Discriminative Decorrelation for Clustering and Classification*

Bharath Hariharan[1], Jitendra Malik[1], and Deva Ramanan[2]

[1] Univerisity of California at Berkeley, Berkeley, CA, USA
{bharath2,malik}@cs.berkeley.edu
[2] University of California at Irvine, Irvine, CA, USA
dramanan@ics.uci.edu

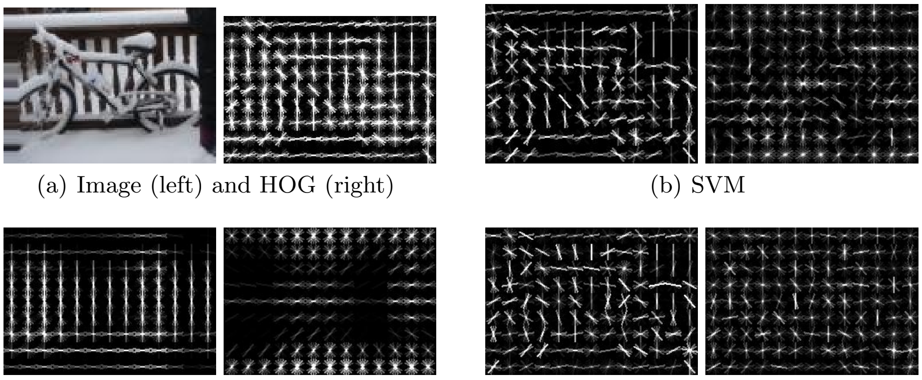
Abstract. Object detection has over the past few years converged on using linear SVMs over HOG features. Training linear SVMs however is quite expensive, and can become intractable as the number of categories increase. In this work we revisit a much older technique, viz. Linear Discriminant Analysis, and show that LDA models can be trained almost trivially, and with little or no loss in performance. The covariance matrices we estimate capture properties of natural images. Whitening HOG features with these covariances thus removes naturally occuring correlations between the HOG features. We show that these whitened features (which we call WHO) are considerably better than the original HOG features for computing similarities, and prove their usefulness in clustering. Finally, we use our findings to produce an object detection system that is competitive on PASCAL VOC 2007 while being considerably easier to train and test.

1 Introduction

Over the last decade, object detection approaches have converged on a single dominant paradigm: that of using HOG features and linear SVMs. HOG features were first introduced by Dalal and Triggs [1] for the task of pedestrian detection. More contemporary approaches build on top of these HOG features by allowing for parts and small deformations [2], training separate HOG detectors for separate poses and parts [3] or even training separate HOG detectors for each training exemplar [4].

Figure 1(a) shows an example image patch of a bicycle, and a visualization of the corresponding HOG feature vector. Note that while the HOG feature vector does capture the gradients of the bicycle, it is dominated by the strong contours of the fence in the background. Figure 1(b) shows an SVM trained using just this image patch as a positive, and large numbers of background patches as negative [4]. As is clear from the figure, the SVM learns that the gradients of the fence are unimportant, while the gradients of the bicycle are important.

* This work was funded by ONR-MURI Grant N00014-10-1-0933 and NSF Grant 0954083.


(a) Image (left) and HOG (right)

(b) SVM

(c) PCA

(d) LDA

Fig. 1. Object detection systems typically use HOG features, as in (a). HOG features however are often swamped out by background gradients. A linear SVM learns to stress the object contours and suppress background gradients, as in (b), but requires extensive training. An LDA model, shown in (d), has a similar effect but with negligible training. PCA on the other hand completely kills discriminative gradients, (c). The PCA, LDA and SVM visualizations show the positive and negative components separately, with the positive components on the left and negative on the right.

However, training linear SVMs is expensive. Training involves expensive bootstrapping rounds where the detector is run in a scanning window over multiple negative images to collect “hard negative” examples. While this is feasible for training detectors for a few tens of categories, it will be challenging when the number of object categories is of the order of tens of thousands, which is the scale in which humans operate.

However, linear SVMs aren’t the only linear classifiers around. Indeed, Fisher proposed his linear discriminant as far back as 1936 [5]. Fisher discriminant analysis tries to find the direction that maximizes the ratio of the between-class variance to the within-class variance. Linear discriminant analysis (LDA) is a generative model for classification that is equivalent to Fisher’s discriminant analysis if the class covariances are assumed to be equal. Textbook accounts of LDA can be found, for example, in [6, 7]. Given a training dataset of positive and negative features (x, y) with $y \in \{0, 1\}$, LDA models the data x as generated from class-conditional Gaussians:

$$P(x, y) = P(x|y)P(y) \quad \text{where} \quad P(y = 1) = \pi \quad \text{and} \quad P(x|y) = N(x; \mu_y, \Sigma)$$

where means μ_y are class-dependent but the covariance matrix Σ is class-independent. A novel feature x is classified as a positive if $P(y = 1|x) > P(y = 0|x)$, which is equivalent to a linear classifier with weights given by $w = \Sigma^{-1}(\mu_1 - \mu_0)$. Figure 1(d) shows the LDA model trained with the bicycle image patch as positive and generic image patches as background. Clearly, like the SVM, the LDA model suppresses the contours of the background, while enhancing the gradients of the bicycle. LDA has been used before in computer vision, one of the earliest and most popular appications being face recognition [8].

