LexisNexis® Academic

Print Request: Current Document: 7
Time Of Request: Friday, February 09, 2018 02:13:22 EST
Send To:

MEGADEAL, ACADEMIC UNIVERSE
MASTER'S COLLEGE
LIBRARY
SAN JOSE, CA 00000

Terms: (autonomous vehicle)

Source: IET Computer Vision
Project ID:

7 of 31 DOCUMENTS

IET Computer Vision

August 2017

Algorithmic optimisation of histogram intersection kernel support vector machine-based pedestrian detection using low complexity features

BYLINE: MuhammadBilalmeftekar@kau.edu.sa

SECTION: Pg. 350 - 357 Vol. 11 No. 5 1751-9632

LENGTH: 5267 words

1

Introduction

Pedestrian detection in images and video frames has garnered considerable attention from computer vision experts in the recent years due to growing interest in the development of smart surveillance systems and **autonomous vehicles.** Since Dalal and Triggs [1] introduced their landmark detector based on histogram of oriented gradients (*HOG*) and linear support vector machine (*SVM*), various researchers have put forward increasingly complex features and classification frameworks to advance the field considerably. It has also been emphasised by notable recent works that no single feature is a better discriminant than *HOG*. Moreover, attempts to reduce the computational complexity of *HOG* inevitably lead to reduction in detection performance. Thus, all the later efforts have necessarily augmented the original *HOG* detector with other features such as colour information [2] and local binary pattern (*LBP*) and so on [3] to increase the sophistication and hence detection accuracy of the overall detector. On the other hand, histogram intersection kernel (*HIK*) *SVM* classifier has been shown to work better than its linear counterpart by many researchers [4, 5]. However, *HIK-SVM*-based classifier is computationally much more complex and requires mathematical approximations to be employed for real-time operation. In a previous work [6], it was shown that the combination of histogram of significant gradient (*HSG*), an integer-valued feature inspired by *HOG*, and *HIK-SVM* is not only computationally much less complex than *HOG-linear SVM* but also gives better detection results on standard datasets. The integer-valued features allow implementation of *HIK* SVM using fast look-up table (*LUT*) without any approximation. The proposed *HSG-HIK* algorithm provides hardware designers with an alternative, first of its kind, framework that is simultaneously much less complex and more accurate than original *HOG-linear SVM*. In this paper, several enhancements to the earlier proposed *HSG-HIK* framework have been described. The feature calculation in the enhanced framework includes pre-processing operations to replace the use of averaging, which was used as a means to determine gradient significance in the previous work. Colour information has also been included in the feature vector. Moreover, the training methodology has been revised to run multiple bootstrapping cycles and use only the most significant negative examples. The enhanced framework, named *HSG-HIK-plus*, yields significantly better results than the original *HSG-HIK* detector as can be noticed from the miss rate (*MR*) versus false positives per image (*FPPI*) curves depicted in Fig. 1.

Fig. 1 Detection results of the proposed detector (HSG-HIK-plus) on standard pedestrian datasets

In fact, the newly proposed framework performs better than the boosting cascades-based aggregate channel feature (*ACF*) [2] and (in some cases) spatially pooled features (SPF) [7] detectors as well. The proposed framework does not incorporate complicated floating point operations and yet achieves detection accuracy at par with some of the best-known detectors reported in the literature.

The remaining of this paper is organised as follows. Section 2 gives an overview of the latest pedestrian detection algorithms as reported in the literature. Section 3 describes the proposed framework and is followed by discussion and analysis of its detection results on standard datasets in Section 4. Section 5 concludes the discussion.

2

Literature review

Pedestrian detection has been an active area of research in computer vision for the past decade and is still considered a challenging problem. As noted in surveys by Dollar *et al.* [8] and more recently by Benenson *et al.* [9, 10], although a variety of detectors employing different sets of features and classification methods have been proposed, the problem of pedestrian detection still has a large room for improvement both in terms of detection accuracy and speed.

HOG feature with linear SVM classifier was popularised when Dalal and Triggs [1] demonstrated its superior performance on the *INRIA* pedestrian dataset, also introduced by them. As mentioned earlier, since then many researchers have enhanced *HOG* by adding more features such as colour information [2], *LBP* [3] and second-order statistical measures [7, 11] and so on to increase the performance beyond that of the original detector. This has necessarily increased the computational complexity of detectors since *HOG* alone requires complex floating point operations. Especially its bilinear interpolation method for histogram population and subsequent normalisation take a heavy toll on the processor. Dedicated full hardware implementations of *HOG* also require complex data flow controllers and arithmetic units [12, 13]. A low complexity variant of the original *HOG*, *HSG-HIK* detector, was proposed in [6] and yields better performance despite being simpler to compute. *HSG-HIK*, the predecessor of the work being presented in this paper, uses *LUT*-based *HIK-SVM* in place of *linear SVM* to increase its discrimination power and renders the original HOG-linear SVM detector, still being employed in dedicated hardware platforms [12, 13], redundant.

