Feature Sets and Dimensionality Reduction for Visual Object Detection

Sibt Ul Hussain, William Triggs

Feature Sets and Dimensionality Reduction for Visual Object Detection

Sibt ul Hussain
Sibt.ul.Hussain@gmail.com
Bill Triggs
Bill.Triggs@imag.fr

Laboratoire Jean Kuntzmann
BP 53, 38041 Grenoble Cedex 9
France

Abstract

We describe a family of object detectors that provides state-of-the-art error rates on several important datasets including INRIA people and PASCAL VOC'06 and VOC'07. The method builds on a number of recent advances. It uses the Latent SVM learning framework and a rich visual feature set that incorporates Histogram of Oriented Gradient, Local Binary Pattern and Local Ternary Pattern descriptors. Partial Least Squares dimensionality reduction is included to speed the training of the basic classifier with no loss of accuracy, and to allow a two-stage quadratic classifier that further improves the results. We evaluate our methods and compare them to other recent ones on several datasets. Our basic root detectors outperform the single component part-based ones of Felzenszwalb *et. al* on 9 of 10 classes of VOC'06 (12% increase in Mean Average Precision) and 11 of 20 classes of VOC'07 (7% increase in MAP). On the INRIA Person dataset, they increase the Average Precision by 12% relative to Dalal & Triggs.

1 Introduction

Despite the substantial advances made during the past decade, the detection of visual object classes in images remains a challenging problem that receives a great deal of attention in the vision community [5]. Although generative models have considerable potential for deep image understanding, the best current object detectors are trained discriminatively, typically taking the 'sliding window' approach in which a detection window is swept across the image at multiple positions and scales, and a window-level object / non-object classifier is evaluated at each window position. We use this approach and assume that it is familiar [2, 7, 13].

Typical window-based detectors can be divided into two stages. First, image processing is used to extract a set of robust visual descriptors that implicitly contains the information needed to make object / non-object decisions while resisting extraneous effects such as changing object appearance, pose, illumination and background clutter. Secondly, a machine learning based classifier uses the descriptors to make window-level object presence decisions, often followed by postprocessing to merge nearby decisions. The classifiers are typically trained using large sets of labelled training examples. The overall performance depends critically on all three elements: the feature set, the classifier & learning method, and the training set.

Work supported by the Higher Education Commission (HEC) of Pakistan and European research project CLASS.

In this work, we present a family of detectors that give state-of-the-art results for human and object class detection on several important datasets. The detectors combine several recent advances including a rich and complimentary set of visual features [1, 2, 12, 14], efficient dimensionality reduction [11], and latent training with parts [7].

Related Work. The literature on sliding window based human and object detectors is vast. Here we mention only a few relevant methods [2, 6, 7, 9, 11, 13, 14]. Dalal & Triggs [2] developed Histogram of Oriented Gradient (HOG) features for human and object detection. Felzenszwalb *et al*. [6, 7] built a sophisticated multi-component part-based object detection framework over Latent SVM and HOG. Vedaldi *et al*. [13] used multiple kernel learning to combine six feature sets (including bag-of-words, dense visual words, self-similarity descriptors and edge based descriptors) in a spatial pyramid framework for object detection. Schwartz *et al*. [11] combined HOG with color histograms and texture co-occurrence features in a QDA based human detector. Wang *et al*. [14] used HOG and Local Binary Patterns (LBP) with partial occlusion handling for human detection.

Contributions. Our work builds on the insights and methods of several of the above approaches. It makes two main contributions: *(i)* we show that an extended feature set incorporating Histograms of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Local Ternary Patterns (LTP) gives state of the art performance on several important datasets; *(ii)* we show that Partial Least Squares dimensionality reduction can be used to further enhance this in several ways including faster training, improved linear detectors and efficient nonlinear ones. We use the Latent SVM learning framework [7]. For simplicity we focus mainly on single component linear SVM based root detectors, although some results are also presented for multiple component and part based detectors.

