NEURAL CELLS INSIGHTS ON PEDESTRIAN DETECTION

Douglas Almonfrey*, Raquel Frizera Vassallo†, Evandro Ottoni Teatini Salles†, Mylène C.Q. Farias‡

*IFES, Av. Vitoria, 1729, Vitoria, ES, Brasil

†UFES, Av. Fernando Ferrari, 514, Vitoria, ES, Brasil

‡UnB, Brasília, DF, Brasil

Emails: dalmonfrey@ifes.edu.br, raquel@ele.ufes.br, evandro@ele.ufes.br, mylene@ieee.org

Abstract— Pedestrian detection has become central in recent years mainly due to the emerging of driver assistance systems. One of the main challenges in this area is how to design a good detector, since there is no general theory that guarantees even regular results in detection and most solutions are developed based on empirical experiments by trial and error. Because of this, some works have proposed bio-inspired or data-driven techniques to solve the problem. In this sense, one of the most important part of a detector is the feature extraction. It is from the feature extraction step that is expected the great improvement in detection quality. This work uses theories related to neural cells to support the generation of features to a classifier that is part of a pedestrian detection system. As baseline, the Filtered Channel Features framework is used to analyse, apply and evaluate the method proposed.

Keywords— Pedestrian Detection, Feature Extraction, Neural Cells, Natural Image Statistics, Filtered Channel Features.

Resumo— A detecção de pedestres apresentou um grande progresso nos últimos anos devido a sua importância para sistemas de carros autônomos. Não existe uma solução geral para o problema que balise a obtenção de resultados satisfatórios e grande parte das soluções são obtidas por tentativa e erro. Alguns trabalhos tem proposto soluções bio-inpiradas ou utilizando técnicas dirigidas aos dados. Uma das etapas que está intimamente ligada ao sucesso da solução proposta é a extração de características e é da evolução dessa etapa que se espera um grande incremento na qualidade da detecção nos próximos anos. Este trabalho emprega teorias relacionadas a células neurais para gerar características para um classificador que é parte de um sistema de detecção de pedestres. Como ponto de partida e patamar de comparação são escolhidas técnicas conhecidas como *Filtered Channel Features* (Canais de Características Filtrados).

Palavras-chave— Detecção de Pedestres, Extração de Características, Células Neurais, Canais de Características Filtrados.

1 Introduction

Pedestrian detection is important due to the relevance of the detection, identification and monitoring of humans in a variety of scenes (Nguyen et al., 2016). However, this research area has become central in recent years, mainly due to the emerging of driver assistance systems (Gerónimo et al., 2010).

One of the main problems in pedestrian detection is how to design an accurate detector. There is no general rule and most of the works in the literature were developed based on trial and error. An important step in a pedestrian detection system is the feature extraction. It is from feature extraction that the improvement in the detection quality is expected for the next years (Dollár et al., 2012). The classifier is also an important part of a pedestrian detection system, but there are many different classifiers that have similar detection quality results, as observed in (Benenson et al., 2015), while the principal distinction between them resides in what are the features being used.

This work addresses the feature extraction step, while the classifier employed is the decision forest (DF) trained via AdaBoost, used by lots of successful state of art detectors (Viola et al., 2003; Dollár et al., 2009; Benenson et al., 2013; Zhang et al., 2015). As baseline for the feature extraction step, the Filtered Channel Features framework is used, more specifically the features used in the *LDCF* (Locally Decorrelated Channel Features) detector proposed in (Nam et al., 2014). *LDCF* is a slight modification of the *ACF* (Aggregated Channel Features) detector (Dollar et al., 2014) which is also used in this work as baseline, and whose source code is publicly available[1]. These two detectors are built using 10 channel features and, after channels computation, a filtering procedure is performed before sending a single pixel lookups as features to the classifier. This is the reason *ACF* and *LDCF* are categorized as part of the "Filtered Channel Features Detectors", as first stated by (Zhang et al., 2015). Another reason to choose *LDCF* as baseline is because it is a data-driven technique as well as the method presented in this work.

