Boosting Detection Results of HOG-based Algorithms Through Non-Linear Metrics and ROI Fusion

Darius Malysiak[1], Anna-Katharina Römhild[2], Christoph Nieß[1], and Uwe Handmann[1]

[1]Computer Science Institute - Hochschule Ruhr West, Germany,[2] Hochschule Bochum, Germany

Abstract Practical application of object detection systems, in research or industry, favors highly optimized black box solutions. We show how such a highly optimized system can be further augmented in terms of its reliability with only a minimal increase of computation times, i.e. preserving realtime boundaries. Our solution leaves the initial (HOG-based) detector unchanged and introduces novel concepts of non-linear metrics and fusion of ROIs. In this context we also introduce a novel way of combining feature vectors for mean-shift grouping. We evaluate our approach on a standarized image database with a HOG detector, which is representative for practical applications. Our results show that the amount of false-positive detections can be reduced by a factor of 4 with a negligable complexity increase. Although introduced and applied to a HOG-based system, our approach can easily be adapted for different detectors.

Keywords: augmentation, object detection, gpgpu, high performance computing, histogram of oriented gradients, HOG, opencl, cuda, meanshift grouping, svm

1 Introduction and previous work

Histograms of oriented gradients[4] are a fundamental building block for many object detection systems. Even with the advent of deep-learning, several of today's state-of-the-art systems, e.g. [9] or [6]. [5]) still continue to use HOGs as supplementary information in order to boost their performance. In other cases, the benefits of HOGs, e.g. less required training data, outweigh those of other systems such as a slightly better recognition rate with much more training data. One can see these effects in the comparison of MultiFtr+CSS and HOG in [2]. Yet, the practical application of object detection systems, in research or industry, favors highly optimized black box solutions. Complex systems require an intrinsic understanding of their parameters in order to boost their performance, due to time constraints it is often unfeasible to study and retrofit an existing system. For industrial applications it is a costly task to train an SVM classifier

and optimize the involved parameters, a similar thought regarding time holds for research applications in which the detector results are merely used as supplementary feature elements. In this paper we show how such a highly optimized system can be further augmented in terms of its reliability with only a minimal increase of computation times, i.e. increasing the detection quality while preserving realtime boundaries. Our solution leaves the initial detector unchanged. Although introduced and applied to a HOG-based system, our approach can easily be applied for other detectors as well.
Section 2 briefly explains the challenge of grouping and selecting detections within the HOG algorithm. Furthermore it introduces the key elements of our approach; metric scaling of SVM weights and ROI fusion. Section 3 introduces a processing pipeline which applies the mentioned elements in order to boost the detectors performance. We conclude this paper with sections 4, 5 and 6 which describe our setup, present our results on a standard image database and give an outlook for additional research, respectively.

2 Boosting Results Through ROI Fusion and Non-Linear Metrics

Let us assume an already trained HOG detector, i.e. all HOG parameters and the involved SVM training have been optimized for some training/verification set of images. Even in case of a huge training set, the pure HOG detection will yield a large amount of false-positive detections. This is often addressed by discarding all detections y_i whose SVM score ω_i lies below a certain theshold t. This can reduce the e.g. ≈10000 positively classified patches down to ≈50, which usually removes many false positives yet keeps many adjacent scales and positions for correct classifications. In order to reduce these detection groups down to an (ideally) single representant one applies clustering methods such as the mean shift approach. Yet as the mean-shift algorithm incorporates the SVM scores, it is also possible to loose true-positives (in case of true positives with small SVM scores). Finetuning the threshold t yields only marginal improvements and results in an increased amount of detections, which in turn can significantly slow down clustering algorithms.

