A Data Driven Method for Feature Transformation

Mert Dikmen, Derek Hoiem, Thomas S. Huang
University of Illinois at Urbana-Champaign
{mdikmen, dhoiem, thuang-1}@illinois.edu

Abstract

Most image understanding algorithms begin with the extraction of information thought to be relevant to the particular task. This is commonly known as feature extraction and has, up to this date, been a largely manual process, where a reasonable method is chosen through validation on the experimented dataset. In this work we propose a data driven, local histogram based feature extraction method that reduces the manual intervention during the feature computation process and improves on the performance of widely used gradient histogram based features (e.g., HOG). We demonstrate favorable object detection results against HOG on the Inria Pedestrian[7], Pascal 2007[10] data.

1. Introduction

Carefully engineered, gradient-based patch descriptors, such as HOG[7] and SIFT[21], have become a staple of computer vision algorithms. Object detection, image classification, registration, and many other applications benefit from local descriptors that enable robust correspondence. Due to their importance, much research has gone into exploring variations on the feature representations, normalization, and pooling of HOG and SIFT. Most efforts take the simple gradient as the basic building block.

Our paper demonstrates that replacing gradient filters with a set of more general, learned 3x3 filters leads to major improvement in object detection. The filters are constrained to be zero-mean and unit-norm, encoding the intuition that contrast is most informative. The filters are learned by K-medoid clustering with a cosine distance on 3x3 patches that are sampled with a bias that favors high-contrast patches. Local descriptors are created by summing the filter responses within a small cell (e.g., 8×8 square group of pixels) and applying L1-sqrt normalization. Our experiments support the importance of these details, and we validate the general utility of our descriptor on several datasets. On INRIA pedestrians, our descriptors outperform HOG with a 50% reduction in miss rate. By replacing HOG with our descriptor in the latest Felzenszwalb et al. detectors, we improve results in PASCAL VOC 2007 for 15 out of 20 categories.

A key advantage of the proposed descriptor is that it can be directly integrated into existing learning frameworks using similar block descriptors such as [13] or multilevel pyramid–like representations (e.g., [1], [16])

We give a brief background of relevant image representations in Section 2. In Section 3, we describe the filter learning process and explain the relation to HOG and SIFT. In Section 4, we describe implementation details for efficient computation and how to apply our representation to object detection and interest point matching. Our experiments validate the design decisions and demonstrate state-of-the-art performance on several datasets in Section 5. Finally, we discuss directions for further evaluation and development in Section 6.

2. Background

Efficient and robust representations of visual data are of major interest in vision research and generally are one of the most important factors defining the ultimate performance of algorithms. Simple wavelet–like filters, which have shown very good performance in face[25] and pedestrian detection[23], are one of the early examples of descriptors with both efficient computation and high discriminative performance. When the contours of the object can be successfully extracted, Shape Contexts of Belongie et al. [5], which is a histogram of edge points, with log polar bins around the center of the object, have shown good performance in matching and recognition. Ahonen et al.[2] proposed binary representations of intensity changes around pixels as a representation of local texture. The so called local binary patterns are especially useful in applications where the texture is the main source of information.

The most relevant type of representation to this work is the histogramming of gradient orientations. Inspired by Scale Invariant Feature Transform (SIFT [21]), which was originally intended for resolving the problem of keypoint correspondence, many variants have been proposed improving on computational efficiency[15, 4] and invariance properties [24]. Furthermore, densely sampled variants of SIFT with no alignment for orientation have been proven very useful for detection of objects with reasonably rigid part appearances[7, 27]. A successful application of representing local appearances through clustering of filter responses has been studied by Leung and Malik [18] in the context of texture recognition, which further suggests that our approach may yield good results under the detection setting.

Recently, there has been interest in optimizing feature representations through learning. LeCun et al.[17] have pioneered use of convolutional neural networks for bottom up learning of object detectors, where the first learned layer consists of patch level filters. Dictionary based image representations can also be improved through iterative subgradient methods[19] or through sparse coding [26, 22]. However such dictionary optimization methods are difficult to apply in low level representations because of various non–convexities caused by the use of heuristics such as block-normalization of histograms, max-clipping of bins and weighted histogramming. Furthermore learned methods tend to yield object or task specific representations, whereas the representation proposed in this work is general.

