A Study on Pedestrian Detection using a Deep Convolutional Neural Network

Ismael Orozco[†], María E. Buemi[††], Julio Jacobo Berlles[††]

[†]Departamento de Informática, Facultad de Ciencias Exactas, Universidad Nacional de Salta, Argentina.
[††]Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina.
ciorozco.unsa@gmail.com, {mebuemi;jacobo}@dc.uba.ar

Keywords: Pedestrian detection, Convolutional Neural Network, Haar-like Features, AdaBoost, Deep Learning

Abstract

Pedestrian detection is presently a topic of interest in computer vision due to its applications as an aid in car driving and in surveillance. The good results obtained using Convolutional Networks for vision tasks make them an attractive tool to improve the capabilities of pedestrian detection systems. In this work we study the use of a Convolutional Network as a refinement for classification of candidate regions previously detected using Haar features embedded in an AdaBoost scheme. The data used for training and testing come from the INRIA pedestrian database. The influence of design parameters, such as, the number of stages of the cascade in the detection stage and the scale factor in the pyramid of the multi-scale method, have been studied.

1 Introduction

The importance of pedestrian detection systems stems from the demand of computer vision applications in fields such as e.g. car driver assistance and surveillance of public places. It constitutes an important challenge due to the variety of scales, positions, illumination, body shapes and contexts at which a pedestrian can be found. Also the problem of partial occlusion adds to the complexity of the task. Many approaches to this problem, based on Convolutional Networks, have been proposed [4],[15],[16]. Some of them have a first stage that selects candidate regions to be passed on to the Convolutional Network for further analysis. Among the methods that can be used to generate these candidate regions we can mention VJ [18], ACF [6], HOG [5], HOG+SVM [19], DPBM [10] as some of the most important ones. The detected candidate regions are taken as inputs to Convolutional Networks that in turn decide the presence/non-presence of a pedestrian in them. In this work we use a LeNet5 network, specially trained for pedestrian detection. This network is fed Regions of Interest (ROIs) selected by a multi-scale method based on Haar features within a cascaded AdaBoost scheme.

In this paper, we combined the speed of the Haar detector for the detection of the ROIs with the robustness of the Convolutional Networks for their classification. The first stage is called the *detection stage* and the second the *classification stage*. That is the basics of the approach outlined in [11]. This paper is organized as follows: 1 states the goal of this work, lists briefly some of the approaches for the detection stage and mentions other works on the use of Convolutional Networks for pedestrian detection. Section 2 explains the implementation of the detection and the classification stages. Section 3 reports the results of the experiments performed using the INRIA pedestrian database and Section 4 analyzes the obtained results, conclusions and future works.

2 Pedestrian Detection Approach

The scheme of the approach used in this work is constituted by two stages. A first stage of ROI recognition, in which rectangular candidate regions, of different sizes and positions, are selected. This stage is followed by a classification stage, in which a Convolutional Neural Network classifies each candidate region as pedestrian or non-pedestrian.

2.1 Detection Stage

In the ROI detection stage, a detector based on Haar features [18] is used. The features of this detector are calculated very efficiently using integral images. It also uses a cascaded structure that combines Adaboost classifiers. These characteristics result in a very high detection speed [11].

In the implementation of the detection stage, the OpenCV library was used. It provided an efficient implementation of the object detection framework based on the work of Viola and Jones.

Training process: In order to develop the detection stage that detects as many pedestrians as possible, the detection rate should be kept high, even if many detections are false positives. For this reason, we carried out a series of training processes, using training samples from different sources that were all re-sized to 12×28 pixels. This also determines the size of the the smallest pedestrian that can be detected in the image. The positive samples show a wide range of different poses and appearances. Throughout this paper, we worked with the INRIA Pedestrian Dataset [5] as well also with the MIT Dataset

Figure 1. Part of the generated samples obtained using the create_samples tool from OpenCV

[14]. Thus, the cascade was trained with the training set given therein: 2416 positive images, 9000 negative images obtained as sub samples of the 1218 negative images from INRIA finally with 1848 positive images from MIT. These 4264 positive images correspond to the original and the reflection of every positive image of the training dataset. The training images were cropped closely around the annotated persons, because Haar-cascades do not benefit from having so much background included.

OpenCV includes a tool *create_samples* that is used to generate a large number of positive samples from our positive images, by applying transformations and distortions. Some of the generated samples are shown in Figure 1.

The training process is influenced by the training data themselves, the minimal detection rate and the maximal false positive rate per cascade level. These parameters implicitly determine the number of evaluated Haar features in each cascade level and the total number of levels, which in turn are responsible for the overall detection speed of the classifier. Experiments showed that, using a large number of training samples and high rates for detections (99.5%) and moderate rates of false positives (50%) per cascade level, yields best results. This led to a Haar classifier cascade with only 19 levels. Even if an input sample has to pass through all stages, its detection speed is acceptable.