HIK-SVM was first used by Maji *et al.* [14] to classify their own low complexity *HOG* like features for pedestrian detection and claimed to obtain significantly better results than original *HOG*. They also employed an approximation technique to reduce the computational complexity of *HIK-SVM*. However, their training procedure and results were later found to be incorrect by Dollar *et al.* [8]. Later Wu and Rehg [15] showed that calculation of HIK-SVM can be sped up significantly with a *LUT* if the input feature vector consists only of integers with limited dynamic range. The major contribution of the previous work [6] is proposing an integer-only feature vector which is discriminative enough to give better results than original *HOG* using *LUT*-based *HIK-SVM* implementation.

In a bid to improve over original *HOG* detector, Dollar *et al*. proposed ACF; a boosting cascade-based detector, augmenting orientation histograms with colour information as discriminating features and Adaboost as classifier [2]. To improve detection speed, ACF approximates features between various scales to avoid redundant computations and depends heavily on *Intel SSE2* instruction set to achieve real-time performance. This detector is considered to be the next major milestone in the history of pedestrian detectors after original *HOG*. Dollar *et al.* also introduced the *Caltech* pedestrian dataset which is very different from *INRIA* in frame resolution, hue and overall picture quality. Detectors trained on one of these datasets generally perform sub-optimally on the other [8, 10].

A notable improvement over *ACF* detector is *SPF* detector recently proposed by Paisitkriangkrai *et al.* [7]. This detector enhances the set of features employed by ACF to include second-order spatial gradients, full 360° orientations of calculated gradients and a variant of *LBP*. This ensemble of features is collected via a covariance matrix to capture a more detailed relationship between various features in a local patch. The robustness of this highly discriminative, albeit computationally expensive, feature is further enhanced through max-pooling, a variant of spatial pooling. Furthermore, they have modified their *SVM* classifier to optimise for a partial region of interest. This highly complex feature-classification framework inevitably requires a region proposal generator as a preliminary detection stage to circumvent the exorbitant computational cost.

Benenson *et al.* [16, 17] have provided a detailed analysis of the individual features employed by various popular detectors (*ACF* and *HOG* etc.) and their impact on the final detection scores. They also list some ways for the practitioners which yield better results using the same set of features as employed by *ACF* and *HOG*. Using *ACF* features as baseline, they have also proposed combining multiple detectors to detect pedestrians of different sizes without rescaling [18]. They also demonstrated that detection speeds up to 100 frames per seconds are possible through engagement of *graphical processing units (GPUs)* using ACF like features.

Bag-of-visual-words feature [19, 20] is another popular approach used for scene classification and object recognition. This approach uses image key points for sparse representation in place of dense pixel based features such as gradient histograms, colour and LBP and so on. Jiang *et al.* [20] propose using *scale invariant feature transform* [21] as features to describe regions around key points and report promising results on *PASCAL* [22] and *TRECVID* [23] datasets using *radial basis function* (RBF) and HIK-SVM. According to their findings, RBF kernels perform much better than their linear counterpart albeit at the cost of more computational time.

Recently, many general object and pedestrian detectors using deep learning approach [24, 25] have been proposed and shown to deliver unprecedented detection results. However, their utility in comparison with hybrid features-based detectors has been questioned by other researchers [26] since these detectors are severely limited by their processing speeds, requiring several seconds per frame and are hence impractical for real-time systems.

Another notable pedestrian detection approach is using deformable parts-based model [27] which is particularly useful in scenarios where humans are either partially occluded or do not maintain upright pose. Yan *et al.* [28] have proposed combining this approach

with image resolution information to gain advantage in detection accuracy. This approach, however, takes heavy toll on processor and is not favourable for real-time systems.

Pedestrian detectors are often combined with trackers for surveillance applications. Recently, Makasai *et al.* [29] and Wang *et al.* [30] have proposed multi-people trackers which can identify complex trajectories as well as motion in groups.