In this paper, the *LDCF* detector is modified to use other filtering process that is inspired in neural cells and uses concepts of natural image statistics. The objective is to have a more grounded filter choice, differently of the work presented in (Zhang et al., 2015), where some filters were used to generate features without any investigation on the problem data structure, as were

[1] http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html

ISSN 2525-8311

the cases of *Checkerboards* and *RandomFilters*.

Independent Component Analysis (ICA) is employed to remove the local image dependence, inspired by the *LDCF* detector that uses LDA (Linear Discriminant Analysis) to remove correlation locally. The objective of *LDCF* detector is to remove local correlation improving the classification accuracy of decision trees with orthogonal splits. Decision trees with orthogonal splits, using decorrelated data, outperform trees with obliques splits while being more computationally efficient. Oblique trees outperform orthogonal trees only when both use correlated data (Nam et al., 2014). By using ICA, it is expected to have a higher statistical order of decorrelation, and taking advantage of the connection between ICA, sparse coding and simple cells theories, when applied to natural images. This will provide a filtering approach that roughly simulates the way the brain, more specifically the visual cortex, processes information. The way our visual system processes data is a very intuitive inspiration to design data processing techniques, like the case of data-driven feature generation.

This paper is organized as follows: in Section 2, a brief review of the literature is presented, and Section 3 details the baselines used to develop and evaluate the approach presented in this paper. Section 4 discusses the ICA theory and its relation to sparse coding and simple cells for natural images. At this time, the way ICA is employed in this work is clarified. Finally, in Section 5, the experiments and results are presented and analysed and Section 6 presents the conclusions of this work.

2 Related Work

The solutions to the pedestrian detection problem can be, in a simplified manner, categorized as: silhouette matching and appearance-based. In the silhouette matching approach, a region of interest (ROI) is extracted before classification, to avoid an excess of background area (Gerónimo et al., 2010). The classification step can be accomplished using a simple similarity measurement, as correlation. Nonetheless, most successful works use an appearance-based approach. Such methods employ features (descriptors) to represent the differences between image patches and train a classifier, often in a supervised manner. Generally, the classifier is applied to the image in a sliding window approach as is the case of the top ranked detectors. Sliding window techniques tend to be part of detectors that outperform those ones that use segmentation or keypoints (Dollár et al., 2012). The focus of this work will also be appearance-based techniques employing the sliding windows methodology.

In 2003, Viola and Jones (Viola et al., 2003) used their cascade general object detector for the pedestrian detection task. The detector employed features generated using Haar-like templates positioned to model pedestrian body parts. Two years later, Dalal and Triggs (Dalal and Triggs, 2005) proposed the widely used histograms of oriented gradients (HOG), which are nowadays some of the most used descriptors in recognition tasks. In (Felzenszwalb et al., 2008), was proposed the Deformable Part Model (DPM). In this work, the pedestrian was modelled by coarse whole body and high resolution parts templates, where the position of the parts were considered dynamic and latent. As classifier, a latent Support Vector Machine (SVM) that is a version of SVM adapted to handle latent information was used.

In 2009, the performance of a variety of channels was investigated for the pedestrian detection task by (Dollár et al., 2009). As a contribution of that work, 10 channels were selected as appropriate for the task being considered: L, U and V color channels, six histograms of oriented gradients and one normalized gradient magnitude. They were found to give the best detection results when associated with a DF framework trained via AdaBoost at that time and the detector was named *ChnsFtrs*. These channels are the same used in the chosen baselines and are known as HOG+LUV in the literature. The key difference between *ChnsFtrs* and the baselines is that, in the former, after channels computation, the channels are passed by a region pooling through 30000 random rectangular regions of different sizes and locations, using integral images. After that, these 30000 candidate features are sent to AdaBoost for feature selection. This region pooling operation can be faced as a filtering approach, a convolution of the channels with a diverse set of masks. In *ACF*, after channels computation, the channels are aggregated and AdaBoost receives the vectorized channels, single pixel lookups, as features. In the case of *LDCF*, in the place of region pooling, local decorrelation filters are used in the channels before sending the vectorized decorrelated channels as features to AdaBoost.