Training an LDA model requires figuring out the means μ_y and Σ. However, unlike an SVM which has to be trained from scratch for every object category, we show that μ_0 (corresponding to the background class) and Σ can be estimated just once, and reused for all object categories, making training almost trivial. Intuitively, LDA computes the average positive feature μ_1, centers it with μ_0, and "whitens" it with Σ^{-1} to remove correlations. The matrix Σ acts as a model of HOG patches of natural images. For instance, as we show in section 2, this matrix captures the fact that adjacent HOG cells are highly correlated owing to curvilinear continuity. Thus, not all of the strong vertical gradients in the HOG cells of Figure 1(a) are important: many of them merely reflect the continuity of contours. Removing these correlations therefore leaves behind just the discriminative gradients.

The LDA model is just the difference of means in a space that has been whitened using the covariance matrix Σ. This suggests that this whitened space might be significant outside of just training HOG classifiers. In fact, we find that dot products in this whitened space are more indicative of visual similarity than dot products in HOG space. Consequently, clustering whitened HOG feature vectors (which we call WHO for Whitened Histogram of Orientations) gives more coherent and often semantically meaningful clusters.

Principal components analysis (PCA) is a related method that has been explored for tasks such as face recognition [9] and tools for dimensionality reduction in object recognition [10]. In particular, Ke and Sukthankar [11] and Schwartz et al [12] examine (linear) low-dimensional projections of oriented gradient features. In PCA, the data is projected onto the directions of the most variation, and the directions of least variation are ignored. However, for our purposes, the directions that are ignored are often those that are the most discriminative. Figure 1(c) shows the result of projecting the data down to the top 30 principal components. Clearly, this is even worse than the original HOG space: contours of the bicycle are more or less completely discarded. Our observations mirror those of Belhumeur et al [8] who showed that in the context of face recognition, the directions retained by PCA often correspond to variations in illumination and viewing direction, rather than variations that would be discriminative of the identity of the face. [8] conclude that Fisher's discriminant analysis outperforms PCA on face recognition tasks. In section 4 we show concretely that the low dimensional subspace chosen by PCA is significantly worse than whitened HOG as far as computing similarity is concerned.

Our aim in this paper is therefore to explore the advantages provided by whitened HOG features for clustering and classification. In section 2 we go into the details of our LDA models, describing how we obtain our covariance matrix, and the properties of the matrix. Section 3 describes our first set of experiments on the INRIA pedestrian detection task, showing that LDA models can be competitive with linear SVMs. Section 4 outlines how WHO features can be used for clustering exemplars. We then use these clusters to train detectors, and evaluate the performance of the LDA model vis-a-vis SVMs and other choices in section 5. In section 6 we tie it all together to produce a final object detection system that performs competitively on the PASCAL VOC 2007 dataset, while being orders-of-magnitude faster to train (due to our LDA classifiers) and orders-of-magnitude faster to test (due to our clustered representations).

2 Linear Discriminant Analysis

In this section, we describe our model of image gradients based on LDA. For our HOG implementation, we use the augmented HOG features of [2]. Briefly, given an image window of fixed size, the window is divided into a grid of 8×8 cells. From each cell we extract a feature vector x_{ij} of gradient orientations of dimensionality $d = 31$. We write $x = [x_{ij}]$ for the final window descriptor obtained by concatenating features across all locations within the window. If there are N cells in the window, the feature vector has dimensionality Nd.

The LDA model is a linear classifier over x with weights given by $w = \Sigma^{-1}(\mu_1 - \mu_0)$. Here Σ is an $Nd \times Nd$ matrix, and a naive approach would require us to estimate this matrix again for every value of N and also for every object category. In what follows we describe a simple procedure that allows us to learn a Σ and a μ_0 (corresponding to the background) once, and then reuse it for every window size N and for every object category. Given a new object category, we need only a set of positive features which are averaged, centered, and whitened to compute the final linear classifier.

