribution of our positive examples by clustering, and to utilize this clustering in a latent SVM formulation to learn the mixture model. The clustering relies on a robust measure of visual similarity which suppresses uninformative clutter by using a novel representation based on the exemplar SVM. This subtle clustering of the data leads to learning better mixture models, as is demonstrated via extensive evaluations on Pascal VOC 2007. The final classifier, using a HOG representation of the global image patch, achieves performance comparable to the state-of-the-art while being more efficient at detection time.

1 Introduction

Object class detection and recognition is a major challenge within computer vision. It has been most successfully tackled with the approach: learn a discriminant function from labelled data sets of positive and negative examples [1]. The decisions about the form of this discriminant function and how it should be learnt are critical. These decisions require one to consider that the appearance of images of the same object class can vary significantly due to clutter, lighting, view-point of the camera and intra-class variation. There is also a strong bias imposed by photographers with their preferences for specific viewpoints and illuminations. These variations and biases lead to a multi-modal distribution of the positive class irrespective of representation. Combined with the almost uniform distribution of the negative class, this results in non-linear decision boundaries. This paper addresses this non-linearity with a mixture of discriminative functions which exploit the multi-modal nature of the positive class.

In order to be able to scale the method to large data set and reduce memory and computational costs of both the training and the testing phases, instead of using non-linear mappings of the data [2–4] or utilizing the invariances inherent in more complex representations e.g. [5], we focus on the use of mixture of linear discriminants. Here each classifier effectively distinguishes one mode of the positive class distribution from the background[1, 6]. This framework is attractive as

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 115–128, 2012.

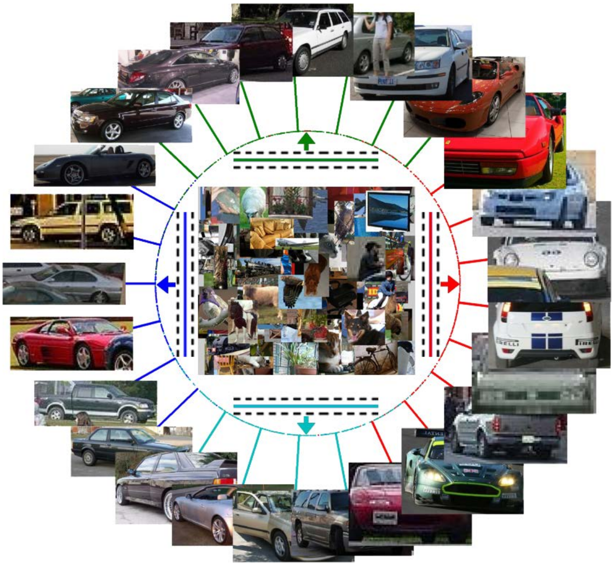

Fig. 1. The high level overview of our approach. We group visually similar positive instances together and for each cluster, learn a linear classifier which separates the cluster from all negative data. Each color represents a different cluster.

the simplicity of the component classifiers serve to regularize the overall classifier and avoid over-fitting.

However, learning such mixture of classifiers is not trivial when the association of each positive training example to a mode of its class distribution is unknown, the case when one only has weakly annotated data. How to achieve this learning robustly in this scenario is the main motivation of this paper. One can try to perform an optimization which simultaneously finds the assignment of each positive training example to a mixture and learns each discriminative classifier. But this is a non-convex and expensive optimization problem bedeviled by local minima. Instead we propose to de-couple the association of the positive examples to the mixture components and the discriminative learning of the classifiers.

We regard the problem as consisting of two stages. The first is associating each example with a mode - for which we use the term *Mixture Component Identification (MCI)* - while the second is learning the mixture of classifiers given the associations which we refer to as *Mixture Component Learning (MCL)*. Figure 1 illustrates our approach: we group visually similar positive samples of a class together and learn linear classifiers for each group of samples.

We show in the experimental section that such a grouping results in learning better classifiers per cluster which in turn improves the performance of a detection system. Extensive experiments are performed on the Pascal VOC-2007 data set where the configuration settings of our algorithm are thoroughly tested. The contributions of this work are: 1- to promote the use of unsupervised clustering - based on visual similarities - in mixture modeling, for the purpose of visual recognition and 2- to propose a new robust visual similarity measure using a representation derived from exemplar SVMs[7].

Following a review of the related work, the organization of the rest of the paper is: section 2 introduces our method, our experiments and results are described and interpreted in section 3 and the paper is concluded in section 4.

Related Work
Related to our work are all the works which address different sources of variations such as view point [8, 9], articulation [10] and sub-categories [11]. We aim to address the sources of variations without explicitly modelling any and without using any extra supervision, in a way that leads to better performance in the detection task. Therefore, we implement a discriminative framework - to perform well in the detection task - combined with a rather generative reasoning - to address the variations - for careful initialization of the discriminative model. A rather similar argument can be found in [12] and a similar approach for a different problem is taken in [13].