As time complexity is a question of particular importance in pedestrian detection, mainly because the need of real time applications for car driver assistance systems, different schemes were developed to cope with the speed-up of detection. In (Dollar et al., 2010), the computationally expensive multiscale detection of *ChnsFtrs* was simplified by an approximating exponential scaling law of the features. Rather than computing features from resampled images at every scale, they were computed at each octave[2] and, between each octave, the classifier was rescaled using the exponential scaling law approximation, avoiding im-

[2] An octave is the interval between one scale and another with half or double its value.

age resampling. This procedure resulted in the called Fastest Pedestrian Detection in the West (*FPDW*), a five frame per second (fps) detector, which represents an improvement of five fold in relation to *ChnsFtrs*. In (Dollar et al., 2014), together with a fast feature pyramids computation, *ACF* detector also employs a rejection stage for examples with a score lower than an empirical threshold. This procedure avoids exhaustive inspection in all trees in the DF classifier. This resulted in a 32 fps detector, approximately six fold faster than *FPDW*.

In (Zhang et al., 2015), the Filtered Channel Features framework is defined and analysed using six different filtering approaches. The *InformedFilters* are manually designed and correspond to human body templates, and *Checkerboards* filters are different samples of checkerboards patterns with varying size. *RandomFilters* are filters generated from random patterns of different sizes and, as *Checkerboards*, are not data-driven. *LDCF*, *LDCF8* and *PCAforeground* are data-driven and estimated using the theory of data factorization. *LDCF* uses four filters that are the four eigenvectors associated with the four larger eigenvalues of the data covariance matrix. *LDCF8* is an additional scheme with eight filters and *PCAForeground* estimates four *LDCF* filters for positive and four *LDCF* filters for negative examples, and tries to simulate the effect of Fisher's Discriminant Analysis, considering the background and foreground having different distributions. In (Zhang et al., 2015), the best results were achieved using *Checkerboards*, *RandomFilters* and *InformedFilters*. *LDCF*, *LDCF8* and *PCAforeground* achieved competitive results, but they had the drawback of having a limited number of eight filters.

Although the use of deep architecture for pedestrian detection was questioned in (Benenson et al., 2015), it was proved that this approach can be competitive with or better than DF based detectors in (Hosang et al., 2015). Finally, for additional and more complete information the reader can refer to (Gerónimo et al., 2010; Dollár et al., 2012; Benenson et al., 2015).

3 Baseline

In this work, the *ACF* and *LDCF* are used as baseline. The workflow of the *ACF* detector in the training and detection stages is ilustrated in Figures 1 and 2, respectively. At the training stage, the HOG+LUV channels features are computed after smoothing, and a downsampling is conducted in the channels after another smoothing. Then, single pixel lookups is passed as feature to AdaBoost. This shrinking operation, often by a factor of four, is used to speedup training and detection.

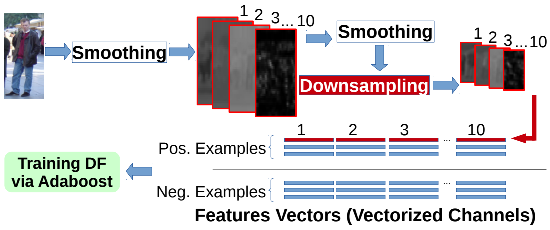


Figure 1: *ACF* training stage.

As the classifier is not trained for multiscale detection, the channels are computed for different image scales at the detection stage, so an image pyramid must be built. This is represented as a red asterisk in Figure 2. After a sliding-window classification, a non-maximum-suppression (NMS) operation is used to remove nearby detections.

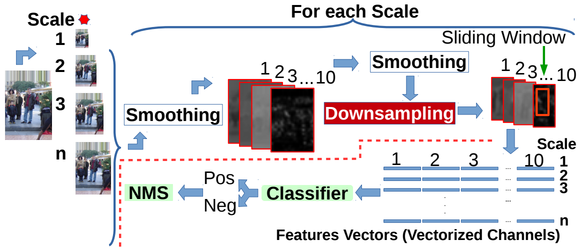


Figure 2: *ACF* detection stage.