2.1 Estimating μ_0 and Σ

Object-independent backgrounds: Consider the task of learning K 1-vs-all LDA models from a multi-class training set spanning K objects and background windows. One can show that the maximum likelihood estimate of Σ is the sample covariance estimated across the entire training set, ignoring class labels. If we assume that the number of instances of any one object is small compared to the total number of windows, we can similarly define a generic μ_0 that is independent of object type. This means that we can learn a generic μ_0 and Σ from *unlabeled* windows, and this need not be done anew for every object category.

Marginalization: We are now left with the task of estimating a μ_0 and Σ for every value of the window size N. However, note that the statistics of smaller-size windows can be obtained by marginalizing out statistics of larger-size windows. Gaussian distributions can be marginalized by simply dropping the marginalized variables from μ_0 and Σ. This means that we can learn a single μ_0 and Σ for the largest possible window of N_0 cells, and generate means and covariances for smaller window sizes "on-the-fly" by selecting subpartitions of μ_0 and Σ. This reduces the number of parameters to be estimated to an $N_0 d$ dimensional μ_0 and an $N_0 d \times N_0 d$ matrix Σ.

Scale and translation invariance: Image statistics are largely scale and translation invariant [13]. We achieve such invariance by including training windows extracted from different scales and translations. We can further exploit translation invariance, or stationarity in statistical terms, to reduce the number of model parameters. To encode a stationary μ_0, we compute the mean HOG feature $\mu = E[x_{ij}]$, averaged over all features x and cell locations (i, j). μ_0 is just μ replicated over all N_0 cells.

Write Σ as a block matrix with blocks $\Sigma_{(ij),(lk)} = E[x_{ij}x_{lk}^T]$. We then incorporate assumptions of translation invariance by modeling Σ with a *spatial autocorrelation function* [14]:

$$\Sigma_{(ij),(lk)} = \Gamma_{(i-l),(j-k)} = E[x_{uv}x^T_{(u+i-l),(v+j-k)}] \tag{1}$$

where the expectation is over cell locations (u, v) and gradient features x. In other words, we assume that $\Sigma_{(ij),(kl)}$ depends only on the relative offsets $(i-k)$ and $(j-l)$. Thus instead of estimating an $N_0d \times N_0d$ matrix Σ, we only have to estimate the $d \times d$ matrices $\Gamma_{s,t}$ for every offset (s, t). For a spatial window with N_0 cells, there exist only N_0 distinct relative offsets. Thus we only need to estimate $O(N_0d^2)$ parameters.

We now estimate μ and the matrices $\Gamma_{s,t}$ from *all* subwindows extracted from a large set of unlabeled, 10,000 natural images (the PASCAL VOC 2010 dataset). This computation can be done once and for all, and the resulting μ and Γ stored. Then, given a new object category, μ_0 can be reconstructed by replicating μ over all the cells in the window and Σ can be reconstructed from Γ using (1).

Regularization: Even given this large training set and our $O(N)$ parametrization, we found Σ to be low-rank and non-invertible. This implies that it would be even more difficult to learn a separate covariance matrix for each positive class because we have much fewer positive examples, further motivating a single-covariance assumption. In general, it is difficult to learn high-dimensional covariance matrices [14]. For typical-size N values, Σ can grow to a $10,000 \times 10,000$ matrix. One solution is to enforce conditional independence assumptions with a Gaussian Markov random field; we discuss this further below. In practice, we regularized the sample covariance by adding a small value ($\lambda = .01$) to its diagonal, corresponding to an isotropic prior on Σ.

2.2 Properties of the covariance matrix

WHO: We define a whitened histograms of orientations (WHO) descriptor as $\hat{x} = \Sigma^{-1/2}(x - \mu_0)$. The transformed feature vector $\hat{x}$ then has an isotropic covariance matrix. An alternative interpretation of the linear discriminant is that w computes the difference between the average positive and negative features in WHO space. Such descriptors maybe useful for clustering because euclidean distances are more meaningful in this space. We explore this further in section 4. We use a cholesky decomposition $RR^T = \Sigma$ and Gaussian elimination (Matlab's blackslash) to efficiently compute this whitening transformation.

Analysis: We examine the structure of Σ in Fig.2. Intuitively, Σ encodes generic spatial statistics about oriented gradients. For example, due to curvilinear continuity, we expect a strong horizontal gradient response to be correlated with a strong response at a horizontally-adjacent location. Multiplying gradient features by Σ^{-1} subtracts off such correlated measurements. Because Σ^{-1} is sparse, features need only be de-correlated with adjacent or nearby spatial locations. This in turn suggests that image gradients can be fit will with a 3rd or 4th-order spatial Markov model, which may make for easier estimation and faster computations. A spatial Markov assumption makes intuitive sense; given we see a strong horizontal gradient at a particular location, we expect to see a strong gradient to its right regardless of the statistics to its left. We experimented with such sparse models [15], but found an unrestricted Σ to work well and simpler to implement.

