RELIABLE PEDESTRIAN DETECTION USING A DEEP NEURAL NETWORK TRAINED ON PEDESTRIAN COUNTS

Sanjukta Ghosh [⋆†] *Peter Amon* [†] *Andreas Hutter* [†] *André Kaup* [⋆]

[⋆] Multimedia Communications and Signal Processing,
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[†] Sensing and Industrial Imaging, Siemens Corporate Technology, Munich, Germany

ABSTRACT

Pedestrian detection is an important task for applications like surveillance, driver assistance systems and autonomous driving. We present a novel approach for detecting pedestrians using a deep convolutional neural network (CNN) trained for counting pedestrians. Our method avoids the need for annotation of the position of the pedestrians in the training data via bounding boxes. The deconvolved outputs of the filters of the trained counting model are used to detect the pedestrians. The average miss rate values on the tested datasets were found to be in the same range as other methods in spite of a simpler training using only pedestrian counts. This method is found to be suitable for detecting pedestrians in crowded scenes with occlusion as well as less crowded scenes.

Index Terms— Pedestrian Detection, Deep Learning, CNN, Counting Model, Deconvolution

1. INTRODUCTION

Detecting objects is critical in surveillance applications, driver assistance systems, autonomous driving or any application where information about the environment is required. Often it is required to detect a specific category of objects like pedestrians or vehicles. Deep learning [1], [2] has shown promising results for object detection [3] in general and specifically also pedestrian detection. Most of the approaches for pedestrian detection require specifying the explicit locations of pedestrians in an image frame during training. This paper comprises of a novel approach for pedestrian detection in which the explicit locations of pedestrians are not required during training. The main contribution of this paper is using a deep model trained for counting pedestrians to detect pedestrians. Transfer learning is used to train a model for counting pedestrians using synthetic images. The deconvolved outputs of the learned filters are then used to detect pedestrians.

The research leading to these results has received funding from the German Federal Ministry for Economic Affairs and Energy under the VIRTUOSE-DE project.

2. RELATED WORK

Multiple solutions to pedestrian detection have been developed over the years and continue to being developed. This section provides an overview of the techniques for pedestrian detection using hand-crafted features and more recently using deep learning approaches.

2.1. Detection using Hand-crafted Features

Viola and Jones applied the VJ detector [4] for the task of pedestrian detection. Dalal and Triggs [5] proposed the histogram of oriented gradients (HOG) detector. Felzenswalb et al. [6] described the use of deformable part models (DPM) for object detection. Various techniques for object detection have been proposed based on DPM.

2.2. Detection using Deep Learning

In [3], region proposals are used along with deep convolutional neural networks (CNNs) trained for image classification to detect objects. The region proposal algorithm proposes regions where objects are possibly present. Each region is then run through a CNN trained for image classification followed by a set of linear SVMs. A bounding box regressor is used to obtain the final bounding box around the object. Training data for the different stages comprise of image frames with objects and metadata corresponding to the bounding boxes of the objects in the image frame. Angelova et al. [7] use a large field of view (LFOV) deep network to detect pedestrians for autonomous driving. The deep network processes large areas of the image and makes decisions about the presence of pedestrians at multiple locations. For training, all possible square boxes around the pedestrian are generated. [8], [9], [10], [11] detail some of the other methods using deep neural networks for pedestrian detection. A common requirement in the training data for these approaches is that the locations of the objects in the image frames need to be annotated by the bounding boxes, for example using the top left and bottom right corner co-ordinates of the bounding boxes.

