Improving the performance of pedestrian detectors using convolutional learning ⋆

David Ribeiro [a] Jacinto C. Nascimento [a] Alexandre Bernardino [a]
Gustavo Carneiro [b]

[a] *Instituto de Sistemas e Robótica*
Instituto Superior Técnico 1049-001 Lisboa
PORTUGAL

[b] *Australian Centre of Visual Technologies, The University of Adelaide*
AUSTRALIA

Abstract

We present new achievements on the use of deep convolutional neural networks (CNN) in the problem of pedestrian detection (PD). In this paper, we aim to address the following questions: (i) *Given non-deep state-of-the-art pedestrian detectors (e.g. ACF, LDCF), is it possible to improve their top performances ?*; (ii) *is it possible to apply a pre-trained deep model to these detectors to boost their performances in the PD context ?* In this paper, we address the aforementioned questions by cascading CNN models (pre-trained on Imagenet) with state-of-the-art non-deep pedestrian detectors. Furthermore, we also show that this strategy is extensible to different segmentation maps (e.g. RGB, gradient, LUV) computed from the same pedestrian bounding box (i.e. the proposal). We demonstrate that the proposed approach is able to boost the detection performance of state-of-the-art non-deep pedestrian detectors. We apply the proposed methodology to address the pedestrian detection problem on the publicly available datasets INRIA and Caltech.

1 Introduction

Outstanding progress has been made in pedestrian detection (PD) in the last decade and reaching state-of-the-art results is becoming harder to achieve.

⋆ This work was partially supported by FCT[UID/EEA/50009/2013], and by IST-ID through a research scholarship for the project INCENTIVO/EEI/LA009/2014,
Email addresses: david.ribeiro@ist.utl.pt (David Ribeiro), jan@isr.ist.utl.pt (Jacinto C. Nascimento), alex@isr.ist.utl.pt (Alexandre Bernardino), gcarneiro@gmail.com (Gustavo Carneiro).

There exist many works in the PD field that testify the intensive study on this subject [4–10]. One of the reasons is that pedestrians are among the most important objects to be detected in images, and a large number of applications could benefit from this. For instance, in the fields of mobile robotics to inform Human-Robot Interaction systems, or automotive providing input to Advanced Driver Assistance Systems. The PD task is complex due to several difficulties that arise in this context. For instance, the high variability that characterizes the pedestrians projections on the camera image plane; the appearance of a pedestrian on the image that is influenced by the person's pose; clothing; the atmospheric conditions that contribute to the illumination changes; and the background clutter, all play a role in making PD a difficult problem to be solved. Another difficulty inherent to this problem is occlusion, which has received special attention in the community, giving rise to three different types of metrics [11] that allow to quantify how occluded the pedestrian is: *no occlusion*, *partial occlusion* and *reasonable* (combination of partial occlusion and no occlusion). This contributed to the standardization of the evaluation methodologies and to provide a fair comparison among different detectors. Also, the occlusion motivated the development of two main types of detectors tailored to deal with this specific issue: (*i*) the ones with prior knowledge of the occlusion types [12,13] and (*ii*) the ones that divide the pedestrian into several parts and infer visibility [14,15].

The performance of conventional, handcrafted features has plateaued in recent years, however, new developments in deep compositional architectures have gained significant attention and have greatly advanced the performance of the state-of-the-art concerning image classification, localization and detection. Indeed, deep learning have brought successful results in several domains of application, being an active research topic in computer vision community. One of the advantages provided by deep learning models, and specifically by deep convolutional neural networks (CNN) [16], is the high-level features produced by the top layers of the model that allow to largely improve the classification results, compared to performance produced by hand-crafted features (see [32] for discussion). However, the training process for CNNs requires large amounts of annotated samples to avoid overfitting. This issue has been tackled with transfer learning, which retrains, via fine-tuning, publicly available models (e.g. models that have been trained with large annotated databases [18]) using smaller datasets [19]. This fine-tuning process has been adopted in several works, since it has been shown to improve the generalization of the model (i.e. regularization) compared to a model that is trained with randomly initialized weights using only the small datasets [13,19]. This strategy has been applied with success in other domains of application, such as in medical image analysis [20].

Despite the popularity of pedestrian detection, only few works have applied deep neural networks to this problem [21]. One of the first published works

using convnets for PD is proposed in [7]. A similar research path is followed in this paper. We use pre-trained models in order to determine if the fine-tuning process improves the generalization of the model, compared to a model trained with randomly initialized weights on small datasets. To accomplish this we use the recently proposed VGG model (see [22]) [1].

One of the popular approach for PD is based on a sliding window paradigm. However, such procedure can easily become intractable. With CNNs becoming more of a commodity in the computer vision field, a number of attempts have been published to overcome the above difficulty. For instance, [23], [24] were the best-performing methods at ILSVRC-2013, using a small receptive window size and small stride of the first convolutional layer. We follow a different strategy, pioneered by [25], where selective search identifies promising image regions where to use more expensive features.

In this paper, our main objective is to answer the question: "Can a deep CNN model improve the performance of non-deep PD detectors?". More concisely, in this paper we propose to cascade non-deep state-of-the-art detectors, namely, ACF [10], LDCF [26] and `Spatial pooling +` [27] (including also the classic Viola and Jones (VJ) [3]) with a deep CNN model. We show that such a scheme improves the performance of the base non-deep detectors and generalizes well to different feature maps. Figure 1 illustrates the proposed methodology.

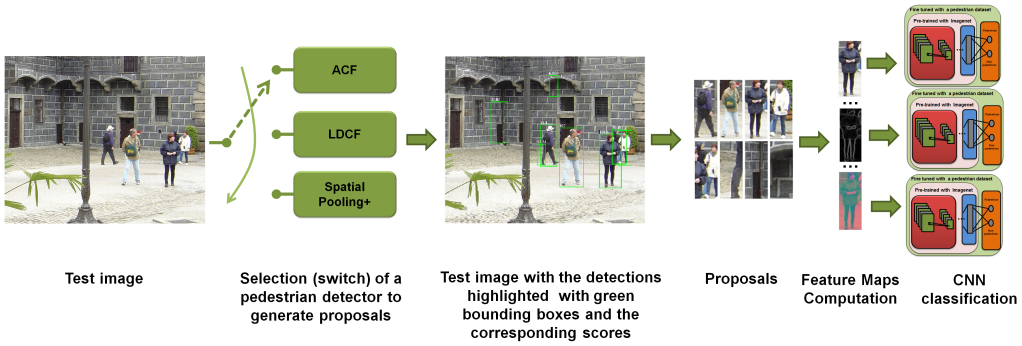

Fig. 1. Illustration of proposed methodology. The switch means that we only select one non-deep pedestrian detector to generate the proposals.