It can be noted from this review that there exists a general trend of adding computationally expensive features to enhance the detection accuracy of existing pedestrian detectors. This has in turn led to increased dependence on hardware (*SSE* and *GPU* support) to achieve real-time performance. While these research endeavours to approach the problem *vertically* are commendable, there exists a need to find solutions *horizontally* as well to explore alternative techniques which might lead to simpler solutions. *HIK-SVM*, for instance, has been largely dismissed by the researchers owing to some initial setbacks [8, 14] while it was recently proved to show promise in combination with low complexity *HSG-HIK* [6] and *Bag-of-visual-words* features [20]. This paper proposes an enhanced detector based on HIK-SVM and demonstrates that this approach can match the performance of the advanced contemporary detectors even with low complexity features and pre-processing methods.

3

Proposed pedestrian detection framework

HSGs and its corresponding LUT-based *HIK-SVM* classification scheme have been described in detail in [6]. For the sake of completeness, only a brief description of the original framework and the enhancements made to it are presented in this section. The following subsections describe the individual operations in the required order, i.e. feature extraction followed by classification and subsequently non-maxima suppression (NMS).

3.1

Pre-processing and feature vector generation

The original HSG-HIK framework did not include any pre-processing steps. However, subsequent experimentation revealed that both low-pass filtering and gamma correction of the input image lead to improved detection results. Thus, the enhanced framework *HSG-HIK-plus* pre-processes the input frame with a 3 × 3 box filter to reduce the fine granular details irrelevant for detecting pedestrian contours. This intentional blurring is followed by gamma correction of all three 8-bit colour channels according to the following equation: (1)

$$\boldsymbol{I}_{\text{out}} = 255 \times \left(\frac{\boldsymbol{I}_{\text{in}}}{255}\right)^{0.5}$$

where I_{in} is the input colour channel intensity vector and I_{out} is the corresponding output. This operation stretches the contrast of the input image and makes edges in the darker regions of the image more prominent. The exponent value, 0.5, was initially chosen based on recommendation in [1] and was found to give the optimal results on test datasets after experimentation.

Gradient magnitude squared and the corresponding orientation bin for every pixel in the blurred, gamma corrected image is then calculated according to the following set of equations: (2)

$$G^2_{\text{Mag}} = \partial x^2 + \partial y^2$$

(3)

$$G_{\text{bin}} = \left\lfloor \frac{\tan^{-1}(\partial y/\partial x) \times k}{\pi} \right\rfloor$$

where

∂x

and

∂y

are the horizontal and vertical gradients obtained through centred filter masks [1 0 -1] and [1 0 -1]T, respectively.

G^2_{Mag}

is the gradient magnitude squared and

G_{bin}

is the bin number in a *k-bin* orientation histogram. We use $k = 9$ to be consistent with the default HOG detector [1]. Also, the greatest gradient magnitude squared among the three colour channels is considered.

For orientation histogram population, we divide the detection window of size *W* (*width*) × *H* (*height*) into overlapping blocks of size 6 × 6 with a spatial stride of three in each direction. Thus, each pixel contributes to a total of four overlapping histograms, two in each direction. This contribution is based on the following criteria: (4)

$$\begin{bmatrix} H[G_{\text{bin}}] = & H[G_{\text{bin}}] + 2 \\ H[G_{\text{bin}} - 1] = & H[G_{\text{bin}} - 1] + 1 \\ H[G_{\text{bin}} + 1] = & H[G_{\text{bin}} + 1] + 1 \end{bmatrix} \quad \text{if} \left(\frac{G_{\text{Mag}}^2}{Q^2} > 0 \right)$$

where '*H*' is the histogram corresponding to each of the overlapping blocks containing the pixel and '*Q*' is a quantisation factor, typically 16. Thus, a pixel casts votes to the orientation bins only if its gradient magnitude is larger than '*Q*', otherwise not. In the original HSG-HIK framework, the average value of gradient magnitudes in a block was used in place of '*Q*' to decide the *significance* of the current pixel's gradient and hence its eligibility to vote. Experimentation has revealed that this simpler technique in combination with the pre-processing steps described earlier leads to better detection accuracy since it captures the shape contours more efficiently by identifying the more *significant* edges. Moreover, the orientation histogram bin, G_{bin}, given by (3) receives two votes while the two adjacent ones receive one vote each as depicted in Fig. 2. So, for an orientation of 56°, according to (3) and (4), $H[G_{\text{bin}}] = H[2]$ gets two votes while $H[G_{\text{bin}} - 1] = H[1]$ and $H[G_{\text{bin}} + 1] = H[3]$ each gets one vote. This scheme is a low-cost alternative to bilinear interpolation employed by original HOG and reduces the impact of aliasing due to the finite number of bins. Experimental data confirms the efficacy of this approach.