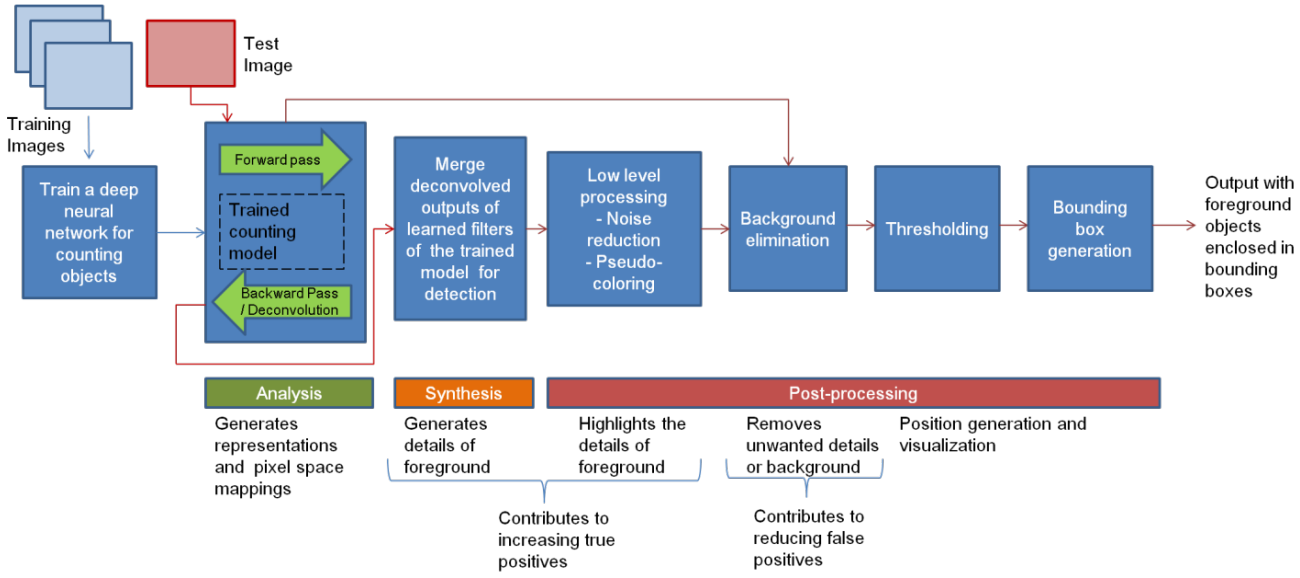

Fig. 1. Steps for pedestrian detection using a deep counting model.

3. PROPOSED METHOD FOR PEDESTRIAN DETECTION

The goal of detection is to locate the objects present in an image frame. Therefore, it is required to maximize the true detections while minimizing the false detections. The first stage of our approach involves training a deep model for counting objects, in this case pedestrians. This means that the labels required in the training set are simply a single number which is the count of objects in a single frame. In the next stage, the filters learned by the different layers of the deep neural network are used to analyze an input image for detecting the foreground objects. Fig. 1 depicts the steps of our approach.

3.1. Training a Deep Model for Pedestrian Counting

Transfer learning is used to train a CNN for counting pedestrians. The base network used is AlexNet [2] which is trained for image classification. The cross entropy loss function with regularization is used for training the deep model for counting and is as follows:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} t_{ij}\log y_{ij} + \frac{\lambda}{2N}\|\mathbf{w}\|_2^2 \qquad (1)$$

where L is the loss which is a function of the parameters, θ, comprising of the weights and biases. Furthermore, N is the number of training images, C is the number of classes where each class represents a count of the pedestrians in the input image, y is the predicted count of pedestrians, t is the ground truth or the actual count of pedestrians and $\mathbf{w}$ represents the weights. Details of the training and performance of the deep counting model can be found in our previous work [12]. The counting model is trained using synthetic images with varying counts of pedestrians. There also exists the possibility of using natural images or a combination of synthetic and natural images to train and tune the CNN. The deep counting model can be tuned for the target dataset using fewer images. Training on synthetic images avoids the need for large annotated training data from the target site during training. The trained counting model was then used to analyze natural images with pedestrians to be detected not experienced by the model during training.