1.1 Related Work

Although pedestrian detection is being addressed in a considerable amount of works, the application of deep neural networks to this problem has only been tackled very recently [2]. Indeed, the success of CNNs is witnessed in several

[1] Details are also available in `http://www.robots.ox.ac.uk/~vgg/research/very_deep/`.

[2] Being the majority of the recent works published in CVPR 2015 and ICCV 2015.

works, but in other context of applications. For instance, *feature matching* is addressed in [28], where it is shown that features learned via convolutional neural networks outperforms SIFT on descriptor matching context. *Stereo matching* is another topic where CNNs show improved performance [29], compared to previous approaches. Another context of application is that of *scene recognition*. Although, large datasets (e.g. ImageNet) are available for object recognition, this is not true in the scene recognition context. One of the first works in this field is proposed in [30], testifying that deep features are efficiently learned in a scene-centric database (called Places) with over seven million labeled pictures of scene. In [31] it is proposed a hybrid architecture to perform *pose estimation*. It is shown that in a monocular image settings, the use of the CNN (jointly used with MRF) is successfully applied to the problem of articulated human pose estimation.

Object detection and classification (using Imagenet) have also received attention in several works. In [18] the large-scale data collection process of ILSVRC is described; datasets such as ILSVRC-2010 and ILSVRC-2012 are used in [17]; SPP-nets [33] and "inception" [34] are proposed in the same context of application. A review is available in [35], where an in depth analysis of ten object proposal methods is available along with four baselines regarding ground truth annotation recall (on Pascal VOC 2007 and ImageNet 2013).

Another related topic of research is *detection proposals for object recognition*. In [25] it is proposed a scheme for generating possible object locations. Object bounding box proposals using edges [36] is shown to be effective in terms of computational efficiency. Another class of relevant works is presented in [37], in which several choices of features (i.e. `HOG,HOF,CSS`) and classifiers (i.e. `linear SVM`) are explored to optimize sliding window based approaches.

Methods based on decision trees (but not using convnets) have been applied with success in PD task. In [38], a collection of the main approaches for PD detection are put together. From this study, decision forest based methods emerge to provide among the best results. By using different combinations of features, we are able to obtain distinct methodologies in this context.

Square channel features are addressed in [8], where it is proposed the `Roerei` detector using `HOG+colour` only, moving away from the classic `HOG+SVM` which has been one of the benchmark methods in the field. Informed Haar-like Features [9] obtain high performance, where the pedestrian shapes are modeled using three rectangles geared towards different body parts. Based on this observation, compact Haar-like features are proposed. However, these (manual) features are specially designed for the pedestrian detection task. In [27] spatial pooling is used, where three visual descriptors are investigated, such as `HOG`, `LPB` and `CSS`. Regionlets [39] is also another method known in this class of approaches, integrating various types of features from competing local regions.

Despite the unquestionable value of the above works, only a few, however, have applied deep learning to the task of pedestrian detection, which is the focus of this paper.

1.2 Contributions

This paper presents the following contributions. First, we aim at using *very deep learning* based approaches to face the problem of PD. Along this goal, we experimentally conduct results with pre-trained vs. randomly initialized models for the same architecture. Second, we propose to alleviate the computational burden that comes from a sliding window exhaustive search, through a selective search approach that significantly improves the computational efficiency. To accomplish this we cascade several state-of-the-art non-deep PD detectors with a deep CNN model. We conduct experiments, to show that such cascade strategy boosts the performance of the non-deep detectors. Finally, we generalize the methodology for several different feature maps. This allows to ascertain the accuracy of an individual feature map, being able to figure out what are the most effective/suitable channel map features for the pedestrian detection task.

2 Methodology

Let us consider a pedestrian dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{m}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ where $\mathbf{x}$ denotes the original RGB CNN input bounding box, i.e. $\mathbf{x} : \Omega \rightarrow \mathbb{R}^p$ [3] and Ω denoting the image lattice; $\mathbf{m}$ is the feature map computed from $\mathbf{x}$. In this paper, we resort to use common features that most state-of-the-art methodologies also use. More specifically we use $\mathbf{m} = \{\mathbf{m}_{\text{RGB}}, \mathbf{m}_{\text{Gm}}, \mathbf{m}_{\text{Gx}}, \mathbf{m}_{\text{YUV}}, \mathbf{m}_{\text{LUV}}\}$. The above notation stands for different feature maps obtained from $\mathbf{x}$ as follows: RGB color space, Gradient magnitude (Gm), Gradient in x (Gx), YUV color space and LUV color space. Finally, $\mathbf{y} \in \mathcal{Y} = \{0, 1\}$ denotes the class (label) indicating if $\mathbf{x}$ contains (or not) the pedestrian; i indexes the ith pedestrian bounding box used for training. Fig. 2 illustrates some of the feature maps used. Also, we have available a dataset for pre-training the CNN, i.e. $\widetilde{\mathcal{D}} = \{(\widetilde{\mathbf{x}})^{(n)}, (\widetilde{\mathbf{y}})^{(n)}\}_n$, with $\widetilde{\mathbf{x}} : \Omega \rightarrow \mathbb{R}^p$ and $\widetilde{\mathbf{y}} \in \mathcal{Y} = \{0, 1\}^C$ (where C is the number of classes in the pre-trained model, in this case 1000).

Deep Convolutional Neural Networks: A CNN model consists of a network containing several processing stages. Each stage comprises two different layers: (i) a convolutional layer with an activation non-linear function, and

[3] p is three for RGB, Gx, YUV and LUV, and is one for Gm.

Fig. 2. Illustration of some of the feature maps used: (a) RGB, (b) Gx, (c) YUV.

(ii) a non-linear subsampling layer. In (i) the convolution filters are applied to the image, whereas in (ii) a reduction in the input image size is performed. These two stages are followed by several fully connected layers and a multinomial logistic regression layer [17]. Fig. 3 illustrates the architecture of our approach. The convolution layers compute the output at location j from the input location i using the filter $\mathbf{W}_k$ and bias b_k at the kth stage using:

$$\mathbf{x}_k(j) = \sigma\Big(\sum_{i \in \Omega(j)} \mathbf{x}_{k-1}(i) * \mathbf{W}_k(i,j) + b_k(j)\Big) \tag{1}$$

where $\sigma(.)$ is a non-linear function, e.g. the logistic or rectification linear unit; the operator $*$ represents the convolution, and $\Omega(j)$ are the input region addresses. The non-linear subsampling layers are defined by $\mathbf{x}_k(j) = \downarrow(\mathbf{x}_{k-1}(j))$, where $\downarrow(.)$ is the pooling function that subsamples the values from the data input region $\Omega(j)$. The fully connected layers consist of the convolution in (1) using a separate filter for each output location, and the whole input from the previous layer. The regression layers compute the probability of the ith class using the features $\mathbf{x}_L$ from the Lth layer with the softmax function $\mathbf{y}(i) = \frac{\exp^{\mathbf{x}_L}}{\sum_j \exp^{\mathbf{x}_L}}$. The inference process is accomplished in a feedforward fashion, and the training stage is carried out with stochastic gradient descent to minimize the cross entropy loss over the training set via back propagation (see [17] for details).

