PEDESTRIAN AND OBJECT DETECTION USING LEARNED CONVOLUTIONAL FILTERS

Anamaria RĂDOI[1] , Dan Alexandru STOICHESCU[2]

Object detection is still a very active field in Computer Vision. Until now, part based models proved to be one of the most interesting and successful approaches in object and pedestrian detection. The method applies a machine learning approach not to the input images themselves, but to histograms of gradients. However, its performances are still limited when compared to what humans can do. The purpose of the present paper is to show that sparse representations can be successfully used in object detection. The main advantage of using this method is related to the possibility of learning only those filters that are able to express the most frequent patterns that appear in the analyzed images. The experiments are carried out on two widely used datasets, namely VOC2007 and INRIA Person.

Keywords: learned filterbanks, stochastic gradient descent, pedestrian detection, object detection, Histogram of Oriented Gradients.

1. Introduction

Object detection is a major challenge for many areas of research, starting from medicine and going to applications such as street surveillance or video applications. In general, this is a difficult task to accomplish as the variability in terms of objects' position, color, illumination, deformation and viewpoint have a great influence over the rate of object detection.

One of the most successful approaches in object detection is the method presented in [1], that uses grammars of models to build an hierarchic structure in order to obtain a better description of the objects to be detected. To be more precise, the mixture of smaller models is built based on a hierarchic histogram of gradients model for each analyzed images. The filters for the base model and for all the components are trained in order to build the respective model, whilst the responses to these filters are computed at different resolutions in the feature pyramid. However, if the responses are computed for only a fixed number of orientations (i.e., as shown in [1]), we are not able to determine the most representative ones to express the

[1]PhD Student, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: rdi.anamaria@gmail.com

[2]Prof., Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: stoich@elia.pub.ro

main characteristics of the image. For this reason, in this paper, we propose to learn these orientations directly from the images under test via dictionary learning techniques, and, afterwards, we show that building an accurate object detector starting from these representations is possible.

Closely related to the present approach used for orientation learning, [2] and [3] deals with a method for dictionary learning by solving an optimization problem. The results presented in [2] show that sparsity might not be helpful for classification, but it is important when learning the dictionary of filters.

The present paper has two main directions of research. The first direction is related to finding the dictionaries needed for the feature extraction part in the object detection flowchart. In this regard, we show that the learned filters are gradient filters or average filters. The second direction of our study analyses the possibility of replacing the histograms of gradients with histograms of the responses to other directional filters, while maintaining a similar rate of detection as in [1]. We would like also to reduce the dimensionality of the feature arrays so that the performance and the speed of the algorithm is increased.

The flowchart considered for object recognition is summarized below:
(1) feature extraction based on sparse representations;
(2) design of the part-based model;
(3) classification using discriminative learning with latent variables (L-SVM).
Each of these parts are detailed in the next sections. The experiments are carried out on two difficult datasets, namely VOC2007 dataset for object detection ([4]) and INRIA Person dataset for human detection ([5]).

The rest of the paper is organized as follows. Section 2 reviews the part based model presented in [1] for pedestrian detection. Section 3 presents the proposed approach for the feature extraction part, whilst the experimental results are reported in Section 4. Section 5 concludes the paper.

2. Part Based Model for Object Detection

Pedestrian detection, or, more general, object detection, are two fields in computer vision under intensive development nowadays. The approach presented in [1] is built upon a framework based on pictorial structures that represent objects by a collection of oriented (or, better to say, deformed) sub-parts so that the constitutive model gives the best approximate of the objects of study. In order to generalize deformable part models, a hierarchy of sub-parts is formed, meaning that each sub-part can be further expressed by another collection of deformable sub-parts. This is done mainly in order to obtain a better representation of the object, but also to provide an additional flexibility in catching the variation in poses of the object. More precisely, the approach builds a dictionary of sub-parts of objects to be further combined in a grammar-like model that forms the searched object in the analyzed images.

As a particularity, it is worth to mention that simple models outperform complex models, where simple representations will give in most cases better results. One of the reasons can be the fact that complex models can be more difficult to train and usually they make use of much more latent information.