2.1 ROI Fusion

The following approach is motivated by the results of [4], who utilized a weighted variant of mean-shift clustering for the grouping of multiple detections in (x, y, s) space. Let D_1, D_2 be existing HOG detectors which are trained to find distinct parts of an object, e.g. an upper-body and a head detector respectively. Just as with classical mean shift algorithms, e.g. [3], in which one iteratively estimates the modes y_m (of n points y_i) of a distribution by

$$y_m = H_h(y_m) \sum_{i=1}^{n} \overline{\omega}_i(y_m) {H_i}^{-1} y_i \tag{1}$$

with

$$\overline{\omega}_i(y_m) = \frac{|H_i|^{-1/2} \exp(-D^2[y_m, y_i, H_i]/2)}{\sum_{j=1}^{n} |H_j|^{-1/2} \exp(-D^2[y_m, y_j, H_j]/2)} \tag{2}$$

Let $y_i = (x, y, s) \in \mathbb{R}^3$ be the elements of the sampled data (i.e. the windows obtained by a complete multiscale HOG run, x, y, s denoting the position of the window center and scale respectively), $H_i = diag(\sigma_x, \sigma_y, \sigma_s)$ the diagonal uncertainty matrix and

$$D^2[y_m, y_j, H_j] := (y_m - y_i)^T {H_i}^{-1} (y_m - y_i) \tag{3}$$

$$= \sigma_x((y_m)_1 - (y_i)_1)^2 + \sigma_y((y_m)_2 - (y_i)_2)^2 + \tag{4}$$

$$\sigma_s((y_m)_3 - (y_i)_3)^2 \tag{5}$$

the Mahalanabois distance between y_m and y_i ($(y)_i$ indicates the i-th vector element). We propose the following weighted extension, let y^1, y^2 denote the resulting windows from D_1, D_2 respectively and ω^1, ω^2 the corresponding SVM scores / weights. First one has to create feasible 5-dimensional features

$$\tilde{y}_k := ((y_i^1)_1, (y_i^1)_2, (y_j^2)_1, (y_j^2)_2, (y_i^1)_3), \quad \tilde{\omega}_k := \omega_i^1 \omega_j^2 \tag{6}$$

by grouping all feasible D_2 windows y_j^2 for a single D_1 window y_i^1. The selection criteria for this combination, which must be fulfilled, are as follows

1. $\alpha_1(y_i^1)_3 \leq (y_j^2)_3 \leq \alpha_2(y_i^1)_3, \quad \alpha_1, \alpha_2 \in (0, 1], \alpha_1 < \alpha_2$
2. Let w, h denote the width and height of y^1:
   $\beta_1 w \leq (y_j^2)_1 - (y_i^1)_1 \leq \beta_2 w, \quad \beta_1 < \beta_2, \ \beta_1, \beta_2 \in (0, 1]$
3. $\beta_3 h \leq (y_j^2)_2 - (y_i^1)_2 \leq \beta_4 h, \quad \beta_3 < \beta_4, \ \beta_3, \beta_4 \in (0, 1]$

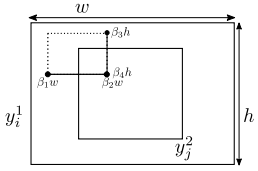


Figure 1. The fusion of detections, each detection y_i^1 of detector D^1 is combined with all detections y_j^2 from detector D^2 if its left upper corner lies in the dashed rectangle. The same applies for the scale.

These rules represent position restrictions which combine windows y_j^2 only if they lie in a certain boundary relative to y_i^1 (see Fig. 1). A practical example would be to consider only head windows which lie completely within the upper-body window. This grouping lifts the upper-body windows into a 5-dimensional space and adds the position variance of each fitting head window y_i^2 to it. The scale remains unchanged since a scale equivalent is defined with criteria 1. The mean-shift clustering was changed to a weighted variant

$$\overline{\omega}_i(y_m) = \frac{|H_i|^{-1/2} \tilde{\omega}_i \exp(-D^2[y_m, y_i, H_i]/2)}{\sum_{j=1}^{n} |H_j|^{-1/2} \tilde{\omega}_j \exp(-D^2[y_m, y_j, H_j]/2)} \tag{7}$$

with $H_i = diag(\sigma_x^1, \sigma_y^1, \sigma_x^2, \sigma_y^2, \sigma_s)$. The uncertainty values σ_x^2, σ_y^2 should be set to a smaller value than σ_x^1, σ_y^1, since might be reasonable to put more certainty into

D_2 so that it might stabilize the detection windows for the upper body. Since the amount of D_1 detections is increased we refer to this combination approach as sample spreading. The evaluation in section 5 shows that this strategy can yield a significant improvement in detection quality compared to D_1 alone. It should be pointed out that k can (in theory) reach values up to $|\{y_i^1\}| \cdot |\{y_j^2\}|$, which can even slow down an efficient implementation.

