



Pedestrian Detection and Gender Recognition from Non-Intrusive Camera Videos using Convolutional Networks

March 06, 2017

Federico Zanetti ^[1]

Supervisor: Nicola Conci

**University of Trento,
38123 Povo (TN) - Italy**

^[1] *federico.zanetti@studenti.unitn.it*

1 Abstract

Gender Recognition is a challenging problem in the field of pattern recognition. In this work, we propose a deep learning model that can learn the joint high-level and low-level features of human body to address this problem. Our convolutional networks apply convolution and subsampling in extracting the local and abstract features of human body and decompose the raw input images to learn global and effective features for a recognition task. Before doing so we focus on the detection challenge posed by real-life scenarios of pedestrians moving in a public environment. The objective of this work is to find out whether the image of a pedestrian from afar can provide by itself useful information to infer the gender.

Keywords: Pedestrian Detection, Gender Recognition, Support Vector Machines, Convolutional Networks, Supervised Training.

2 Introduction

Gender recognition is an intelligent video surveillance task that provides the soft-biometric gender information of a person. In this work such task is applied to a real-case scenario of a pedestrian freely circulating in a public area among other peers. As such the task is closely related to the detection of such pedestrians. Pedestrian detection is a key problem in computer vision, with several applications that have the potential to positively impact the quality of life. In recent years, the number of approaches to detecting pedestrians in images has grown steadily, but multiple datasets and varying evaluation protocols are used and make comparisons difficult. Some of the most popular datasets include Caltech, ETH, TUD-Brussels, Daimler and the INRIA datasets [3].

We make the following contributions: (1) we put together a large pedestrian gender dataset from another realistic crowd scenario dataset to support the gender recognition task. (2) We develop a detection method and then use deep learning methods to carry out the recognition of the gender. (3) We propose an evaluation experiment that allows to perform informative comparisons and (4) we evaluate the performance on our newly developed dataset. Our experiments show that despite encouraging results, performance still has much room for improvement. Undoubtedly challenges to gender recognition include: different clothing appearance, different postures, different point of views on the pedestrians, the presence of occluding obstacles and overlapping between pedestrians in crowds. In real-case scenario additional challenges are due to baby strollers, bags, wheel bags, walking sticks and other tools that are not necessarily connected to one of the two sexes.

In particular, detection is disappointing at low resolutions as the method does not fully adapt to multi-scale pedestrians. The training set that was used

for detection has in fact a limited number of scales of pedestrians, and those that are moving afar from the camera are particularly challenging to detect.

3 Theoretical framework

3.1 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for regression or preferably classification challenges. In this algorithm, each data item is plotted as a point in the n-dimensional space, being n number of available features with the value of each feature being the value of a particular coordinate. Then classification is carried out by finding the hyperplane that best differentiates the two classes. If the training data are linearly

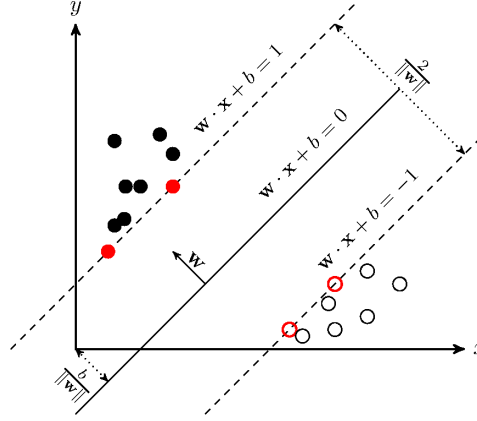


Figure 1: SVM hard margin visualization.

separable hard margins are used. Two parallel hyperplanes that separate the two classes of data can be selected, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. These hyperplanes can be described by the equations: \vec{w}

$$\vec{w} \cdot \vec{x} - b = 1 \quad (1)$$

$$\vec{w} \cdot \vec{x} - b = -1 \quad (2)$$

To extend SVM to cases in which the data are not linearly separable, the hinge loss function is used to compute a measure of the loss:

$$\max(0, 1 - y_i \cdot (\vec{w} \cdot \vec{x}_i - b)) \quad (3)$$

This function is zero if the \vec{x}_i lies in the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin. The equation to minimize is:

$$\left[\frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)) \right] + \lambda \|\vec{w}\|^2 \quad (4)$$

where the parameter λ determines the tradeoff between increasing the margin-size and making sure that the \vec{x}_i lies on the correct side of the margin. For sufficiently small values of λ , the soft-margin SVM will behave identically to the hard-margin SVM if the input data are linearly classifiable, but will still learn a viable classification rule if not. SVMs have wide applications in image and video recognition and can be trained for detection tasks.