2 Feature Sets

Object detectors are critically dependent on the visual features that they use, which must capture the information needed to identify objects of the class despite highly variable object appearance, pose, lighting, clutter, background texture, *etc*. Advances in feature sets have been a constant source of progress over the past decade. Our detector owes much of its accuracy to the use of a combination of three recent high-dimensional feature sets: Histograms of Oriented Gradients (HOG); Local Binary Patterns (LBP); and Local Ternary Patterns (LTP). Each of these is strong in its own right, and together they turn out to complement one another, leading to exceptionally good performance.

Histograms of Oriented Gradients (HOG). HOG [2] is one of the most successful recent feature sets for sliding window based visual recognition. Inspired by the features used by SIFT interest points, it is computed by spatially pooling oriented image gradient strengths into a grid of overlapping cells, with careful attention to normalization. It captures mainly coarse shape (object contour) information, with very strong resistance to illumination variations and some robustness to small spatial variations.

Our implementation of HOG is similar to [2]. We use a simple grid of 8×8 pixel cells with 9 bins of unsigned gradient orientation over color images. Each cell is grouped into four 2×2 cell blocks for SIFT-style normalization, giving 36 feature dimensions per cell. Reducing to 9-D per cell (normalization by the average of the 4 neighbouring blocks) increases the miss rate on the INRIA test set at 10^{-5} False Positive Per Window (FPPW) from 20% to 23%, while reducing to 13-D as in [7] by summing over the 4 normalization and the 9 orientation channels increases it to 22%. Although these performance losses are small given the

reduction in feature vector size, we achieve a similar effect with our PLS projection scheme without suppressing any information.

Local Binary Patterns (LBP). LBP features capture microscopic local image texture. They have proven very successful for texture classification and face recognition [1, 10, 12], and they have recently been used for human detection [14]. LBP descriptors are built by converting each pixel to an 8 bit binary code by thresholding the 8 surrounding pixels at the value of the central pixel, then histogramming the resulting codes over a spatial pooling region, typically one cell of a local grid covering the detection window. Common refinements include using a circle of 8 surrounding pixels rather than a square, with pixel values obtained by interpolation, and histogramming *uniform patterns* rather than full 8-bit ones. A pattern is 'uniform' if it contains at most one contiguous group of 1's within the 8 pixel circle. In practice the vast majority of pixels generate uniform patterns. Uniform LBP histograms have 59 bins, one each for the 58 possible uniform patterns and one for all of the nonuniform ones.

Our detector uses uniform LBP codes based on circles of radius 1 pixel, pooled into 8×8 pixel cells, *c.f.* [1, 14]. LBP codes are computed separately on the R, G and B color channels then pooled into the same histogram. For the best performance[1], the resulting histograms are normalized to sum 1 then square-rooted ("L1Sqrt" normalization). Using L2 instead of L1Sqrt normalization increases the miss rate on INRIA from 30% to 38% at 10^{-5} FPPW.

Local Ternary Patterns. LTP is a simple generalization of LBP introduced by [12]. It uses the same sampling structure as LBP but instead of making binary pixel comparisons it makes 3-way ones depending on whether the pixel is above, within a threshold τ of, or below the central pixel. For simplicity the resulting 8-digit ternary codes are split into 'above' and 'below' binary codes, which are separately histogrammed as uniform LBP patterns as in [12]. The introduction of τ breaks the monotonic illumination invariance of the descriptor, but it helps to suppress the noise that dominates LBP responses in near-uniform regions and it provides an additional parameter that can be tuned to extract complementary information. Empirically, a threshold of $\tau = 5$ gray-levels (out of 255) gave the best performance. Thresholds between 3 and 10 give very similar results, while larger ones tend to discard too much texture information and smaller ones give descriptors that are too strongly correlated with LBP for complementary. As Fig. 1 and the experiments below show, the LBP and $\tau = 5$ LTP responses have rather different characters, with LBP capturing mainly dense local texture and LTP putting more emphasis on strong textures and object contours. Although still a local texture descriptor, LTP provides relatively complementary information to LBP.

The datasets tested here have only a limited range of illumination variations and we found that preprocessing the images using the method of [12] did not enhance the performance. More precisely, adding gamma correction reduced the miss rate at 10^{-5} FPPW for LBP alone from 28% to 20%, but it did not improve the performance of combinations including both LBP and LTP. Adding DoG filtering reduced the performance.