The starting point for [1] is the construction of Histogram of Oriented Gradients (HoG) proposed by Dalal and Triggs in [5]. The local descriptors are taken on a dense grid basis and, for each patch (or, "cell") of the dense grid, the features are extracted using standard orientation (or, edge detection) filters. In order to avoid the variance to illumination, a normalization over neighboring patches is performed, whilst the features are used for training a set of filters at different scales and positions.

One of the most difficult problems dealt in [1] is training a model starting just from a partly labeled training data. To be more precise, one has literally access only to the bounding boxes around the objects of interest together with the corresponding labels, but not to the labels regarding the parts that build a model. The algorithm used for training the model is L-SVM (Latent Support Vector Machine), which makes use of latent variables as side information regarding the objects as, for example, position or scale.

In this section, we briefly review the construction of the algorithm presented in [1], that consists of three steps, namely HoG extraction, part-based models, and L-SVM. We start with the second step as this is the nucleus of the entire construction.

2.1. Part-based Model Design

Designing the models, as presented in [1], is equivalent to finding the coefficients of some linear filters applied to the feature map extracted from each image. A feature map associates to each entry, a d-dimensional feature vector. Thus, the filter is just a rectangular window defined by an array of d-dimensional weight vectors.

In order to define a hierarchy-type model to be applied on a multi-scale feature map a set of filters have to be found. A model with n parts is defined by a $(n+2)$-tuple $(F_0, P_1, P_2, ..., P_n, b)$, where:

- F_0 is the root filter
- $P_1, ..., P_n$ are the part filters whose resolution is double than the resolution of the root filter. Each P_i is given by the 3-tuple (F_i, v_i, d_i), where F_i is the filter for the i^{th} part defined by the anchor position v_i relative to the root position and the quadratic deformation cost, specified by the 4-dimensional bi-variate column vector d_i.
- b is a bias.

The total score resulted from applying the set of filters is given by the score of all filters (root and part). The deformation cost depends on the relative distance between the position of the root filter and the position of each part filter, whilst the

bias term represents just a correction added to the result. So, we can write:

$$score((x_0,y_0,l_0),(x_1,y_1,l_1),...,(x_n,y_n,l_n)) = \sum_{i=0}^{n} \langle F_i, \Phi(x_i,y_i,l_i)\rangle - \sum_{i=1}^{n} d_i^T \begin{pmatrix} \delta_{x_i} \\ \delta_{y_i} \\ \delta_{x_i}^2 \\ \delta_{y_i}^2 \end{pmatrix} + b \quad (1)$$

where (x_i,y_i,l_i) specifies the coordinates of the point (x_i,y_i) at level l_i, $(\delta_{x_i},\delta_{y_i}) = (x_i,y_i) - (2(x_0,y_0)+v_i)$.

2.2. Histogram of Oriented Gradients (HoG)

As mentioned before, the construction presented in [1] starts from computing the gradients on horizontal (Ox) and vertical (Oy) of the image by a simple filtering with $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$. Afterwards, one can extract the orientation $\theta(x,y) \in \{k\pi/9 : k = 0...17\}$ and the magnitude $r(x,y)$ of the intensity gradient for each pixel and build a map with the indexes of the best orientations for each pixel. Two types of orientations can be depicted: contrast-sensitive (18 orientations) and contrast-insensitive (9 orientations, where each orientation is a weighted sum over the two orientations $\{\pm k\pi/9 : k = 0...8\}$)

Thus, the features array is made of $m \times n$ feature vectors corresponding for each block of the image, where m and n are the number of horizontal, respectively vertical, blocks. Each feature vector is a histogram of orientations and, additionally, 4 normalizations over neighboring blocks for each orientation.

Considering a 108-dimensional (27 orientations $\times$ 4 normalizations) histogram is redundant and time consuming. [1] shows that it is possible to transform the product into a sum (27 orientations + 4 normalizations), forming a 31-dimensional feature vector in the following way:

- summing over all normalizations around each block, by making distinction between contrast-sensitive features for $\theta(x,y) \in \{k\pi/9 : k = 0...17\}$ and contrast-insensitive features for $\theta(x,y) \in \{k\pi/9 : k = 0...8\}$;
- summing over all orientations for each type of normalization;
- 0-valued bit to form an 32-bit feature vector.