3. Overview

We propose a new feature transformation for encoding local blocks of image information for discriminative purposes. Inspired by block-descriptors such as HOG and SIFT, multiple local histograms of appearance information is grouped and normalized together to form a local descriptor. The key difference from aforementioned feature transformations is that the locally pooled appearance information is not an orientation histogram, but rather a less restrictive set of ”snippets” of appearance.

When studying SIFT and HOG closely, a general pattern of feature extraction emerges. The first step usually involves extraction some low level image information through utilization of basic image filters (e.g., directional gradient filters). In the next step, this local information is spatially pooled through a local histogramming process, which groups the information from these local patches according to a pre–defined vocabulary. Finally, one frequently normalizes locally neighboring groups of these histograms. This process is useful for histograms built using the magnitudes of local information patches as weights because the normalization process ameliorates feature magnitude variations due to changes in local contrast.

A feature of the directional gradient filters is that they are 0-mean and have unit norm. Unit norm property ensures that no filter response is unfairly biased in building the histogram since the magnitude of the filter response is used as the histogram voting weight. The 0-mean property has biological justification because it is well known that human visual system is more sensitive to local changes in the contrast than the absolute brightness of the signal[6].

3.1. Patch Dictionary

We propose to replace the gradient orientations with general filters that preserve the unit norm and 0-mean properties but are empirically learned from data and thus can capture the statistical properties of the underlying visual structures better than manually constructed filters.

The first step is to define a vocabulary of image patches through clustering. Staying faithful to gradient based image representations, we define the similarity measure in the space of $d \times d$ image patches to be the dot product of vectorized representations of the image patches:

$$s(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i^T \mathbf{p}_j \qquad (1)$$

, where i and j denote the pixel indexes and $\mathbf{p}_i$, is the vectorized representation of a $d \times d$ patch centered at the pixel p_i.

Dot product as the similarity measure is frequently utilized in the document retrieval research under the term ”cosine similarity” and is tied naturally to the popular variant of the k-means algorithm known as spherical k-means [9], which groups points based on their cosine similarity. This method is different from regular k-means because the underlying probability distribution of the points is no longer assumed to be mixture of k unit variance Gaussians, and furthermore, as the name implies, spherical k-means produces clusters on the unit hypersphere, whereas cluster centers of regular k-means are the means of the clusters.

However spherical k-means can not be applied directly to build representations of local image patches, because unlike word frequency histograms used in document retrieval, the values of convolving the filters with an image does not necessarily have positive or 0 responses, and therefore the cosine similarity is not guaranteed to be positive in all cases. This difference in data domain does not affect the convergence of the spherical k-means clustering algorithm because the objective function is still bounded, but presents the practical question whether an image patch is more similar to a patch with 0 or a small positive correlation value than a patch with a very high negative correlation (i.e., its contrast negative). Another way of thinking about this is whether one prefers the representation to be contrast sensitive or insensitive. In the case of former, cosine similarity shall be used, while for the latter, taking the absolute value of the cosine similarity is appropriate. Using the absolute cosine similarity is equivalent to assuming that dark-to-bright gradient is equivalent to bright-to-dark gradients in the same direction. As shown by Dalal and Triggs [7] for pedestrian detection, the direction of the contrast change is not very relevant for detecting objects of objects with large variation in appearance (e.g., pedestrians appear in arbitrary clothing standing against arbitrary backgrounds). Unless otherwise noted, we will be assuming the use of absolute cosine similarity in the rest of the paper.

In order to cluster image patches using absolute cosine similarity as the similarity measure, we propose a modified spherical k-means algorithm using sample medoids as opposed to sample means. The proposed spherical k-medoids algorithm is summarized in Algorithm 1. The main motivation for using medoids instead of means is due to the lack of definition for a meaningful center of mass of points when using cosine similarity. But an additional advantage is obtained in terms of speed. Since the solution has to be a subset of initial points, the local minima are quite stable. This leads to quick convergence in practice. In our experience, we have found that the algorithm with 1 million initial points quickly converges to a local minimum within 10–20 iterations regardless of k.