Previous works have often utilized mixture models and - either explicitly or implicitly - dedicated mixture components to modes of the aforementioned multi-modal distributions e.g. [1, 6, 12, 14]. Unlike the greedy optimization steps in boosting based approaches, we use the latent SVM formulation of [1] - which is essentially a mixture of linear SVMs - for our MCL step. The latent SVM formulation minimizes a convex objective once the latent variables, which include the data-component associations, are fixed. However, once the latent variables are allowed to vary, the problem is non-convex and is referred to as semi-convex [1]. This non-convexity makes the latent SVM initialization-dependent.

Most similar to our work is [12], which - in the unsupervised case - initializes a latent SVM using a clustering of the positive examples. In comparison to our work: 1- the similarity measure in [12] does not perform any feature selection and therefore is clutter sensitive, 2- the focus of [12] is view-point classification and therefore, very little experiment is done in the direction of object recognition, 3- the objective being minimized in [12] is slightly different: ℓ^2 regularization for large number of components leads to over-regularization for the same cache size; therefore the variables C_{Neg} and C_{Pos} are included in (3) of [12] which probably require extra cross-validation while Cs in our case are fixed for different number of components, thanks to the max regularization.

Unsupervised MCI is possible either by explicitly using a generative model or by unsupervised clustering of the positive data. Current approaches in the second direction include the clustering according to the Aspect Ratio of the bounding boxes [1], a combination of HOG and AR similarity [12] and the recent ensemble of exemplar SVM approach [7] which essentially treats each positive sample as a mixture component. The AR clustering is a very crude estimate of the visual similarity of the data and therefore, clusters based on aspect ratio do not necessarily contain visually similar samples. HOG based similarity - without feature selection - is sensitive to clutter, as it will be shown later in sections 2 and 3. Therefore, linear combination of the two - as suggested in [12] - cannot overcome the mentioned shortcommings. On the other hand, MCL based on one positive sample inherently cannot generalize well. We now describe how to measure and utilize visual similarity to group the positive data and learn a mixture model with one linear classifier per cluster which discriminates better than the former and generalizes better than the latter.

2 Visual Similarity Based Mixture Model Learning

2.1 Mixture Learning Framework

Our learning framework consists of two de-coupled steps: MCI and MCL. The MCI step, given a desired number of components c, assigns to each training example, x_i, a mixture component number $m_i \in \{1, \ldots, c\}$. We further describe the elements of the MCI step in sub-sections 2.3 and 2.2.

The MCL step, given the data-component associations, learns a model for each component using a latent SVM [1] formulation. The training data in this step consists of the following. There is a set of positive examples and their associated mixture components $\mathcal{D}_p = \{(x_1, m_1), \ldots, (x_N, m_N)\}$, a set $\mathcal{D}_n = \{x'_1, \ldots, x'_{N'}\}$ of negative examples and finally a set $\mathcal{Z}(x)$ containing all the candidate bounding boxes which overlap more than 50% with the annotated bounding box of x[1].

Let $\Phi(x, z)$ denote the modified HOG [15] feature vector of [1] extracted from the bounding box z. The MCL step learns the parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_c)$ by minimizing the objective function:

$$L(\boldsymbol{\beta}) = \frac{1}{2} \max_i \|\boldsymbol{\beta}_i\|^2 + C \sum_{i=1}^{N} \max\big(0, 1 - f^+_{\boldsymbol{\beta}}(x_i, m_i)\big) + C \sum_{i=1}^{N'} \max\big(0, 1 + f^-_{\boldsymbol{\beta}}(x'_i)\big) \quad (1)$$

where the scalar C controls the relative weight of the regularization with respect to the hinge loss and

$$f^+_{\boldsymbol{\beta}}(x, m) = \max_{z \in \mathcal{Z}(x)} \boldsymbol{\beta}_m \cdot \Phi(x, z), \qquad f^-_{\boldsymbol{\beta}}(x) = \max_m f^+_{\boldsymbol{\beta}}(x, m). \quad (2)$$

In (2), the data-mixture component associations ("m_i" s) for positive samples were fixed to those found in the MCI step. The "m_i" s can also be treated as latent variables. This increases the non-linearity of the objective function which in turn increases the number of local minima. However, we expect a careful initialization to result in better minima. This is empirically validated later in section 3.

We use a slightly modified version of [16] to optimize (1) and unless stated otherwise, we use the same parameters as in the original implementation.

2.2 Measuring Visual Similarity

To perform successful clustering one must have a good way of measuring similarity between examples. This is a tricky task as background and foreground

[1] Here, the set of valid bounding boxes should be a function of the dimensionality of the corresponding filter. This was neglected in the notations for the sake of brevity.

clutter affect the appearance of an object instance within its bounding box. Hence, to robustly measure the visual similarity between two examples from the same visual class one needs to disregard the irrelevant clutter.