Fig. 1 explains how the HoG-based features are extracted over an example taken directly from one of the studied datasets.

2.3. L-SVM

Continuing the work in [1], L-SVM (Latent-Support Vector Machine) is used for training and testing the models. As mentioned in the beginning of this chapter, training the models having access to a weakly-labelled data is a difficult task to accomplish.

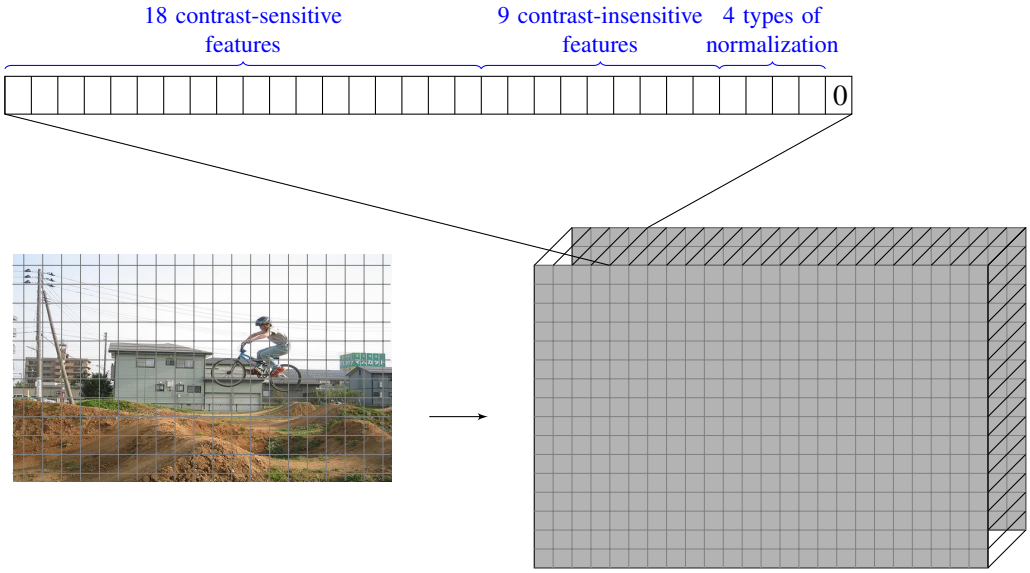


Fig. 1. HoG-based feature extraction for object detection

Similar to SVM, we use N examples $x_1,...,x_N$ along with their correspondent labels $l_1,...,l_N$, where $l_i \in \{-1,+1\}$ represents the absence/presence of the searched object. The task is to minimize the function:

$$L_D(\beta) = 1/2||\beta||^2 + C\sum_{i=1}^{N} \max(0, 1 - l_i F_\beta(x_i)), \tag{2}$$

where

$$F_\beta(x) = \max_{z \in Z(x)} \beta^T \cdot \Phi(x,z), \tag{3}$$

is the score associated to each item x. Here z is the latent (hidden) variable, $Z(x)$ is the valid set of latent values, $\Phi(x,z)$ is the feature vector and β is the filter that contains actually the model parameters.

This function is convex for negative examples ($l_i = -1$) because the maximum over convex functions is also convex. Because of this property, the L-SVM is called to be *semi-convex*. The function becomes convex over all examples (positive and negative) if there exists only one latent value z, in which case the function F_β is a linear function in β.

3. Proposed Approach: Learning Filters for Sparse Representations

As mentioned in the first section, the first step in object recognition is feature extraction. [1] considers HoG features, but the computation of these feature maps

uses only a fixed number of orientations for the gradients, that, in addition, might not be the ones that offer the best representations of the images under test.

Similar to [1], in this section we try to implement a new method of extracting the features. Sparse representations were considered to be desirable for category recognition and for medical image segmentation [6]. A recent method for learning separable filters directly from the dataset of images is presented in [2] and [7]. Moreover, [2] shows that imposing sparsity for classification is compulsory when learning filters, but, by contrary, it is not required for classification.