Algorithm 1 spherical k-medoids clustering

```
Input: Set of training points with zero mean and unit norm to be clustered:
$\chi = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$
Initialize K cluster centers $M = \{\mathbf{m}_1, \ldots, \mathbf{m}_k\}$ by selecting K exemplars from the set $\chi$ [3].
Initialize clustering fitness: $C = 0$
while $\triangle C > 0$ do
    For each $\mathbf{x}_i$ set $y_i = \texttt{argmax}_k \mid \mathbf{x}_i^T \mathbf{m}_k \mid$
    Update each cluster $k$:
    $\mathbf{m}_k \leftarrow \texttt{argmax}_{\mathbf{x}_i} \sum_j \delta(k - y_j) \mid \mathbf{x}_i^T \mathbf{x}_j \mid$
    $C \leftarrow \sum_i \sum_k \delta(k - y_i) \mid \mathbf{x}_i^T \mathbf{m}_k \mid$
end while
```

Similar to well known k-means algorithm, the first step in the outer loop associates each training sample with the maximally similar cluster center. In the second step, for each cluster, the training sample that is most similar to all other samples assigned to the same cluster is chosen as the new cluster exemplar. Again, similar to k-means, k-medoids is also sensitive to initialization. In order to remedy the effect of bad initialization, we pick the initial k centers using a maximum dissimilarity criterion analogous to the k-means++ method [3].

Sample selection

The training set for dictionary learning is collected from a set of natural, greyscale images, and we used the same dictionary in all of the experiments. We limited the size of the training set of patches to be around one million, which seems to produce clusters with good variety. Appearance prior of image patches in natural images is not uniform. Smooth structures such as uniform patches or ramp discontinuities comprise most types of patches. Unfortunately, from a discriminative standpoint, such patches are rarely interesting. Boundary patches with sharp edge content are more useful. Thus for dictionary learning, we perform a biased sampling of image patches when collecting the training set. The probability of the patch being sampled as an exemplar is proportional to the strength of the pixel intensity contrast within the patch:

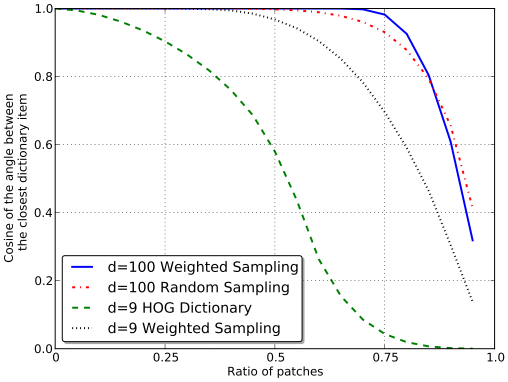


Figure 1. Visualization of patch modeling accuracy of representations. Each curve shows the ratio of randomly sampled patches in the training dataset for which there is at least one dictionary item, whose cosine similarity is at least the value on the vertical axis.

$$P^{(\mathbf{p}_i)}_{\texttt{sampling}} \propto \|\mathbf{p}_i - \mu(\mathbf{p}_i)\mathbf{1}\| \tag{2}$$

, where $\mu(\mathbf{p}_i)$ denotes the average pixel intensity within the patch $\mathbf{p}_i$.

The modeling properties of the learned dictionaries can be readily observed on Fig. 1. With a learned dictionary size of 100 codewords, only less than 10% of randomly sampled patches do not have a cluster center, whose dot product with the patch is larger than 80% of the patch magnitude. Also, the dictionary trained with bias-sampled data is slightly better at modeling than the dictionary trained with uniformly sampled patches. The modest gap of 100 item dictionaries with the HOG dictionary (Section 3.2), is not surprising because as the number of dictionary items approaches infinity, all patches should have an arbitrarily close neighbor in the dictionary. However, even a trained dictionary of 9 items, where the size is equal to the HOG dictionary, is still able to demonstrate significantly better modeling performance.