The detection stage returns bounding boxes for all the potential pedestrians in the picture, which are sent on to the classification stage. An example of the output of the detection stage can be seen on Figure 2.

2.2 Classification Stage

In this stage, we used a Convolutional Neural Network that takes the detected regions as inputs and classifies them as pedestrian or non-pedestrian. We used the architecture proposed for [12]. The implementation of the Convolutional Network used in the classification stage was done in Python using the Theano Library [2], [3].

The architecture used in this paper is a case of a LeNet-5 network, proposed by LeCun [12]. Sparse convolutional layers and max-pooling are at the heart of the LeNet family of models. While the exact details of the model can vary greatly, the

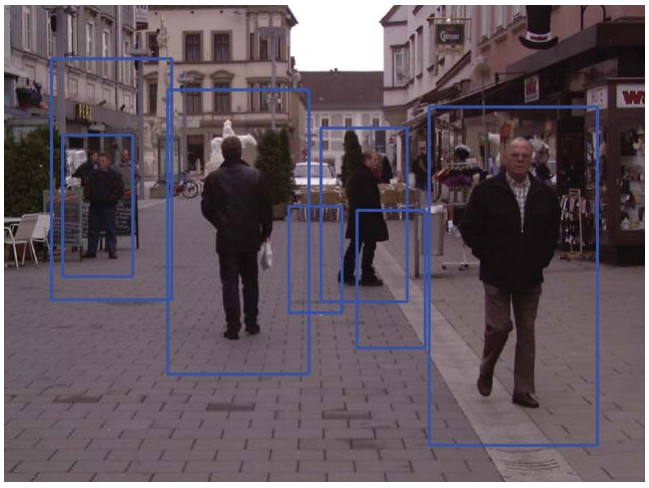


Figure 2. Example of the output from the detection stage. Blue boxes indicate the detections produced as the output of the detection stage.

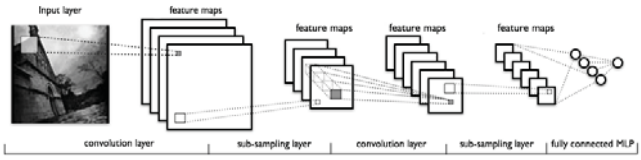


Figure 3. Architecture of the DCNN - LeNet5, showing the alternance of convolutional layers and subsampling layers followed by a Multi Layer Perceptron.

figure below shows a graphical depiction of a LeNet model in Figure 3.

The proposed architecture has 4 layers. It takes input images of 64×128 pixels in size, so first we re-size the sub-images obtained by the first stage, the transformed images are filtered using 20 kernels of size $6 \times 6 \times 3$ with a stride of 2 pixels, the max-pooling is applied in a 2×2 grid. The second layer has 50 kernels of size $5 \times 5 \times 20$, and a max-pooling stage with a 2×2 grid. The last two layers constitute a fully connected Perceptron. In it, the input layer has 1050 nodes, the hidden layer has 500 nodes, and there output is made of 2 nodes.

Training process: We trained the network using only the INRIA pedestrian dataset [5]. Regions of interest with pedestrians were extracted generating windows that were re-sized to 64×128 pixels. Each pedestrian image was mirrored along the horizontal axis to expand the dataset. Similarly, we added 7 variations of each original sample using deformations such as translations and scaling. We obtained a total of 33810 positive samples. Background samples were extracted at random from the negative images. We obtained a total of 50123 negative samples. The final dataset contained 83933 samples, 10% of which were used for validation.

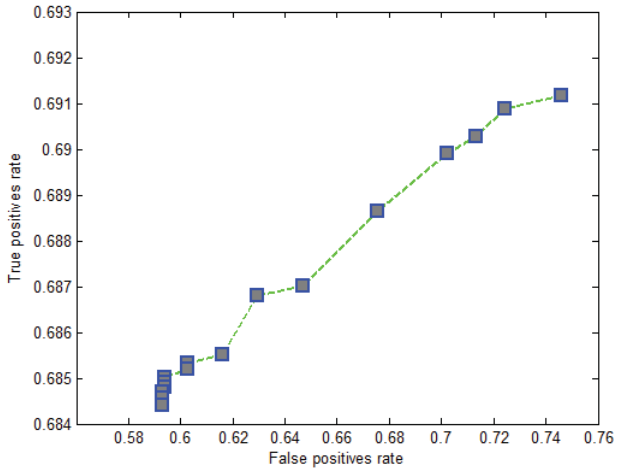


Figure 4. Receiver-Operating-Characteristic (ROC) graphic for the full system with varying k from 25 (point on the left) and 10 (point on the right).

3 Experiments and Results

In order to set various parameters so that the best possible performance is achieved, several experiments have been performed. While the training part of the INRIA dataset was used to train both the detection stage and the verification stage, the test part has been used as base for these experiments. It contains 742 images in total, of which 289 contain one or more persons. In total the test set contains 589 persons that should be detected by a perfect system.