2.2 A Nonlinear Metric for SVM weights

In order to reduce the computation time and increase the detection quality in the context of HOG applications, one usually first filters out all result windows with a SVM weight below a given threshold t_1. This strategy can be applied to accommodate the problem of too many items for the mean shift clustering. Yet this simple method can also remove a large amount of correct detections. The reason for this lies in the HOG algorithm, for large objects it will scale the corresponding image area down to the detection window size, this removes a large quantity of high-res image information. Such windows will exhibit a smaller SVM weight compared to smaller regions. Filtering according to t_1, which obviously will be chosen according to the higher SVM values (and thus the smaller windows), will remove many if not all candidates for large objects. Our approach addresses this problem under the assumption that not all large windows have been filtered out. We developed a simple strategy by rescaling the SVM weights (after filtering with t_1) according to

$$\omega'(\omega_i) := f(\omega_i)\omega_i \tag{8}$$

with for example

$$f(\omega_i) := \begin{cases} \tau(-\theta((I_h - (y_i)_2)/I_h))(y_i)_3 & (y_i)_3 \geq \rho \\ 1 & else \end{cases} \tag{9}$$

with I_h being the image height and θ a constant. The general effect of this transformation should be that SVM weights of large windows will be increased to rival with those of smaller windows during the mode estimation. Not only can this approach retain large windows but also remove infeasible small windows in an area of large windows. The scaling function f must be chosen to accommodate this goal. In the example above the scaling function f exhibits a linear behaviour, yet one might also use an nonlinear tesselated transformation for more complex scenes. Large objects usually appear in the lower part of the image while small objects inhibit the upper portion. Thus SVM weights of window candidates in the lower region should be scaled up, while the scaling vanishes linear in the upper image area. Furthermore only the weights of windows above a certain scale will be transformed, this prevents small windows in the lower area to be transformed as well. An example for this can be seen in Fig.2.

3 A Detection-Pipeline for Boosting the Detection Quality

In order to further motivate the techniques from section 2 we conducted a thorough evaluation by embedding them in a detection pipeline. This section shows

Figure 2. Weighting of near field windows: The left image shows the use of a single upper-body detector; the person in the lower part is not detected due to small amounts of candidate windows. Using transformed SVM weights one can see on the right image that the same detector now finds the person in the lower part and suppresses the small false detection.

that an efficient implementation of the previously described algorithms and ideas can boost HOG-based systems in terms of their detection quality while still maintaining previous time constraints.
The detection pipeline is depicted in Fig. 3. The first step may consist of any form of image preprocessing, the output is directed into the HOG detector, which performs the initial detection (of at least one feature) without any form of detection grouping. All detections are forwarded into a metric-based selector, which consists of two steps; a thresholded reduction of detections and a metric scaling of SVM weights. After this point the results are forwarded into two parallel grouping branches, each consisting of two steps; detection grouping and a sanity check, which may use any available scene information in order to remove detections with impossible positions. The calculated detections from both branches are finally fused with a mean-shift grouping, these detections are forwarded into a so called *streaking* block. The streaking block is utilized in video streams and applies a simple heuristic (which is described in Alg. 1) in order to predict detections and eliminate short term detection gaps. The algorithm leaves out the details of how to find corresponding detections, i.e. it does not specify the form of feature vectors. In the following evaluation normalized intensity histograms have been used, additionally the distance between detections has been checked. More precisely

- The feature vector f_i consists of $n \cdot 256$ real numbers, 256 for each color channel.
- Feature vectors are compared by calculating the euclidean distance, which can not exceed a threshold t_f
- The distance between the upper left corner for two detections can not exceed a pixels along the x-axis and b pixels along the vertical.