3.1. Learning convolutional filters for sparse representations

In the following, we present a variant for learning convolutional filters. Let $\{f^j\}_{1\leq j\leq N}$ be N 2-D filters and x_i the set of images from which the filter-bank is learned. Finding a set of separable filters can be regarded as an minimization problem of the following type:

$$\min_{f^j,t_i^j}\sum_i\left(\|x_i-\sum_{j=1}^N f^j * t_i^j\|_2^2+\lambda\sum_{j=1}^N\|t_i^j\|_1\right), \tag{4}$$

where $*$ represents the convolution operator, $\{t_i^j\}_{j=1\ldots N}$ is the set of extracted coefficients during filter learning and λ is a regularization parameter.

This problem is solved via a stochastic gradient descent algorithm, in several steps. With the initialized filters (the first one is set to be uniform and the rest are random), the coefficients t_i^j are computed directly by:

$$t_j^i = f_j * x_i, \quad \forall i, \forall j. \tag{5}$$

Inserting the found coefficients, the minimization is done over the filters using the gradient descent algorithm. New coefficients are computed in the same way as it was mentioned before and the procedure is continued until a convergence condition or the maximum number of steps are met. The algorithm is detailed below.

Require: n patches $\{x_i\}_{i=1,\ldots,n}$
Ensure: Learned convolutional filters $\left[f_j\right]_{j=1,\ldots,P}$
1: Randomly choose a patch x_i, $i=1,\ldots,n$
2: Initialize randomly $\left[f_j\right]_{j=1,\ldots,P}$
3: Maximum number of iterations $Count_{max}$
4: Maximum number of reconstruction steps Rec_{max}
5: **for** $k=1,\ldots,Count_{max}$ **do**
6: Randomly choose a patch x_i, $i=1,\ldots,n$
7: **for all** $j=1,\ldots,P$ **do**
8: $t_j^i \leftarrow f_j * x_i$
9: **end for**
10: **for** $l=1,\ldots,Rec_{max}$ **do**

$Residual_i = x_i - \sum_{j=1}^{P} f_j * t_j^i$
for all $j = 1, \ldots, P$ **do**
Update $t_j^i = t_j^i - \eta_t \left[2(f_j * Residual_i) + \lambda\right]$
end for
end for
for all $j = 1, \ldots, P$ **do**
Update $f_j = f_j - 2\eta_f (Residual_i * t_j^i)$
end for
end for
return $\left[f_j\right]_{j=1,\ldots,P}$

In order to remove the dependencies between the images in the analysed dataset and to speed up the convergence, the images are whitened before. The whitening filters can be easily deduced by finding the eigenvectors and eigenvalues of the covariance matrix, C. The eigenvalue decomposition of C is nothing but $C = E\Lambda E^T$, where E represents the matrix of eigenvectors in the decreasing order of the eigenvalues and Λ is the matrix of eigenvalues in decreasing order. Therefore, the whitening matrix can be written as $W = E\Lambda^{-1/2}E^T$.

3.2. Learning filters for VOC2007 and INRIA Person datasets

We apply the algorithm mentioned above to a set of 150 photos selected randomly from the VOC 2007 database, and INRIA person database, respectively. Examples of learned filters are presented in Fig. 2.

(a) VOC: 27 filters (b) VOC: 11 filters (c) INRIA: 16 filters

Fig. 2. Examples of learned filterbanks from VOC2007 and INRIA person datasets.

For both learning algorithms, the following values for parameters were used: $\lambda = 0.02$ and a gradient step size for the filters of 0.01 so that the minimization of (4) is effective. The size of the 2-D filters is 7×7 pixels for which we obtain the lowest mean error reconstruction.

As it can be easily observed in Fig. 2(a), the degree of redundancy is high for a 27-dimensional filterbank, that is, many filters have the same orientation. Using a smaller dimensionality for the filterbank (as in Fig. 2(b)) is argued by the decay of the eigenvalues of the covariance matrix of the learned filters, represented in Fig. 3.