Implications: Our statistical model, though quite simple, has several implications for scanning-window templates. (1) One should learn templates of larger spatial extent than the object. For example, a 2^{nd}-order spatial Markov model implies that one should score gradient features two cells away from the object border in order to de-correlate features. Intuitively, this makes sense; a pedestrian template wants to find vertical edges at the side of the face, but if it also finds vertical edges above the face, then this evidence maybe better explained by the vertical contour of a tree or doorway. Dalal and Triggs actually made the empirical observation that larger templates perform better, but attributed this to local context [1]; our analysis suggests that decorrelation may be a better explanation. (2) Current strategies for modeling occlusion/truncation by "zero"ing regions of a template may not suffice [16, 17]. Rather, our model allows us to properly marginalize out such regions from μ and Σ. The resulting template w will not be equivalent to a zero-ed out version of the original template, because the de-correlation operation must change for gradient features near the occluded/truncated regions.

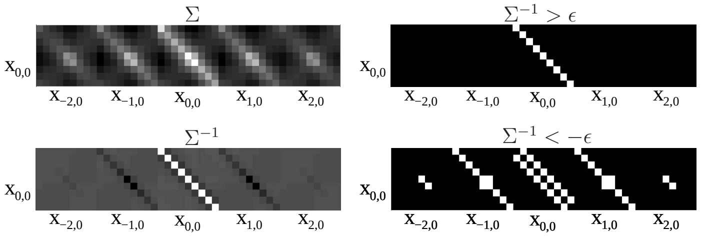
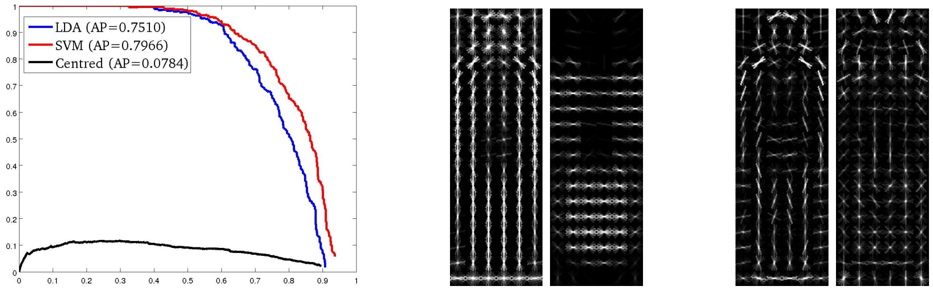


Fig. 2. We visualize correlations between 9 orientation features in horizontally-adjacent HOG cells as concatenated set of 9 × 9 matrices. Light pixels are positive while dark pixels are negative. We plot the covariance and precision matrix on the **left**, and the positive and negative values of the precision matrix on the **right**. Multiplying a HOG vector with Σ^{-1} decorrelates it, subtracting off gradient measurements from adjacent orientations and locations. The sparsity pattern of Σ^{-1} suggests that one needs to decorrelate features only a few cells away, indicating that gradients maybe well-modeled by a low-order spatial Markov model.



(a) AP (b) Centered (c) LDA

Fig. 3. The performance (AP) of the LDA model and the centered model (LDA without whitening) vis-a-vis a standard linear SVM on HOG features. We also show the detectors for the centered model and the LDA model.

3 Pedestrian detection

HOG feature vectors were first described in detail in [1], where they were shown to significantly outperform other competing features in the task of pedestrian detection. This is a relatively easy detection task, since pedestrians don't vary significantly in pose. Our local implementation of the Dalal-Triggs detector achieves an average precision (AP) of 79.66% on the INRIA dataset, outperforming the original AP of 76.2% reported in Dalal's thesis [18]. We think this difference is due to our SVM solver, which implements multiple passes of data-mining for hard negatives. We choose this task as our first test bed for WHO features.

We use our LDA model to train a detector and evaluate its performance. Figure 3 shows our performance compared to that of a standard linear SVM on HOG features. We achieve an AP of 75.10%. This is slightly lower than the SVM performance, but nearly equivalent to the original performance of [18]. However, note that compared to the SVM model, the LDA model is estimated only from a few positive image patches and neither requires access to large pools of negative images nor involves any costly bootstrapping steps. Given this overwhelmingly reduced computation, this performance is impressive.

Constructing our LDA model from HOG feature vectors involves two steps, i.e, subtracting μ_0 (centering) and multiplying by Σ^{-1} (whitening). To tease out the contribution of whitening, we also evaluate the performance when the whitening step is removed. In other words, we consider the detector formed by simply taking the mean of the centered positive feature vectors. We call this the "centered model", and its performance is indicated by the black curve in Figure 3. It achieves an AP of less than 10%, indicating that whitening is crucial to performance. We also show the detectors in Figure 3, and it can be clearly seen that the LDA model does a better job of identifying the discriminative contours (the characteristic shape of the head and shoulders) compared to simple centering.