Fig. 2 Orientation histogram voting example for an orientation of 56°. Each dot represents a single vote

For each block, an average value of the '*U*' component of the colour information in the *LUV* colour space is also collected as a feature point. The bin values of orientation histograms and the averaged '*U*' colour component information from the overlapping blocks in the whole detection window are concatenated to form the final feature vector without any further processing. Fig. 3 depicts the whole feature extraction process described above. In summary, the HSG-HIK-plus feature augments the earlier proposed HSG-HIK feature by pre-processing the input image, adding colour information and reducing the aliasing effects of a finite number of histogram bins. In addition, it simplifies the operation by removing the averaging of gradient magnitudes in a block which was earlier used to determine the significance of individual gradients. The augmented feature is much more discriminative than its predecessor on standard datasets as shown by detection results in Section 4. Despite being more powerful, the new feature is still computationally much simpler than original HOG since it does not require complex operations such as bilinear interpolation and normalisation for histogram population. Moreover, the final descriptor retains its predecessor's *integer-only* property since the final integer count in all the respective histogram bins does not undergo any further processing. This aspect is the key to efficient LUT-based implementation of HIK-SVM.

Fig. 3 HSG-HIK-plus feature calculation flow

3.2

LUT-based HIK-SVM classification

The SVM classification of an input feature vector, '*x*', consisting of '*n*' elements can be represented by a function '*h*' with the following formulation: (5)

$$h(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \alpha_j K(x(i),\ SV_j(i)) \right) + b$$

where SV_j is the '*j*th' support vector out of a total of '*m*', '*i*' is the index to access individual elements of input feature vector and support vectors, '*[#x3b1]*$_j$' is the learned coefficient for each corresponding support vector, '*b*' is the learned bias and '*K*' represents the *kernel* function [15]. The linear kernel function is simply multiplication and simplifies to (6)

$$h_{\text{linear}}(x) = \sum_{i=1}^{n} \boldsymbol{x}(\boldsymbol{i}) * \left(\sum_{j=1}^{m} \alpha_j * \boldsymbol{SV_j(i)} \right) + b$$

In (6), '$x(i)$' can be brought out of the inner summation due to linearity. The term inside the bracket can be computed beforehand since it only involves the learned coefficients and identified support vectors. This leads to (7)

$$h_{\text{linear}}(x) = \sum_{i=1}^{n} \boldsymbol{x}(\boldsymbol{i}) * \boldsymbol{C(i)} + b$$

Thus, for the linear case, SVM classification is the dot product between the pre-computed coefficient vector, '$C(i)$', and input feature vector, '$x(i)$', added to the learned bias, 'b'.

The HIK-SVM, on the other hand, has the following form: (8)

$$h_{\text{HIK}}(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \alpha_j \min(\boldsymbol{x}(\boldsymbol{i}),\ \boldsymbol{SV_j(i)}) \right) + b$$

i.e. the kernel function finds the *minimum* of the corresponding elements from input feature vector and the current support vector in the inner summation. Due to the non-linearity of this function, $x(i)$ cannot be brought outside the inner summation for simplification and hence the run-time complexity of this function is *[#x3b8]*($m \times n$). This is in general true for all non-linear kernel functions such as *Gaussian* kernel, discouraging their use for real-time applications despite their better accuracy over *linear SVM*. However, this run-time computation complexity can be dramatically reduced through the use of a LUT if the input feature vector, $x(i)$, only takes on integer values with limited dynamic range. This can be done by pre-computing the inner summation of (8) for all the possible values $x(i)$ take and storing inside a finite two-dimensional *LUT* [15] such that (9)

$$T(i,\ k) = \left(\sum_{j=1}^{m} \alpha_j \min(k,\ \boldsymbol{SV_j(i)}) \right)$$

(10)

$$h_{\text{HIK}}(x) = \sum_{i=1}^{n} T(i,\ \boldsymbol{x(i)}) + b$$

Here 'T' stores the pre-computed values of the inner summation in (8) for the whole dynamic range of $x(i)$. Equation (9) assumes only positive integer values of $x(i)$ so that 'k' varies from 0 to the maximum possible value that $x(i)$ can take. Thus 'T' is a '$n \times \max(x(i)) + 1$' *LUT* and can be pre-computed. The *HSG* feature vector proposed in [6] and its augmented version, described in the previous subsection, are both well-suited for this *LUT*-based *HIK-SVM* implementation since these comprise of integer-only elements with limited dynamic range. For a block size of 6 × 6, the maximum attainable value of $x(i)$ according to (4) is 72. It should be noted that 'T' contains floating point values due to real-valued '$[\#x3b1]_j$' being multiplied with integer-valued '
$\min(k,\ \mathrm{SV}_j(i)$
'. Thanks to this *LUT*, the number of operations required per classification is only 'n' floating point additions as evident from (10). Compared to 'n' floating point multiplications and 'n' additions required by *linear SVM*, described in (7), this results in significant speed up while gaining better discrimination power of *HIK-SVM* at the same time.