Let us denote the following set of CNN parameters as $\widetilde{\mathbf{\Theta}} = [\widetilde{\theta}_{\text{cnv}}, \widetilde{\theta}_{\text{fc}}, \widetilde{\theta}_{\text{lr}}]$. Each element in $\widetilde{\mathbf{\Theta}}$ is a set of parameters representing: the convolutional and non-linear subsampling layers ($\widetilde{\theta}_{\text{cnv}}$), the fully connected layers ($\widetilde{\theta}_{\text{fc}}$), and multinomial logistic regression layer ($\widetilde{\theta}_{\text{lr}}$). Thus, the process of pre-training a CNN can be formalized as $\widetilde{\mathbf{y}}^* = f(\widetilde{\mathbf{x}}, \widetilde{\mathbf{\Theta}})$. The above pre-training process $\widetilde{\mathbf{y}}^*$, is defined as training M_1 stages using $\widetilde{\theta}_{\text{cnv}}$, followed by M_2 fully connected layers, using $\widetilde{\theta}_{\text{fc}}$, and by one multinomial logistic regression layer that minimizes the cross-entropy loss function $\widetilde{\theta}_{\text{lr}}$, over the data set $\widetilde{\mathcal{D}}$ (see Fig. 3 for an illustration of the proposal). This pre-trained model can be further used by taking the first $M_1 + M_2$ layers to initialize the training of a new model [19], in a process

that is known as *fine tuning*. This fine tuning is crucial to achieve the best classification results in a transfer learning context. Using the above strategy, we take the set of parameters $\{\widetilde{\theta}_{\text{cnv}}, \widetilde{\theta}_{\text{fc}}\}$ and introduce a new multinomial logistic regression layer θ_{lr}, and fine tune the CNN model by minimizing the cross-entropy loss function using the training set $\mathcal{D}$. The process described above will produce the following models for each pedestrian bounding box: $\mathbf{y}^*_{\text{RGB}} = f(\mathbf{x}, \mathbf{\Theta}_{\text{RGB}})$ $\mathbf{y}^*_{\text{Gm}} = f(\mathbf{x}, \mathbf{\Theta}_{\text{Gm}})$, $\mathbf{y}^*_{\text{Gx}} = f(\mathbf{x}, \mathbf{\Theta}_{\text{Gx}})$, $\mathbf{y}^*_{\text{YUV}} = f(\mathbf{x}, \mathbf{\Theta}_{\text{YUV}})$ and $\mathbf{y}^*_{\text{LUV}} = f(\mathbf{x}, \mathbf{\Theta}_{\text{LUV}})$.

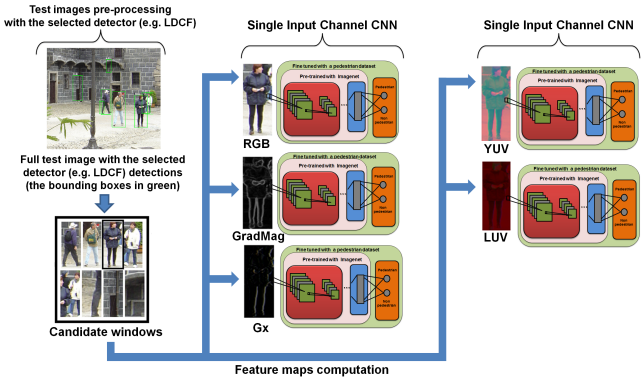

Fig. 3. The proposed methodology uses several CNN models, one for each feature map.

3 Experimental setup

This section discusses results obtained with the proposed methodology in INRIA and Caltech datasets. Since, the detection quality depends on the effect of positive and negative training sets, a detailed description is provided on how these two sets are built: in Sec. 3.1 we describe how the positives and negative training sets are generated in the INRIA dataset, whereas in Sec. 3.2 we describe how this procedure in conducted in the case of Caltech dataset.

3.1 CNN training for INRIA dataset

The experiments are performed on the full INRIA dataset [4], comprising 614 positive and 1218 negative images for training, and 288 images for testing. The methodology used to obtain the training and validation sets is as follows. We use a positive training set, $\mathcal{P}_{\text{os}} = 1237$ samples [4] extracted from the 614 positive images using the original ground truth bounding boxes. Then, we perform data augmentation comprising: (i) an additional set of 1237 by

[4] By samples we mean a bounding box or window of *width* $\times$ *height* pixel size.

performing a horizontal flipping over the set $\mathcal{P}_{\text{os}}$, obtaining a set of $\mathcal{P}_{\text{os}}^{(1)} = 2474$ samples, and (ii) performing random deformations (comprising translations and scaling) in the interval $\mathcal{R} =]0, 5[$ pixels over the set $\mathcal{P}_{\text{os}}^{(1)}$, obtaining a new set of $\mathcal{P}_{\text{os}}^{(2)} = 4948$ samples. The deformations consist in performing cubic interpolation between randomly chosen starting and end values for the width and height of the image (i.e. obtained from a uniform distribution on the interval $\mathcal{R}$).

To obtain the negative sample set $\mathcal{N}_{\text{eg}}$, we run the non-fully-trained LDCF detector. By non-fully-trained, we mean that, the LDCF detector was taken after the first training stage as in [1], being only trained with a segment of the total number of images. This is applied on the 1218 negative full images, obtaining a set of $\mathcal{N}_{\text{eg}} = 12552$ samples. An upper-bound of 18 negative candidate samples per image for negative detection is used.

Such procedure (i.e. using the training proposals of the detector as in [21]), provides a richer set of negatives than the standard method of randomly extracting windows. The latter approach, provides both informative (e.g. resembling pedestrians) and non-informative (e.g. sky or blank patches) negatives. However, the number of non-informative samples is quite large, and from the experiments conducted, they have a negative impact in the performance. Fig. 4, illustrates the negatives selection mechanism: the two leftmost negative samples (in figures (a) and (b)) result from randomly extracting patches from the image and the two rightmost negative samples (in figures (a) and (b)) result from running the non-fully-trained LDCF detector. The above procedure amounts to obtain a total of 17500 samples, from which 15751 are taken for training (90%) and 1749 for validation (10%). Here the proposals have a 100×41 size.