4 Clustering in WHO space

Owing to large intra-class variations in pose and appearance, a single linear classifier over HOG feature vectors can hardly be expected to do well for generic object detection. Hence many state of the art methods train multiple "mixture components", multiple "parts" or both [3, 2]. These mixture components and parts are either determined based on extra annotations [3], or inferred as latent variables during training [2]. [4] consider an extreme approach and consider each positive example as its own mixture component, training a separate HOG detector for each example.

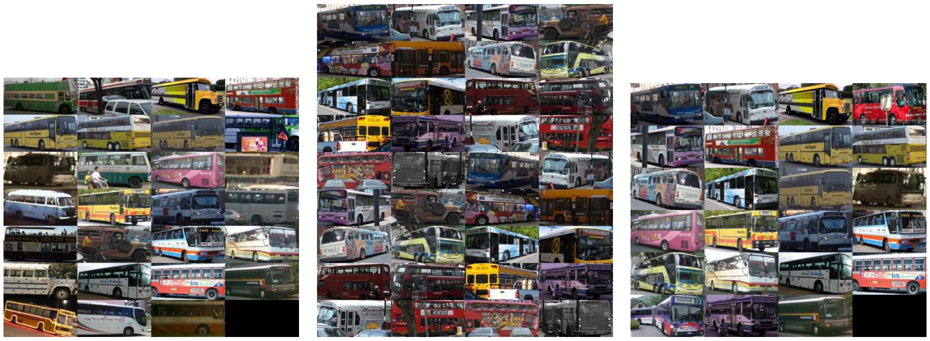
In this section we consider a cheaper and simpler strategy of producing components by simply clustering the feature vectors. As a test bed we use the PASCAL VOC 2007 object detection dataset (train+val) [19]. We first cluster the exemplars of a category using kmeans on aspect ratio. Then for each cluster, we resize the exemplars in that cluster to a common aspect ratio, compute feature vectors on the resulting image patches and finally subdivide the clusters using recursive normalized cuts [20]. The affinity we use for N-cuts is the exponential of the cosine of the angle between the two feature vectors.

We can either cluster using HOG feature vectors or using WHO feature vectors ($\hat{x} = \Sigma^{-1/2}(x - \mu_0)$, see section 2). Alternatively, we can use PCA to project HOG features down to a low dimensional space (we use 30 dimensions), and cluster in that space. Figure 4 shows an example cluster obtained in each case for the 'bus' category. The cluster based on WHO features is in fact semantically meaningful, capturing buses in a particular pose. HOG based clustering produces less coherent results, and the cluster becomes significantly worse when performed in the dimensionality-reduced space. This is because as Figure 1 shows, HOG overstresses background, whereas whitening removes the correlations common in natural images, leaving behind only discriminative gradients. PCA goes the opposite way and in fact *removes* discriminative directions, making matters worse. Figure 5 shows some more examples of HOG-based clusters and WHO-based clusters. Clearly, the WHO-based clusters are significantly more coherent.

5 Training each cluster

We now turn to the task of training detectors for each cluster. Following our experiments in section 3, we have several choices:

1. Train a linear SVM for each cluster, using the images of the cluster as positives, and image patches from other categories/background as negatives (SVM on cluster).
2. Train an LDA model on the cluster, i.e, use $w = \Sigma^{-1}(x_{mean} - \mu_0)$ (LDA on cluster).
3. Take the mean of the centered HOG features of the patches in the cluster, i.e use $w = x_{mean} - \mu_0$ ("centered model" on cluster).