3.2. Connection to Gradient Orientations

Gradient histogram based feature transformations can be thought of as dictionary based representations of local image patches. To illustrate this, note that image gradient orientations are computed through measuring the responses of two filters: h_x and h_y corresponding to horizontal and vertical gradient filters. Equation 3 shows an example pair of commonly used gradient filters. While a more sophisticated filter pair [14] can potentially yield more accurate gradient information, in practical cases, statistics obtained by ac-

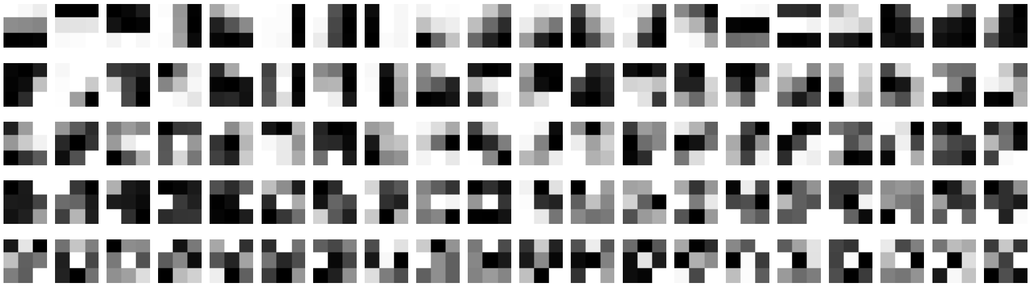


Figure 2. Visualization of a patch dictionary of 3 × 3 patches learned using spherical k-medoids. The dictionary contains 100 elements and the elements are ordered by the frequency they appear in the training set with the most frequent dictionary item being on the top left and the least frequent on the bottom right. The most frequent patches appear to be horizontal and vertical boundary segments followed by diagonal boundaries and line-like shapes. Corner, and point-like segments make up the rest of the words.

cumulating the responses of filters with very small support work better for capturing local appearance information.

$$h_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, h_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3}$$

Let $h_x^{(\mathbf{p}_j)}$ be the response of the filter h_x centered at the pixel p_j. The gradient orientation for the interval $[-90°, 90°]$ at pixel p_j is given by:

$$\tan^{-1}\left(h_y^{(\mathbf{p}_j)}/h_x^{(\mathbf{p}_j)}\right) \tag{4}$$

Quantizing this orientation information is equivalent to finding the maximally responding filter from the following set:

$$h_n = \begin{bmatrix} 0 & -\sin(\theta_n) & 0 \\ -\cos(\theta_n) & 0 & \cos(\theta_n) \\ 0 & \sin(\theta_n) & 0 \end{bmatrix} \tag{5}$$

$$\theta_n = \frac{180° \times n}{N+1}, n \in \{1, \dots, N\} \tag{6}$$

where N is the number of quantization levels.

The histogram descriptor of one cell can be expressed as the following:

$$f_n = \sum_{\mathbf{p}_j \in c_l} g(\mathbf{p}_j, \mathbf{h}_n) \tag{7}$$

$$g(\mathbf{p}_j, \mathbf{h}_n) = \begin{cases} \mathbf{p}_j^T \mathbf{h}_n^2, & \text{if } n = \text{argmax}_m(\mathbf{p}_j^T \mathbf{h}_m^2) \\ 0 & \text{else} \end{cases} \tag{8}$$

Note that this is essentially a histogram based representation, by doing a locally constant estimation of 3 × 3 patch appearances. According to the default feature parameters, the appearance constants in the case of SIFT is 8 and HOG is 9 respectively. This is arguably a small number for piecewise constant modeling of a 9-dimensional signal. Further, since they all have the structure in Equation 5, they are placed on a one dimensional manifold (parameterized by θ) on $\Re^9$.

