Traffic Sign Detection from Video: A Fast Approach with Tracking

Dongdong Wang, Xinwen Hou and Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences
{ddwang, xwhou, liucl}@nlpr.ia.ac.cn.

Abstract

This paper proposes a fast approach for traffic sign detection from video. First, we modify the image-based detector HHVCas to improve its accuracy and speed, then apply it to video-based detection with further acceleration by tracking. For the image-based detector, by optimizing the parameters in the cascade using an unsupervised approach, we achieve performance comparable to the state-of-the-art while keeping the speed advantage. Parallelizing some steps in the HHVCas detector leads to $1.5\times$ *speedup and* 20 *fps detection. In video, the detector achieves* $2.8\times$ *speedup and performs* 35 *fps by tracking every other frame. It also obtains significant precision increase by* $5\sim8\%$ *at high recall when exploiting temporal coherence of results in multiple frames.*

1. Introduction

Traffic sign detection from image and video has been studied by many researchers. Some adopt segmentation algorithms based on color [4, 13]. Others exploit shape information by Hough transform derivatives [5]. An important family of methods perform by sliding window. In this case, many works extend the famous Viola-Jones detector using different features [10, 8]. The idea of coarse-to-fine search is used by [7, 16, 15], where nonsigns are pruned sequentially by cascading stages. Some of these methods show appealing performance in images.

For traffic sign detection in video, more information such as spacial and temporal cues between multiple frames is provided. These cues can be captured by tracking algorithms. Generally, tracking predicts the positions of candidate signs in the following frame where detection can perform locally. Many method [11, 1, 14] worked in this way. The authors of [11] also used the tracking information to update a Pixel Relevance Model which further boosts a shape-based sign detector. The method in [1] exploited belief functions for ROIs association. Additionally, the temporal information of the ROIs is analysed to reduce false alarms. The work [14] employed a trained discriminative model to classify obtained tracks. The tracking module in our research bears some similarity to the work [1] but uses different data association methods.

Our system is based on the detector [15], which consists of four cascading stages where HOG variants are adopted. Here we refer it as HHVCas (Hybrid HOG Variants Cascade) for abbreviation. Although it has an advantage over state-of-the-art methods in speed, it cannot achieve real-time detection. In addition, the original HHVCas suffers from the dilemma of parameter tuning. In this work, we modify the HHVCas and apply it to video-based detection with tracking. The contributions of this work are as follows. Firstly, we optimize the parameters of the HHVCas by introducing an unsupervised method [2]. The performance of the optimized detector is comparable to the state-of-the-art and keeps speed advantage. Further, the detector is parallelized and performs 20 fps on 720×576 images. Finally, the detector is applied to video and boosted by tracking. It gains great speedup and runs at 35 fps. By exploiting temporal coherence of results in multiple frames, it obtains significant precision improvement by $5\sim8\%$ at high recall.

The remainder of this paper is organized as follows. Section II describes the improvements on the HHVCas detector involving parameter optimization and a parallelized implementation. Section III details the tracking part in video where search region maintaining and data association are used. Experiments on image and video datasets are discussed in Section IV. At last the work is summarized in Section V.

2. Detection

Improvements about parameter tuning and speed are made for the HHVCas [15]. A principled framework [2] is applied to the detector to optimize the parameter selection. Additionally, the detector is accelerated by a parallelized implementation.

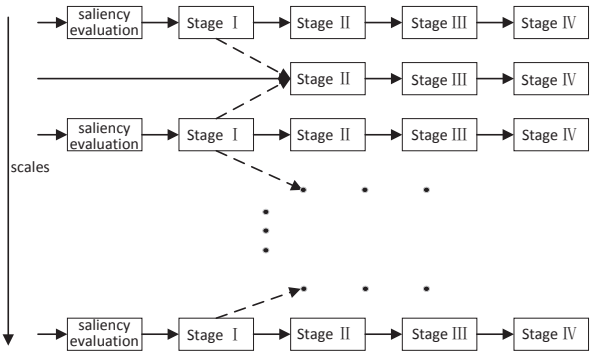

Figure 1. System pipeline. The HHVCas detector includes four cascade classifiers. Sub-windows only surviving current stage will forward to the next stage. A robust saliency test is firstly adopted to eliminate background regions. Then compressed integral HOG is used at the first stage to eliminate most irrelevant windows. For subsequent stages, multiple HOG variants, including integral HOG, HOG and color HOG features, are used and several classifiers are learned to prune non-signs. A technique called neighbor scales awareness shown by dash lines is adopted to speed up evaluation.

