Shallow learning methods for pedestrian detection

Master's degree in Computer Engineering for Robotics and Smart Industry
Machine Learning and Artificial Intelligence course, a.a. 2021/2022
Lorenzo Busellato

Abstract

This work presents a pipeline for shallow-learning based pedestrian detection. The Histogram of Oriented Gradients (HOG) method for feature extraction is described and implemented. Two classification techniques, Support Vector Machines (SVMs) and k-Means clustering, are described and benchmarked with different sets of hyperparameters. The benchmarks are then compared against each other to determine the best performing method in terms of execution time and accuracy in detection. Finally the best performing method is showcased by applying it to a video sequence.

Contents

List of Figures

List of Tables

Shallow learning methods for pedestrian detection

I. Introduction

Pedestrian detection, i.e. the task of detecting and localizing pedestrians within an image, is one of the most tackled instances of the broader class of object detection problems. This is due to its natural applications in autonomous driving, surveillance and robotics. For instance, in the US alone about 5000 of the 35000 annual car accident fatalities involve pedestrians [1], therefore the need for automated methods for pedestrian detection is apparent.

The task of pedestrian detection presents a series of complexities regarding the high variability in scale and size of the person within the image, occlusions with the rest of the scene, susceptibility to lighting conditions and the real time detection requirement.

The main models that are used for pedestrian detection are hand-crafted models and deep learning models. Hand-crafted models are based on classification of hand-crafted features [2] [3] [4]. Deep learning models on the other hand typically use convolutional neural networks for the extraction of features, therefore producing high-level semantic representations of objects [5] [6] [7].

In this work the first class of models will be analyzed and implemented. Specifically the Histogram of Oriented Gradients (HOG) method will be used to extract features from images. The classification of images will then be implemented using non-deep learning methods, namely Support Vector Machines (SVMs) and k-Means clustering.

The main objectives of this work are described in section II. The dataset and methodology employed are described in sections III and IV. The experimental evaluation is presented in section V. The conclusions are summarized in section VI.

II. Objectives

The main objective of this work is the implementation of a pedestrian detection pipeline, that is essentially made up of two main components: a feature extractor and a classifier.

The feature extractor chosen for this task is the HOG, given its improved performance with respect to other common feature extractors employed in image processing [8]. HOG implementations are available in popular machine learning libraries, such as *scikit-image* [9] and *opencv* [10], but in this work it will be implemented from scratch.

Regarding the classifier, two models will be implemented: SVMs and k-Means clustering. Both models will be applied with different combinations of hyperparameters in order to obtain a benchmark that will allow for their comparison.

Finally the best performing classifier will be showcased against a video sequence in order to demonstrate its capabilities.

III. Dataset

The dataset used in this work is the Daimler Pedestrian Detection Benchmark Dataset [11]. The set contains 15560 48x96 grayscale image samples of pedestrians, to be used as positive samples, and 6744 640x480 grayscale images not containing pedestrians, to be used for the extraction of negative samples. From each image in the non-pedestrian dataset, 4 randomly picked 64x128 image patches are extracted, for a total of 26976 negative samples. Of both subsets 80% of samples will be kept as training data, while the rest will be used as testing data to benchmark the classifiers.

IV. Methodology

A. Feature extraction

For a given input image or image patch I, the first step is the conversion to grayscale and the resizing to a width of 64 pixels and an height of 128 pixels.

Then for each pixel $(i, j) \in I$ the gradients G_x and G_y in the x and y directions are computed. This is done with a convolution operation:

$$G_x = \omega_x * I \quad G_y = \omega_y * I \tag{1}$$

where ω_x and ω_y are the kernels associated to the convolution operation. [8] has shown that the best performing kernel in terms of detected false positives per moving window (FFPW) is the monodimensional:

$$\omega_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad \omega_y = \omega_x^T \tag{2}$$

After the convolution operation, two 128×64 gradient matrices G_x and G_y are obtained. For each pixel (i, j) in the image, the magnitude μ and angle θ are then computed:

$$\mu(i, j) = \sqrt{G_x(i, j)^2 + G_y(i, j)^2} \tag{3}$$

$$\theta(i, j) = \left| \tan^{-1} \left(\frac{G_y(i, j)}{G_x(i, j)} \right) \right| \tag{4}$$

The resulting gradient magnitude and gradient angle matrices are then divided into 8×8 patches, for each of which a 9-point histogram of oriented gradients is computed. A histogram is a series of bins labeled with angle values. The number of bins is determined by the angle step between the bins, for instance with an angle step of 20 degrees the histogram will have $180^\circ/20^\circ = 9$ bins.

For each pixel in the patch the corresponding gradient angle determines two bins on the histogram that the pixel votes for. The two bins selected are those which represent the two closest angles to the given gradient angle value. The value added to the selected bins is proportional to the distance of the gradient angle to the bin angle, the gradient magnitude and the step between the bins:

$$v_{closest} = \frac{(binAngle - \theta)\mu}{angleStep} \quad (5)$$

$$v_{\textit{2nd-closest}} = \frac{(\theta - binAngle)\mu}{angleStep} \quad (6)$$

Repeating the process for each 8×8 patch, a 16×8 matrix of histograms is constructed.

To reduce the variability in gradient intensities due to lighting conditions and foreground-background contrast, a block normalization procedure is introduced. The 16×8 histogram matrix is scanned with a 2×2 window producing $15 \cdot 7$ blocks, for each of which a normalized histogram vector is computed by concatenating the four histograms contained in the block and normalizing the resulting vector.

Finally, the feature descriptor for the image is constructed by concatenating the normalized histogram vectors. From each 8×8 patch a 1×9 vector is collected, which is then concatenated with another three 1×9 vectors composing a 1×36 vector, which is finally concatenated with another fourteen 1×36 vectors, resulting in a 1×3780 feature vector.

B. Classification

C. Hard-negative mining

D. Non-maximum suppression

Non-maximum suppression (NMS) is a technique for the filtering of predictions made by an object detector. An object detector typically produces a list B of proposed bounding boxes along with the confidence score S assigned to each of them. The algorithm is:

1) Select from B the proposal with the highest confidence score, remove it from B, and add it to an initially empty list D.
2) Compare the proposal with all other proposals in B by computing the Intersection Over Union (IOU):
$$IOU = \frac{\text{Intersection Area}}{\text{Total Area}}$$
3) Remove from B all the proposals with an IOU greater than some threshold.
4) Repeat steps 1-3 until B is empty.

V. Results

A. Adopted metrics

B. Results

VI. Conclusions

References

[1] U. Shankar, "Pedestrian roadway fatalities," Tech. Rep., 2003.

[2] 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, 2014.

[3] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[4] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2011.

[5] Y. Pang, J. Cao, and X. Li, "Learning sampling distributions for efficient object detection," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 117–129, 2016.

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] 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)*, vol. 1, 2005, pp. 886–893 vol. 1.

[9] "Histogram of oriented gradients." [Online]. Available: https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html

[10] "Cv::hogdescriptor struct reference." [Online]. Available: https://docs.opencv.org/3.4/d5/d33/structcv_1_1HOGDescriptor.html

[11] 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.