The *LDCF* detector is a slight variation of the *ACF* detector, and has four decorrelation filters just after the downsampling operation. After decorrelation, another downsampling by a fixed factor of two is conducted. These decorrelation filters are the eigenvectors of the data covariance matrix associated with the four larger eigenvalues. Uncorrelated data are more convenient for trees with orthogonal splits as mentioned before, and this is the inspiration to use decorrelation filters (Nam et al., 2014). A final number of 40 channels are obtained with the four filters of *LDCF* being applied to the 10 HOG+LUV channels. These channels features are also passed as single pixel lookups to Adaboost.

4 ICA, Sparse Coding and Simple Cells

Natural images (I) can be represented through a basis formed by a set of m features A_i and independent components s_i, as shown in

$$I(x,y) = \sum_{i=1}^{m} A_i(x,y) s_i, \tag{1}$$

for all x and y. This equation can also be represented by an inverse transformation

$$s_i = \sum_{x,y} W_i(x,y) I(x,y). \tag{2}$$

In (Hyvarinen et al., 2009, Chapters 7 and 8) is shown that, for natural images, the latent linear coefficients s_i are as independent and sparse as possible, when the features detectors W_i are obtained through entropy minimization or sparseness maximization of a set of linear components s. Thus, in this case, Equation 2 becomes an ICA and a sparse coding (SC) of the image. When obtained from natural scenes, these SC models have been shown to represent receptive fields (RFs) similar to that of simple cells in macaque primary visual cortex (V1) (Zhu and Rozell, 2013; Hyvarinen et al., 2009, Chapters 3 and 6). This minimum entropy representation of the V1 information is related to the brain minimum energy consumption during V1 activity. Nevertheless, it is important to mention that, as recently shown by (Zylberberg and DeWeese, 2013), these SC models are not the only way to interpret the sensory computation in brain, since, in some experiments, ferret demonstrated dense neural activity (less sparse) in V1, as they matured viewing natural scenes movies.

In this work, the data-driven features detectors W_i will be estimated for each one of the HOG+LUV channels, and will be applied locally in the channels, as filter masks, in a process similar to *LDCF*. However, instead of providing a local decorrelation, it is expected that these filters provide a local independence between the pixels of the channels. Examples of masks generated from each channel can be seen in the Figure 3. The filters are generated using the FastICA algorithm, described in (Hyvarinen et al., 2009, Chapter 18), whose implementation is available online[3]. To resize ICA filter masks to the required size, the nearest neighbour resampling strategy is employed, because it gives more consistent results in relation to interpolation-based resampling. As the filters are sparse, interpolation seems to attenuate filters magnitude information. Figure 4 shows the four L filters of the Figure 3 before the nearest neighbour resampling.

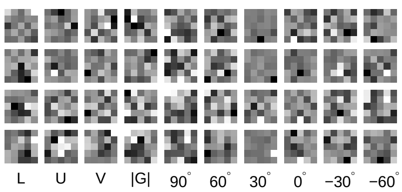


Figure 3: Filters estimated using ICA (INRIA).

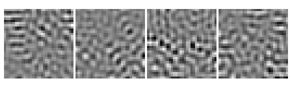


Figure 4: Filters of channel L before resampling.

[3] http://www.naturalimagestatistics.net/

5 Experiments and results

5.1 Methodology of Experiments

To validate the results, the INRIA (Dalal and Triggs, 2005) dataset is used. It has a training set with 1208 positive examples (pedestrians) in 614 positive images, besides 1218 negative images without any pedestrian. The test set has 589 pedestrians in 288 positive images and also 453 negative images. The negative examples are random sampled in the first stage of the training and the additional negative samples are obtained from the detector during the hard negative mining stages.