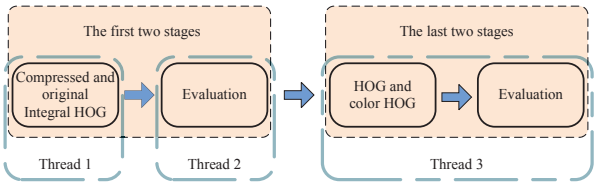

Figure 2. The three-thread HHVCas. The four-stage detector is divided into two parts. The first part consists of the first two stages where integral HOG and its compressed variant are computed and evaluated. The second part includes the remaining two stages. Two child threads are occupied by the first part. One is responsible for compressed and original integral HOG extraction and the other for hypothesis evaluation. A main thread maintains the detector and is responsible for the last part to further evaluate obtained ROIs.

2.1. Detector

The HHVCas detector introduced by [15] consists of a preprocessing step using mid-level saliency and four classification stages (Fig. 1). They form a cascade framework with carefully-designed HOG variants.

2.2. Parameter Optimization

We want to obtain optimal parameters instead of those chosen empirically [15]. We follow the work [2], where an unsupervised, data-driven way is presented to select thresholds of a soft cascade. Here we consider the first three stages and define the HHVCas as a K ($K = 3$) stages detector. In this case, H_k ($k \in \{1, 2, ..., K\}$) denotes classifiers associated with thresholds θ. A set θ^* is chosen as the base rejection thresholds according to the performance on the training set and then is used to choose the optimal parameters.

There are two modules should be optimized. *Cascade:* It rejects a hypothesis x if its score $H_k(x)$ is lower than a per-stage rejection threshold $\theta_k^R \geq \theta^*$. *Neighbor scale awareness:* It focuses on the communication between windows of neighbor scales. A window x is pruned if in the previous stage all its neighbors fall below the rejection threshold $\theta_k^S \geq \theta^*$.

The HHVCas with base thresholds is then evaluated on the training images. Detected windows $x \in \mathcal{X}$ are referred as quasi-positives and the quasi miss rate (QMR) γ is defined as the fraction of quasi-positives with $H_k(x)$ rejected by a cascade. Let $\gamma^{'} = 1 - (1 - \gamma)^{1/K}$. If at each rejection stage the QMR is $\leq \gamma^{'}$, the overall QMR of the cascade module will be $\leq \gamma$. Let $\mathcal{X}_1 = \mathcal{X}$ be the set of quasi-positives and define $\mathcal{H}_1 = \{H_1(x)|x \in \mathcal{X}_1\}$. The first rejection threshold θ_1^R is obtained by:

$$\theta_1^R = \lfloor \mathcal{H}_1 \rfloor_r - \epsilon \quad \text{where } r = \lfloor \gamma^{'} \cdot |\mathcal{H}_1| \rfloor. \tag{1}$$

Here $\lfloor \mathcal{H} \rfloor_r$ denotes the r^{th} smallest value in $\mathcal{H}$ and $\epsilon = 10^{-5}$. For other stage $1 < k \leq K$ we define $\mathcal{X}_k = \{x \in \mathcal{X}_{k-1} | H_{k-1}(x) > \theta_{k-1}^R\}$ and $\mathcal{H}_k = \{H_k(x)|x \in \mathcal{X}_k\}$, obtaining θ_k^R via:

$$\theta_k^R = \lfloor \mathcal{H}_k \rfloor_r - \epsilon \quad \text{where } r = \lfloor \gamma^{'} \cdot |\mathcal{H}_k| \rfloor. \tag{2}$$