(a) HOG (b) PCA (c) WHO

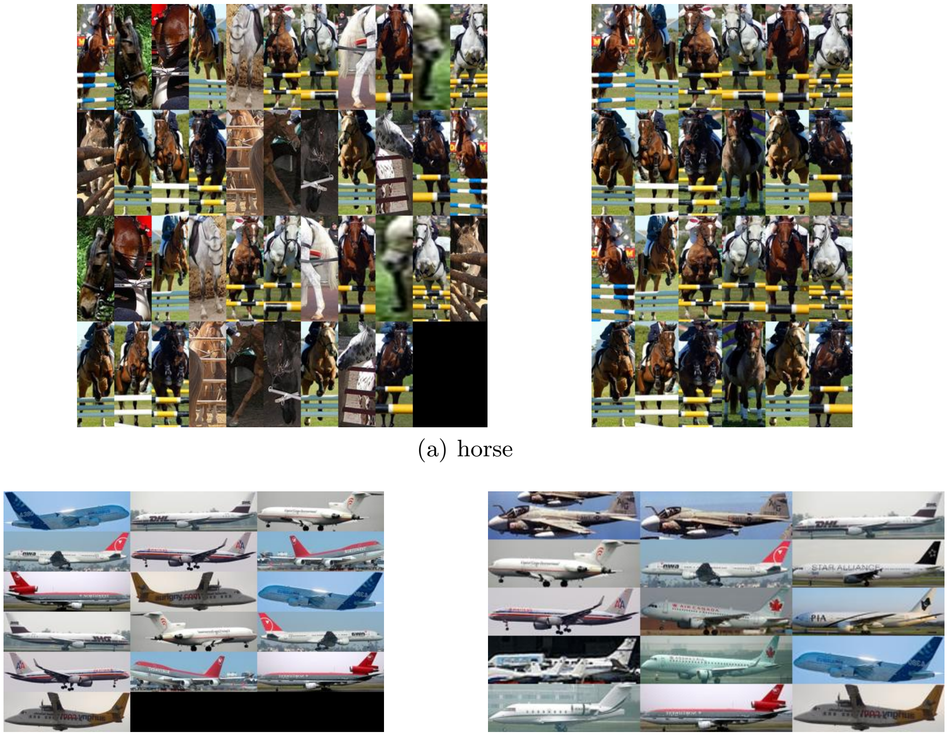
Fig. 4. Clusters obtained using N-cuts using HOG feature vectors, HOG vectors projected to a PCA basis and WHO feature vectors. Observe that while all clusters make mistakes, the HOG-based cluster is much less coherent than the WHO-based cluster. The PCA cluster is even less coherent than the HOG-based cluster.

[4] treat each exemplar separately, and get their boost from training to discriminate each exemplar from the background. On the other hand we believe that we can get bigger potential gains by averaging over multiple positive examples. In order to evaluate this, we also consider the following choices:

4. Train an LDA model on just the medoid, i.e $w = \Sigma^{-1}(x_{medoid} - \mu_0)$ (LDA on the medoid).
5. Take the medoid of the cluster and train a linear SVM, using the medoid as positive and image patches from other categories/background as negative.

We take the clusters obtained as described in the previous section for three categories : horse, motorbike and bus. For each cluster we train detectors according to the five schemes above. We then run each detector on the test set of PASCAL VOC 2007, and compute its AP. The ground truth for each cluster consists of all objects of that category.

Table 1 shows a summary comparison of the five schemes, and Figure 6 compares the performance of the LDA model with the other four schemes in more detail. First note that both single-example schemes perform worse than the LDA model. Indeed, for all but 6 of the 77 clusters tested, the LDA model achieves a higher AP than a single SVM trained using the medoid. This clearly shows that simple averaging over similar positive examples helps more than explicitly training to discriminate single exemplars from the background. This also provides an indirect validation of our clustering step, since it indicates that each cluster is coherent enough to be better than any single individual example. In our experimental results, we further quantitatively evaluate our clusters by demonstrating that they perform similarly to "brute-force" methods that train a separate exemplar template for every member of every cluster [4]. Our clustered representation performs similarly while being faster to evaluate.



(a) horse

(b) aeroplane

Fig. 5. Examples of clusters obtained for aeroplane and horse using HOG feature vectors (left) and WHO feature vectors (right). Note how the clusters based on WHO are significantly more coherent than the clusters based on HOG.

Secondly, observe that on average the performance of the LDA model is very similar to the performance of a linear SVM, and is also highly correlated with it. This reiterates our observations on the pedestrian detection task in section 3. This also indicates that our LDA model can be used in place of SVMs for HOG based detectors with little or no loss in performance, at a fraction of the computational cost and with very little training data.

Finally, the performance of the centered model without whitening is much lower than the LDA model, and is in fact significantly worse than even the single-example models. This again shows that decorrelation, and not just centering, is crucial for performance.

6 Combining across clusters

In this section we attempt to tie the previous two sections together to produce a full object detection system. We compare here to the approach of [4], who show competitive performance on PASCAL VOC 2007 by simply training one

	LDA on cluster	SVM on cluster	LDA on medoid	SVM on medoid	Centered
Mean AP	7.59 ± 4.86	6.75 ± 4.80	4.84 ± 4.13	4.05 ± 4.12	0.74 ± 2.02
Median AP	9.25 ± 3.86	9.16 ± 4.04	4.65 ± 3.71	2 ± 3.6	0.06 ± 0.7

Table 1. Mean and median AP (in %) of the different models.