3.3

Non-maxima suppression

The classification of feature vector is performed several times as the detection window slides over multiple scaled version of the input frame following specified spatial and scale strides to detect pedestrians of various sizes at different locations. This typically results in multiple detections of the same object at close-by scales and spatial locations. Various *NMS* techniques have been described in the literature to output a single location and size of the object with an associated confidence score. These schemes are generally built around the notion that of all the overlapping detected windows, the one with the highest score is the optimal one. Mostly this window coincides with the centroid of all the detected windows in the vicinity. Moreover, false positive detections are also considered more likely to be *lone samples* than clustered multiple detections generated by true objects. Thus, the number of overlapping detected windows also serves as a heuristic for confidence score. Both of these concepts are mathematically incorporated in the *mean shift clustering* employed in the original *HOG-linear SVM* detector. Recently, however, a simpler *NMS* technique which simply suppresses the window with lower confident score has been proposed and shown to work as effectively [8]. The criteria for sufficient overlapping of the two competing windows in this technique are calculated according to *PASCAL* formulation. For *HSG-HIK-plus* detector, it is proposed that the final confidence score, *S*, be a function of both number of detections and the highest score among overlapping

windows, given as (11)

$$S = n^{\alpha} \times \max\{W\}$$

where '*n*' is the number of overlapped detection windows, '*W*' is the set of individual scores of these overlapped windows and 'α' is an empirically determined constant. For $\alpha = 2$, this function gives better detection accuracy than the above-mentioned *NMS* schemes for false positive rates less than 10-2 when tested on standard datasets.

The enhanced HSG-HIK-plus framework described in this section does not require computationally expensive floating point multiplication operations and yet makes a powerful pedestrian detector as evidenced by detection results presented in Section 4.

4

Experimental setup, results and discussion

This section discusses the detection performance of the proposed *HSG-HIK-plus* pedestrian detection framework on standard datasets and discusses important results. The framework has been implemented in C++ using Microsoft® Visual Studio 2013 and integrates open source libraries, SVMlight [31] and OpenCV [32], for SVM training and video interfacing, respectively. Three detectors with detection window sizes 18 × 54, 30 × 78 and 36 × 102 with pedestrian heights 48, 72 and 96, respectively, were trained using INRIA pedestrian dataset [33]. For these detectors, same block size of 6 × 6 and stride of three in both directions, as described in the previous section, were used leading to feature vector sizes of 850, 2250 and 3630, respectively. Thus, these require LUT sizes of 850 × 73, 2250 × 73 and 3630 × 73, respectively, for HIK-SVM classification as described in the previous section.

During SVM training, a soft margin of 0.00025 was found to give the best results and hence used for all the experiments. Furthermore, a significant improvement in detection was observed when more than four bootstrapping cycles were run during the training. For this purpose, an iterative algorithm, depicted in Fig. 4, was employed to systematically include only the most relevant hardest negative examples in the training pool at each stage. The training process starts with an initial sample of 1000 randomly extracted windows from *INRIA* training dataset negative images. All 2416 positive examples from the same dataset are utilised. A naïve detector is trained with this minimal pool of negative examples and then successively improved by mining the negative images for false positives and retraining. The SVM score threshold is iteratively decreased from 2.5 (empirically determined) with a step of 0.2 to extract hard false positive examples. This approach ensures that multiple hard retraining cycles are run without inclusion of excessive negative examples. In contrast, running fewer bootstrapping cycles by including a large number of negative examples at each stage and starting with a large initial pool size both lead to poorer performance of the final detector.

Fig. 4 SVM training process with bootstrapping

Fig. 5 displays three sample HIK-SVM functions (9) learned by the training process for the detector with window size 36 × 102. These functions correspond to the orientation histogram bins centred around 50°, 0° and 10° angles for blocks at the location of right shoulder, belly and right leg, respectively. Thus, the presence of shoulder is strongly indicated by detection of edges oriented around 50° angle at the marked location as expected. Notice that the HIK function at this spot lowers the score if too many votes are cast in this bin indicating a high texture patch instead of an actual shoulder edge. The 0° angle bin at the belly location gives a negative score for any number of cast votes since this area should not depict a vertical edge normally, as reported by other research works as well. Similarly, detection of slightly vertical edges around 10° angles near the leg region leads to positive contribution to the final score. These *HIK-SVM* functions are sensitive to any alterations and lead to lower detection rate if processed by polynomial fitting or any other form of smoothing.