Results. Fig. 2 shows results for various combinations of these feature sets, for window-level classifiers on the INRIA test set and for complete root detectors on the VOC'06 Person test set. The window-level classifiers were trained and evaluated using the method of [2], *i.e.* traditional linear SVM trained on cropped positive and negative windows with several iterations of search for hard negatives, followed by a window-level scan over the test set to evaluate a DET curve (lower curves are better). The complete root detectors were trained and

[1] Using squares instead of circles reduces the performance. Using a larger radius improves it slightly on some problems but reduces it on others. Histogramming each color channel separately sometimes increases it but not enough for us to recommend it as a general practice given the increase in feature vector size.

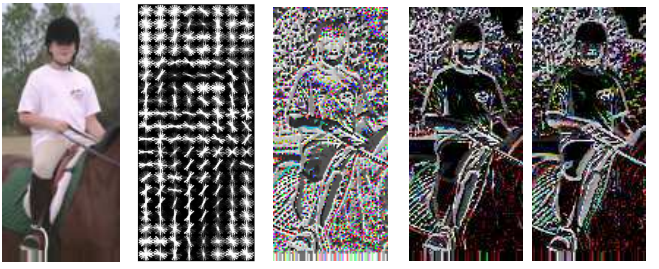

Figure 1: Different feature channels, from left to right: (a) original image; (b) HOG image; (c) color LBP image; (d) & (e) positive & negative color LTP images.

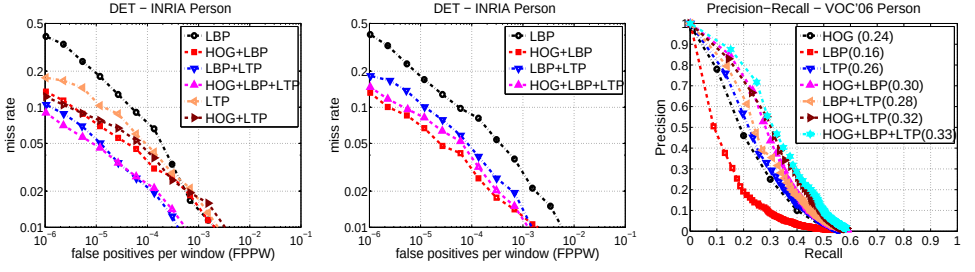

Figure 2: Performance of various combinations of features. DET curves on the INRIA test set for (left) LBP/LTP block size 16×16, (middle) LBP/LTP block size 8×8. (right) Precision-Recall on the VOC'06 Person class for LBP/LTP block size 8×8.

evaluated using the method of [7], *i.e.* with linear Latent SVM learning, scale-space merging of overlapping detections, and evaluation based on Precision-Recall curves and Average Precision / Area Under Curve. Combinations of features give the best results in all cases, with HOG+LBP+LTP being best in all but the middle plot, where it came second. There is no clear winner between HOG+LBP, HOG+LTP and LBP+LTP, but all perform well and LBP and LTP are quite complementary[2]. As far as we know, the results of the window-level classifier on the INRIA dataset are the best reported so far.

Despite fig. 2 (left) and (middle), an LBP/LTP block size of 8×8 gives better performance than 16×16 for the complete root detector. It increases the Average Precision from 29.1% to 32.8% on the VOC'06 Person test set and by 80.1% to 81.6% on the INRIA Person test set. When used in combination with HOG, the LBP/LTP block normalization schemes L1Sqrt and L2 give similar results – respectively 32.8% and 33.5% AP on VOC'06 person and 81.4% and 81.6% AP on INRIA Person – but when LBP is used alone, L1Sqrt normalization is preferred. Provided that *some* normalization method is used, the relative weighting of the HOG, LBP and LTP sections of the feature vector has little impact on the performance of final detector. For HOG+LBP+LTP, replacing color descriptors with grayscale ones increases the miss rate at 10^{-5} FPPW on the INRIA test set from 4.8% to 6.5%.