3.2 Convolutional Networks

A Convolutional Network (CNN) is a feed-forward artificial network in which the connectivity pattern between its neurons is obtained by sliding a set of filters on an input space between subsequent layers. Individual neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation. In convolutional layer, each neuron is connected locally to inputs from the previous layer, which functions like a 2D convolution with a certain filter, then its activation could be computed as the result of a nonlinear transformation:

$$\alpha_{i,j} = \rho(f * x) = \rho\left(\sum_{i'=1}^n \sum_{j'=1}^n f_{i',j'} x_{i+i',j+j'} + b\right) \quad (5)$$

where f is an $n \times n$ weight matrix of the convolutional filter, x refers to the activations of the input neurons connected to the neurons (i,j) in the following convolutional layer. $\rho()$ is a nonlinear activation function (usually sigmoid or hyperbolic tangent), b is the bias, $*$ is the convolution operator.

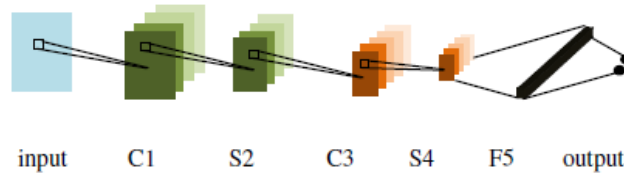


Figure 2: CNN used in [2] for gender recognition.

3.3 Background on Gender Recognition

Video surveillance is an important topic in modern-day society. In recent years, the number of installed surveillance cameras has increased drastically in urban and general living environments. This results in an increasing demand for automatic and intelligent video content analysis. Gender is semantically a very interesting characteristic. First, people are detected in a video stream and secondly people are classified from the patches of people's body profile or face. Gender recognition from facial images and videos has been explored in works such as Khryashchev *et al.*[4] based on adaptive feature extraction and support vector machine classification. They reach top accuracies around 90% but the need of a clear face image need to deploy more "intrusive" methods of detection. Full body gender recognition has been done in Geelen *et al.*[5] using entirely SVMs trained on HOG and other features, reaching accuracies approximately around 80% on the MIT CBCL dataset. As deep learning studies and CNNs have developed in recent years, these have been exploited in this context as well. In Ng *et al.*[2] a 4 layer CNN (two convolutional layers and two pooling layer) has been trained on the MIT dataset reaching on average about 75% of test accuracy on different features (RGB, YUV, grayscale). CNNs have also been employed for the detection task itself, to basically perform pedestrian segmentation [1]. Here different techniques (HOG+SVM and AdaBoost, CNNs) have been tested on the Caltech Dataset.

4 Our work

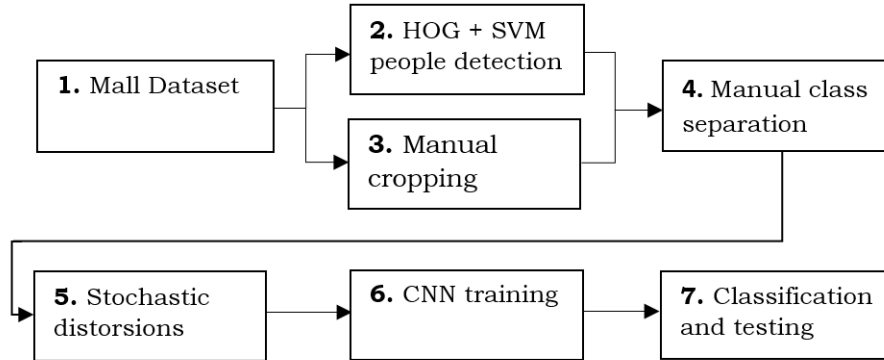


Figure 3: Block scheme of this work.