We use the recently developed exemplar SVM [7] to suppress this clutter. The aim of the exemplar SVM is to learn a classifier which best separates a single positive example from the large set of negative examples. The classifier learnt based on this premise effectively performs feature selection on that particular example. It suppresses the uninformative detail inside the bounding box, see figure 3, which is not useful when discriminating it from the negative class. The exact details of how we robustly measure visual similarity now follow.

Let $\ = \{\mathbf{w}_i \mid i = 1, \dots, n\}$ be the set of n sparse basis filters (in this paper these filters correspond to the weights of the exemplar SVMs learnt for each training example). Each one is applied linearly to the feature extracted from the image patch in x defined by the bounding box z as $\mathbf{w}_i \cdot \Phi(x, z)$. A calibration process is then required to ensure the scores from the different basis filters are comparable. This is achieved with the sigmoid function and we define our basis functions as

$$F_i(x, z) = \frac{1}{1 + \exp(-\alpha_i(\mathbf{w}_i \cdot \Phi(x, z) - \gamma_i))} \tag{3}$$

where α_i and γ_i are the calibration parameters learnt as in [7][2] and $\mathbf{w}_i$ is the i-th sparse basis filter. Let $E_i(x)$ be the maximum score of $F_i(.,.)$ over the valid latent positions of x:

$$E_i(x) = \max_{z \in \mathcal{Z}(x)} F_i(x, z) \tag{4}$$

This maximization process corresponds to finding the best alignment over scale and translation, the search is over bounding boxes of different size and position, of the sparse filter with the test image patch and can be found via convolution.

If there is a one-to-one association between the basis functions and the positive training examples, which is the case if an exemplar SVM is trained for each positive example, we can directly use the bases to evaluate *visual structural similarity* between the i-th and j-th positive training instance. Assuming the same order for the bases and the positive examples in this case, we can define

$$K^E(x_i, x_j) = \frac{1}{2}\left(E_i(x_j) + E_j(x_i)\right) \tag{5}$$

where symmetry is achieved by averaging between two model responses. However, if a one-to-one association between the bases and the positive training samples does not exist or cannot be established, other measures need to be utilized as the K^E measure cannot be evaluated on such cases. Let $\mathbf{E}_x = (E_1(x), \dots, E_n(x))$ be the vector of all basis functions aligned and evaluated on x. With this new fixed length representation of x, we can utilize any kernel to measure similarity between two instances without directly associating either of the instances with the bases. Applying the Intersection Kernel on this representation, the visual similarity between two image patches becomes:

[2] We used the models provided by the authors.

$$K^E_{\text{MI}}(x,y) = \sum_{i=1}^{n} \min\left(E_i(x), E_i(y)\right) \tag{6}$$

As a specific example is usually visually similar to only a limited number of examples, averaging (mean pooling) the intersection measure on all the bases will unnecessarily smoothen out the responses. Therefore, if the responses of the bases are calibrated *with respect to each other*[3], we can make use of measures which are more sensitive to the responses of the bases. Therefore, we utilize ℓ^∞ on the intersection measures and define the K^E_{MMI} as the max pooling of the intersections:

$$K^E_{\text{MMI}}(x,y) = \max_i \min\left(E_i(x), E_i(y)\right) \tag{7}$$

Figure 2 shows the top nearest neighbors using each similarity measure evaluated on several classes. Similar to the results reported in [7], feature selection according to the exemplar SVMs results in better visual similarity measures which in turn leads to visually more appealing nearest neighbors. It is evident from the figure that unlike K^E_{MMI} which is sensitive to subtle variations in basis responses, the averaging behavior of K^E_{MI} does not result in visually appealing nearest neighbors if the class exhibits high variations.

Let $L = \frac{1}{N}\sum_{x\in\mathcal{D}_p} |\mathcal{Z}(x)|$ be the average number of latent positions over the positive training set and D be the average dimensionality of the linear weights of the basis filters . The computational complexity of evaluating a full affinity matrix using K^E is $\mathcal{O}(Dn^2L)$. Assuming the same number of bases as positives i.e. $N = n$, the computational complexity of evaluating a full affinity matrix using K^E_{MI} and K^E_{MMI} is $\mathcal{O}(Dn^2L + n^3)$. However, as usually $DL \gg n$, the dominating factor is still the convolutions which makes the computational complexity of all measures equivalent. We now describe how these similarity measures can be used to identify mixture components.