3 Classifiers

We use sliding window based detectors over high-dimensional feature sets, with SVM classifiers trained using the Latent SVM approach of Felzenszwalb *et al.* [7] and incorporating multiple stages of search for hard negatives as in [2]. Multiple root detectors and parts can be

[2]Here, the LTP threshold was tuned to optimize the combination HOG+LBP+LTP. This is not necessarily the best setting when using LTP alone or with just one other feature set.

incorporated along the lines of [7]. Most of our classifiers are linear, but we also test some quadratic ones based on Partial Least Squares (PLS) dimensionality reduction. Moreover, we show that our classifiers can be sparsified with negligible loss of precision.

Latent SVM. Latent SVM training [7] refines the training examples by running a local search around their labelled image positions and scales, iteratively finding the best match against the current classifier estimate and then retraining the classifier using the resulting matches. This provides cleaner training data and, by mimicking the way in which the classifier will actually be used in practice, it often significantly improves its performance, particularly for classes with highly variable spatial layout. For example on the INRIA dataset, Latent SVM learning increases the average precision from 75% to 79% for HOG features and from 77% to 80% for HOG+LBP+LTP features.

Dimensionality Reduction using Partial Least Squares (PLS). We use very high dimensional feature vectors (20448 dimensions for our full feature set over 48×128 windows). These are bulky to store and slow to process during training, and despite the use of linear SVM classifiers there is always a risk of overfitting. One way to handle this is to introduce some form of dimensionality reduction as a preprocessing stage before learning. Here we use Partial Least Squares (PLS), *c.f.* [11].

PLS is an iterative algorithm for building low-dimensional orthogonal projections that have good predictive power in a given regression task (here, linear classification). It was developed in chemometrics where it is used to produce linear predictors in situations with few observed examples and many highly-correlated variables [15]. Formally, it finds a linear regressor of the form $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$ by a low-dimensional projection $\mathbf{T} = \mathbf{X}\mathbf{W}$ followed by regression $\mathbf{Y} = \mathbf{T}\mathbf{Q} + \mathbf{E}$, so that $\mathbf{B} = \mathbf{W}\mathbf{Q}$. Here, $\mathbf{X}$ and $\mathbf{Y}$ are input and target matrices with each row representing one example, $\mathbf{W}$ is the orthogonal projector, $\mathbf{Q}$ the corresponding regressor, and $\mathbf{E}$ is the residual error matrix. The algorithm scales the variables in $\mathbf{X}$ and $\mathbf{Y}$ to zero mean and unit variance, then iteratively evaluates the best new column that can be added to the current $\mathbf{W}$ and $\mathbf{Q}$ to minimize the prediction (regression) error, and adds this.

In the context of our detector, in each training iteration, after Latent SVM alignment, we use PLS of feature vectors against class labels to project out a discriminative subspace of feature space for SVM training. Both linear and nonlinear classifiers can be trained in the reduced space. PLS is useful even for linear ones. Even including the cost of PLS learning and reduction, it speeds up batch SVM training by a factor of 10-15 or more, *e.g.* reducing training times from 10-15 minutes per stage to 30-90 seconds per stage in the last stages of the full detector. In our experiments it leads to little or no loss of accuracy. Indeed, it often slightly *increases* the accuracy[3]. The resulting (linear) classifier $\mathbf{f}$ can be pulled back through the PLS projection $\mathbf{W}$ to give an equivalent classifier $\mathbf{W}\mathbf{f}$ on the original feature space: this can be run more efficiently than $\mathbf{f}$ because it does not require test examples to be projected through $\mathbf{W}$. By default we use 30-D PLS projection when training linear classifiers, but other values give similar results.

Type of Classifier. Our baseline classifier is (latent) linear SVM trained using SVMLight [8], but we also tested a number of other linear and nonlinear classifiers. We tried several variants of L1 and L2 linear Logistic Regression, but none of them outperformed linear SVM. For example L2 Logistic Regression trained using LibLinear ('L2LR') gave AP on VOC'06 Person of 30%, as compared to 33% for linear SVM. For conventional kernel SVMs, the

[3] SVM detectors are based on limited numbers of noisy support vectors, for example training examples with background clutter. PLS dimensionality reduction implicitly averages over many examples. Presumably, this sometimes projects away part of the noise and hence reduces overfitting.