Fig. 5 Sample HIK-SVM functions corresponding to histogram bins centred around different orientation angles superimposed on 'average' positive training image

The detection performance of the three trained detectors corresponding to 48 (*small*), 72 (*medium*) and 96 (*large*) pedestrians on *INRIA* dataset has been illustrated in Fig. 6. As expected, the detector corresponding to the largest window size (36 × 102) gives the best performance while the smallest window size (18 × 54) gives the least. Since all three detectors use the same block size (6 × 6) and block overlapping ratio, they can simultaneously use the same block level feature vector extracted from a single scale to detect pedestrians of different heights. The *INRIA* test dataset contains pedestrians both more and less than 100 pixels high. Thus, if only the detector with the largest window size is run, it necessarily requires the input image to be scaled up to detect smaller pedestrians as well. The detection results presented in Fig. 1 were obtained by simultaneously running all three detectors without scaling up. This combined detector (Fig. 1) has a better detection accuracy than those of the individual detectors (Fig. 6) on *INRIA* test dataset. On *ETH* (Fig. 1) and *Caltech* (Fig. 7) datasets [34, 35], results for the proposed *HSG-HIK-plus* detector were obtained using only the *large* detector

corresponding to window size 36 × 102 (with upscaling for smaller pedestrians). These results can be seen to be either better or at par with those for *ACF* boosting cascade-based detector. The detector results are significantly better on *ETH* and *INRIA* because of the superior frame quality of these two datasets. Due to the very different picture quality of *INRIA* and *Caltech* datasets, *ACF* and other notable detectors described in the literature train different detectors using the training samples from these respective datasets to report the best results for each. *HSG-HIK-plus* detector proposed in this work, however, is trained using only *INRIA* dataset and the detection results for *ETH* and *Caltech* datasets are reported for the same detector. This demonstrates the universal applicability of the proposed detector in a variety of scenarios.

Fig. 6 Effect of detector window size on detection performance for INRIA pedestrian dataset

Fig. 7 Detection performance on Caltech pedestrian dataset

The legends in Figs. 1, 6 and 7 also list the *average MR* for the compared detectors in *FPPI* range [10-3 × 10-1]. This range is more likely to be used by practical detectors and the performance curves of the proposed detector show its suitability in this regard specially for pedestrians with height more than 80 pixels. This is important because it is deemed that with prevalence of high definition video cameras, detection of larger pedestrians (>80 pixels) will become more relevant.

Table 1 compares the detection speed of the proposed *HSG-HIK-plus* detector with those of other reported detectors. The results for *ACF*, *HSG-HIK, HSG-HIK-plus* and *SPF* were obtained on Intel® Core i7-3630QM CPU (@2.4[#x2005]GHz) with 8[#x2005]GB installed *RAM*. Results for *MT-DPM + context* and *RPN + BF* are taken from the respective published sources and are only available for VGA resolution. The proposed *HSG-HIK-plus* detector, implemented without multi-threaded parallel processing, *SSE2*-based optimisation or GPU support, achieves 2.0[#x2005]fps (frames per seconds) for the VGA resolution (640 × 480) video. The boosting cascade-based *ACF* pedestrian detector has been heavily optimised for *Intel SSE2* instruction set and hence achieves up to 13.5[#x2005]fps detection speed on the same video resolution despite being more computationally expensive. Respective processing speeds on *full-HD* resolution are significantly less for both *ACF* and *HSG-HIK-plus*. *SPF* detector uses parallel processing paradigm as well as *Intel SSE2* support and still runs at merely 0.5 and 0.07[#x2005]fps for VGA and FHD resolutions, respectively. Moreover, SPF spends additional time on pre-processing steps, i.e. optical flow calculation and region proposals generation. Thus, the final speeds are even lower. Fig. 7 includes detection results for two other detectors, i.e. deformable parts-based MT-DPM + context [28] and deep learning-based RPN + BF [25]. These two detectors show promising results but, as shown in from the data reported in Table 1, perform sub-optimally in terms of processing speed. *RPN + BF* gives the lowest detection MR on *Caltech* dataset but despite using parallel processing and GPU support takes more than 2[#x2005]s per VGA resolution frame due to its slow pre-processing proposal generation step as reported in [24].