3.2 CNN training for Caltech dataset

This section describes the training data collection process in the Caltech dataset. To obtain the negatives we take every $30th$ frame in the Caltech dataset. As previously, we apply the non-fully-trained LDCF detector over the selected negative images and collect negative samples based on the following rules: (i) a training proposal is considered to be a negative example if it does not exceed an Intersection-over-Union (IoU) of 0.1, that is, if IoU < 0.1; (ii) for the consistency of the experiments we maintain the negative selection mechanism used in the INRIA dataset, i.e., resorting to the non-fully-trained LDCF detector; and (iii) we define an upper bound of five windows per image. Under these conditions, we obtain $\mathcal{N}_{\text{eg}} = 17540$ samples. To obtain the positives, we use a sampling in which we consider every $3rd$ frame in the sequence of the Caltech dataset. We take the initial set of $\mathcal{P}_{\text{os}} = 16376$ samples,

extracted from the images using the original ground truth bounding boxes. From this set, we take 1000 samples forming an auxiliar set $\mathcal{P}_{\text{os}}^{(1)} = 1000$ samples, and perform a horizontal flipping over $\mathcal{P}_{\text{os}}^{(1)}$, obtaining an additional set $\mathcal{P}_{\text{os}}^{(2)} = 1000$ samples. Finally, from $\mathcal{P}_{\text{os}}^{(1)}$ and $\mathcal{P}_{\text{os}}^{(2)}$, we perform a random deformation in the range of $\mathcal{R} =]0, 5[$ pixels (as in the INRIA dataset). This amounts to obtain, $\mathcal{P}_{\text{os}}^{(3)} = 1000$, and $\mathcal{P}_{\text{os}}^{(4)} = 1000$, respectively. To summarize, we have the final set of positives $\mathcal{P}_{\text{os}} = 16376 + 3000$ (i.e., additional samples obtained from $\mathcal{P}_{\text{os}}^{(i)}$, with $i \in \{2, 3, 4\}$). Here the proposals have a 50×20 size.

The training data obtained as detailed in Sec. 3.1, Sec. 3.2 will be used for all the detectors to provide a fair comparison of the results.

3.3 Testing phase

For testing purposes, we follow the evaluation protocol as in [11] that uses, as performance metric, the log average miss rate over nine points in the range of 10^{-2} to 10^{0} False Positives Per Image (FPPI). The evaluation for the Caltech dataset is done according to the *reasonable* setting [11].

To show the benefits of the approach, with respect to the base detectors, we apply the CNN over their test proposals (see Fig. 1). More specifically, we run the following detectors over the INRIA test set: ACF and LDCF detectors [5]. Then, we compute the feature maps (see Section 2) over the ACF and LDCF test proposals (of size 100×41 pixels) and we run the CNN. The number of ground truth bounding box annotations is 589.

The same procedure is also applied to the Caltech dataset, but adding the `Spatial pooling +` detector. Also, the previously used `ACF` detector is replaced by the `ACF` Caltech +, denoted by `ACF+` [26]. The proposals have size of 50×20 pixels. We measure the performance on the *reasonable* setting, i.e., no occlusion or partial occlusion in pedestrians with more than 50 pixels of height. The number of ground truth bounding box annotations is 1014.

Original network model: we use the VGG Very Deep 16 architecture (VGG-VD16) [6], that corresponds to the configuration D in [22]. The architecture consists of a CNN that takes a $224 \times 224 \times 3$ input image and contains: 13 convolutional layers, five non-linear sub-sampling operations (i.e., max-pooling), three fully connected layers and a final multinomial logistic regression layer. More specifically, all max-pooling layers sub-sample a 2×2 input window by a

[5] We do not run the detector `Spatial pooling +` since no optical flow information is available in the INRIA dataset

[6] See additional details in `http://www.robots.ox.ac.uk/~vgg/research/very_deep/`

(a) (b)

Fig. 4. Leftmost position in (a) and (b): Two negative windows randomly extracted from full INRIA images. Rightmost position in (a) and (b): Two negative windows obtained by running the non-fully-trained LDCF detector in full INRIA images.

factor of 2 and the convolutional and fully connected layers have the Rectified Linear Unit (ReLU) as the activation function. Moreover, all convolutional layers from 1 to 13 have a receptive field of size 3×3, but different number of filters (or channels). In fact, layers 1 and 2 have 64 filters (each), followed by max-pooling; layers 3 and 4 contain 128 filters (each), followed by max-pooling; layers 5, 6 and 7 have 256 filters (each), followed by max-pooling; layers from 8 to 13 have 512 filters (each), followed by max-pooling after layers 10 and 13; layers 14 and 15 have 4096 filters (each), and layer 16 has 1000 filters corresponding to the ILSVRC [18] classes, followed by the soft-max. This model is pre-trained with Imagenet [18] (1K visual classes, 1.2M training, 50K validation and 100K test images).

Network changes: In order to reduce the computational demands of the VGG-VD16 model, the original CNN input size of $224 \times 224 \times 3$ was reduced to $64 \times 64 \times 3$. However, this input size does not allow to perform inference after the first fully connected layer, unless the size of the weights is changed. [7] Furthermore, for the PD task, we need to replace the layer 16 and the soft-max, with a new layer and soft-max adapted for only two classes (pedestrian or non-pedestrian). Consequently, the parameters of the three fully connected layers were randomly initialized from a Gaussian distribution with zero mean and variance equal to 0.01, allowing to perform the mentioned changes. Afterwards, we fine-tune the modified network using all feature maps (see Fig. 2) obtained from the pedestrian dataset (as depicted in Fig. 3). The use of the proposed pre-training can be seen as a regularization approach, which can be compared to other forms of regularization, such as data augmentation, obtained by artificially augmenting the training set with random geometric transformations. Finally, the minibatch size is 100, the number of epochs is 10, the learning rate is set to 0.001, and momentum is 0.9. No special attention was made to fine-tune the above hyperparameters.

Section 4 performs a comparison between the baseline performance of these

[7] We conducted additional experiments in which an alternative was to adequately change the stride of some of the previous layers. This however, results in more expensive and time consuming computations.

detectors and with the application of the very deep VGG network.

4 Results

This section shows the results, where we first perform a comparison between the randomly initialised and the pre-trained networks. In this comparison stage, we also study different modes of initializing the CNN model, namely random initialization and *Xavier Improved* initialization [40]. This is done for one feature map (i.e. RGB). Second, we experimentally show that, deep architectures are useful since they improve the performance of all of the tested state-of-the-art pedestrian detectors. Furthermore, this can be generalized for several and quite different feature maps. Finally, we perform a comparison with state-of-the-art applied in the PD task, and show that our technique exhibits very competitive results in the field.