Method	Linear	FastIKSVM	CoordinateQuad	FullQuad	FullQuadCascade	L2LR
Space	PLS	Full	Full	PLS	PLS	Full
Avg.Prec	33	35	34	**35**	**35**	30
Run Time (sec)	**1.4**	74	2.4	5	1.7	2.7

Table 1: Performance of various nonlinear classifiers on the VOC'06 Person class.

training and testing times are often too high to permit large scale experiments. Instead we tested two forms of embedded quadratic classifier that can be run efficiently.

The first approach is inspired by the fact that Intersection Kernel SVM (IKSVM) can be viewed as a method for learning a nonlinear function $f_i(x_i)$ for each feature dimension i such that $\sum_i f_i(x_i)$ is a good classifier. This can be achieved more directly by expanding each $f_i(x)$ in terms of some convenient set of basis functions $b_{ij}(x)$ such as polynomials, and learning the resulting basis coefficients using a standard linear SVM over the features $\{b_{ij}(x_i)\}$. Our 'CoordinateQuad' method does just this, with features x_i, x_i^2. In practice we find that it gives identical accuracy to IKSVM (and only slightly better accuracy than simple linear SVM) in about 2.4 seconds per image, which is much faster than FastIKSVM [9]. Moreover, the learned x_i^2 coefficients turn out to be very small so there is little evidence that the noisy and highly nonlinear 1D functions learned by IKSVM are actually needed for good classification. Adding higher powers of x_i gives little further improvement. In general, for the feature sets commonly used in visual object recognition, we remain sceptical of the advantages of IKSVM relative to simple feature set expansions of the kind suggested above.

Our second quadratic classifier, 'FullQuad', is a full quadratic SVM in the PLS-projected feature space, *i.e.* it is a linear SVM over a feature set containing the projected features and all bilinear products of them [11]. This is feasible because PLS projections are highly discriminant even with quite small numbers of dimensions (here, 14). The relative weighting of the linear and quadratic components of the feature vector has some effect on performance, but for good performance we found that it sufficed to normalize the range of each to unity.

Table 1 summarizes the performance of various nonlinear classifiers on the VOC'06 person class. The full quadratic classifier in the reduced space has the best overall accuracy but it is rather slow, requiring about 5 seconds per image because each test window must be projected to the reduced feature space. We can reduce this to about 1.7 seconds per image by using a two stage cascade, training a linear SVM on the final stage training set of the quadratic classifier to remove as many negatives as possible while keeping 95% of the positives, and using this as a prefilter before the PLS projection ('FullQuadCascade'). The converse approach – learning a PLS based quadratic classifier as a postfilter on the hard negatives of a linear SVM – also works but it is less accurate.

In general, on these datasets we find that nonlinear methods are slightly more accurate than the linear ones but that the differences are small. In particular, the challenging 'hard negatives' found in the later stages of detector training do not appear to have systematic properties that yield easily to the nonlinearities tested here.

Sparsity. In our experiments, if the learned SVM weight vectors are examined, many of their components are small. The smallest 50% of the components typically carry only about 20% of the total weight. This is to be expected because most visual classes are characterized by relatively sparse cues such as outlines or particular types of textures. To the extent that the small components represent noise inherited from the training set, it can be useful to suppress them. We are currently working on an efficient L1 regularized hinge-loss SVM method for this (existing algorithms that we tested proved too inefficient for problems of this size), but

in the mean time a similar effect can be achieved by taking the trained linear SVM, deleting the features whose SVM weights have small absolute values, and optionally retraining the SVM over these features alone. Such retraining helps. For VOC'06 People, it gives almost identical performance to the original detector (34% AP) with 49% of the features active, and 32% AP with 30% active. Similarly, an L1 regularized L2 SVM gave 31% AP with 14% of the features active and L1 logistic regression gave 29% AP with 24% active. Further work is needed here.

