Computer Vision and Image Understanding 120 (2014) 46–58

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Macrofeature layout selection for pedestrian localization and its acceleration using GPU ☆

Woonhyun Nam *, Bohyung Han, Joon Hee Han

Department of Computer Science and Engineering, POSTECH, Republic of Korea

ARTICLE INFO

Article history:
Received 13 July 2012
Accepted 26 October 2013
Available online 7 November 2013

Keywords:
Macrofeature selection
Object localization
Pedestrian detection

ABSTRACT

Macrofeatures are mid-level features that jointly encode a set of low-level features in a neighborhood. We propose a macrofeature layout selection technique to improve localization performance in an object detection task. Our method employs line, triangle, and pyramid layouts, which are composed of several local blocks represented by the Histograms of Oriented Gradients (HOGs) features in a multi-scale feature pyramid. Such macrofeature layouts are integrated into a boosting framework for object detection, where the best layout is selected to build a weak classifier in a greedy manner at each iteration. The proposed algorithm is applied to pedestrian detection and implemented using GPU. Our pedestrian detection algorithm performs better in terms of detection and localization accuracy with great efficiency when compared to several state-of-the-art techniques in public datasets.

1. Introduction

Object detection involves the localization (*where*) as well as the identification (*what*) of predefined objects such as face, pedestrian, vehicle and so on. Although various object detection algorithms have been proposed so far, the research in object detection has been focused mainly on the identification with loose requirement of localization; the bounding boxes of detected objects are sometimes poorly aligned. However, the localization of an object is also a very important issue in object detection since it may affect the performance of subsequent procedures significantly. For example, multiperson tracking-by-detection approaches [1,2] rely on the outputs from the object detector and tracking problem is formulated as a data association task across adjacent frames [2], where the accuracy of object localization affects tracking performance directly. Since localization quality in object detection is critical for many computer vision applications, it is worthwhile to investigate the object localization problem for the development of a robust object detector. Fig. 1 illustrates some examples of true positives with variations of alignment accuracy in pedestrian detection. In this paper, we propose a macrofeature layout selection algorithm in a boosting framework to improve localization performance, and apply our technique to pedestrian detection for validation.

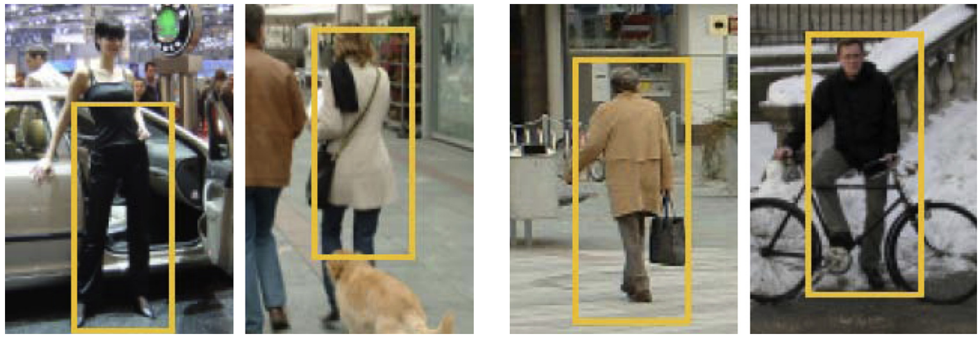
Object detection has been widely studied in computer vision to identify various objects such as faces [4,5], pedestrians [4,6–18], and others [3,6,11,19]. Recently, pedestrian detection has received much attention and many algorithms have achieved successful results. Viola et al. [13] proposed an efficient pedestrian detection framework using a boosted cascade with simple and efficient Haar-like features. As a classifier, AdaBoost [20] was employed to select a number of discriminative features and their corresponding weak classifiers among a huge number of candidates. Dalal and Triggs [7] proposed a good feature for pedestrian detection, which is called the Histogram of Oriented Gradients (HOG), and published the INRIA dataset for pedestrian detection. The HOG feature is combined with other types of features successfully in object detection [8,10,14,15,18]. Walk et al. [14] employed a new feature based on self-similarity of low-level features and combined the new feature with several different features. Dollár et al. [10] introduced the Caltech Pedestrians dataset and benchmarked existing detection algorithms in images—not in windows—with several performance metrics [10,21]. Felzenszwalb et al. [19] proposed an algorithm based on deformable part models, which presents the state-of-the-art detection performance in broad object categories.

Finding good features is crucial in various computer vision problems. The combination of low-level features, which is also called mid-level features, has been widely studied in object detection and recognition community. Boureau et al. [22] presented a supervised learning method for mid-level feature extraction by sparse coding and tested it on object recognition benchmark datasets. Laptev [11] introduced boosted histograms that combine local histogram features by boosting and demonstrated competitive performance on the PASCAL VOC 2005 dataset [23]. The feature mining strategy was discussed in [4], where a pool of informative and complementary features is obtained from the huge feature

☆ This paper has been recommended for acceptance by Tinne Tuytelaars.

* Corresponding author.
E-mail addresses: xgene@postech.ac.kr (W. Nam), bhhan@postech.ac.kr (B. Han), joonhan@postech.ac.kr (J.H. Han).