2.3 Mixture Component Identification via Unsupervised Clustering

With our similarity measure K, we can cluster our positive data using spectral clustering [17]. We construct fully connected similarity graphs and use the similarity measure as the affinity measure s.t. $\mathbf{W} = (w_{ij})$ and $w_{ij} = K(x_i, x_j)$. Let $\mathbf{L}_{sym}$ denote the symmetric normalized Laplacian:

$$\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \tag{8}$$

where $\mathbf{D}$ is the degree matrix - a diagonal matrix with diagonal entries $d_{ii} = \sum_j w_{ij}$. In order to identify c components, we compute the first $c+1$ eigenvectors $\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_c$ of $\mathbf{L}_{sym}$ and ignoring the first eigenvector, construct $\bar{\mathbf{U}} = (\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_c)$. Let $\mathbf{U}$ be the matrix obtained by normalizing the rows of $\bar{\mathbf{U}}$:

[3] We need to emphasize here that while the exemplar SVMs in [7] are not calibrated with respect to each other, we found out the independent calibrations to be sufficiently accurate to be used in K^E_{MMI} (7).

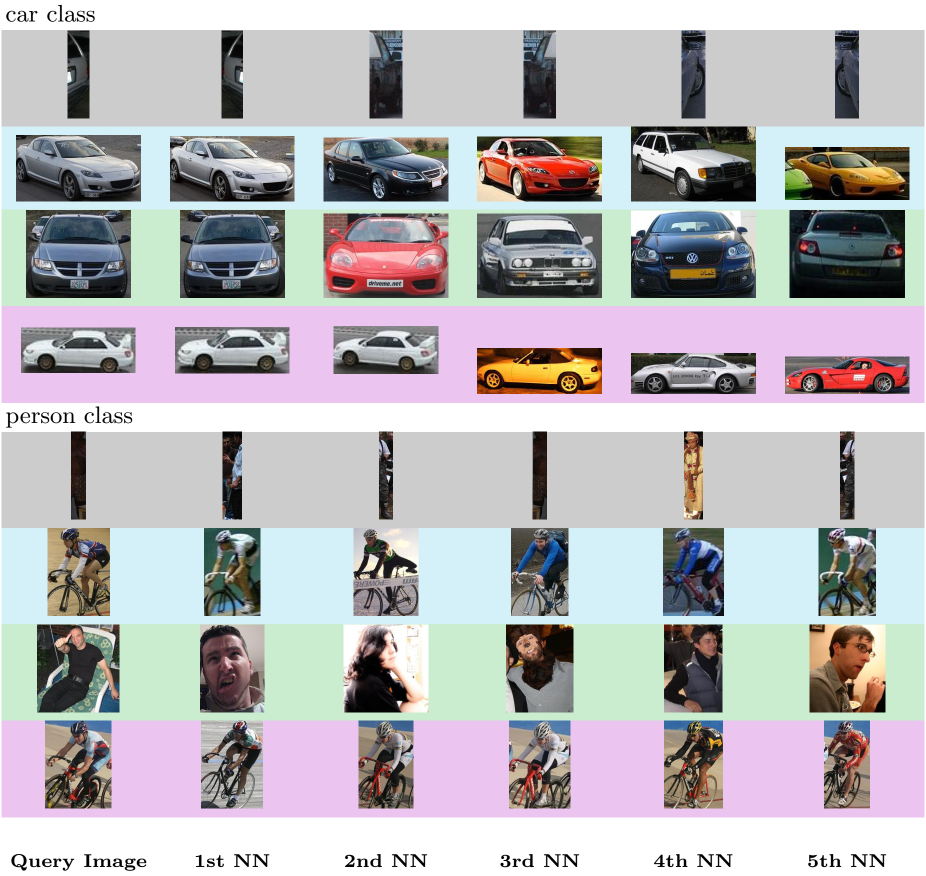

Fig. 2. Nearest neighbors produced by different visual similarity measures. The similarity measures within each block, from top to bottom are: HOG similarity without feature selection, K^E, K^E_{MI} and K^E_{MMI}. The leftmost column shows the image with highest similarity to its 10 nearest neighbors and to its right are its 5 nearest neighbors. Note how feature selection based on exemplar SVM results in better measures of visual similarity.

$u_{ij} = \bar{u}_{ij} / \left(\sum_k \bar{u}^2_{ik}\right)^{\frac{1}{2}}$. We refer to the i-th row of $\mathbf{U} \in \mathbb{R}^{n \times c}$ as $\mathbf{u}_i$ and the mapping - according to K and c - from x_i to $\mathbf{u}_i$ as the (c, K)-*spectral projection.*

The ℓ^2 distance is well suited to the spectral projection ($\mathbf{u}$) representation and therefore, as suggested in [17], k-means on this representation gives a good clustering of the data. The 2D coordinates of the instances in Figure 1 depict the $(2, K^E_{\mathrm{MMI}})$-spectral projection of a subset of the car examples. It can be observed that the ℓ^2 distance on this representation reflects the visual similarity between instances: points close in this space are expected to be visually similar. Because of this fact, we can measure the quality of a cluster by computing the average distance between two samples in the cluster. The colors in Figure 1 reflect the