Although simple in its design, this approach yields significant improvements compared to the canonical HOG algorithm. Yet, depending on the situation, e.g. image quality, a color histogram might become unfeasible since it is susceptible to image noise effects. This histogram based metric can be exchanged for more complex descriptors and similarity measures, e.g. a gradient based descriptor with a weight based metric, thus one can adapt the described pipeline for such a scenario. The streaking results will be grouped by a last mean-shift step after having been filtered by a final sanity check.
Note that the classic HOG algorithm can be obtained by setting less restrictive parameters for the metric selection, defining the sanity check of the fusion

branch to filter all results, setting the streaking history size $s = 1$, adapting the parameters of the last two grouping steps and the last sanity check.

Algorithm 1 Streaking

Require: Image I, detections $\mathcal{D}$, history size s, detection history $\mathcal{H} = \{(d_1, f_1, x_1, c_1), ..., (d_k, f_k, x_k, c_k)\}$ with shift buffer x_i of size s, feature vector f_i, match counter c_i and mismatch threshold t

```
b = 0;
for each d ∈ D do
    for each e_i ∈ H do
        compare the feature vector e_i and that of d (include additional sanity checks)
        if If the vectors match then
            Add position x of d to x_i;
            Update histogram of f_i with data at the position of d;
            Set c_i = 0;
            b = 1;
            break;
        else
            Set c_i = c_i + 1;
        end if
        if If c_i ≥ t then
            remove e_i from H;
        end if
    end for
    if b == 0 then
        Add new entry for d to H;
    end if
    b = 0;
end for
```

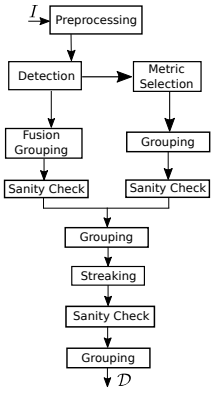


Figure 3. A detection pipeline with ROI fusion and metric scaling of SVM weights. The image I will be preprocessed before being forwarded into the HOG-based detector without grouping. The third block removes all detections below a certain threshold and rescales the weights according to a scene specific metric. The initial and remaining detections are sent into two parallel grouping branches; a fusion grouping and a canonical grouping, respectively. The grouped results of both branches will be merged afterwards. In case of video streams from a static scene the streaking block can be utilized for reduction of detection gaps. All sanity checks are scene specific heuristics which remove detections at unfeasible places. The pipelines output consists of a grouped detection set $\mathcal{D}$.

4 Evaluation

The motivation behind the design of the pipeline was to demonstrate the applicability of the developed algorithmic concepts. Since the work in this thesis mainly targets the improvement of efficiency while preserving scalability we will show the works potential by improving the detection rate of an existing HOG implementation while preserving the previous time constraints.

Two HOG detector were trained on the INRIA training set, one for the detection of entire human bodies (H_B) and one for detecting upper bodies (H_{UB}), the training was done according to the original protocol by [4]. Each corresponding SVM was obtained by a decadic grid search over $C \in [10^{-5}, 10^3]$ with a 10-fold cross-validation at every step. Since the INRIA database only provides labels for complete bodies, all upper body labels were extracted by using the upper third of each rectangular label ROI. Both HOG detectors have been utilized in the pipeline's detection block. Table 1 states the HOG parameters in more detail. The pipeline was evaluated on the CAVIAR [1] database since it provides targets for all blocks in the pipeline;

1. The metric scaling of SVM weights becomes applicable due to strong size differences between objects in near and far field.
2. The ROI fusion is trivially applicable.
3. The streaking is applicable since the images are extracted from a continuous video stream.