3.2. Deep Counting Model for Pedestrian Detection

The trained deep counting model is used as an analysis tool as shown in Fig. 1. On visualizing the learned filters of the deep model for counting objects and the features of the input image causing activations, it was found that the trained counting model represents well features relevant for detecting the foreground objects which in this case are pedestrians. To map the features of the inputs causing activations, a backward pass through the trained model was done using the deconvolution approach from [13]. This method involves successive unpooling, rectification and filtering to map to the layers lower in the hierarchy causing the activations. If the deconovlution is propagated all the way to the input, a mapping is obtained in pixel space which reveals the features causing activations. The analysis stage involves a forward pass through the network resulting in various representations and a backward pass through the network for selected channels as can be seen in Fig. 1. Though our deep network has not been explicitly trained for detecting objects, in order to achieve the task of counting objects, it learns filters that are activated by the foreground objects. This fact is exploited to achieve the detec-

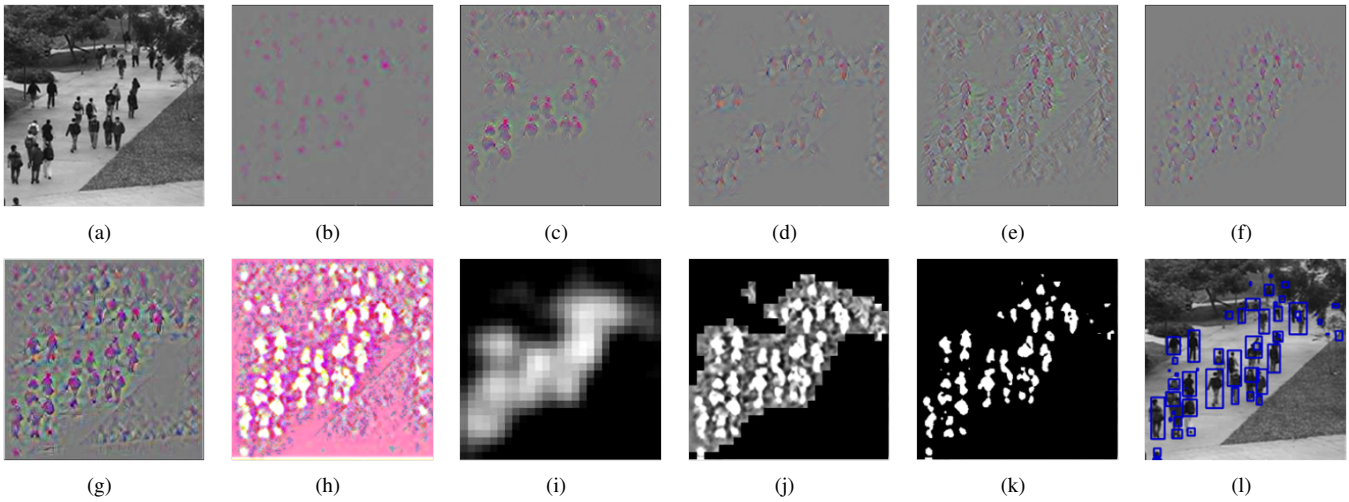

Fig. 2. Pedestrian detection on a frame from the UCSD dataset.

tion. The lower layers of the deep neural network represent low-level features and using these would require additional intelligence to separate the foreground from the background. With the increasing hierarchy of the layers, higher level concepts are learned until the highest layers which are the most task-specific. A combination of channels from the mid-level layers and the higher layers are used to detect the foreground objects. Different channels of the mid-level layers are activated by various parts of the foreground objects, for example, some of the filters are activated by the feet while some by the head and so on. By merging these deconvolved outputs, it is possible to obtain an indication of the foreground objects. Merging can be achieved by image compositing techniques for blending images. The synthesis stage comprises of merging the results of the backward pass or the deconvolved outputs of the selected channels in order to highlight the foreground objects more clearly. Here a combination of multiply and screen blending has been done to achieve hard light blending.