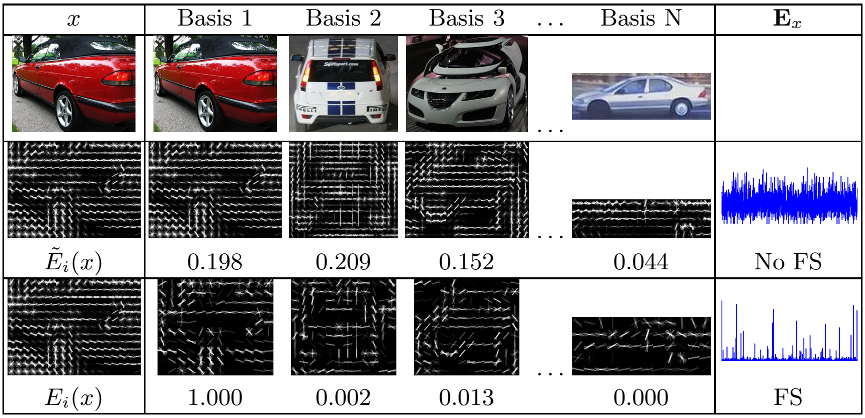

Fig. 3. Visualization of x projected onto a set of basis filters. In this figure the feature vector, $\Phi(x, z)$, extracted from example x is projected onto two different sets of basis filters. The first is a non-sparse basis and corresponds to the original HOG feature representation of each training example, while the second is a sparse one based on its exemplar SVM weight vector. The suppression of the clutter in the sparse basis allows for a more precise matching w.r.t. visual similarity (compare E_i s with $\tilde{E}_i$ s).

association of samples to the top 4 clusters from the 5 clusters produced by k-means on the $(5, K^E_{\text{MMI}})$-spectral projection of the data. The 5th cluster had a high average distance measure as it mainly contained everything which was not visually similar to samples of any of the other clusters and therefore, it was omitted for visualization purposes.

Example clusters found using the K^E_{MMI} similarity measure are shown in figure 4. Shown are the top 5 samples of the top 4 from the 5 clusters for four classes and the filters (the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_c$ from equation (1)) learned for each cluster. Here, the top sample refers to samples with the highest average visual similarity, using K^E_{MMI}, to all instances associated with the same component. The top cluster is considered as the cluster with the highest average visual similarity between the samples assigned to the cluster. It can be observed that the MCI step groups together examples that are visually similar.

It is worth noting that while the K^E and K^E_{MMI} visual similarity measures are not kernels i.e. they do not result in positive definite affinity matrices, they can be utilized in the spectral clustering as the spectral projection utilizes (the normalized version of) the largest eigenvalues of the affinity matrix. Let $\tilde{\mathbf{d}}$ refer to the vector of ordered eigenvalues of the $\mathbf{L}_{sym}$ and $\bar{\mathbf{d}}$ refer to the average $\tilde{\mathbf{d}}$ values over all classes in Pascal VOC 2007. Figure 5 shows $\bar{\mathbf{d}}$ and its derivative when using K^E and K^E_{MMI} as similarity measures. It can be observed that K^E results in higher rank affinity matrices leading to lower rank normalized Laplacians; which means that K^E_{MMI} is potentially preferable for coarser clusterings (less number of clusters). This is also experimentally validated later in Figure 6.

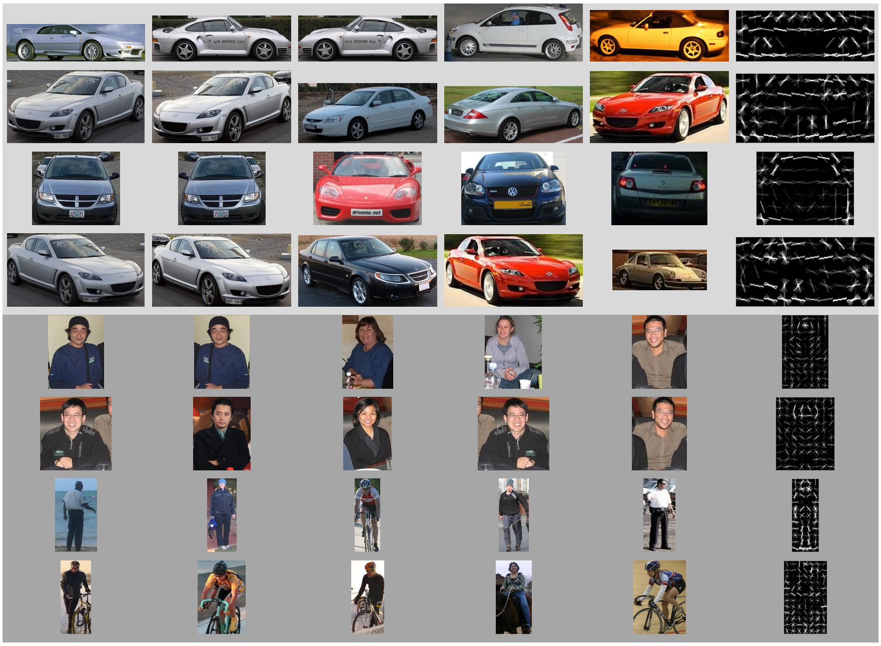

Fig. 4. Visualization of the clusters. Each row shows the top 5 samples of the top 4 clusters based on the highest average kernel similarity after $(5, K^E_{\mathrm{MMI}})$-spectral clustering of the car and person classes (see text for details). The last column depicts the positive weights of the model learnt for each cluster in the MCL step.