For the neighbor scale awareness module, we consider the score $H_1(x)$ and the threshold θ^S. Let $\mathcal{N}(x)$ be neighbors of x at nearby scales. Let $x^{'} \in \mathcal{N}(x)$ and define $x_i^{'}$ by:

$$H_1(x_i^{'}) = \max_{x_i^{'}}(H_1(x^{'})) \tag{3}$$

The score $H_1(x_i^{'})$ is the maximum of all the $x^{'}$. If $H_1(x_i^{'})$ is above the base threshold θ_1^*, corresponding $x_i^{'}$ is collected leading to another quasi-positives set $\mathcal{X}_2$. Let $\mathcal{H}_1 = \{H_1(x_i^{'})|x_i^{'} \in \mathcal{X}_2\}$. We set θ^S by:

$$\theta^S = \lfloor \mathcal{H}_1 \rfloor_r - \epsilon \quad \text{where } r = \lfloor \gamma \cdot |\mathcal{H}_1| \rfloor \tag{4}$$

The $\lfloor \mathcal{H}_1 \rfloor$ is computed as above. Since multi-resolution models are adopted for detection at different scales, the above methods work separately for different models. Thus a single QMR can be used to tune the parameters of different modules and models.

2.3. Parallelization

The HHVCas detector is parallelized since it cannot achieve real-time detection with a single thread. We find that the feature extraction and the evaluation modules run at comparable speed and can be easily separated. In addition, modern PCs with multiple cores are prevalent, so we develop a three-thread HHVCas as depicted in Fig. 2. The system is divided into two parts. The first part consists of the first two stages where integral HOG and its compressed

variant are computed and evaluated. The second part consists of the last two stages. Two child threads are occupied by the first part for feature extraction and evaluation. A main thread maintains the detector and is responsible for the last part to further evaluate obtained ROIs.

3. Tracking

Tracking algorithms capture the temporal and the spacial redundance between detected signs in multiple frames and provides much useful information. Kalman Filter with constant speed motion is adopted. Its prediction is used to generate small search regions for local detection every other frame. The ROIs tracked by Filtering are maintained by data association, then are analyzed to reduce false positives. Consider the raw detection signs (without NMS) at frame t, they are divided into different groups according to pairwise overlap. The resulting groups are presented $gp = \{ur, r\}$, where ur is the union of all the sub-windows and r is the sub-window with the maximum detection score. A trajectory is defined by $Rt = \{r_i\}(i = 1, ...n)$.

3.1. Search Region Maintenance

For a sub-window r in a trajectory, let us define its state vector $\mathbf{x}(t) = (i, j, s, v_i, v_j, v_s)^T$, where (i, j, s), (v_i, v_j) and v_s denote its position and size, velocities and scale variation respectively. The prediction state vector $\hat{\mathbf{x}}(t+1)$ is obtained by Filtering and used to predict a search region $\hat{ur}(t+1)$.

$$\hat{\mathbf{x}}(t+1) = A\mathbf{x}(t) + \mathbf{q}(t+1) \quad (5)$$

$$\hat{ur}(t+1) \xleftarrow{\hat{\mathbf{x}}(t+1)} ur(t) \quad (6)$$

Here equation (5) is the prediction step of a Kalman Filter, where A and $\mathbf{q}$ denote the state matrix and the Gaussian noise related to the dynamic system. The velocity and size variation in $\hat{\mathbf{x}}(t+1)$ is used to compute the prediction region $\hat{ur}(t+1)$. The final search region at frame $t+1$ is the union of $ur(t)$ and $\hat{ur}(t+1)$.

3.2. Data Association

The detected trajectory is maintained by data association which assigns detected signs to trajectories. A ROI trajectory is accepted as valid if it has at least $N_{trConfir}$ consecutive associations and ends up when N_{trEnd} consecutive failures appear. The affinity of two sub-windows is defined as in [17]:

$$A(r_i, r_j) = A_{pos}(r_i, r_j)A_{size}(r_i, r_j)A_{appr}(r_i, r_j) \quad (7)$$

where A_{pos}, A_{size} and A_{appr} are affinities based on position, size and appearance respectively. The histogram distance on HOG is adopted in A_{appr}. The match between trajectories and detected signs is implemented as in [17].