4.1 Initialization of the very Deep CNN

To perform a comparison between random initialization and pre-trained models, we start by performing a random initialization of the deep CNN. More specifically, we adopt the methodology as in [20,32], that is, random weights drawn from Gaussian distributions with fixed standard deviation. However, we experienced that such initialization does not help the deep CNN to converge, drastically hampering its performance. Thus, we confirm what is already reported in [22,40], i.e., such initialization strategy can stall learning due to the instability of gradient in deep nets (in our case 13 conv layers). However, alternatives are available in the literature. Glorot and Bengio [41] proposed the so called *Xavier* initialization in the attempt to overpass the above mentioned difficulty. However, we follow the robust initialization method as proposed in [40], that particulary considers the rectifiers nonlinearities - *Parametric Rectified Linear Unit (PReLU)*. Recall, that the *Xavier* initialization is based on the linearity assumption, which is not valid in ReLU (as in our case) and PReLU. This methodology helps the convergence of very deep models, being possible to train it from scratch. We denote this initialization as *Xavier improved* initialization.

Table 1 shows a comparison of the two modes for the network initialization for the RGB feature map.

Table 1
Log average miss rate % using two modes for the initialization (see text). Results are shown for the detectors, using the RGB feature map, and for both data sequences.

	Random Init. Xavier Improved			VGG Pre-training		
	ACF	LDCF	SP+	ACF	LDCF	SP+
INRIA	21.14%	21.83%	-	15.13%	12.45%	-
Caltech	31.21%	30.15%	23.77%	23.34%	21.66%	16.66%

Table 2
Log average miss rate % for the proposals generated by several detectors on the INRIA and Caltech datasets, and the CNN post-processing, using different feature maps to perform the fine-tuning.

Dataset	**Detector**	**Proposals**	**RGB**	**GM**	**Gx**	**YUV**	**LUV**
INRIA	ACF	16.83	15.13	15.74	14.82	15.94	14.99
	LDCF	13.89	**12.45**	13.33	12.61	12.78	12.66
	SP+	-	-	-	-	-	-
	VJ	72.48	63.47	62.20	64.04	64.17	63.84
Caltech	ACF+	29.54	23.34	27.63	27.24	26.31	25.68
	LDCF	25.19	21.66	24.59	24.67	24.31	23.24
	SP+	21.48	**16.66**	20.43	20.04	19.87	18.71
	VJ	94.73	86.50	89.97	91.73	89.74	88.81

4.2 Improving the pedestrian detectors

We now consider the results with the pre-trained VGG model (VGG-VD16) as detailed in Sec. 4.1, that provided the best results as shown in Table 1. Table 2, summarizes the performance of all the detectors considered in this paper, for several feature maps and for both of the datasets. The column proposals corresponds to the baseline performance of the detectors, the remaining columns correspond to the improvement of having the cascade of the detector with the very deep CNN model. [8]

Figure 5 shows the results for the INRIA dataset, displaying a comparison with state-of-the art pedestrian (non-deep) detectors and our approach that cascades the ACF and LDCF with a very deep CNN. Recall that for this dataset the optical flow is not available, not being possible to use the `Spatial pooling +`. The results of the cascade are shown in black bold. It is shown

[8] The results were obtained with either 256 or 512 as the third dimension in the first fully connected layer.

the best performances for these detectors. A log average miss rate of 14.82% is achieved for the ACF detector using the Gx feature map, and a log average miss rate of 12.45% is achieved for the LDCF detector using the RGB feature map. Notice that the CNN can always improve the performance of any non-deep detector. Recall that the worst detector (Viola and Jones) cannot deteriorate the cascade (i.e. VJ + CNN) performance.

Figure 6 shows the results for the Caltech dataset. Again, it is shown the performances cascading the three detectors with the very deep CNN. The top performance is reached 16.66%, cascading the Spatial pooling + and VGG.

Figure 7 shows a comparison with the state-of-the art methodologies for the Caltech dataset. Here, the results of non-deep detectors are shown, as well as the deep methodologies that were proposed very recently. Our proposal has the third best result.

We have computed the obtained running time figures for both datasets. Table 3, shows the computational effort for both the datasets considering the cascade of the non-deep detectors with the CNN. These training and testing are measured using Matconvnet training [42] on a 2.50 GHz Intel Core i7-4710 HQ with 12 GB, a 64 bit architecture and graphics cards NVIDIA GeForce GTX 850M (main memory) and Intel HD Graphics 4600 (secondary memory).

The training time, refers to the time spent training the CNN model (not the non-deep full detector). The test time concerns the time spent to forward the proposals through the CNN model. Recall that, the test time depends on the number of proposals generated by the non-deep detector. Thus, for the INRIA dataset, the number of proposals for the four detectors are the following: $\text{Prop}_{\texttt{ACF}} = 1835$, $\text{Prop}_{\texttt{LDCF}} = 940$, and $\text{Prop}_{\texttt{VJ}} = 12323$. For the Caltech data set the number of proposals are: $\text{Prop}_{\texttt{ACF}} = 114K$, $\text{Prop}_{\texttt{LDCF}} = 54K$, $\text{Prop}_{\texttt{SP+}} = 228K$, and $\text{Prop}_{\texttt{VJ}} = 190867$. This justifies the larger test time spent for the ACF, VJ and SP+ detectors.

4.3 Analysis of the obtained results

In this section we analyse the obtained results and the improvement introduced by the application of the CNN over the proposals. More specifically, we experimentally show that the CNN is able to:

- Significantly reduce the number of False Positives (FP)
- Slightly reduces the number of True Positives (TP)

The ideal scenario would be to decrease the number of FP, while maintaining the TP.

Table 3
Running time figures for the two initialization modes, obtained for both datasets and corresponding miss rate (MR), using the RGB feature map.

Dataset	**Rand. Init. Xavier Improved**	**Pre-trained VGG-VD16**
INRIA	MR LDCF+VGG proposals = 21.83% MR ACF+VGG proposals = 21.14% train time = 5.1 hrs. LDCF test time = 40.1 sec. ACF test time = 63.9 sec.	MR LDCF+VGG proposals = 12.45% MR ACF+VGG proposals = 15.13% train time = 5.1 hrs. LDCF test time = 33.0 sec. ACF test time = 67.3 sec.
Caltech	MR LDCF+VGG proposals = 30.15% MR ACF+ +VGG proposals = 31.21% MR SP+ +VGG proposals = 23.77% train time = 12.17 hrs. LDCF test time = 30.86 min. ACF+ test time = 1.37 hrs. SP+ test time = 3.11 hrs.	MR LDCF+VGG proposals = 21.66% MR ACF+ +VGG proposals = 23.34% MR SP+ +VGG proposals = 16.66% train time = 11.4 hrs. LDCF test time = 34.6 min. ACF+ test time = 1.5 hrs SP+ test time = 3 hrs.