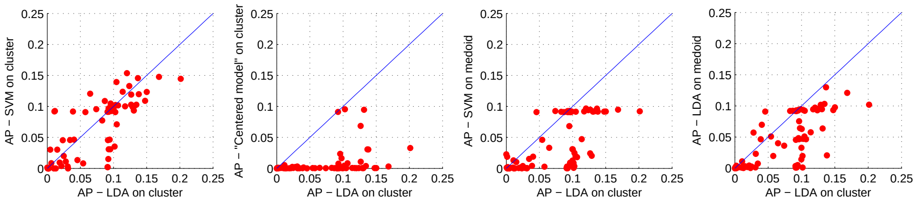


Fig. 6. Performance (AP) of the LDA model compared to (from left to right) an SVM trained on the cluster, the centered model trained on the cluster, an SVM trained on the medoid and an LDA model trained on the medoid. The blue line is the $y = x$ line. The LDA performs significantly better than both the single-example approaches and is comparable to an SVM trained on the cluster.

linear SVM per exemplar. This performance is impressive given that they use only HOG features and do not have any parts [2, 3].

We agree with them on the fact that using multiple components instead of single monolithic detectors is necessary for handling the large intra-class variation. However, training a separate SVM for each positive example entails a huge computational complexity. Because the negative class for each model is essentially the background, one would ideally learn background statistics just once, and simply plug it in for each model.

LDA allows us to do precisely that. Background statistics in the form of Σ and μ are computed just once, and training only involves computing the mean of the positive examples. This reduces the computational complexity drastically: using LDA we can train all exemplar models of a particular category on a single machine in a few minutes. Table 2 shows how exemplar-LDA models compare to exemplar-SVMs [4]. As can be seen, there is little or no drop in performance.

Replacing SVMs by LDA significantly reduces the complexity at train time. However at test time, the computational complexity is still high because one has to run a very large number of detectors over the image. We can reduce this computational complexity considerably by first clustering the positive examples as described in Section 4. We then train one detector for each cluster, resulting in far fewer detectors. For instance, the 'horse' category has 403 exemplars but only 29 clusters.

To build a full object detection system, we need to combine these cluster detector outputs in a sensible way. Following [4], we train a set of rescoring functions that rescore the detections of each detector. Note that only detections that score above a threshold are rescored, while the rest are discarded.

We train a separate rescoring function for each cluster. For each detection, we construct two kinds of features. The first set of features considers the dot product of the WHO feature vector of the detection window with the WHO feature vector of every exemplar in the cluster. This gives us as many features as there are examples in the cluster. These features encode the similarity of the detection window with the purported "siblings" of the detection window, namely the exemplars in the cluster.

The second set of features is similar to context features as described in [4, 3]. We consider every other cluster and record its highest scoring detection that overlaps by more than 50% with this detection window. These features record the similarity of the detection window to other clusters and allow us to boost scores of similar clusters and suppress scores of dissimilar clusters.

These features together with the original score given by the detector form the feature vector for the detection window. We then train a linear SVM to predict which detection windows are indeed true positives, and fit a logistic to the SVM scores. At test time the detections of each cluster detector are rescored using these second-level classifiers, and then standard non-max suppression is performed to produce the final, sparse set of detections. Note that this second level rescoring is relatively cheap since only detection windows that score above a threshold are rescored. Indeed, our cluster detectors can be thought of as the first step of a cascade, and significantly more sophisticated methods can be used to rescore these detection windows.

As shown in Table 2, our performance is very close to the performance of the Exemplar SVMs. This is in spite of the fact that our first-stage detectors require no training at all, and our second stage rescoring functions have an order of magnitude fewer parameters than ESVM+Co-occ [4] (for instance, for the horse category, in the second stage we have fewer than 2000 parameters, while ESVM+Co-occ has more than 100000). Although our performance is lower than part-based models [2], one could combine such approaches and possibly train parts with LDA.

Finally, each detection of ours is associated with a cluster of training exemplars. We can go further and associate each detection to the closest exemplar in the cluster, where distance is defined as cosine distance in WHO space. This allows us to match each detection to an exemplar, as in [4]. Figure 7 shows examples of detections and the training exemplars they are associated with. As can be seen, the detections are matched to very similar and semantically related exemplars.