Part Detector. Our part-based detectors are represented and trained in almost the same way as [7], using 2× finer features and locally quadratic penalties on their displacements. We select initial part regions by greedily maximizing the sum of the positive SVM weights of the learned root detector within the region, then initialize the part detectors by densely interpolating the root weights. For the PLS based approaches, we learn separate PLS reductions for each part and for the root at each round of training, appending the (unreduced) deformation costs to these to make the complete reduced feature vector. Unlike [7], we limit the range of possible part displacements by truncating the quadratic displacement penalties. We find that this makes part learning and detection significantly more reliable.

Further Implementation Details. For all of the classes reported here, we force the root detector to be left-right mirror symmetric, and force the parts to come in pairs that are mirror images of one another with mirror image displacement penalties. This allows us to reduce the effective feature dimensions during training by a factor of 2. The transformations are implemented using look-up tables that map HOG components and LBP/LTP codes to their mirror images. We used the densified version of SVMLight [8] to train linear SVM's. We tried several more recent SVM packages including LibLinear, but SVMLight proved to be the most reliable. When training Latent SVM root detectors, we pre-calculate and cache the features of the (typically 10–14) detection windows with more than 70% overlap with the original annotation boxes in position and scale, as this saves a good deal of computation by avoiding the need to recompute the local scale-space pyramids. Moreover, for each positive example in Latent SVM, we select the $k > 1$ best detection windows near the example for use in SVM training, not just the single best window. For $k \leq 3$ this gives the same final performance while allowing the number of training iterations to be reduced by a factor of nearly k. It also compensates to some extent for the imbalanced ratio of hard negatives to positives in the training set. Taking $k \geq 4$ progressively reduces the accuracy.

4 Experiments

We compared our methods with the root and part-based detectors of Felzenszwalb *et al*. [6, 7] and the extended feature set approaches of Schwartz *et al*. and Wang *et al*. [11, 14] on various standard datasets including PASCAL VOC'06 and VOC'07 [5], INRIA People [2] and ETHZ [3]. In this section, unless otherwise noted, the root detectors are trained using linear Latent SVM over PLS reduced features, the HOG and LBP/LTP features are computed over 8×8 pixel cells, L2-hysteresis normalization is used for HOG and L1Sqrt normalization is used for LBP/LTP, and no image preprocessing is done before feature extraction. For the part based approaches we report variants trained using PLS features and unreduced features. We use the standard evaluation protocols, *e.g*. for VOC'06 and VOC'07 we score by Average Precision (AP) on the test set. Fig. 3 summarizes some of the results for three classes from the INRIA and VOC'06 datasets.

VOC'06. We performed our most detailed experiments on VOC'06, using the training and validation sets for training. Table 2 summarizes the results from several of our (rows 1-7) and

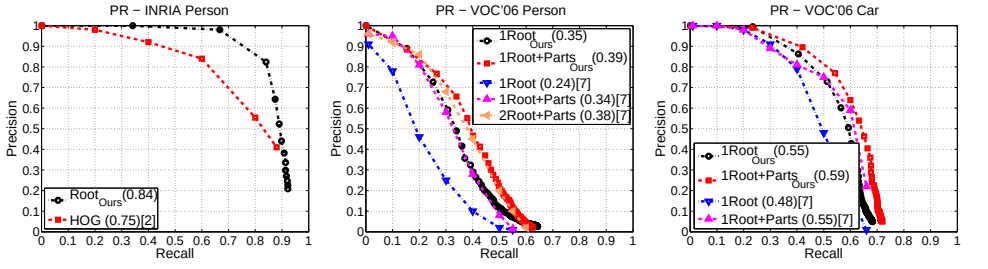

Figure 3: Precision-Recall of several root and part detectors on three datasets: (left) INRIA Person; (middle) VOC'06 Person; (right) VOC'06 Car.