Table 1 Detection speed (fps) comparison

Detector	Hardware support			VGA resolution (640?×?480)	Full-HD resolution (1920?×?1080)
	Intel SSE	Parallel processing	GPU		
ACF [2]	?	?	?	13.5	1.8
HSG-HIK [6]	?	?	?	2.2	0.32
HSG-HIK-plus (proposed)	?	?	?	2.0	0.3
MT-DPM?+?context [28]	?	?	?	?1	NA
SPF [7]	?	?	?	<0.5	<0.07
RPN?+?BF [25]	?	?	?	<0.4	*NA*

The proposed HSG-HIK-plus detector is slightly slower than the HSG-HIK detector because of the additional pre-processing steps depicted in Fig. 3. However, despite these additional steps, the speed is not affected too much because of the optimisations in the detection window size, which has been reduced from 64 × 128 to 36 × 102 and the removal of the averaging step in the histogram population stage. The proposed detector implementation does not employ multi-threading, SSE instruction set or GPU support and is hence a strong candidate for speed optimisation in the future work using either of these approaches.

Fig. 8 visually depicts the detection results of the proposed HSG-HIK-plus detector on sample images from the tested datasets. It can be noticed that objects with strong vertical edges (bearing similarity to human limbs) are more likely to trigger false alarms. False negatives, on the other hand, are expected to be caused by the absence of strong discriminating edges due to carried baggage, occlusion or loose clothing items.

Fig. 8 Visual detection results of HSG-HIK-plus pedestrian detector on standard datasets

5

Conclusion

This paper has described a pedestrian detection framework which employs low complexity features and pre-processing steps and yet achieves detection accuracy at par with the computationally complex contemporary detectors, e.g. *ACF* and *SPF* detectors. Moreover, the experimental results demonstrate that the proposed detector while trained using a single standard dataset, i.e. *INRIA*, performs competitively on other datasets such as *ETH* and *Caltech* as well. In conclusion, the developed framework proves that the discriminative power of HIK-SVM, efficiently harnessed through multiple bootstrapping and re-training cycles, can be used to make a powerful detector even with low complexity features and pre-processing operations.

LOAD-DATE: July 17, 2017

LANGUAGE: ENGLISH

BIBLIOGRAPHY:

REFERENCES

1 Dalal N., Triggs B.: 'Histograms of oriented gradients for human detection'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, San Diego, CA, USA, June 2005, pp. 886-893

2 Dollar P., Appel R., Belongie S., : 'Fast feature pyramids for object detection', IEEE Trans. Pattern Anal. Mach. Intell., 2014, 36, (8), pp. 1532-1545

3 Wang X., Han T.X., Yan S.: 'An HOG-LBP human detector with partial occlusion handling'. Proc. IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, September 2009, pp. 32-39

4 Wu J.: 'Efficient HIK SVM learning for image classification', IEEE Trans. Image Process., 2012, 21, (10), pp. 4442-4453

5 Zhao Y., Zhang Y., Cheng R., : 'An enhanced histogram of oriented gradients for pedestrian detection', IEEE Intell. Transp. Syst. Mag., 2015, 7, (3), pp. 29-38

6 Bilal M., Khan A., Khan M.U.K., : 'A low complexity pedestrian detection framework for smart video surveillance systems', IEEE Trans. Circuits Syst. Video Technol., 2016, PP, (99), pp. 1-1

7 Paisitkriangkrai S., Shen C., Hengel A.V.D.: 'Pedestrian detection with spatially pooled features and structured ensemble learning', IEEE Trans. Pattern Anal. Mach. Intell., 2016, 38, (6), pp. 1243-1257

8 Dollar P., Wojek C., Schiele B., : 'Pedestrian detection: an evaluation of the state of the art', IEEE Trans. Pattern Anal. Mach. Intell., 2012, 34, (4), pp. 743-761

9 Zhang S., Benenson R., Omran M., : 'How far are we from solving pedestrian detection?'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, June 2016, pp. 1259-1267

10 Benenson R., Omran M., Hosang J., : 'Ten years of pedestrian detection, what have we learned?'. Proc. ECCV Workshop on Computer Vision for Road Scene Understanding and **Autonomous** Driving, Zurich, Switzerland, September 2014, pp. 613-627

11 Watanabe T., Ito S.: 'Two co-occurrence histogram features using gradient orientations and local binary patterns for pedestrian detection'. Proc. 2nd IAPR Asian Conf. on Pattern Recognition, Okinawa, Japan, November 2013, pp. 415-419