We conducted experiments on both INRIA and Caltech datasets to obtain the number of FP and TP, before and after the application of the CNN. Tables 4, 5, 6 and 7, present the mentioned metrics for each of the four methods: VJ, ACF, LDCF and SP+, respectively.

In general, the number of false positives is significantly reduced after the application of the CNN framework in the initial non-deep detectors proposals. This shows that our classification refinement strategy for the proposals is effective. As a negative side effect, the true positives number tend to decrease. However, these changes are substantially less significant to the methods overall performance when compared with the false positives reduction achieved by the CNN framework.

For instance, in the SP++RGB case (our best result for the Caltech dataset), the false positives after the application of the CNN represent only approximately 5% of its original value (i.e., more than a tenfold FP reduction), while the true positives still constitute approximately 95% of its original value. The impact experienced by the method's performance (using the usual metrics [11]) is positive, leading to an improvement of 4.82 percentage points, comparing to the base SP+ detector (as shown in the Table 2).

Table 4
Number of True Positives (TP) and False Positives (FP) for the proposals generated by the Viola and Jones (VJ) detector on the INRIA and Caltech datasets, and the CNN post-processing, using different feature maps to perform the fine-tuning.

Dataset	**Metrics**	**VJ Proposals**	**RGB**	**GM**	**Gx**	**YUV**	**LUV**
INRIA	TP	464	450	427	438	443	452
	FP	11859	939	631	816	773	809
Caltech	TP	212	202	194	178	183	191
	FP	17108	902	2622	3778	1894	1517

Table 5
Number of True Positives (TP) and False Positives (FP) for the proposals generated by the ACF (ACF+) detector on the INRIA and Caltech datasets, and the CNN post-processing, using different feature maps to perform the fine-tuning.

Dataset	**Metrics**	**ACF Proposals**	**RGB**	**GM**	**Gx**	**YUV**	**LUV**
INRIA	TP	551	546	533	542	540	546
	FP	1284	249	197	237	242	284
Caltech	TP	952	900	889	866	889	899
	FP	111087	7742	28596	16187	17234	12840

Table 6
Number of True Positives (TP) and False Positives (FP) for the proposals generated by the LDCF detector on the INRIA and Caltech datasets, and the CNN post-processing, using different feature maps to perform the fine-tuning.

Dataset	**Metrics**	**LDCF Proposals**	**RGB**	**GM**	**Gx**	**YUV**	**LUV**
INRIA	TP	554	550	543	549	549	552
	FP	386	127	104	119	122	140
Caltech	TP	947	894	896	874	889	891
	FP	51460	3712	14020	8847	8473	6528

Table 7
Number of True Positives (TP) and False Positives (FP) for the proposals generated by the Spatial Pooling+ (SP+) detector on the INRIA and Caltech datasets, and the CNN post-processing, using different feature maps to perform the fine-tuning.

Dataset	**Metrics**	**SP+ Proposals**	**RGB**	**GM**	**Gx**	**YUV**	**LUV**
INRIA	TP	-	-	-	-	-	-
	FP	-	-	-	-	-	-
Caltech	TP	985	939	932	913	917	926
	FP	223968	11576	58092	34775	34605	22257

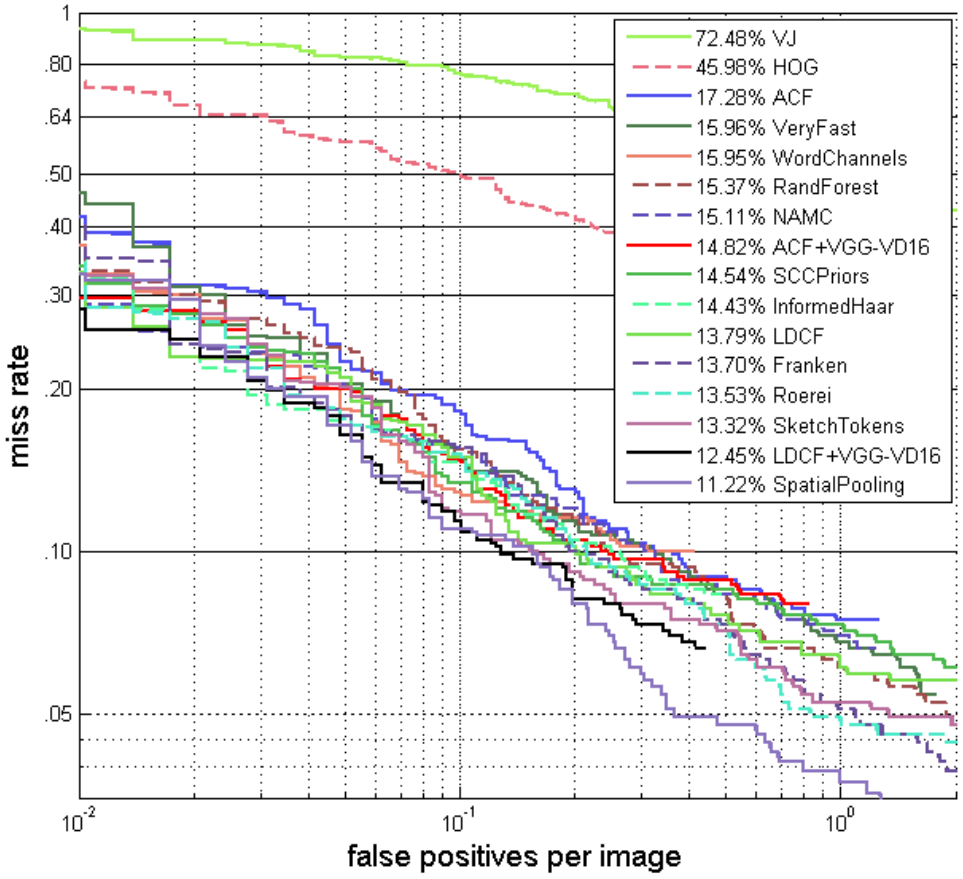

Fig. 5. Results in INRIA dataset. Improvement over the state-of-the-art non-deep detectors, cascading the ACF and LDCF with very deep CNN. The best result (shown in black bold) is 12.45% log average Miss Rate, achieved for the RGB feature map, cascading the LDCF with VGG-VD16.

5 Discussion and Conclusions

A novel methodology for pedestrian detection based on very deep convolutional learning is presented. The main outcome is that, it is experimentally observed that an improvement over the top non-deep state-of-the-art detectors for the task of PD is achieved. This is accomplished by cascading state-of-the-art non-deep detectors with a deep compositional architecture. This architecture not only allows fast test times but also actually improves the performance of the base detectors.