At a particular frame t, we firstly match detection signs to the trajectories. For an unmatched trajectory Rt, we search a potential r_p in the region of the Filtering prediction. The association score $A(Rt(r), r_p)$ and the detector confidence $H(r)$ are combined to obtain a weighted confidence $H_w(r_p)$ via:

$$H_w(r_p) = \alpha H(Rt(r)) + (1-\alpha)H(r_p) \quad (8)$$

$$\alpha = \lambda exp(\frac{A(Rt(r), r_p)}{2\sigma^2}) \quad (9)$$

where λ is used to control the weight of association score. If r_p with the raised confidence is accepted as a positive, it will be associated with Rt. Finally unmatched Rt suffers an interruption in trajectory.

4. Experiments

Our system is trained and evaluated on the GTSDB [6] and the MASTIF [14] databases. The GTSDB contains 600 training images and 300 testing images, including 846 and 360 traffic signs respectively. Most of the signs are grouped into three main categories: prohibitory, danger and mandatory signs. The MASTIF consists of some datasets including both images and videos. The biggest TS2010 video is used in our experiments. It lasts about 1.5 hours containing more than 130, 000 frames and a few hundreds of signs. As in [14], we focus on the superclass of triangular signs which correspond to the danger category in the GTSDB. In the original annotation [14], each physical sign is sparsely annotated at several frames. We re-annotate the video and obtain per frame annotation using the excellent tool [3]. In total, about 11,000 frames containing signs are annotated.

Experiments are provided to demonstrate the parameter optimization and the effectivity of tracking. All our experiments are preformed on an i7 CPU.

4.1. Detector Evaluation

Per-stage parameter optimization for the HHVCas is performed on the GTSDB training set. Negatives are gathered by randomly selecting sub-windows with no overlap with ground truth. Positives are generated by jittering signs cut from the training images. Base thresholds θ^* are selected for classifiers in each stage when training. Setting $\theta_1^* = -1$ is used for SVM classifiers. For LDA classifiers, values with zero false negative rate and the lowest false positive rate are selected. We then gather two quasi-positives sets for the cascade and the neighbor scale awareness modules. Several QMR values in the logarithmic coordinate are tested. The resulting detector is validated on the GTSDB testing set. Statistics are given in Fig. 3, including recall, precision, candidate sub-windows number per image and detection time. Detected results after the first two stages are merged into a big region where stage III is performed. This results in less computation complexity when

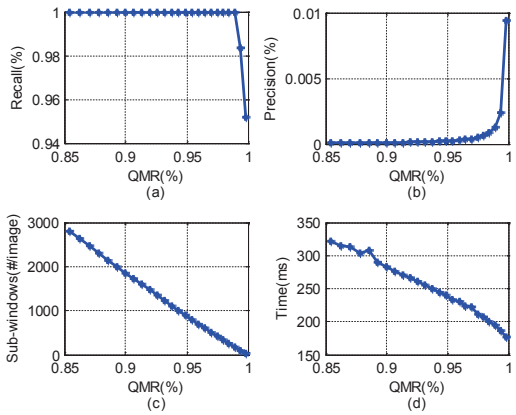

Figure 3. Statistics at different QMR values. High QMR leads to fast evaluation and high precision. Excessively high QMR hurts the recall dramatically.

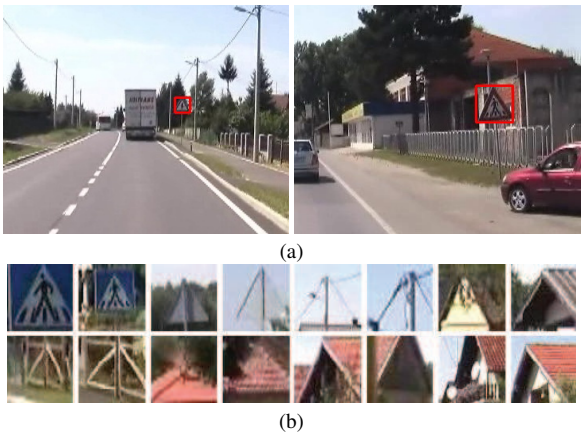

Figure 4. Examples of detection result in the TS2010 video. Detected signs and false alarms with high detection score are given in (a, b) respectively.