4. Block Descriptor

The construction of the proposed descriptor is verbatim to the construction of widely used histogram based descriptors. The atomic unit of information for the descriptor is the ”bag of words”-like histogram of dictionary similarities for all pixels in a cell. Each block descriptor is the concatenation of $c \times c$ neighboring but disjoint cell histograms. The location of these blocks can either be defined on a dense rectangular grid for rigid object detection applications or they can be constructed around interest points produced by an affine interest point detector.

To construct the descriptor for an image block (Algorithm 2), the image is first convolved with all elements of the dictionary. Because of the 0-mean property, the elements of the dictionary can be used directly as filters and it is not necessary to subtract the mean from each patch. The image block region contains $c \times c$ cells. For each pixel in these cells, the dictionary element with highest similarity value is found and a weighted vote equal to this similarity value is accumulated on the histogram bin on the corresponding cell. After concatenation of cell histograms in each block, the block descriptors are normalized with respect to an appropriate norm (the selection of this norm is further discussed in Section 5). The cell histograms are computed by pooling the information from $d \times d$ pixels. In the case of SIFT, $c = 4$ and $d = 4$, whereas the original HOG paper sets $c = 2$ and $d = 8$ for optimum performance on INRIA-Pedestrians dataset.

Algorithm 2 Overall construction process of the block descriptor

```
Given $k$ item dictionary $M = \{\mathbf{m}_1, \dots \mathbf{m}_k\}$
Set Cell Size $d = 8$
Set Block Size $c = 2$
Initialize $N = c^2$ cell histograms ($H^{C_n}$) of length $k$
Build the cell histograms:
for all $p_i$ in cell $C_n$ do
  $b = \text{argmax}_k \mid \mathbf{p}_i^T \mathbf{m}_k \mid$
  $H^{C_n}[b] += \mid \mathbf{p}_i^T \mathbf{m}_b \mid$
end for
Perform within block normalization:
$n^{B_l} = \left(\sum_{C_n \in B_l} \sum_b H^{C_n}[b]\right)^{0.5}$
for all $C_n$ in block $B_l$ do
  $H^{C_n} \leftarrow H^{C_n} / n^{B_l}$
end for
```

4.1. Computation and Memory

Dictionary sizes on the order of 100 can produce seemingly high dimensional representations. However the number of non–zero entries in the cell histograms is upper bounded by the number of sampled pixels in the cell. Therefore using a sparse vector representation yields a worst case maximum of $2 \times d^2$ units of memory footprint per cell histogram, independent of the dictionary size.

Computation of the filter responses for each of the dictionary items can be performed in parallel very efficiently using a GPU. Straightforward convolution is the preferred method for convolutions with small convolution kernels up to 7 pixels wide [20]. Our CUDA version of the filterbank code performs 100 convolutions and reductions with 3×3 filters in approximately 2 milliseconds on a GeForce GTX560 for a 640 by 480 image, whereas the CPU version of the corresponding code takes 75 milliseconds on a Quad–Core i5 in multi–threaded code. In our implementation of HOG features, the filterbank size is 9. The GPU algorithm takes approximately 0.5 milliseconds.

5. Evaluation

We test the performance of our descriptor on INRIA-Pedestrian dataset as well as on the 2007 dataset of the Pascal VOC.

INRIA-Pedestrians

INRIA-Pedestrian dataset contains 1208 training images of pedestrians with their reflections along the vertical median axis. 566 images of pedestrians with reflections are provided for testing. The dataset also contains negative training and testing images that do not contain any people.

For training, we extract the block descriptors based on concatenation of 2×2 cell histograms on a dense grid of 8×8 pixels inside a 128×64 window centered around the pedestrian images, which consists our positive training set. This yields 105 block descriptors of $2 \times 2 \times k$ dimensions, where k is the size of the dictionary. Unless otherwise specified we set $k = 100$. Negative features are collected from random 128×64 subimages from the negative training set, which does not contain any images with people in them. We learn the initial dictionaries from a set of one million 3×3 image randomly collected image patches. First a support vector machine classifier is learned with linear kernel. Then this classifier is used to densely scan the negative training set to look for "difficult" samples, which are falsely classified as pedestrians by this first stage classifier. A second stage classifier is then trained using the initial training set with the addition of all difficult samples detected by the first classifier. All of the SVM training is performed through LIBLINEAR [11], which we have slightly modified to increase memory efficiency tailored to take advantage of the sparse nature of our proposed representation.