3 Experiments

Data Set: We evaluate our method on the Pascal VOC 2007 [18] data set, training on the train + validation set, and testing on the test set and using the Average Precision (AP) and mean Average Precision (mAP) as performance measures. We report the performance of the MCI + MCL framework based on different visual similarity measures and different number of mixture components. Therefore we review the visual similarity measures considered and our acronyms for them : 1- aspect ratio (AR) as a very crude measure of visual similarity, 2- visual similarity *without feature selection (HOG)*: linear kernel on HOG feature vectors with latency on the position and scale and 3- visual similarity *with feature selection* (K^E, K^E_{MI} and K^E_{MMI}). A "+L" in the results denotes an MCL step with latent data-component association, initialized from the data-component associations of the MCI step.

Performance vs. Number of Components: Figure 6 (left) shows the *mAP vs the number of mixture components* when different visual similarity measures are used in the MCI step. We point out the following observations: *1-* Clustering based on AR performs well only for low numbers of components i.e. 3 and 5 components. Unlike other visual similarity measures however, it fails to provide

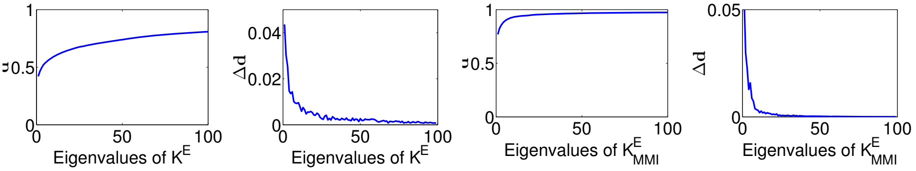

Fig. 5. Rank analysis of K^E and K^E_{MMI}: average of the (oredered) eigenvalues of $\mathbf{L}_{sym}$ ($\bar{\mathbf{d}}$) and its derivative ($\Delta\bar{\mathbf{d}}$) when using K^E and K^E_{MMI} as visual similarity measures.

good initializations when the non-linearity of the objective increases. *2*- Latent (positive) data-component association is beneficial almost consistently (with the exception of AR:5). The extra non-linearity introduced to the objective via this latent formulation is initialization dependent (compare K^E_{MMI}+L and AR+L). *3*- Feature selection in visual similarity measure improves the performance (compare HOG with K^E_X). *4*- The performance tends to improve when more mixture components are utilized in combination with MCI based on visual similarities.

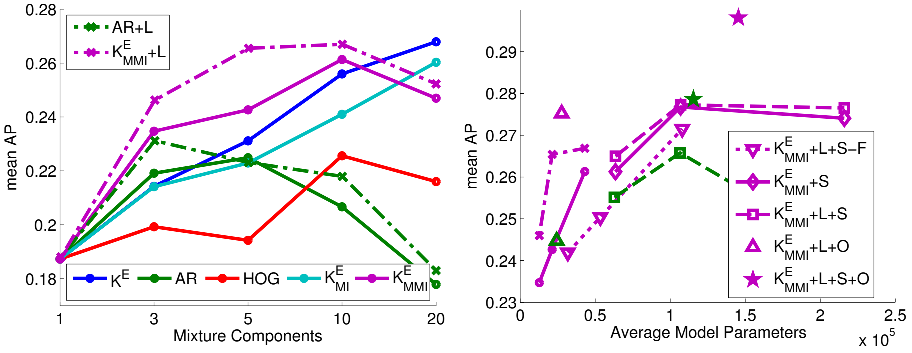

Fig. 6. The performance of the MCI + MCL framework using different visual similarity measures on Pascal VOC 2007 classes. Left: results achieved by varying the number of components for each visual similarity measure. Right: performances vs model complexity for 3, 5 and 10 component mixture models in different configurations(see text for details).

We did not experiment with higher number of mixture components mainly because of the computational expense. We observed, though, that the performance of K^E_{MMI} - which outperforms all other measures consistently up to and including 10 mixture components - degrades after 10 components while the smoother measures K^E and K^E_{MI} continue to benefit from more mixture components. The main reason of failure in these cases is the domination of the ℓ^2 distance in the k-means clustering (after the spectral projection step) by the eigenvectors associated with large eigenvalues of the normalized Laplacian (small eigenvalues of the affinity matrix which tend to be noisy). Addressing this issue is out of the scope of this work but, a potential solution is to use less eigenvectors than the desired number of clusters, in the spectral clustering step.