extracting HOG. It interprets some irregularities appearing in Fig. 3(d). High QMR leads to fast evaluation and high precision. Thresholds at the QMR turn point in Fig. 3(a) are used. The overall HHVCas is trained and used in all our experiments. Detection performance in term of AUC(Area Under the precision-recall Curve) and time on the GTSDB is shown in Table 1. The optimal HHVCas achieves 1.8% performance improvement than the pervious one. It is more fast and runs as $2\sim6\times$ fast as most of the state-of-the-art methods. The method [8] is more appealing and the time is for three sign categories together. It trained on a bigger training set beyond the GTSDB, however.

To test the effectiveness of parallelization, detectors are assessed on 720×576 images. The signs of $20\sim90$ pixels wide are detected. Table 2 gives the time costs. The parallel HHVCas (par_HHVCas) achieves $1.5\times$ speedup over the non-parallel version.

Team/Method	Danger		
	AUC	Time(ms)	
wgy@HIT501 [16]	99.91%	∼1179	
visics [9]	100%	∼400*	*GPU
LITS1 [7]	98.85%	400∼1000	
BolognaSVLab [12]	98.72%	∼794	
milan	96.55%	-	
SFC-tree[8]	99.20%	∼192*	*Total
WaDe+MSER[13]	99.79%	∼794	
HHVCas	95.72%	∼301	
Optimal HHVCas	97.58%	∼207	

Table 1. Performance and runtime on the GTSDB. The optimal HHVCas achieves 1.8% performance improvement than the pervious detector. Its performance is comparable to most state-of-the-art methods and obtained at high speed.

	HHVCas	par_HHVCas	tr_HHVCas
Time(ms)	$\sim 79ms$	$\sim 51ms$	$\sim 28ms$
Comparison	$1\times$	$1.5\times$	$2.8\times$

Table 2. Detection time. The parallel HHVCas (par_HHVCas) achieves $1.5\times$ speedup over the non-parallel version. By tracking, the tr_HHVCas gains $2.8\times$ speedup over the HHVCas and $1.8\times$ over the par_HHVCas.

4.2. Tracking Evaluation

In detail, the detection on a whole frame is performed every other frame. At the next frame, detection operates locally in the small region maintained by tracking.

The tr_HHVCas (parallelized HHVCas with tracking) runs at 35 fps on average as shown in Table 2.

Performance improvement is expected by incorporating temporal context in tracking. Two implementations of tracking, the tr_HHVCas and the tr_HHVCas_trj, are evaluated. In the first case, tracking only provides a priori knowledge for detection searches. For the latter, trajectories are obtained and the ROI life is exploited to reduce false positives with $N_{trConfir} = 3$ and $N_{trBreak} = 3$. In experiments, all the 130,000 frames of the TS2010 are used. The precisions of the three detectors (two tracking-based ones and the one without tracking) at the same recalls are given in Table 3. There is a small performance difference between the HHVCas and the tr_HHVCas due to slightly different implementation: the tr_HHVCas eliminates some speedup strategies used on full frames when performing local search. The tr_HHVCas_trj achieves obvious precision increase compared with both the other detectors. This demonstrates that temporal coherence is effective in reducing false alarms. Note that our detectors are trained on the GTSDB training set. This results in inferior performance on the TS2010 dataset. Partly due to some difference between the two datasets in image quality and color. The other reason is that signs are annotated from a small

Recall	91.60%	89.57%	87.06%	83.06%	77.70%	72.27%	65.40%	56.71%	48.20%
HHVCas	21.35%	28.50%	37.09%	47.82%	62.96%	72.47%	81.56%	89.03%	93.39%
tr_HHVCas	22.49%	30.08%	38.28%	49.26%	61.48%	72.75%	81.32%	88.42%	92.66%
tr_HHVCas_trj	**27.28%**	**35.49%**	**46.13%**	**57.55%**	**68.42%**	**78.86%**	**86.59%**	**91.65%**	**95.33%**

Table 3. Precision comparison at some recall values. The tr_HHVCas_trj taking account of temporal coherence achieves obvious precision increase compared with other detectors. Compared with the tr_HHVCas which only uses tracking for search, the tr_HHVCas_trj obtains about 5∼8% precision improvement at high recall.

scale (20×20 pixels) which are prevalent in videos. They are more difficult to be detected [14]. Some examples of detection results in the TS2010 are shown in Fig. 4. Many false positives appear nearby roofs as in Fig. 4(b).