7 Conclusion

Correlations are naturally present in features used in object detection, and we have shown that significant advantages can be derived by accounting for these correlations. In particular, LDA models trained using these correlations can be used as a highly efficient alternative to SVMs, without sacrificing performance. Decorrelated features can also be used for clustering examples, and we have

	ESVM	ESVM	ELDA	Ours-only 1	Ours-only 2	Ours-full
	+Calibr	+Co-occ	+Calibr			
aeroplane	20.4	20.8	18.4	17.4	22.1	23.3
bicycle	40.7	48.0	39.9	35.5	37.4	41.0
bird	9.3	7.7	9.6	9.7	9.8	9.9
boat	10.0	14.3	10.0	10.9	11.1	11.0
bottle	10.3	13.1	11.3	15.4	14.0	17.0
bus	31.0	39.7	39.6	17.2	18.0	37.8
car	40.1	41.1	42.1	40.3	36.8	38.4
cat	9.6	5.2	10.7	10.6	6.5	11.5
chair	10.4	11.6	6.1	10.3	11.2	11.8
cow	14.7	18.6	12.1	14.3	13.5	14.5
diningtable	2.3	11.1	3	4.1	12.1	12.2
dog	9.7	3.1	10.6	1.8	10.5	10.2
horse	38.4	44.7	38.1	39.7	43.1	44.8
motorbike	32.0	39.4	30.7	26.0	25.8	27.9
person	19.2	16.9	18.2	23.1	21.3	22.4
pottedplant	9.6	11.2	1.4	4.9	5.1	3.1
sheep	16.7	22.6	12.2	14.1	13.8	16.3
sofa	11.0	17.0	11.1	8.7	12.2	8.9
train	29.1	36.9	27.6	22.1	30.6	30.3
tvmonitor	31.5	30.0	30.2	15.2	12.8	28.8
Mean	19.8	22.6	19.1	17.0	18.3	21.0

Table 2. Our performance on VOC 2007, reported as AP in %. We compare with ESVM+Calibr and ESVM+Co-occ [4]. "ELDA+Calibr" constructs exemplar models using LDA, followed by a simple calibration step [4]. The last three columns show the performance using our clusters instead of individual exemplars. "Ours-only 1" is our performance using only the "sibling" features, while "Ours- only 2" is our performance using only the context features. Clearly both sets of features give us a boost. Our full model performs similarly to [4], but is much faster to train and test.

shown that the combination of these two ideas allows us to build a competitive object detection system that is significantly faster not just at train time but also at run time. Our work can be built upon to produce state-of-the-art object detection systems, mirroring the developments in SVM-based approaches [2, 3]. Our statistical models also suggest that natural image statistics, largely ignored in the field of object detection, are worth (re)visiting. For example, gradient statistics may be better modeled with heavy-tailed distributions instead of our Gaussian models [13]. However, the ideas expressed here are quite general, and as we have shown, can also be applied to tasks other than object detection, such as clustering.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)

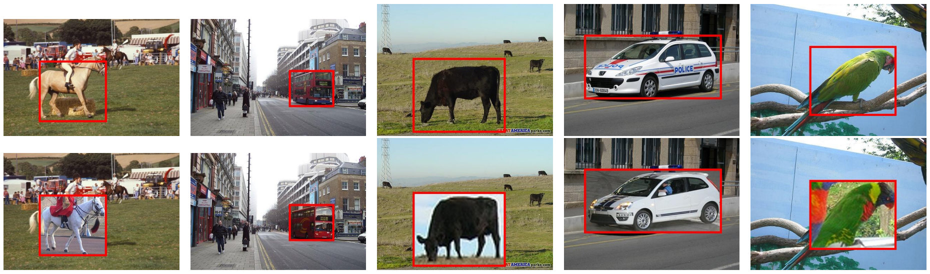


Fig. 7. Detection and appearance transfer. The top row shows detections while in the bottom row the detected objects have been replaced by the most similar exemplars.

2. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI **32** (2010)
3. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV. (2009)
4. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV. (2011)
5. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Human Genetics (1936)
6. Hastie, T., Tibshirani, R., Friedman, J.J.H.: The elements of statistical learning. Springer (2009)
7. Duda, R., Hart, P.: Pattern recognition and scene analysis (1973)
8. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. TPAMI **19** (1997)
9. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of cognitive neuroscience (1991)
10. Murase, H., Nayar, S.: Visual learning and recognition of 3-d objects from appearance. IJCV **14** (1995)
11. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: CVPR. (2004)
12. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: ICCV. (2009)
13. Hyvärinen, A., Hurri, J., Hoyer, P.: Natural Image Statistics: A probabilistic approach to early computational vision. (2009)
14. Rue, H., Held, L.: Gaussian Markov random fields: theory and applications. (2005)
15. Marlin, B., Schmidt, M., Murphy, K.: Group sparse priors for covariance estimation. In: UAI. (2009)
16. Vedaldi, A., Zisserman, A.: Structured output regression for detection with partial truncation. In: NIPS. (2009)
17. Gao, T., Packer, B., Koller, D.: A segmentation-aware object detection model with occlusion handling. In: CVPR. (2011)
18. Dalal, N.: Finding people in Images and Videos. PhD thesis, INRIA (2006)
19. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. (`http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`)
20. Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI **22** (2000)