Performance vs. Model Complexity: Figure 6 (right) shows the performance of the MCI + MCL framework vs the models' parameters (averaged over the 20 classes) using K^E_{MMI} (magneta) and AR (green) visual similarities and for models with 3, 5 and 10 mixture components. In the figure, a "-F" refers to a model without the flip heuristic[4] and a "+S" refers to a finer (2 scale) HOG representation instead of a coarse representation (the same scale as the root filters in [16]). Additionally, the performance of the model is shown if an oracle were available to tell the model the optimal number of mixture components for each class (assign each class a number $c_i \in \{3, 5, 10\}$); shown with a "+O" in the legend entries.

The analysis of the figure is as follows: *1*- Models with the flip heuristic outperform equally complex models based on the same similarity measure and without the flip heuristic (compare K^E_{MMI}+L+S-F with the rest of models based on K^E_{MMI}). The reason for this is probably the reduced degrees of freedom imposed on the model using the flip heuristic which prevents model from over-fitting. *2*- The use of an oracle (the "+O" entries) improves the performance of a coarse representation by approximately 0.01 mAP and that of a fine representation by approximately 0.02 mAP in case of K^E_{MMI} and by 0.015 mAP and 0.012 mAP in case of AR. These are encouraging results for future work on adapting/estimating the number of mixture components and at the same time emphasize on the use of subtle visual similarity measures: K^E_{MMI}+L+O and K^E_{MMI}+L+S+O perform 0.03 mAP and 0.02 mAP better than their AR based counterparts. *3*- Fine scale representation improves the performance of K^E_{MMI} by approximately 0.01, but improves that of AR by 0.025 in case of 3 components and 0.035 in case of 5 components. Nevertheless, AR+L+S+O is only 0.003 better than K^E_{MMI}+L+O, while it is more than 4.1 times more complex!

Performance vs. Intra-class Variation: In order to analyze the performance of our models in presence of different bias and variation levels of the positive classes, we need to be able to approximate the intra-class variation[5]. In the following, we assumed the intra-class variation is negatively correlated with the performance of K^E_{MMI}+L+O and we made our arguments reasonably invariant to the actual measure we used to approximates intra-class variation by considering the ordering of the classes instead of the exact measured values. This makes the estimates invariant to any monotonic transformation of the measure. It is worth mentioning that similar overall conclusions can be drawn using other reasonable

[4] In [16], for each mixture component by default two filters are learnt that are flipped horizontally with respect to each other i.e. a 3 component mixture contains 6 (root) filters. This constraint essentially reduces the degrees of freedom in comparison to a model with the same number of filters without the flip constraint.

[5] Here, we neglect the effect of the inter-class variations.

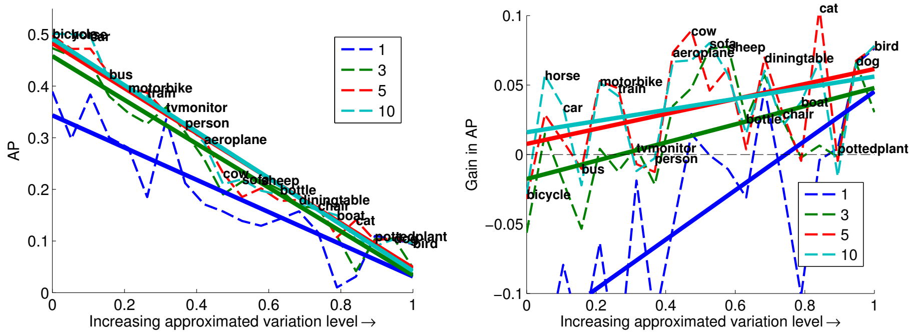

Fig. 7. Performance (of K^E_{MMI}) vs approximate intra-class variation level on (left) and AP gains in comparison to AR+L:3 (right).

measures e.g. the performance of a one component latent SVM model or the results of AR+L:3, leads to similar overall conclusions.

Figure 7 (left) shows how the performance of K^E_{MMI}+L decreases when intra-class variation increases. The solid lines are fitted to the actual data depicted by dashed lines via linear regression. Higher bias (simpler) models are expected to work better when intra-class variation is large and sufficient data is not available for the classifier to efficiently learn the discriminative structures. As expected, more complex models perform worse in presence of larger intra-class variation: slope of the lines increases when more mixture components are utilized and also, a 5 component model performs better than a 10 component model on classes with more intra-class variation than "diningtable". At the same time 1 and 3 component models are almost consistently outperformed by 5 and 10 components; except the last 3 classes: bird, dog and plant which probably require other representations, more data or more supervision.

Figure 7 (right) shows how K^E_{MMI}+L compares with AR+L:3. It can be observed that in all cases, the gain has a positive slope i.e. improvement gets more as intra-class variation increases. However, the slope decreases when the complexity of the model increases. Considering the slope and intercept, we can conclude that K^E_{MMI}+L with 5 and 10 components almost consistently outperform AR+L:3.