Detailing now the experiments conducted in this paper, one of the first issues to be tackled is that of initializing the architecture. It was experimentally demonstrated, that the initialization of the very deep architecture plays a relevant role in the performance. Although, we have used a robust initialization

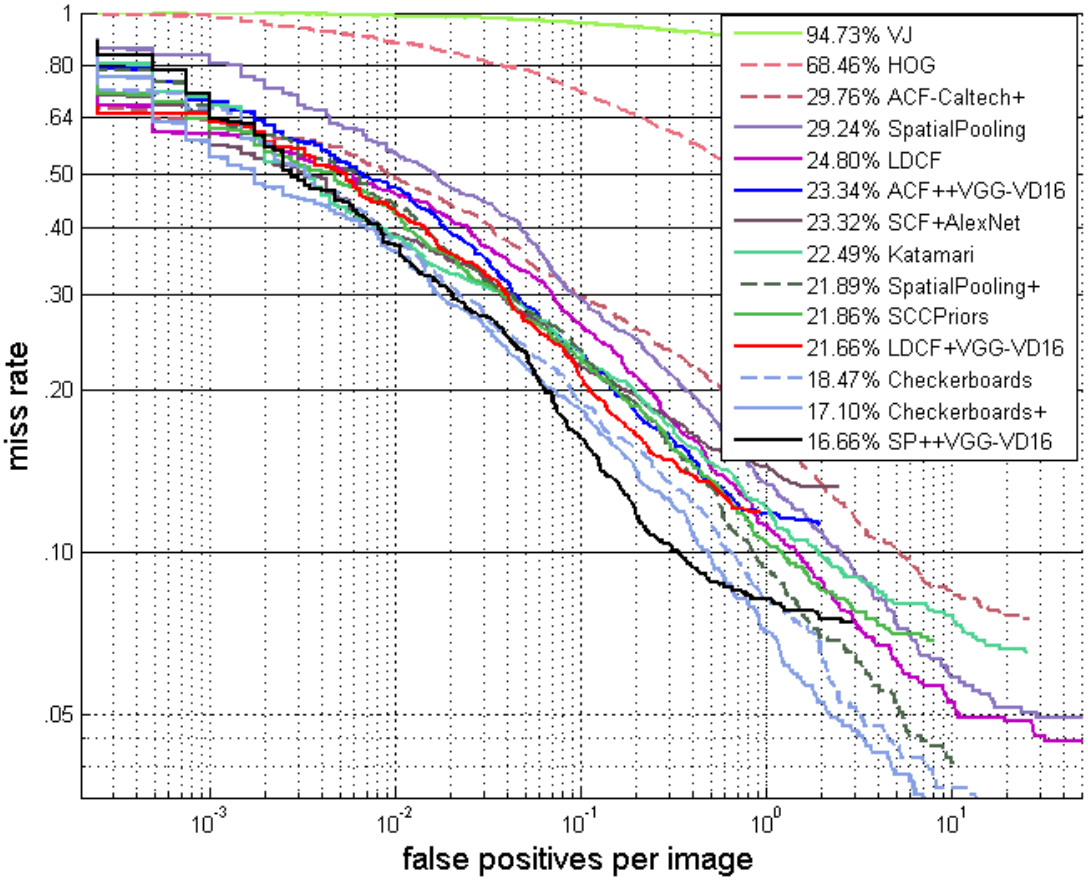

Fig. 6. Results in Caltech dataset with reasonable setting. Improvement over the state-of-the-art non-deep detectors cascading the ACF, LDCF and SP+ with very deep CNN. The best result (shown in black bold) is 16.66% log average miss rate, achieved for the RGB feature map, cascading the SP+ with VGG-VD16. Here we have included a cascaded deep detector (SCF + AlexNet - 23.32%) since it is similar to our approach.

that allows for a deep CNN to be trained from scratch , better results are achieved using the pre-trained models. Table 1 shows this issue for the RGB feature map. Concerning the adopted cascade approach, we also concluded that the improvement of the detectors is observed for both of the datasets, and also, that these results can be generalized varying the feature map computed from the bounding box. Table 2 testifies precisely this fact for different feature maps. Figures 5 and 6 show a comparison with other well known non-deep detectors used for PD. We note that, the (`ACF,VGG`), (`LDCF,VGG`) and (`Spatial pooling +, VGG`) exhibit top performances for both of the datasets.

Although, not being the main goal of this paper, we also compared the performance with deep methodologies recently proposed. Figure 7 shows these results, where we plot our best result obtained cascading `Spatial pooling`

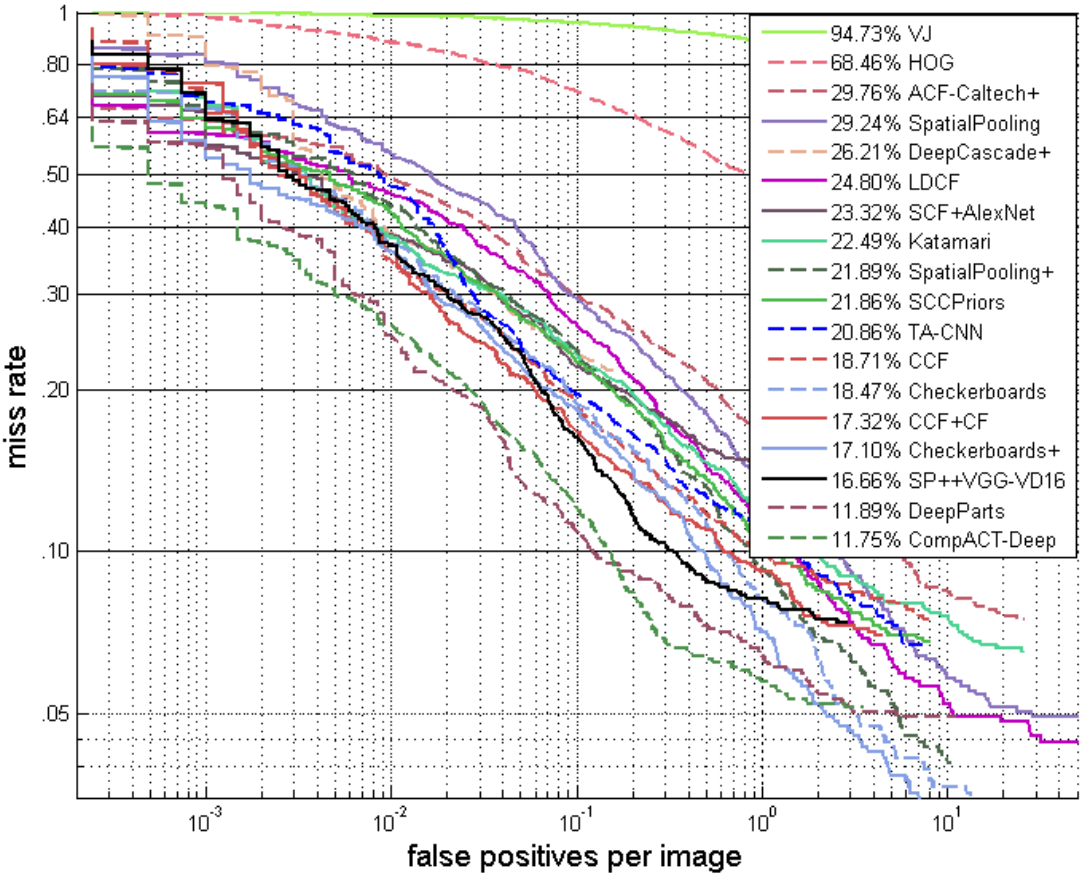

Fig. 7. Comparison with state-of-the-art approaches.