We verify the pedestrian detector by running it on 566 pedestrians (and their mirrors) on the testing set, as well as all image windows of size 128×64 on the negative testing set at the original scales of the images and downscaled versions by a scaling step of 1.2 until a no object window can fit. The window stride length of 8 in horizontal and vertical directions, which yields 2 million image windows with no pedestrians for testing. The results are reported in the form of detection error tradeoff curves, where the logarithmically scaled x-axis shows the false positives per image window on the negative set versus the y-axis which plots the miss rate ($1-$true positive). First experiment (Figure 3) shows the effectiveness of the proposed descriptor versus the standard HOG, which is implemented verbatim to the description in [7]. When building gradient orientation histograms, soft assignment of weights is known to boost the performance relative to the histograms build using hard assignment. However, soft assignment requires a normalization of assignment weights over each dictionary element per sample point, which becomes costly as the dictionary size increases. Therefore in our implementation, the soft assignment method is not utilized.

The normalization of the concatenated histograms in a block has a profound effect on the performance. While authors in [8] found $L2$ normalization to work the best on the normalization of the gradient orientation histograms, $L1$–$sqrt$ norm works significantly better in our case (Fig 4). This can be explained by the high dimensional structure and sparse nature of our cell histograms. Normalizing with respect to the $L2$ norm reduces the small values in the histogram too aggressively.

For the effects of the dictionary size on the overall detection performance, conventional wisdom from bag-of-words

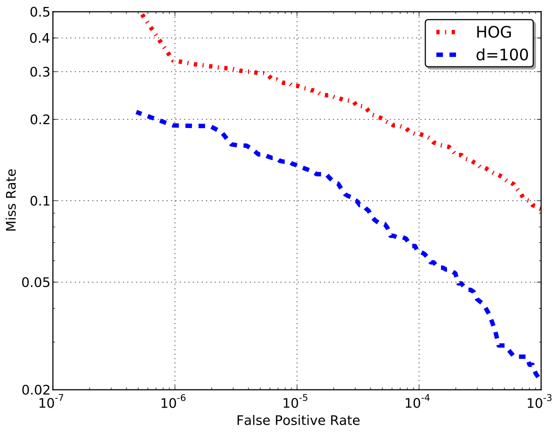


Figure 3. The proposed descriptor yields a false positive rate of 10^{-4} false positives per window at 6.5% miss rate.

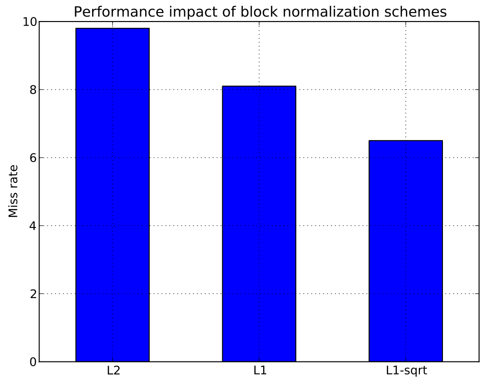


Figure 4. Normalization of cell histograms within each block is essential for optimum performance. Since our histogram is large, a gentle norm like $L1 - sqrt$ works better.

research carries over, and is also reflected in our results, where larger dictionaries outperform the smaller ones for otherwise the same parametrization of the feature. The automatically learned dictionary of about 9 words narrowly beats the HOG baseline, which uses a manually constructed dictionary of 9 items[1], however as the dictionary size increases, the proposed descriptor becomes more discriminative. Figure 5 shows the miss rate at 10^{-4} false positives per window rate for varying dictionary sizes. The returns start diminishing after 200 words and furthermore increased computation becomes another consideration for tradeoff.