5. Conclusions

In this work, we propose a fast approach for traffic sign detection from video based on an improved version of the detector HHVCas. By optimizing the parameters in the cascade of HHVCas, we achieve performance comparable to state-of-the-art methods and keep speed advantage. Parallelizing the detector leads to $1.5\times$ speedup and 20 fps detection. By tracking every other frame in video, the detector performs 35 fps. It also achieves significant precision increase by $5\sim8\%$ at high recall while exploiting temporal context captured from multiple frames. To further improve the performance, we are seeking for strategies for better exploiting temporal and spatial information in video.

6. Acknowledgement

This work is supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316302, the Strategic Priority Research Program of the CAS (Grant XDA06040102) and National Natural Science Foundation of China (NSFC) Grant 61271306.

References

[1] M. Boumediene, J.-P. Lauffenburger, J. Daniel, C. Cudel, and A. Ouamri. Multi-roi association and tracking with belief functions: application to traffic sign recognition. *IEEE Trans. on Intelligent Transportation Systems*, 15(6):2470–2479, 2014. 1

[2] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *Proc. of European Conference on Computer Vision*, pages 645–659. 2012. 1, 2

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012. 3

[4] J. Greenhalgh and M. Mirmehdi. Real-time detection and recognition of road traffic signs. *IEEE Trans. on Intelligent Transportation Systems*, 13(4):1498–1506, 2012. 1

[5] S. Houben. A single target voting scheme for traffic sign detection. In *Proc. of Intelligent Vehicles Symposium*, pages 124–129. IEEE, 2011. 1

[6] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *Proc. of International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2013. 3

[7] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu. Traffic sign detection by roi extraction and histogram features-based recognition. In *Proc. of International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2013. 1, 4

[8] C. Liu, F. Chang, and Z. Chen. Rapid multiclass traffic sign detection in high-resolution images. *IEEE Trans. on Intelligent Transportation Systems*, 15(6):2394–2403, 2014. 1, 4

[9] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool. Traffic sign recognitionhow far are we from the solution? In *Proc. of International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2013. 4

[10] G. Overett, L. Tychsen-Smith, L. Petersson, N. Pettersson, and L. Andersson. Creating robust high-throughput traffic sign detectors using centre-surround hog statistics. *Machine Vision and Applications*, 25(3):713–726, 2014. 1

[11] A. Ruta, Y. Li, and X. Liu. Detection, tracking and recognition of traffic signs from video input. In *Proc. of Conference on Intelligent Transportation Systems*, pages 55–60. IEEE, 2008. 1

[12] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano. A traffic sign detection pipeline based on interest region extraction. In *Proc. of International Joint Conference on Neural Networks*, pages 1–7. IEEE, 2013. 4

[13] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano. Traffic sign detection via interest region extraction. *Pattern Recognition*, 48(4):1039–1049, 2015. 1, 4

[14] S. Šegvić, K. Brkić, Z. Kalafatić, and A. Pinz. Exploiting temporal and spatial constraints in traffic sign detection from a moving vehicle. *Machine Vision and Applications*, 25(3):649–665, 2014. 1, 3, 5

[15] D. Wang, S. Yue, J. Xu, X. Hou, and C.-L. Liu. A saliency-based cascade method for fast traffic sign detection. In *Proc. of Intelligent Vehicles Symposium*, pages 180–185. IEEE, 2015. 1, 2

[16] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang. A robust, coarse-to-fine traffic sign detection method. In *Proc. of International Joint Conference on Neural Networks*, pages 1–5. IEEE, 2013. 1, 4

[17] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007. 3