Class	**Mean**	Bike	Bus	Car	Cat	Cow	Dog	Horse	Mbike	Person	Sheep
1 Root	38.3	57.9	44.4	56.0	18.7	37.0	16.0	29.0	51.2	32.8	39.8
1 Root+Parts	42.2	62.1	50.0	58.7	24.6	40.8	17.1	32.5	59.4	38.7	38.2
1 Root+Parts+BB	**43.6**	**63.3**	**50.7**	**63.5**	**25.6**	**42.1**	17.1	32.5	**60.5**	**40.7**	40.3
PLS+Parts	41.9	62.0	50.4	58.2	21.6	38.9	20.7	33.3	56.1	39.2	39.0
PLS+Parts+BB	42.9	62.1	49.7	62.1	23.9	39.6	**20.8**	33.5	56.2	40.4	**40.6**
2 Roots(PLS)	39.1	58.9	48.6	58.0	18.2	37.2	12.5	27.9	57.1	32.3	40.1
2 Roots(Linear)	36.5	57.1	41.3	59.1	12.7	40.6	6.9	25.7	54.1	34.5	33.0
Root+Parts[6]	34.3	59.2	40.7	54.5	8.1	34.6	7.0	28.3	48.5	32.2	30.3
2 Root+Parts[7]	42.1	61.9	49.0	61.5	18.8	40.7	15.1	**39.2**	57.6	36.3	40.4
[7]+BB	42.5	62.0	49.3	**63.5**	19.0	41.7	15.3	38.6	57.9	38.0	40.2

Table 2: Average Precision for some of our detectors and [6, 7] on VOC'06.

Class	**Mean**	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
1 Root	22.8	25.7	39.8	9.2	6.3	23.6	33.2	37.3	12.6	10.6	17.3	25.7	9.6	35.5	35.4	14.2	13.1	16.6	24.6	27.0	38.0
2 Roots	23.0	25.9	**43.6**	1.0	**11.7**	20.4	32.7	43.2	3.9	13.6	18.7	18.3	5.2	43.5	37.7	25.1	10.4	19.7	22.8	24.4	38.7
1 Root+Parts	25.6	27.5	42.4	**10.1**	5.6	27.5	**37.5**	**47.4**	**18.8**	12.1	19.9	23.1	**15.8**	36.4	38.7	14.8	15.7	20.8	25.0	**32.0**	**41.4**
HOGParts	**26.0**	**28.3**	43.3	2.9	4.9	**29.1**	36.2	44.8	15.7	14.3	**20.4**	**28.5**	14.5	**43.6**	**39.4**	**26.7**	**16.1**	**21.2**	**26.4**	23.3	40.1
Root+Parts[6]	21.3	18.0	41.1	9.2	9.8	24.9	34.9	39.6	11.0	**15.5**	16.5	11.0	6.2	30.1	33.7	**26.7**	14.0	14.1	15.6	20.6	33.6

Table 3: Average Precision for some of our detectors and [6] on VOC'07.

Felzenszwalb *et al.*'s (rows 8-10) detectors. Rows 1-3 show that progressively adding parts and bounding box prediction [7] improves the accuracy. Rows 4-5 show that the PLS based parts detectors have performance similar to, but slightly lower than, the corresponding non-PLS ones. Rows 8-10 show that our single-component object detectors give better results than the multiple-component ones of [7], the previous best performers on this dataset. The improvement is largely due to the high precision of our PLS-based root detectors. Indeed, these outperform the part-based methods of [6] on 9 of the 10 classes[4]. On VOC'06, with our single threaded implementation on a 2.6 GHz P4 workstation, our root person detector runs in about 1.4 seconds per image.

VOC'07. Table 3 gives the results of our detectors on various VOC'07 classes ([7, 13] represent the state of the art). Our root detector (row 1) already outperforms the part based detector of [6] (row 5) on 11 of the 20 classes, and adding an additional root (row 2) or parts

[4]Several variants of [6] are available on their website. Here we use their basic method as it is the one whose training algorithm is most similar to ours.

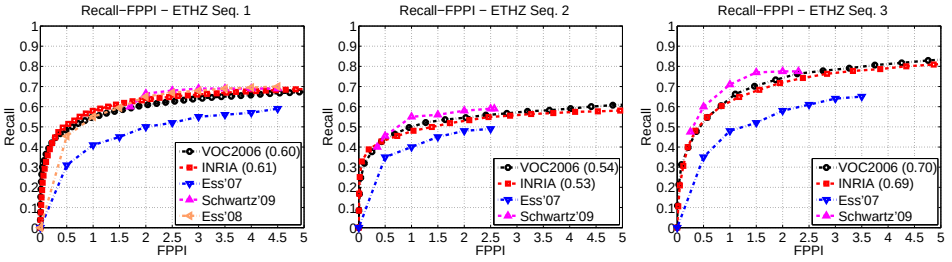

Figure 4: Performance on the ETHZ test set for our linear SVM root detectors trained on the VOC'06 and INRIA sets, versus some competing methods: (left) Sequence 1 (999 images); (middle) Sequence 2 (450 images); (right) Sequence 3 (354 images).