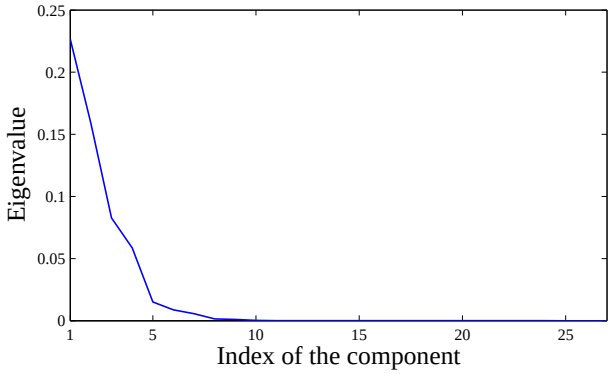


Fig. 3. The decay of the eigenvalues of the covariance matrix of the learned filters.

Analyzing Fig. 2, among the filters that are learned and that form the filterbank, most of them are gradient based filters along different directions. These directions are, in fact, the ones that are the most representative for the analyzed images—the filters are learned directly from the images.

The feature extraction procedure is straightforward and a scheme of the algorithm is shown in Fig. 4.

The main steps of the feature extraction algorithm are detailed below.

(1) *Convolutions with learned filters*

The learned filters are applied on each image and on each channel (red, green, and blue) of the RGB images. The maximum over the channels is kept for each pixel.

(2) *Normalizations*

In order to reduce the influence of the local variations in illumination and foreground-background contrast, we perform a normalization over 4 neighboring blocks, considering that each block is of 8×8 pixels. Denoting by $M(i,j)$ the value of the feature map for block (i,j), the normalizations for the block $M(i,j)$ are the following:

$$N_1(i,j) = (\|M(i-1,j-1)\|^2 + \|M(i,j-1)\|^2 + \|M(i-1,j)\|^2 + \|M(i,j)\|^2)^{1/2}$$
$$N_2(i,j) = (\|M(i,j-1)\|^2 + \|M(i+1,j-1)\|^2 + \|M(i,j)\|^2 + \|M(i+1,j)\|^2)^{1/2}$$
$$N_3(i,j) = (\|M(i-1,j)\|^2 + \|M(i,j)\|^2 + \|M(i,j+1)\|^2 + \|M(i-1,j+1)\|^2)^{1/2}$$
$$N_4(i,j) = (\|M(i,j)\|^2 + \|M(i+1,j)\|^2 + \|M(i,j+1)\|^2 + \|M(i+1,j+1)\|^2)^{1/2}$$

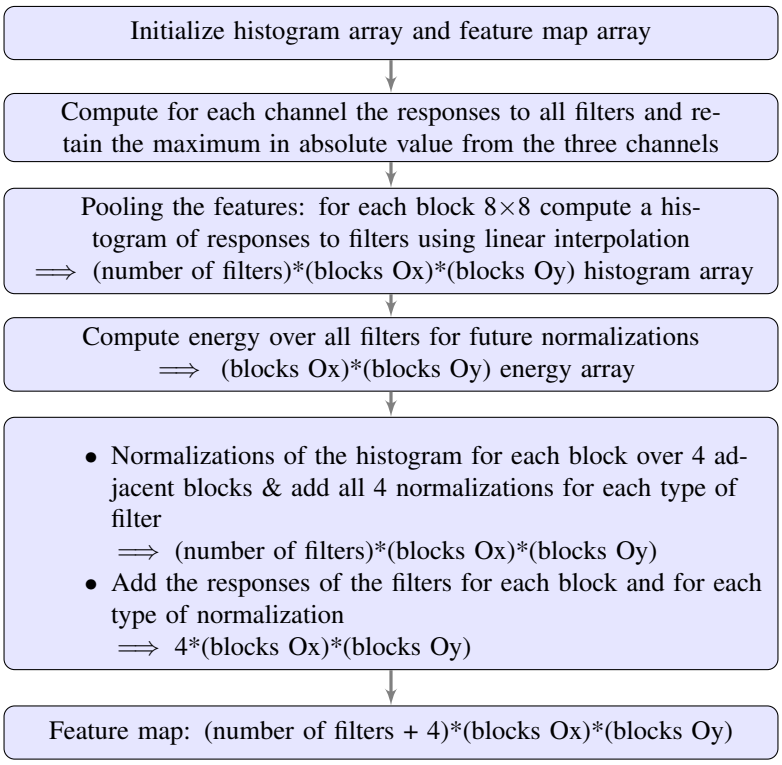
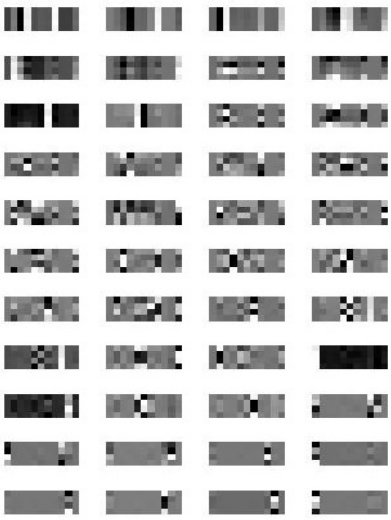