The size of the sampled patches also has a noticeable effect on the performance. We tested 3×3, 5×5 and 7×7 patch sizes for training dictionaries. For a constant dictionary size of 100 and all other feature parameters kept constant, in terms of the miss rate at 10^{-4} false positive per window operating point, the 5×5 patch dictionary performed 2.53% worse than the dictionary of 3×3 patches, whereas the 7×7 dictionary was the worst with 7.67% increase in miss rate over the 3×3 dictionary baseline. These findings suggest that modeling more complex patches can become very difficult very quickly as the patch size increases. Furthermore as the base patch size increases, less and less variety can be captures in terms of local appearance and texture properties.

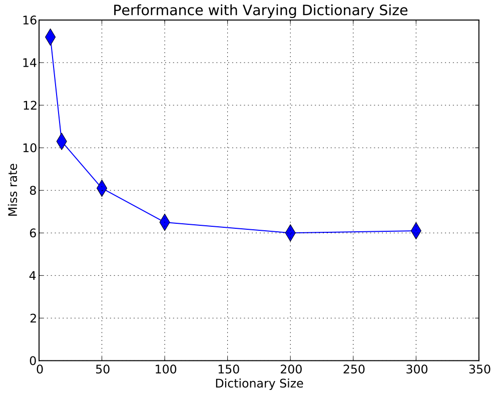


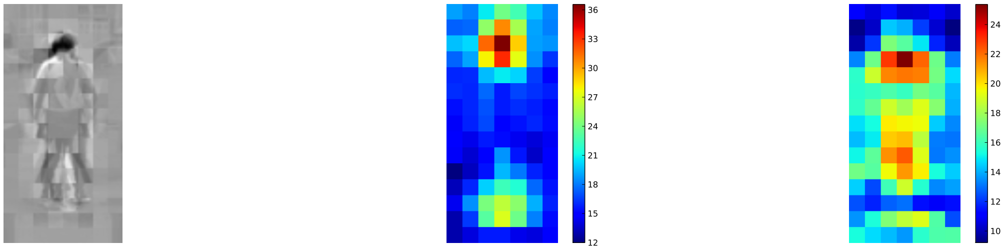
Figure 5. The miss rate drops as larger dictionaries are utilized. However the computational cost also increases linearly for histogram computation as k convolutions are required for each feature transformation.

Finally a visualization of the trained linear SVM classifier can be seen on Figure 6. The most positively performing appearance resembles the averaged human pose of the training images with the positive label. Slight variations in pose such as bent legs are also seem to be successfully captured by the learning stage.

Pascal VOC

Pascal VOC provides a challenging dataset and a good testbed for measuring object detection performance. The authors of [13], which is one of the state of the art object detectors, opened the source-code of their parts based object detector for free use. The original detector uses HOG, sign variant (contrast sensitive) HOG and a texture measure as the base features of their parts detectors. We modified the source code[12] of the parts based detector to operate with the proposed feature transformation instead of the original features. The size of the atomic histograms were set to be 8×8 pixels. The size of the dictionary used was 50, learned from the same training set as previous experiments. For each cell, we produced two histograms: one with cosine similarity as the similarity measure and the other with absolute cosine similarity. These contrast sensitive and insen-

[1] See [8] for the empirical study of dictionary size on the classification performance for HOG descriptor.



(a) (b) (c)

Figure 6. A visualization of the classifier trained on the Inria Pedestrians dataset. a) The collage of image blocks in the training samples with the highest response to the classifier weights for the corresponding block, b) Spatial distribution of the positive classifier weights. c) Spatial distribution of the negative classifier weights. The most distinguishing features of pedestrians are around the head and shoulders area as well as around the feet. The negative weights are distributed more uniformly, with a slight absence of weights around the detection window boundaries, which are expected to be mostly background regions regardless of the class label of the detection window.

sitive representations were finally concatenated to produce the final cell descriptor. All other training parameters were kept fixed. As can be observed in Fig. 7, our descriptor improves or matches the performance of the baseline detector at all but 5 object categories.