(row 3) further improves the results in most cases. Row 4 shows that solely HOG based parts detectors lead to the best overall performance (slightly better than complete feature sets based part detectors). Nevertheless, the erratic results for some of the classes (*Bird*, *Cat*) suggest that the heuristics used to initialize multiple roots and parts could still be improved.

INRIA Person. As in [2], we evaluate window-level classifiers using Detection Error Tradeoff (DET) curves and complete detectors using Precision-Recall curves and Average Precision (AP) scores. Fig. 2 (left) shows the performance of the window-level root classifier: 4.7% miss rate at 10^{-5} FPPW, compared to 5.8% for [11] and 5.6% for [14]. Fig. 3 (left) shows the corresponding Precision-Recall curves with an AP of 84.1% for our two stage linear-quadratic root classifier and 82% for our linear root one. (The best known AP on this dataset is 86.9% for the part-based approach of [7]). Our root detector runs in about 2.7 seconds/image on INRIA.

ETHZ. For comparison with other methods on this dataset, we plot Recall vs. False Positives Per Image (FPPI). Fig. 4 presents results for our linear root detectors trained on the INRIA and VOC'06 Person sets and tested on three ETHZ sequences. Our methods outperform the complex approach of [3, 4] and, despite being linear and having a 40× lower feature dimension, they have higher recall at low FPPI than the QDA based method of [11] on Sequence 1 and near-identical performance on the other sequences. This allows them to process an image every 3 seconds, whereas [11] takes 120 (or 60 for the 2 stage method).

5 Discussion

We have presented a family of sliding window object detectors that combine a rich visual feature set (Histogram of Oriented Gradient, Local Binary Pattern and Local Ternary Pattern features) with Latent SVM training and Partial Least Squares dimensionality reduction to give state of the art performance on several important datasets including PASCAL VOC'06 and '07, INRIA People and ETHZ. The main findings are as follows. *(i)* HOG, LBP and LTP are strong feature sets in their own right, but they capture somewhat complementary information so combinations of them are even stronger, with HOG+LBP+LTP being strongest of all and much stronger than HOG alone. The complementarity of LBP and (with a suitable threshold) LTP was perhaps not evident a priori. *(ii)* As others have confirmed, the framework of [7] with root+parts detectors and Latent SVM training is very effective. *(iii)* PLS dimensionality reduction is a useful tool that can both speed linear SVM training with little loss (and perhaps even a small gain) in accuracy, and allow efficient nonlinear classifiers. In contrast, we are sceptical of the benefits of IKSVM relative to simple feature embeddings.

(iv) Even using rather naive methods, the feature set can be further sparsified by a factor of 2-3 with little loss of accuracy.

Future work. We are working on improved large-scale L1 regularized SVM algorithms for enforcing sparsity and on including spatio-temporal features into our detectors. The batch-mode algorithms that we use to calculate PLS bases are inconvenient for very large feature sets and on-line algorithms would be desirable.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE T-PAMI*, 28(12), 2006.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005.

[3] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *ICCV*, October 2007.

[4] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*. IEEE Press, June 2008.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge Results. http://www.pascal-network.org/challenges/VOC, 2009.

[6] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, June 2008.

[7] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, 2009.

[8] T. Joachims. Making large-scale svm learning practical. advances in kernel methods - support vector learning. In *B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press*, 1999.

[9] S. Maji, A. C. Berg, and J Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[10] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE T-PAMI*, 24(7):971–987, July 2002.

[11] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009.

[12] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Processing*, 19(6):1635–1650, 2010.

[13] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[14] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[15] S. Wold, M. Sjöström, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001.