In this work we used the Mall Dataset [8] originally intended for crowd counting applications. The need of large training data is a common issue in deep learning research. The frames from the mall Dataset have a sufficiently large number of pedestrians circulating in a realistic environment. In order to

feed training data to the CNN, this data needs to be extracted and pre-labeled. As a first step to facilitate the data extraction an HOG features based and SVM decision method is used. The automatic method provides a large number of image patches of people from both genders. The method is applied to half of the frames available. On top of that the errors must be considered as well and need to be separated from authentic correct data. For this purpose a manual selection was performed over the so automatically extracted patches. Manual cropping has been used in the remaining frames. No specific attention has been paid to cropping all possible people on each frame. For the sake of this work a sufficiently large set of training images was needed and our focus was on the quality of the patches. This means that patches should be representative of their category (male or female gender) in all possible contexts (partial occlusions, reflections, overlapping of multiple pedestrians in dense crowds, different postures).

Stochastic distortions are frequently and successfully used in deep learning research in order to artificially enlarge the size of the dataset. Newly originated images will have a higher correlation but CNN training nonetheless benefits from a larger variety of images. The distortions introduced are the following: (1) scaling 0.9 - 1.1, (2) rotation $\pm 3^\circ$, shift $\pm 10\%$. For each patch three distorted versions of it are generated. In one case only we found that accuracy was higher if we trained on the manually extracted images only. For this reason as it will be reminded in the Results Section, we used this reduced training set for training and testing as in Test-2.2. Once the training set is ready we implemented a CNN similar to the one used in Ng *et al.* [2]. The layer structure of the CNN is visible in Figure 2. The RGB input images are preprocessed in a simple way. Firstly, they are resized to 80x40 and central cropping is applied to reduce them to the 64x32 central patch. Secondly, the RGB channels intensities are stretched to the maximum range using the *skimage.rescaleintensity* method from the scikit skimage module [11]. The weights of the filters producing the highest accuracy on the validation set were saved and used for the experiments.

4.1 SVM Pedestrian Detection

In this work we use the implementation from Matlab R2016b Computer Vision System Toolbox "PeopleDetector" [6] as an instrument to support manual extraction of the training set. This Matlab System Object uses Histogram of Oriented Gradient (HOG) features to detect upright, unoccluded people and a trained Support Vector Machine (SVM) classifier for the final decision. The implementation is based on [7] where the training data is taken from the MIT Pedestrian dataset where all people stand upright (front or back view). For this reason the detection works best for upright people and preferably from a front or back view. Since the training data was 128x64 images the frames from the Mall dataset have been resized to 960x720. This is in order to both preserve the

aspect ratio and to have the average size of the bounding box of people in the frames match the size of the training data (128x64), in order to ease the detection. In Figure 4 and 5 are some examples of correct detections, in Figure 6 are some errors. In some cases the lay-figures on the left are mistaken for people. In others reflections of people are falsely detected as people.

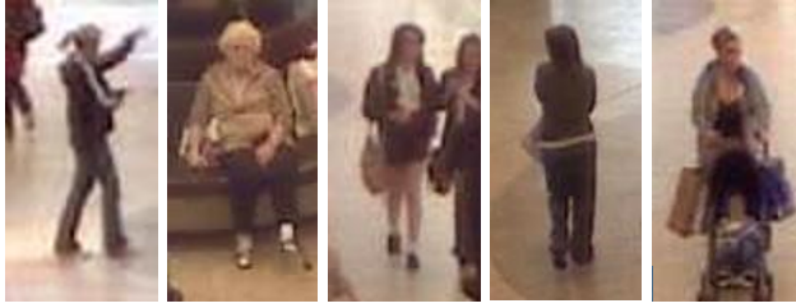


Figure 4: Detection samples from female gender class.



Figure 5: Detection samples from male gender class.

4.2 Convolutional Network Description

Our CNN was implemented using Python with Theano [9] and Lasagne [10] Libraries. The layer structure of our CNN is visible in Figure 2. Adaptations have been made to address our dataset. Layer C1 contains 100 5x5 filters and generates 100 feature maps. No zero-padding is used so the output spatial size of the output volume is reduced to 60x28 (from 64x32). The following maximum pooling layer subsamples to 30x14. Convolutional layer C3 uses 200 5x5 filters and outputs a 200x26x10 volume. The S4 pooling layer returns a 200x13x5 volume. Eventually a single fully-connected layer with softmax non-linearity is used as a classifier. The number of parameters in layers C1, C3 and F5 is respectively 2500, 5000 and 1300000. It is to be expected the large majority of weights to belong to the fully-connected layer. The nonlinearity in use is the



Figure 6: Examples of errors in the detection.

hyperbolic tangent. Although rectification layer units would be faster, computational time was not a strict constraint in this work. In some cases this CNN has been noticed to reduce the training loss during training by always classifying samples as female (the majority of samples). To solve this problem a second version of this CNN was used. This type of CNN uses an additional convolutional layer with 200 filters on top of the S4 layer and a dropout layer (dropout probability 0.5) was inserted between the fully connected and the output layer.