6. Conclusions

We have described a robust alternative to gradient orientation based image features. Our new proposed feature transformation adopts many of the carefully engineered properties of previous feature transformations (e.g., blockwise contrast normalization, locally constructed histograms), while improving on the power of the unit histograms in representing the underlying image appearance. The proposed method is directly applicable to all existing methods using the aforementioned descriptors through a simple substitution. As it is experimentally demonstrated, the proposed feature transform offers robust performance on object detection tasks. We would like to further investigate whether similar performance gains can be obtained in other areas such as keypoint correspondence problems and representations of spatio-temporal data.

References

[1] A. Agarwal and B. Triggs. Hyperfeatures–multilevel local coding for visual recognition. *Computer Vision–ECCV 2006*, pages 30–43, 2006. 1

[2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006. 1

[3] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 3

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006. 1

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002. 1

[6] F. Campbell and J. Kulikowski. Orientational selectivity of the human visual system. *The Journal of physiology*, 187(2):437, 1966. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 886, 2005. 1, 2, 5

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:886–893, 2005. 5, 6

[9] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001. 10.1023/A:1007612920971. 2

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 1

[11] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 5

[12] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://www.cs.brown.edu/ pff/latent-release4/. 6, 8

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2009. 1, 6

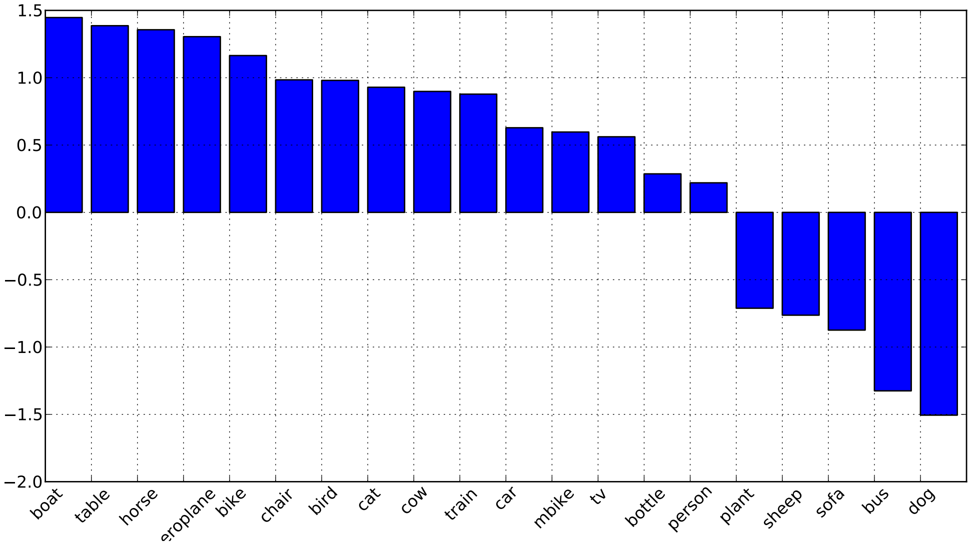


Figure 7. Relative performance difference on the Pascal VOC2007 classes over the baseline detector[12]. Training the parts based detector with the proposed feature set increases the average precision performance over the baseline parts based detector with HOG family of features on most classes in the 2007 Pascal VOC dataset.

[14] W. Freeman, E. Adelson, M. I. of Technology. Media Laboratory. Vision, and M. Group. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991. 3

[15] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. Ieee, 2004. 1

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. Ieee, 2006. 1

[17] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–97. IEEE, 2004. 2

[18] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001. 2

[19] X. Lian, Z. Li, B. Lu, and L. Zhang. Max-margin dictionary learning for multiclass image categorization. *Computer Vision–ECCV 2010*, pages 157–170, 2010. 2

[20] D. Lin, X. Huang, Q. Nguyen, J. Blackburn, C. Rodrigues, T. Huang, M. Do, S. Patel, and W. Hwu. The parallelization of video processing. *Signal Processing Magazine, IEEE*, 26(6):103–112, 2009. 5

[21] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1

[22] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791 –804, april 2012. 2

[23] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997. 1

[24] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):815–830, 2010. 1

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, 2001. 1

[26] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3517–3524. IEEE, 2010. 2

[27] Q. Zhu, M. Yeh, K. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006. 2