The metric employed to compare detectors is the *log average miss rate* (LogAvrMR), obtained from the receiver operation characteristic (ROC) curve called *miss rate* (MR) *versus false positive per image* (FPPI) (Figure 7). The LogAvrMR is obtained from the average of nine points between 10^{-2} and 10^{0} FPPI in the ROC curve. A bounding box detection bb_{dt} is considered positive if it matches any valid bounding box ground truth bb_{gt} having a pascal measure higher than a threshold, as expressed below

$$match(bb_{dt}, bb_{gt}) = \frac{area(bb_{dt} \cap bb_{gt})}{area(bb_{dt} \cup bb_{gt})} > 0.5. \qquad (3)$$

It is worth mentioning that the evaluation setup is in par with (Dollár et al., 2009), which provides the reference used in the area.

The model is obtained using as classifier a DF trained via Real AdaBoost with 2048 level-3 binary decision trees, and a pedestrian model size of 128x64 pixels, before downsampling by a factor of four. The model is built via three rounds of hard negative mining (starting from 32 and then 128, 512 and 2048 trees). Each round adds 5000 additional negatives to the training set with a maximum of 10000 additional negatives. The sliding window stride is 4 pixels. During experiments, different values for configurable parameters, as filter sizes and number of filters are considered. Otherwise mentioned, all experiments are executed 10 times for different random seeds and then the average results and standard errors are presented. This is a robust way of evaluating a detector, and in most works presented in the literature is not clear if this procedure is used or only the best result is considered. In some cases, the best result can be much better than the average result, but it should not be considered alone when evaluating a detector.

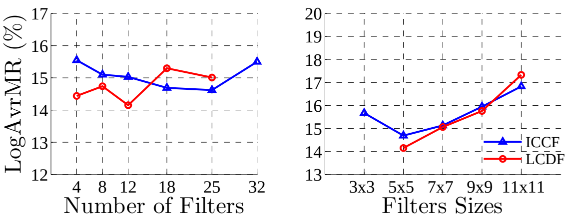
5.2 Experiments

In the next experiments, the analysis of ICA theory in the estimation of filters to be applied to the HOG+LUV channels is done. For convenience, the method developed in this work is referred as Independent Components Channel Filters (*ICCF*)

ISSN 2525-8311

over experiments. In Figure 5, the performance[4] of *ICCF* filters are compared to the *LDCF* one when varying the number and size of filters. As can be seen from Figure 5(a), for 5x5 mask size, the best performance of *ICCF* becames close to the *LDCF* one for 18 and 25 filters. As the number of filters increases, *ICCF* seems to have a better performance, while *LDCF* decreases its performance. As mentioned in (Zhang et al., 2015), for *LDCF*, past eight filters per channel, the eigenvalues decrease to negligible values and the eigenvectors become essentially random filters. Besides that, for a 5x5 mask size, the maximum number of *LDCF* filters that can be estimated is 25[5], and to have more filters the mask size must be increased. As will be shown, this is a problem because the increase of the mask size corresponds to a reduction in the detection quality. So, for *LDCF*, increasing the number of filters to generate more features seems to be a problem, while the same is not true for *ICCF*.

Another observation from Figure 5(b) is that detection quality of both methods decreases as the size of the filter is increased. In fact, this is also an observation of (Nam et al., 2014), (Zhang et al., 2015) and (Yang et al., 2015), and, according to the last one, it seems that, as the mask size increases, the focus on local and important cues as edges and some texture information is lost. The number of filters used for *ICCF* in Figure 5(b) was set to 18 while for *LDCF* it was set to 12. As can be observed from Figure 5, the best configuration obtained for *ICCF* is when using a 5x5 mask size and 25 filters. However, since for 18 filters the result is almost the same, this is preferred and used from now on. The reason is that an increase in the number of filters results in an increase in detection and training times. For 3x3 mask size, the maximum possible number of *LDCF* filters is nine, and, thus, there is no result for this setup.



(a) (b)

Figure 5: Influence of the (a) number of the filters (5x5 mask size) and (b) size in detectors performance (INRIA).

[4]The terms "detection quality" and "performance" are related to the variation of the MR (lower is better), FPPI or LogAvrMR.