Fig. 4. Scheme of the algorithm

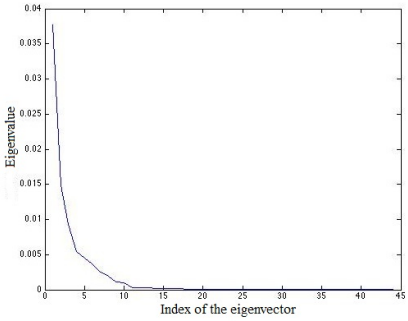
(3) *Linearity of subspaces*

An interesting observation has to be made with respect to the normalizations and filtering. As in [1], we can decompose the subspace on which we make the projections into the subspace spanned by the filters and the subspace of normalizations. This can be done because the eigenvectors with highest eigenvalues have a special form as it can be seen in Fig. 5 for learned filters and for the four normalizations considered. The exemplification is done over the VOC2007 dataset, but the same holds for the INRIA case.

Therefore, in order to further reduce the dimensionality of the feature vectors, we can exploit the linearity characteristic of the subspace. We choose to add all the normalizations for each block for each filter and, for each type of normalization, to add the normalized values over all filter responses. The reduction in dimensionality of the feature vectors corresponding for each 8×8 block is the following: from 44 features/8×8 block to 15 features/8×8 block for the object detection application in the VOC2007 dataset, and from 64 features/8×8 block to 20 features/8×8 block for the person detection application in the INRIA dataset.



(a) Eigenvectors of the covariance matrix for 11 filters × 4 normalizations



(b) The decrease of the eigenvalues of the covariance matrix

Fig. 5. Dimensionality reduction for the learned filterbank from the VOC2007 dataset.

4. Experimental Results in Object/Person Detection

In this section, we compare the results obtained for object/person detection in two scenarios. The first scenario is based on the HoG features, whilst the second one integrates the features extracted using the proposed algorithm in a newly formed part-based model.

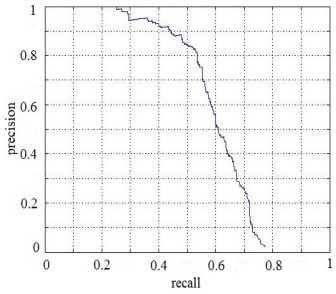
In the first scenario, the object detection method presented in [1] is applied for the two considered databases (VOC2007 and INRIA) are presented in Fig. 6.

In second scenario, the features are extracted with the learned filterbanks and, afterwards, they are inserted in the modified part-based model for object and pedestrian detection. Fig. 7 presents the precision-recall curves along with the average precision accuracy for recognition.