While the synthesis stage serves to generate the details of the foreground, the next stages of noise reduction and pseudo-coloring are aimed at further highlighting the foreground region over the background in order to increase the true detections. Noise reduction can be achieved on the composite image by using techniques where different color channels are treated separately. In our case the details of the red and blue channels are preserved while suppressing the details of the green channel. This is because the deconvolution results in a heatmap with features of the input image causing activations to be highlighted in colors dominated by red and blue.

An unwanted side effect of the previous stages is a highlighting of certain parts of the background. The background elimination stage as shown in Fig. 1 is responsible for reducing false detections. With the increasing hierarchy of the layers of the deep neural network, the filters learn the concept of the foreground increasingly better. It is observed that filters in the convolutional layer 5 of the current example have an output that is able to localize the foreground well while being able to distinguish from large parts of the background. It is possible to obtain information about the separation between the foreground and background of the image frame at a global level. By comparing the channel output of convolutional layer 5 with the output obtained after low-level processing (noise removal and pseudo-coloring), parts of the background are eliminated. The concept is similar to that of guided filtering where the content of a guide image is used to filter the actual image. Likewise, in this case the filter output of the convolutional layer 5 serves as the guide image. For regions with low or zero pixel value in the guide image, the output of the image being filtered is discarded or considered part of the background while the rest are considered as foreground. This is followed by thresholding to segment the foreground region. Finally the segmented foreground is analyzed for closed contours and enclosed by bounding boxes.

4. EXPERIMENTS

The proposed approach was tested on different datasets. The deep model for counting was trained using the Caffe framework [16]. The metric used to measure the performance is average miss rate (AMR). The receiver operating characteristics (ROC) curves plotting miss rate versus false positives per image (FPPI) and the AMR values obtained by the area under the ROC curve in the range of 10^{-2} to 10^{0} FPPI are computed using the toolbox from [17].

4.1. Pedestrian Datasets with Crowds

The UCSD pedestrian dataset [18] has a high count of pedestrians as compared to datasests like Caltech [19], [20], ETH

Table 1. AMR on CUHK08.

Method	Average Miss Rate(%)
DPM [6]	47
ACF [14]	42
DPM + Scale Prior [9]	39
DPM + Scale Prior + Occlusion Analysis [9]	33
Our Method	26

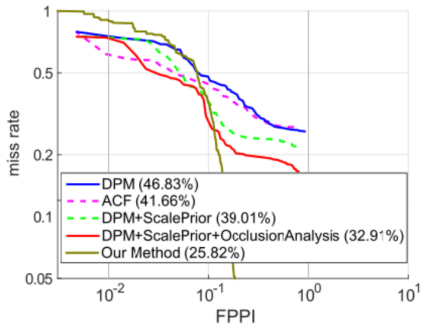

Fig. 3. ROC for CUHK08.

Table 2. AMR on Caltech dataset.

Method	Average Miss Rate(%)
VJ [4]	95
HOG [5]	68
Checkerboards [15]	18
DeepParts [10]	12
MS-CNN [8]	10
F-DNN [11]	9
Our Method	14

[21] and Daimler [22] used for pedestrian detection. Fig. 2 illustrates the outputs at each stage of the algorithm described in Section 3 to arrive at the output with detected pedestrians. Fig. 2a is an input image from the UCSD dataset. Figs. 2b to 2f represent some of the deconvolved outputs of different learned filters from the convolutional layer 3 and 5. Fig. 2g shows the image obtained after merging the deconvolved outputs of the different learned filters. Fig. 2h shows the image obtained after noise removal and pseudo-coloring. Fig. 2i represents the output of a learned filter of convolutional layer 5. As can be observed, the foreground region is localized. This is used to eliminate parts of the background. The resulting image with the eliminated background is as shown in Fig. 2j. As can be observed, detections are obtained for partially occluded pedestrians and numerous pedestrians close to each other in a single image frame. Fig. 2k shows the image obtained after thresholding. This is followed by detecting the closed contours and enclosing in bounding boxes. Fig. 2l shows the bounding boxes superimposed on the input image.