[5]Due to the implementation of the data covariance matrix of the *LDCF* detector in (Nam et al., 2014), for a *RxR* mask size, the maximum number of filters that can be estimated is R^2.

For simplicity, Table 1 presents the corresponding LogAvrMR and StdErr of *ICCF* and *LDCF* in relation to Figure 5. For each one of the methods, the lower LogAvrMR is marked in bold in Tables 1(a) and 1(b).

Table 1: *ICCF* and *LDCF* number and size of filters comparison (INRIA).

(a) Influence of the number of filters for 5x5 mask size.

Number of filters	LogAvrMR and StdErr (%)	
–	***ICCF***	***LDCF*s**
4	15.55 ± 0.88	14.44 ± 0.71
8	15.10 ± 0.65	14.74 ± 0.59
12	15.03 ± 0.80	**14.15 ± 0.65**
18	14.69 ± 0.90	15.30 ± 0.82
25	**14.62 ± 1.01**	15.01 ± 0.87
32	15.50 ± 0.95	–

(b) *ICCF* and *LDCF* for 18 and 12 filters, respectively.

Filters Sizes	LogAvrMR and StdErr (%)	
–	***ICCF***	***LDCF***
3x3	15.67 ± 0.59	–
5x5	**14.69 ± 0.90**	**14.15 ± 0.65**
7x7	15.13 ± 0.79	15.06 ± 0.55
9x9	15.95 ± 1.18	15.76 ± 1.11
11x11	16.83 ± 1.02	17.33 ± 0.99

For 5x5 mask size and 18 filters, Figure 6 shows the percentage of features derived from the HOG+LUV channels after filtering, and which are selected by the AdaBoost for the classification task. The channels features are almost equally used, what indicates no redundancy.

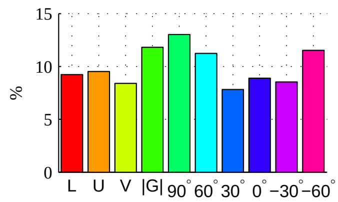


Figure 6: The percentage of features from each channel selected by AdaBoost.

Table 2 shows the effect of *ICCF* when using different channels configurations and employed together with *LDCF*. Note that the result of application of *ICCF* only to the four most used channels of Figure 6 is also presented. It is clear that the diversification of the HOG channels (seven) is more important than the LUV ones (three), but the best result is achieved when all channels are used, showing their complementarity. So, the LogAvrMR is observed to decrease as the number of channels is increased.

The last two lines of Table 2 presents the results of *ICCF* (18 filters) when employed together with *LDCF* (four and 12 filters). We can observe some complementarity between the *ICCF* and *LDCF* with 4 filters (22 filters in total) because the average of LogAvrMR is reduced from 14.44% to 14.13% for *LDCF*. However, the same is not observed for the case of *LDCF* with 12 filters

Table 2: *ICCF* filters employed in different channels configuration (INRIA).

Setup	N° of Channels	LogAvrMR / StdErr (%)
LUV	18·3+7	17.97 ± 0.87
HOG	3+18·7	15.74 ± 1.5
|G| 90°60°-60°	18·4+6	16.92 ± 0.67
HOG+LUV	18·10	14.69 ± 0.90
ICCF+LDCF	18·10+4·10	**14.13 ± 0.76**
ICCF+LDCF	18·10+12·10	14.56 ± 0.76

(30 filters in total) whose average of LogAvrMR is increased from 14.15% to 14.56%.

Finally, for complementarity and to stay in par with the experimental methodology of the literature, Figure 7 presents the best results of our methods when applied to the INRIA dataset. In the legend, in front of each method is indicated the LogAvrMR. *ACF-ours* (random seed 13) and *LDCF-ours* (four *LDCF* filters - random seed 55) were trained and tested in the same machine of *ICCF+LDCF* (18 *ICCF* filters and four *LDCF* filters - random seed 92) and *ICCF* (25 *ICCF* filters - random seed 28). The detection results of the other methods are all online available[6], however, there is no information about what is the random seed used in the experiment. Note that the best result does not mean that the method is better than other "in an average sense" for different random seeds and in different experiments runs.