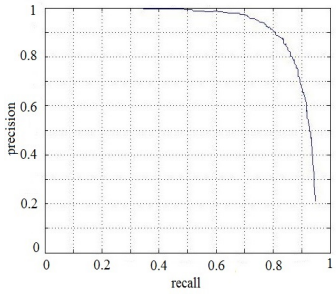
The plots represent the precision-recall curves for the two applications taken into consideration. The precision and recall measures are computed as below:

$$Precision = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{6}$$

$$Recall = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{7}$$

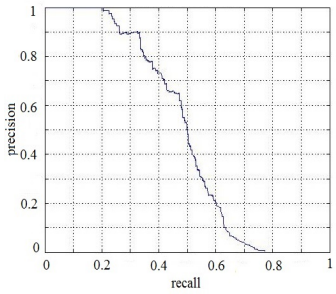


(a) VOC2007 dataset - bicycle class (50.5%)

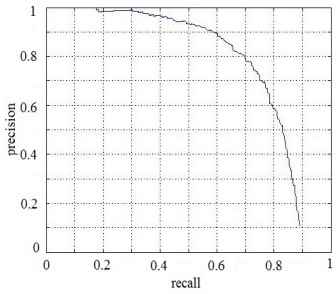


(b) INRIA person dataset (82.4%)

Fig. 6. Precision-recall curves of the part based model & HoG features used for object/person detection. In the parenthesis, the average precision for detection is marked for each class.



(a) VOC2007 - 11 learned filters (49.5%)



(b) INRIA - 16 learned filters (77.2%)

Fig. 7. Precision-recall curves of the part based model & Sparse Representations used for object/person detection. In the parenthesis, the average precision for detection is marked for each class.

5. Conclusions

We evaluated the proposed filtering technique of feature extraction on the VOC2007 and INRIA databases, taking as reference the bicycle class. Compared to Fig. 6, Fig. 7 shows similar results in terms of precision-recall. The meaningful difference is the learning process of the part based models is faster as the vector of features extracted per block is significantly smaller, and, thus, the part-based model is easier to be learned. For example, in the case of 11 filters, the feature vector is of dimension $11+4=15$/block compared to 32 features/block as in [1]. Moreover, the decrease in the average precision is not significant in comparison to the decrease in the dimensions of the feature vectors.

Among the advantages of using the learned orientation filters is that they are learned directly from the analysed images. The most interesting aspect is that the

learned filters are directional filters, which proves the effectiveness of the method also for datasets that have a large degree of variability as the databases considered in this paper. The best results for the filterbank learning part were obtained for randomly selected images, which included objects from many classes, and not for images containing a single type of object. These filters work better for the detection part also because they could make a better distinction between different types of orientations characteristic for each type of object.

Improvements can still be made by using richer methods that would describe better each part of the object that is being modelled, but this would increase the computational burst. Another way of getting better results would be to use a mixture of filters where each subset of the filter is designed to be applicable only at a certain scale.

Acknowledgment

The work has been funded by the Sectoral Operational Program Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398.

REFERENCES

[1] *P. F. Felzenszwalb and R. B. Girshick*, "Object detection with discriminatively trained part-based models," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **32**, no. 9, 2010, 1627–1645.

[2] *R. Rigamonti and M.A. Brown*, "Are sparse representations really relevant for image classification?," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, 1545–1552.

[3] *B. A. Olshausen and D. J. Field*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, Vol. **381**, no. 6583, 1996.

[4] *M. Everingham and L. Van Gool*, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007)*.

[5] *N. Dalal and B. Triggs*, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. **2**, 2005, 886–893.

[6] *B. A. Olshausen and D. J. Field*, "Sparse coding with an overcomplete basis set: a strategy employed by v1," Vision Research, Vol. **37**, 1997, 3311–3325.

[7] *Learning separable filters*, Technical Report EPFL-REPORT-178712, École Polytechnique Fédérale de Lausanne (EPFL), 2012.

[8] *M. Vetterli and J. Kovacevic*, *Fondations of Signal Processing*, Cambridge University Press, 2014.

[9] *B. Ionescu and P. Lambert*, "Classification of animated video genre using color and temporal information," UPB Scientific Bulletin Series C, Vol. **75**, no. 3, 2013, 63–74.

[10] *L. Rotariu*, "On the time-frequency simultaneous alignment of the signals comportament," UPB Scientific Bulletin Series A, Vol. **69**, no. 4, 2007, 23–30.

[11] *A. Fereydooni and A. Safapour*, "Adjoint of pair frames," UPB Scientific Bulletin Series A, Vol. **74**, no. 4, 2012, 131–140.