5 Experiments

5.1 The Dataset

The dataset we used is the Mall Dataset [8]. This dataset consists of 2000 640x480 frames taken with a 2 Hz frequency. The scene is filmed from a static RGB camera inside a mall. Each frame contains about 10 to 20 people on average. The pedestrians in these frames show all attributes of an "average" pedestrian population. Both men and women (although with a consistent majority of women) are present. Pedestrians are filmed from different possible angles, front, back and sides. A number of artifacts are seen in the reflections of the shop windows. Occlusions and overlapping of body parts in dense crowds are noticed as well. This dataset serves well the purpose of gender recognition from a non-invasive acquisition process as pedestrians are filmed from afar. Their faces are hardly detectable and do not provide sufficient information for our purpose as they are barely visible in most cases. In Figure 7 a frame from the dataset is shown.

5.2 Experimental Setup

For our tests we used an 8-core CPU at 2.40GHz Intel(R) Xeon(R) E5-2630, a GPU Tesla K40c, 64 GB of RAM. In this work our training set, assembled as described in the previous Section, consists of the following number of images:



Figure 7: Frame 350 from Mall Dataset.

Table 1: Training set.

	Errors	Female	Male
Number of samples	3066	5010	2295

In the experimental setup we made use of a testing set obtained in the following way. In order to have a sufficiently large testing set that would also be representative of all frames, one frame every ten was taken and HOG+SVM pedestrian detection was performed. All patches were saved and labeled by hand as male, female or as an error. The weights of the CNNs were obtained with a training as described from our method in the previous Section. In the training the weights were initialized with Glorot Initialization [12]. Batch training was used with 16-elements batches. For the updates, the gradient descent with Nesterov momentum was used. For the training and test loss estimation we used the categorical crossentropy loss function. The testing set is composed as follows:

Table 2: Testing set.

	Errors	Female	Male
Number of samples	507	1583	770

In this work a twofold testing method was used. First (test-1), a single CNN was used to evaluate the performances given three possible outcomes: sample is classified as an error, as a male or as a female. Second (test-2), a single CNN was used to separate people from non-people (e.g. detection errors), in test-2.1. Afterwards (test-2.2) the samples that were classified (correctly) as people are fed to another CNN that separates male and female pedestrians. In test-2.2 it is also worth noticing that (as mentioned in the previous Section) the training used a training set without distortions and the CNN was slightly varied (CNN description Subsection).

6 Results

The test set consists of a total of 2353 detections of pedestrians plus 507 erroneous detections (2860 total). The total number of possible correct detections was manually computed and adds up to 3177 across all 200 considered frames. This results in a True Positive Rate of pedestrians equal to 74.06% (all correct pedestrian detections out of all possible pedestrians in the 200 frames). To obtain this value it must be specified that the count of people on each frame follows the following principles: (1) at least *half* of the person’s body must be clearly visible and (2) people *far away* from the camera (close to the upper edge of the camera’s field of view) whose gender is ambiguous even to the human observer have been left out of the count. Hence the estimates for the detection accuracy should be valued as limited by these considerations.

Table 3: Confusion test-1.

		Decision		
		Errors	Female Pedestrian	Male Pedestrian
Correct label	Errors	444	52	11
	Female	212	1084	287
	Male	87	198	485

Table 4: Confusion test-1, values in %.

		Decision		
		Errors	Female Pedestrian	Male Pedestrian
Correct label	Errors	87.57	10.26	2.17
	Female	13.39	68.48	18.13
	Male	11.30	25.71	62.98

Table 5: Confusion test-2.1.

		Decision	
		Errors	Pedestrians
Correct label	Errors	462	45
	Pedestrians	325	2028

The results from tests-2.2 as mentioned in the previous section have been generated from weights that were trained on manually extracted pedestrian patches only.