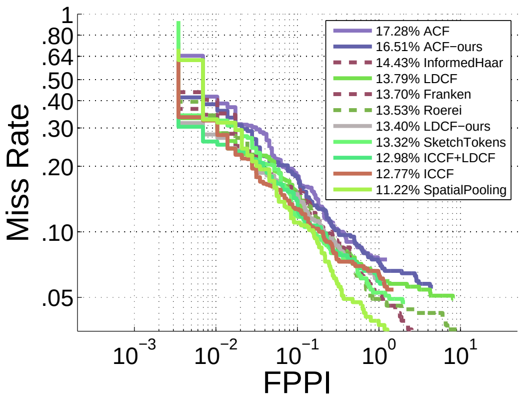


Figure 7: MR vs FPPI roc curve for different methods (INRIA).

6 Conclusions

This work presented a pedestrian detection solution based on the Filtered Channel Features Framework that used ICA to generate features from the HOG+LUV Channels. In addition to this feature extraction procedure, a DF classifier trained via AdaBoost was used. The method was tested in the INRIA dataset and tens of experiments were conducted. The results are in par with the state of art methods and detection quality improvements were obtained in relation to the chosen baselines in some scenarios. As future work the method can be evaluated with different datasets.

[6]`http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`

References

Benenson, R., Mathias, M., Tuytelaars, T. and Van Gool, L. (2013). *Seeking the strongest rigid detector*, CVPR.

Benenson, R., Omran, M., Hosang, J. and Schiele, B. (2015). *Ten Years of Pedestrian Detection, What Have We Learned?*, ECCV.

Dalal, N. and Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection*, CVPR.

Dollar, P., Appel, R., Belongie, S. and Perona, P. (2014). *Fast Feature Pyramids for Object Detection*, TPAMI.

Dollar, P., Belongie, S. and Perona, P. (2010). *The Fastest Pedestrian Detector in the West*, BMVC.

Dollár, P., Tu, Z., Perona, P. and Belongie, S. (2009). *Integral Channel Features*, BMVC.

Dollár, P., Wojek, C., Schiele, B. and Perona, P. (2012). *Pedestrian detection: An evaluation of the state of the art*, TPAMI.

Felzenszwalb, P., McAllester, D. and Ramanan, D. (2008). *A Discriminatively Trained, Multiscaled, Deformable Part Model*, CVPR.

Gerónimo, D., López, A. M., Sappa, A. D. and Graf, T. (2010). *Survey of Pedestrian Detection for Advanced Driver Assistance Systems*, TPAMI.

Hosang, J., Omran, M., Benenson, R. and Schiele, B. (2015). *Taking a deeper look at pedestrians*, CVPR.

Hyvarinen, A., Hurri, J. and Hoyer, P. O. (2009). *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, 1st edn, Springer.

Nam, W., Dollár, P. and Han, J. H. (2014). *Local Decorrelation For Improved Detection*, NIPS.

Nguyen, D. T., Li, W. and Ogunbona, P. O. (2016). *Human Detection from Images and Videos*, Pattern Recognition.

Viola, P., Jones, M. J. and Snow, D. (2003). *Detecting pedestrians using patterns of motion and appearance*, IJCV.

Yang, B., Yan, J., Lei, Z. and Li, S. Z. (2015). *Convolutional Channel Features For Pedestrian, Face and Edge Detection*, ICCV.

Zhang, S., Benenson, R. and Schiele, B. (2015). *Filtered channel features for pedestrian detection*, CVPR.

Zhu, M. and Rozell, C. J. (2013). *Visual Nonclassical Receptive Field Effects Emerge from Sparse Coding in a Dynamical System*, PLoS Computational Biology.

Zylberberg, J. and DeWeese, M. R. (2013). *Sparse Coding Models Can Exhibit Decreasing Sparseness while Learning Sparse Codes for Natural Images*, PLoS Computational Biology.