12 Ma X., Najjar W.A., Roy-Chowdhury A.K.: 'Evaluation and acceleration of high-throughput fixed-point object detection on FPGAs', IEEE Trans. Circuits Syst. Video Technol., 2015, 25, (6), pp. 1051-1062

13 Chen P.Y., Huang C.C., Lien C.Y., : 'An efficient hardware implementation of HOG feature extraction for human detection', IEEE Trans. Intell. Transp. Syst., 2014, 15, (2), pp. 656-662

14 Maji S., Berg A.C., Malik J.: 'Classification using intersection kernel support vector machines is efficient'. Proc. IEEE Computer Society Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, June 2008, pp. 1-8

15 Wu J., Rehg J.M.: 'Beyond the Euclidean distance: creating effective visual codebooks using the Histogram Intersection Kernel'. Proc. IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, September 2009, pp. 630-637

16 Benenson R., Mathias M., Tuytelaars T., : 'Seeking the strongest rigid detector'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Portland, Oregon, USA, June 2013, pp. 3666-3673

17 Zhang S., Benenson R., Schiele B.: 'Filtered channel features for pedestrian detection'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, June 2015, pp. 1751-1760

18 Benenson R., Mathias M., Timofte R., : 'Pedestrian detection at 100 frames per second'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, June 2012, pp. 2903-2910

19 Yang J., Jiang Y.-G., Hauptmann A.G., : 'Evaluating bag-of-visual-words representations in scene classification'. Proc. Int. Workshop on Multimedia Information Retrieval, Augsburg, Bavaria, Germany, September 2007, pp. 197-206

20 Jiang Y.-G., Ngo C.-W., Yang J.: 'Towards optimal bag-of-features for object categorization and semantic video retrieval'. Proc. 6th ACM Int. Conf. on Image and Video Retrieval, Amsterdam, The Netherlands, July 2007, pp. 494-501

21 Lowe D.G.: 'Distinctive image features from scale-invariant keypoints', Int. J. Comput. Vis., 2004, 60, (2), pp. 91-110

22 Everingham M., Van Gool L., Williams C.K.I., : 'The Pascal Visual Object Classes (VOC) challenge', Int. J. Comput. Vis., 2010, 88, (2), pp. 303-338

23 National Institute of Standards and Technology (NIST), 'TREC Video Retrieval Evaluation (TRECVID)', http://www-nlpir.nist.gov/projects/trecvid/, accessed 27 December 2016

24 Zhang L., Lin L., Liang X., : 'Is faster R-CNN doing well for pedestrian detection?'. Proc. the 14th European Conf. on Computer Vision, Amsterdam, The Netherlands, October 2016, pp. 443-457

25 Ren S., He K., Girshick R., : 'Faster (R-CNN): towards real-time object detection with region proposal networks', IEEE Trans. Pattern Anal. Mach. Intell., 2015, PP, (99), pp. 1-1

26 Hosang J., Omran M., Benenson R., : 'Taking a deeper look at pedestrians'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Boston, Massachusetts, USA, June 2015, pp. 4073-4082

27 Felzenszwalb P.F., Girshick R.B., McAllester D., : 'Object detection with discriminatively trained part-based models', IEEE Trans. Pattern Anal. Mach. Intell., 2010, 32, (9), pp. 1627-1645

28 Yan J., Zhang X., Lei Z., : 'Robust multi-resolution pedestrian detection in traffic scenes'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June 2013, pp. 3033-3040

29 Maksai A., Wang X., Fleuret F., Fua P.: 'Globally Consistent Multi-People Tracking using Motion Patterns', https://arxiv.org/abs/1612.00604, accessed 27 December 2016

30 Wang X., Türetken E., Fleuret F., : 'Tracking interacting objects using intertwined flows', IEEE Trans. Pattern Anal. Mach. Intell., 2016, 38, (11), pp. 2312-2326

31 Joachims T.: 'Making large-scale support vector machine learning practical', in Schölkopf B., Burges C.J.C., Smola A.J. (EDs.): 'Advances in kernel methods' (MIT Press, 1999), pp. 169-184

32 Bradski G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000

33 Dalal N., Triggs B.: 'INRIA Pedestrian Dataset', http://pascal.inrialpes.fr/data/human/, accessed 1 September 2016

34 Dollár P.: 'Caltech Pedestrian Dataset', http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/, accessed 1 September 2016

35 Ess A., Leibe B., Schindler K., Van Gool L.: 'A mobile vision system for robust multi-person tracking', 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-8

PUBLICATION-TYPE: Magazine

---- End of Request ----
Print Request: Current Document: 7
Time Of Request: Friday, February 09, 2018 02:13:22 EST