Test-2 (Table 5 and 6) first separates pedestrians from non-pedestrians (detection errors). On a second stage this test inputs to another CNN only the previously classified pedestrians for the gender test. Table 7 and 8 summarize the

Table 6: Confusion test-2.1, values in %.

		Decision	
		Errors	Pedestrians
Correct label	Errors	91.12	8.87
	Pedestrians	13.82	86.18

Table 7: Confusion test-2.2.

		Decision	
		Female Pedestrian	Male Pedestrian
Correct label	Female Pedestrian	1008	341
	Male Pedestrian	318	361

Table 8: Confusion test-2.2, values in %.

		Decision	
		Female Pedestrian	Male Pedestrian
Correct label	Female Pedestrian	74.72	25.28
	Male Pedestrian	46.83	53.17

gender classification in test-2. Test-2 classifies correctly 86.18% of pedestrians as such. It obtains a gender classification accuracy (average of true positive rate of male and female classes) of 63.95%. Table 9 shows the final performances in pedestrian detection rate, gender classification and the estimate for correct gender recognition among all *detectable* people. We notice that for gender classification the setup used in test-1 (Table 3 and 4) yields better performances. The average true positive rate between Male Pedestrians and Female Pedestrians classes is 65.75%. The CNN correctly classifies as non-people the 87.57% of the erroneous detections. Given an estimated detection rate of pedestrians of 74.06%, for test-1 we obtain an estimate of correct gender recognition of 48.69% of *detectable* people (the estimate is still subjected to the specifications described at the beginning of this Section).



Figure 8: Test-1, examples of errors: female pedestrians classified as males.

Table 9: Test comparisons in %.

	Test-1	Test-2
Pedestrian detection	74.06	
Gender classification	65.75	63.85
Correct gender pedestrians	48.69	40.98



Figure 9: Test-1, examples of errors: male pedestrians classified as females.

7 Conclusions

In this work we performed pedestrian detection and non-intrusive gender recognition on the Mall Dataset [8]. Our method uses fully supervised training with a simple 4-layer CNN. The question that this work wants to answer is whether the image of a person’s body alone provides enough information to *contribute* to a soft-biometrics classification systems for the gender. The results show that the person’s body image can in fact provide useful information and our classifier is not randomly guessing. This information alone is though not sufficient by itself to reach top accuracies. We reached 65.75% on the gender recognition part taking into account all possible point of views of the camera with respect to the subject. In an open public environment roughly one person out of three would still be misclassified. In particular many errors seem to be connected to the lower resolution of some test samples (Figure 8 and 9). This work is a hint to explore the possibility of combining heterogeneous input information including body’s image as seen by a surveillance camera to generate a non-invasive ensemble gender classifier.

References

- [1] Denis Tomé, Federico Monti, Luca Baroffio, Luca Bondi, Marco Tagliasacchi, Stefano Tubaro, *Deep convolutional neural networks for pedestrian detection*, arXiv:1510.03608. October 2015.

- [2] Choon-Boon Ng, Yong-Haur Tay, Bok-Min Goi, *A Convolutional Neural Network for Pedestrian Gender Recognition*, in: Lecture Notes in Computer Science pp 558-564. July 2013.
- [3] Piotr Dollar, Christian Wojek, Bernt Schiele, *Pedestrian Detection: An Evaluation of the State of the Art*, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 743 - 761. April 2012.
- [4] Vladimir Khryashchev, Andrey Priorov, Lev Shmaglit, Maxim Golubev, *Gender Recognition via Face Area Analysis*, in: Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I. October 2012.
- [5] Christopher D. Geelen, Rob G. J. Wijnhoven, Gijs Dubbelman and Peter H. N. de With, *Gender Classification in Low-Resolution Surveillance Video: In-Depth Comparison of Random Forests and SVMs*, in: Video Surveillance and Transportation Imaging Applications 2015. May 2015.
- [6] it.mathworks.com/help/vision/ref/vision_peopledetector-class.html
- [7] Navneet Dalal, Bill Triggs, *Histograms of Oriented Gradients for Human Detection*, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). June 2005.
- [8] http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html
- [9] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, Yoshua Bengiol, *Theano: A CPU and GPU Math Compiler in Python*, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010.
- [10] lasagne.readthedocs.io/en/latest/
- [11] <http://scikit-image.org/docs/>
- [12] Xavier Glorot, Yoshua Bengio, *Understanding the difficulty of training deep feedforward neural networks/*.