The proposed method was also tested on the CUHK Occlusion dataset [23]. Table 1 shows the AMR values and Fig. 3 the corresponding ROC curves of our method in comparison with some of the other techniques on CUHK08 which has scenes with multiple pedestrians moving together and hence occluded by each other. Our method performed better than the other methods with an AMR of 26%.

4.2. Caltech Pedestrian Dataset

The proposed approach was also tested on the test set of the Caltech pedestrian dataset [19], [20] comprising of set06-set10 for the 'Reasonable' setting for pedestrians measuring 50 pixels or greater in height and with a visibility starting from 65% as is the commonly used condition for evaluation of the pedestrian detection algorithms. The counting model trained on synthetic images was tuned using the Caltech training set (set00-set05) to detect presence or absence of pedestrians in an input image frame. Only the frames where the model predicts a presence of pedestrians are further processed to detect pedestrians. This step helps to reduce false detections in frames where no pedestrians are present while also accelerating the processing since only the relevant frames are processed. For the frames that are processed using the algorithm described in Section 3, in order to further reduce false detections, the model tuned for detecting presence or absence of pedestrians is run on crops around each of the detected bounding boxes. Based on the prediction of this deep model, the final detections are done. Table 2 shows the comparison of the AMR using different pedestrian detection techniques on the Caltech pedestrian dataset. While some of the methods outperform our method, our method with an AMR of 14% shows that it is in the same range of values and is also effective in spite of using a simplified annotation of the count of the pedestrians in training. DeepParts [10], MS-CNN [8] and F-DNN [11] all involve training multiple networks with the need for annotations involving bounding boxes of the entire pedestrian or different parts of the pedestrian or pixel level masks. So training the multiple networks for each method to effectively use them is much more complex than our method.

5. CONCLUSION

We present a novel approach for detecting pedestrians. By using a deep model trained for counting, extensive annotations of the locations of pedestrians by way of bounding boxes can be avoided. The deep neural network does not need to be trained for detection explicitly. This is advantageous in the case of training deep models which require a large amount of training data that needs to be annotated for supervised learning. The annotations required for training the counting model are simple since only a single number which is the count of the objects is required. Moreover, no region proposal algorithm is used as is common with a lot of detection algorithms. In our case, the localization of the pedestrians is achieved by exploiting the internal representations of the deep model trained by the pedestrian counts. Our method is effective for detecting pedestrians in less crowded as well as crowded scenes with occlusion.

6. REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems.*, 2012, pp. 1106–1114.

[3] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[4] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 734–741 vol.2.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 1, pp. 886–893 vol. 1.

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[7] A. Angelova, A. Krizhevsky, and V. Vanhoucke, "Pedestrian detection with a large-field-of-view deep network," in *Proceedings of ICRA 2015*, 2015.

[8] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, 2016, pp. 354–370.

[9] L. Wang, L. Xu, and M. H. Yang, "Pedestrian detection in crowded scenes via scale and occlusion analysis," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1210–1214.

[10] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1904–1912.

[11] X. Du, M. El-Khamy, J. Lee, and L. S. Davis, "Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection," *CoRR*, vol. abs/1610.03466, 2016.

[12] S. Ghosh, P. Amon, A. Hutter, and A. Kaup, "Pedestrian counting using deep models trained on synthetically generated images," accepted for the 12th International Conference on Computer Vision Theory and Applications (VISAPP), Feb 2017.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, 2014, pp. 818–833.

[14] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[15] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," *CoRR*, vol. abs/1501.05759, 2015.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[17] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," https://github.com/pdollar/toolbox.

[18] A. B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–7.

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

[20] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, vol. 34, 2012.

[21] 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)*. June 2008, IEEE Press.

[22] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[23] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3258–3265.