Comparison to Related Works: Table 1 shows the performance of the MCI+MCL framework using 2 configuration settings on each class of the data set compared to the ESVM approach and 3 part based models. It can be observed that without using parts, we outperform the state-of-the-art part based models - based on the HOG representation - in 2 classes and outperform 2 part based models in mean AP. It should be noted that although the training process is expensive for a visual similarity based MCI step, the testing phase consists of convolutions of linear filters learnt in the MCL step; without any dynamic programming step to account for deformation of the parts. This, without requiring a cascade or hierarchical model, is cheaper and better paralellizable compared to

Table 1. Results on the Pascal VOC 2007 data set. LDPM , CFHPM and DTDPM-R4 are part based models. Without any post-processing and without using parts, we outperform state of the art in 2 classes and two part based models in mean AP.

Method Class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	mAP
ESVM+Co-occ[7]	.208	.480	.077	.143	.131	.397	.411	.052	.116	.186	.227
LDPM [1]	.290	.546	.006	.134	**.262**	.394	.464	.161	.163	.165	.262
CFHPM [19]	.277	.540	.066	.151	.148	.442	.473	.146	.125	.220	.269
DTDPM-R4 [16]	.289	**.595**	**.100**	.152	.255	**.496**	**.579**	**.193**	**.224**	**.252**	**.323**
K^{E}_{MMI}+L:10	.290	.501	.096	.150	.189	.411	.497	.103	.160	.210	.267
$\mathbf{K}^{\mathbf{E}}_{\text{MMI}}$**+L+S+O**	**.333**	.536	.096	**.156**	.229	.488	.515	.163	.163	.200	.298
Method Class	table	dog	horse	bike	person	plant	sheep	sofa	train	monitor	mAP
ESVM+Co-occ[7]	.111	.031	.447	.394	.169	.112	.226	.170	.369	.300	.227
LDPM [1]	**.245**	.050	.436	.378	.350	.088	.173	.216	.340	.390	.262
CFHPM [19]	.242	**.120**	.520	.420	.312	.106	**.229**	.188	.353	.311	.269
DTDPM-R4 [16]	.233	.111	**.568**	**.487**	**.419**	**.122**	.178	**.336**	**.451**	**.416**	**.323**
K^{E}_{MMI}+L:10	.170	.103	.500	.396	.330	.090	.198	.220	.382	.343	.267
$\mathbf{K}^{\mathbf{E}}_{\text{MMI}}$**+L+S+O**	.238	.110	.553	.438	.369	.107	.227	.235	.386	.410	.298

part based models and more sophisticated approaches such as [2]. Furthermore, the same framework can potentially be utilized to train better root filters for any part-based model and to provide better initialization for their non-convex optimization.

Acknowledgements. This work has been funded by the Swedish Foundation for Strategic Research (SSF); within the project VINST (Wearable Visual Information Systems) and by the European Commission KIC: EIT ICT labs. We thank PDC for letting us use their supercomputers, without which the thorough experiments presented in this work were impossible to perform. We also would like to thank Carl Henrik Ek for the constructive discussions around the topic.

4 Conclusions

In this paper, we introduced the MCI + MCL mixture learning framework and promoted the use of visual similarity measures for the MCI step. We performed extensive evaluations of the proposed framework based on different visual similarity measures on the Pascal VOC 2007 data set. The framework achieved very promising results, outperforming the bases we used - the exemplar SVMs - in the detection task and 2 part based models without using parts.

Future work includes estimating the optimal number of clusters for each class, automatic refinement of the ”junk” clusters - clusters which contain samples not similar to those of any other cluster's; but not sharing any structural similarities, investigating the use of other methods for the purpose of feature selection, and learning the mixture of discriminants with methods other than the latent SVM.

References

1. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2010)
2. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
3. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV (2007)
4. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. PAMI (2011)
5. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV (2005)
6. Kim, T.K., Cipolla, R.: Mcboost: Multiple classifier boosting for perceptual coclustering of images and visual features. In: NIPS (2008)
7. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)
8. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: CVPR (2009)
9. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3d feature maps. In: CVPR (2008)
10. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
11. Bar-Hillel, A., Weinshall, D.: Subordinate class recognition using relational object models. In: NIPS (2006)
12. Gu, C., Ren, X.: Discriminative Mixture-of-Templates for Viewpoint Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 408–421. Springer, Heidelberg (2010)
13. Naderi Parizi, S., Oberlin, J.G., Felzenszwalb, P.F.: Reconfigurable models for scene recognition. In: CVPR (2012)
14. Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: ICCV (2005)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
16. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Discriminatively trained deformable part models, release 4, `http://people.cs.uchicago.edu/~pff/latent-release4/`
17. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS (2001)
18. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007)
19. Pedersoli, M., Vedaldi, A., Gonzalez, J.: A coarse-to-fine approach for fast deformable object detection. In: CVPR (2011)