+ and VGG-VD16 architecture. It is seen that our methodology is in the top-three best methods in the literature for the Caltech reasonable dataset. The best approaches are [43,44] that were very recently proposed.

Although our approach is highly competitive, there is still room for improvement. Possible extensions of the methodology should be studied. For instance, and since, there is an improvement for various different feature maps, one approach could be to fuse in a principled way the individual responses of these feature maps. Other alternative could be to select more heterogeneous input channels (such as pedestrian body parts, or different views of the pedestrians), including distinct detectors for proposals and integrating a multiscale scheme in the network. Moreover, we could further demonstrate the generality of our methodology in improving any non-deep pedestrian detector. Therefore, we could apply the proposed framework to other top-performing non-deep detectors in the literature, for example, Checkerboards and Checkerboards+ [2]. Finally, our approach can benefit from recent results in deep learning, namely, novel ideas in deep-state-of-the-art pedestrian detectors could be adapted to

improve our architecture. For example, the shift handling procedure, using fully convolutional neural networks, employed in the Deep Parts method [43] could be used to further enhance our approach. We could also change the network model (e.g. to the GoogLeNet as in [43]), and extract and combine features from the last CNN layers to perform the classification.

References

[1] Piotr Dollár, Piotr's Computer Vision Matlab Toolbox (PMT), URL `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`

[2] S. Zhang, R. Benenson, B. Schiele, Filtered channel features for pedestrian detection, in: CVPR, 2015.

[3] P. Viola, M. Jones, Robust Real-Time Face Detection, in: IJCV, 2004.

[4] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005.

[5] P. Dollar, Z. Tu, P. Perona, S. Belongie, Integral channel features, in: BMVC, 2009.

[6] R. Benenson, M. Mathias, R. Timofte, L. V. Gool, Pedestrian detection at 100 frames per second, in: CVPR, 2012.

[7] S. C. P. Sermanet, K. Kavukcuoglu, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: CVPR, 2013.

[8] R. Benenson, M. Mathias, T. Tuytelaars, L. V. Gool, Seeking the strongest rigid detector, in: CVPR, 2013.

[9] S. Zhang, C. Bauckhage, A. Cremers, Informed haarlike features improve pedestrian detection, in: CVPR, 2014.

[10] P. Dollar, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection 36 (8), 1532–1545, 2014.

[11] P. Dollar, C. Wojec, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Machine Intell., 2012.

[12] C. Wojek, S. Walk, S. Roth, B. Schiele, Monocular 3d scene understanding with explicit occlusion reasoning, in: CVPR, 2011.

[13] M. Mathias, R. Benenson, R. Timofte, L. V. Gool, Handling occlusions with franken-classifiers, in: ICCV, 2013.

[14] W. Ouyang, X. Wang, A discriminative deep model for pedestrian detection with occlusion handling, in: CVPR, 2012.

[15] W. Ouyang, X. Wang, Joint deep learning for pedestrian detection, in: ICCV, 2013.

[16] Y. Bengio, Learning deep architectures for ai, Foundations and Trends in Machine Learning 2 (1), 1–127, 2009.

[17] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int. Journal of Comp. Vision (IJCV) 115 (3), 211–252, 2015. doi:10.1007/s11263-015-0816-y.

[19] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: NIPS, pp. 3320–3328, 2014.

[20] G. Carneiro, J. C. Nascimento, A. P. Bradley, Unregistered multiview mammogram analysis with pre-trained deep learning models, in: MICCAI, 2015.

[21] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, CoRR abs/1501.05790.
URL http://arxiv.org/abs/1501.05790

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR, 2015.

[23] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, in: ICLR, 2014.

[25] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders., Selective search for object recognition., IJCV, 2013.

[26] W. Nam, P.Dollar, J. H. Han, Local decorrelation for improved pedestrian detection, in: NIPS, 2014.

[27] S. Paisitkriangkrai, C. Shen, A. van den Hengel, Strengthening the effectiveness of pedestrian detection with spatially pooled features, in: ECCV, 2014.

[28] P. Fischer, A. Dosovitskiy, T. Brox, Descriptor matching with convolutional neural networks: a comparison to SIFT, CoRR abs/1405.5769, 2014.

[29] J. Zbontar, Y. LeCun, Computing the stereo matching cost with a convolutional neural network, CoRR abs/1409.4326.

[30] X. Chen, P. Wei, W. Ke, Q. Ye, J. Jiao, Pedestrian detection with deep convolutional neural network, in: ACCV workshop, 2014.

[31] J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, 2014.

[32] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[33] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition., in: ECCV, 2014.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CoRR abs/1409.4842, 2014.

[35] J. Hosang, R. Benenson, B. Schiele., How good are detection proposals, really?, in: BMVC, 2014.

[36] C. L. Zitnick, P. Dollar, Edge boxes: Locating object proposals from edges, in: ECCV, 2014.

[37] S. Walk, N. Majer, K. Schindler and B. Schiele, New features and insights for pedestrian detection, in: CVPR, 2010.

[38] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten years of pedestrian detection, what have we learned?, in: ECCV workshop, 2014.

[39] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for generic object detection, in: ICCV, 2013.

[40] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[41] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Int. Conf. on Artificial Intelligence and Statistics, pp. 249–256, 2010.

[42] A. Vedaldi, K. Lenc, Matconvnet – convolutional neural networks for matlab, in: Proceeding of the ACM Int. Conf. on Multimedia, 2015.

[43] Y. Tian, P. Luo, X. Wang, X. Tang, Deep learning strong parts for pedestrian detection, in: ICCV, 2015.

[44] Z. Cai, M. Saberian, N. Vasconcelos, Learning complexity-aware cascades for deep pedestrian detection, in: ICCV, 2015.