The baseline for the comparison is the performance of the system in a configuration similar to the one used by Geismann and Schneider [11]. One of the most important parameters in the system is the number of stages in the Haar-cascade, in this paper designated as k. It is interesting to see what impact the changes in k have on the complete system. In Figure 4, an ROC curve is shown for the full system with varying depths in the detection stage. The importance of this parameter is evident. As k decreases, the number of detections grow, but at a large cost in false positives. The final system uses $k = 19$, since it seems to give an acceptable trade-off between true positives and false positives.

We tested the combined system with the images in INRIA test dataset. To compare each result with the ground truth we used a measure also used in the PASCAL object challenge [9]. A detection was considered correct if the overlap between the area of the ground truth bounding box and the area of the detected bounding box was more than 50%. This is:

$$\frac{|B_{dt} \cap B_{gt}|}{|B_{dt} \cup B_{gt}|} > 0.5,$$

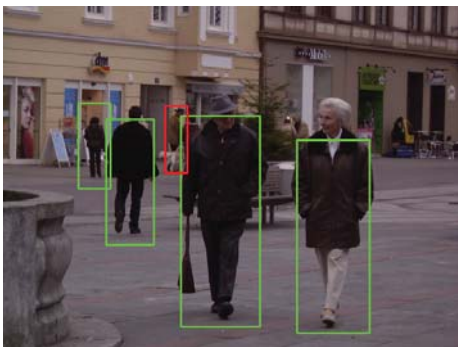
where, B_{dt} is the detected region and B_{gt} is the ground truth region.
In the detection stage, from the original image, we generated all possible windows of size 12×28, using sliding window and pyramidal analysis. The candidate windows were all re-sized to 64×128 and fed to the input of the classification stage. One of the most important parameters in the system is the scale factor which re-scales the image of the pyramid. Table 1. shows how the choice of this parameter affects the detection rate, FPPI and total time per image.

Scale	detection rate	FPPI	time per image (secs.)
1.01	0.27	2.34	0.41
1.02	0.59	1.30	0.29
1.03	0.66	0.84	0.21
1.04	0.64	0.69	0.16
1.05	0.65	0.55	0.14
Total number of pedestrians in the ground truth = 588			
Total number of images in the test set = 289			

Table 1. Result of detection rate, FPPI and time per image, for test INRIA dataset

Examples of the final detection can be seen in the two images in Figure 5. In both cases, the green bounding boxes of the detected pedestrians exceed the 50% overlay with the ground truth. Red bounding boxes indicate non-detected pedestrians.

(a)



(b)

Figure 5. Examples of pedestrian detection with the combined system for two images of the INRIA dataset. Green bounding boxes indicate detected pedestrians. Red bounding boxes indicate non-detected pedestrians.

4 Conclusions

This paper proposes the study of a processing scheme for pedestrian detection, based on a cascade classifier with Haar-like features and a Convolutional Network. The cascade classifier was trained to determine the optimal value of the number of stages, denoted as k. The Convolutional Network was implemented using the Theano Library [2], [3], and was trained with the INRIA dataset. Acceptable results were attained using the INRIA dataset, enriching it by the generation of additional images resulting from rotations and translations to the existing ones. The purpose of this was to make the system more robust. The influence of the scale factor on several performance parameters, such as, detection rate, FPPI (false positives per image) and processing time per image, was assessed. Among the possible future work we could mention the use of other datasets like Caltech-USA [7] and KITTI [8]. Also, for the detection stage, other methods, like Objectness [1], Selective Search [17], ACF [6], LDCF [13] should be explored.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2189–2202, November 2012.

[2] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[3] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[4] Xiaogang Chen, Pengxu Wei, Wei Ke, Qixiang Ye, and Jianbin Jiao. *Computer Vision - ACCV 2014 Workshops: Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part I*, chapter Pedestrian Detection with Deep Convolutional Neural Network, pages 354–365. Springer International Publishing, Cham, 2015.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[6] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, Aug 2014.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[8] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE Press, June 2008.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[11] P. Geismann and G. Schneider. A two-staged approach to vision-based pedestrian recognition using haar and hog features. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 554–559, June 2008.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[13] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved detection. *CoRR*, abs/1406.1134, 2014.

[14] M. Oren, C.P. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. pages 193–99, 1997.

[15] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. *CoRR*, abs/1212.0142, 2012.

[16] Denis Tomè, Federico Monti, Luca Baroffio, Luca Bondi, Marco Tagliasacchi, and Stefano Tubaro. Deep convolutional neural networks for pedestrian detection. *submitted to Elsevier Journal of Signal Processing: Image Communication*, abs/1510.03608, 2015.

[17] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[18] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[19] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1491–1498, Washington, DC, USA, 2006. IEEE Computer Society.