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.10.011



(a) Localizations with overlap ratio ~ 0.5 (b) Localizations with overlap ratio ~ 0.7

Fig. 1. Localization examples of true positives in pedestrian detection for INRIA pedestrian dataset. Localization accuracy is measured by PASCAL VOC [3] overlap ratio. In general, detections are considered to be correct if the overlap ratio is more than 0.5.

space and the optimal feature set is selected by AdaBoost. They introduced generalized Haar-like features that are similar to the original ones [5] but allow arbitrary configurations and numbers of rectangles. Random configurations of heterogeneous low-level features were integrated into integral channel features [9], which improve pedestrian detection performance. Multi-scale generalizations of low-level features was introduced in [6], where the multi-scale features outperformed the best single-scale features in object detection.

We propose a macrofeature layout selection to improve object localization. Our macrofeature layout selection employs the feature layouts representing lines, triangles, and pyramids. The selected layouts are composed of several low-level feature blocks closely located to each other in a multi-scale feature pyramid. According to our observation, features with local high-order information such as curves and surfaces are more discriminative than features with the zero-order information such as points. As a feature selection strategy by boosting, our macrofeature layout selection is closely related to the integral channel features [9] and the boosted histograms [11]. We combine low-level features by boosting similar to [9,11] and our weak learner is same with [11]. However, our method extracts low-level features from neighborhood blocks rather than random ones without spatial constraints [9]. Also, it selects discriminative block layouts corresponding to geometric primitives such as lines, triangles, and pyramids instead of spatial grid blocks [11]. The pedestrian detection algorithm based on the selected layouts improves detection and localization performance in our experiments, particularly on Caltech and Daimler datasets which contain low resolution pedestrians.

The preliminary version of our work appeared in [24], and this paper contains the following additions and updates. First, we change the evaluation protocol of localization performance. Miss rates with respect to PASCAL overlap criteria were reported previously, but we now present average overlap ratios between detections and ground-truths given false positives per image to focus on localization performance. Second, we test performance of several types of layouts to verify the effectiveness of the proposed feature layouts and report the advantage of our layouts in Section 5.2. Third, our detection algorithm based on the selected local layouts is implemented using GPU and its computational efficiency is presented.

The rest of this paper is organized as follows. We describe our boosting framework for object detection in Section 2. Section 3 defines macrofeature layouts and present a macrofeature layout selection method by boosting. We present pedestrian detection algorithm based on the macrofeature layouts and its parallelization in GPU in Section 4. Section 5 illustrates the performance of our macrofeature layout selection strategy and demonstrates the pedestrian detection results in several challenging datasets.

2. Boosting for object detection

An object detector is a binary classifier that estimates a class label $y \in \{1, -1\}$—whether the target object exists or not—given an observed feature vector $\mathbf{x}$. The classifier is learned from training data, which comprise a set of feature vectors $\mathbf{x}_i$ and their corresponding labels y_i, i.e., $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$, where N is the number of training examples.

The discrete AdaBoost algorithm [20] constructs a strong classifier $f(\cdot)$ as a weighted sum of weak classifiers $h_t(\cdot)$, which is given by

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}), \tag{1}$$

where α_t is the weight for the weak classifier $h_t(\mathbf{x})$ and T denotes the total number of weak classifiers. The weak classifiers are added in a greedy manner and the procedure at each iteration focuses on the data samples that have still been misclassified.

To build weak classifiers and select discriminative features simultaneously, we extract feature subvectors using a binary feature selection matrix, $\mathbf{\Phi}_{B^{(j)}}$, whose non-zero elements are parametrized by a feature layout $B^{(j)} \in \{B^{(j)}\}_{j=1,\cdots,M}$. In other words, each feature subvector is extracted by the product of feature selection matrix $\mathbf{\Phi}_{B^{(j)}}$ and feature vector $\mathbf{x}$, $\mathbf{\Phi}_{B^{(j)}}\mathbf{x}$, and a weak classifier is learned to minimize the weighted training error defined by

$$\mathrm{err}[h^{(j)}] \triangleq \sum_i w_i \delta\left(y_i \neq h^{(j)}(\mathbf{\Phi}_{B^{(j)}}\mathbf{x}_i)\right), \tag{2}$$

where $\mathbf{w} = (w_1, \ldots, w_N)^\top$ is a normalized weight vector of samples, and $\delta(\cdot)$ is the Kronecker delta function. The weak classifier with the minimum error and its corresponding feature subvector are selected in each iteration and the weight of the selected weak classifier is given by

$$\alpha_t = \ln \frac{1 - \mathrm{err}[h_t]}{\mathrm{err}[h_t]}. \tag{3}$$

After a weak classifier is selected, the weights of data samples are updated as

$$w_i \leftarrow \begin{cases} w_i & \text{if } y_i = h_t(\mathbf{\Phi}_{B_t}\mathbf{x}_i) \\ w_i \exp(\alpha_t) & \text{if } y_i \neq h_t(\mathbf{\Phi}_{B_t}\mathbf{x}_i) \end{cases}, \tag{4}$$

and the weights of the samples are renormalized so that $\sum_i w_i = 1$.

As a weak learner, we employ the Weighted Fisher Linear Discriminant (WFLD) [11], which is a variant of the Fisher Linear