Futhermore, this database represents a typical field for many parallel video streams; surveillance. In order to get comparable results between the classic standalone HOG detectors and the pipeline results, the mean-shift grouping parameters were kept identical for all grouping steps, $\sigma_x = 16, \sigma_y = 8, \sigma_s = 1.05, \epsilon = 1.0$. The fusion grouping was done with $\alpha_1 = \alpha_2 = 0.05, \beta_1 = \beta_2 = 0.1, \beta_3 = \beta_4 = 0.1$ and $\sigma_x^1 = \sigma_x^2 = \sigma_x, \sigma_y^1 = \sigma_y^2 = \sigma_y$. The iteration count was limited to a maximum of 100.
A detection d is considered to be a true positive if it can be associated with a ground truth date d_{gt} such that

$$\frac{|d \cap d_{gt}|}{|d \cup d_{gt}|} \geq 0.7 \tag{10}$$

The parameters for the metric scaling were set to $\tau = \beta = 1.0$ and $\rho_{Body} = -0.2, \rho_{UpperBody} = -0.1$, this represents an entirely linear scaling over the complete image. One should note that ρ represents an individual thresholding for each detector. All sanity checks consist of checking if a detection lies in an unfeasible region, which are defined through a polygon set $P = \{p_1, ..., p_n\}$.
Since all images within the CAVIAR dataset have been recorded with a rather low resolution of $640x480$ the preprocessing consists of a GPU accelerated upscaling to $1600x1200$. This step is reasonable in the sense of applicability, the already trained detector should not need to be retrained for each different image format. Furthermore one would need to shrink the training images even more for smaller window sizes, this would introduce further information loss and a reduction in detection quality.
Two image sets of a specific scene were chosen, *WalkByShop1Cor* and *ThreePastShop1Cor*, the scene and the corresponding set P is visualized in Fig. 4. Besides the parameters for the metric weight scaling and the defined polygons no additional scene specific optimizations have been utilized.

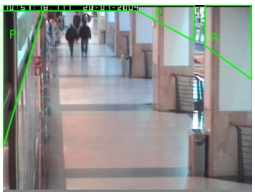


Figure 4. A scene from the CAVIAR image database, the surveillance camera's position induces a significant size difference between objects in near and far field. Three polygons (P_1, P_2, P_3) have been defined and enclose image regions which should not contain pedestrians.

Table 1. Parameters for both detectors, i.e. body and upper body HOG detectors

Set	Cell size	Block size	Window size	Block stride	Window stride	Bin count	Scale	σ
H_B	8×8	16×16	64×128	$(8, 8)$	$(8, 8)$	9	1.05	1.0
H_B	8×8	16×16	96×88	$(8, 8)$	$(8, 8)$	9	1.05	1.0

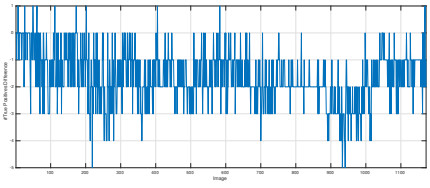
5 Results

The plots in Fig. 5 illustrate the differences between the classic HOG algorithm and elements of the pipeline. Let $TP_i = (x_0, x_1, ..., x_k, ...)$ be the sequence of image-wise true positive counts obtained with algorithm i for each image, analogously let FP_i be the sequence of false positives. Both plots in the first row of Fig. 5 depict the difference $\delta^{TP}_{HOG,Metric} := TP_{Metric} - TP_{HOG}$ and $\delta^{FP}_{HOG,Metric} := FP_{Metric} - FP_{HOG}$, respectively. If $\delta^{TP}_{HOG,Metric}(k) > 0$ for some image index k then more true positives were obtained by using the HOG algorithm than with metric scaling of the weights. The same holds for the false positive count. It becomes obvious that less true positives were obtained with metric scaling, yet one has to accept significantly more false positives. This indicates a stabilizing effect onto the canonical HOG approach, which is also visible in the second row of plots, i.e. the results of comparing the HOG against the pipeline's fusion branch. The last row in Fig. 5 depicts the comparison between the HOG and the entire pipeline. The amount of false positives is significantly reduced while keeping the amount of false positives close to that of the classic HOG algorithm. One obtains a mean value of 4.3643 less false positives per image and 1.02 less true negatives per image.

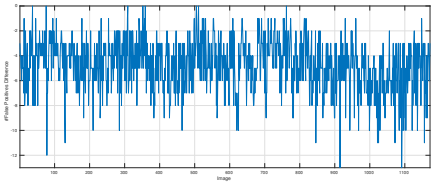
These results illustrate the potential of the constructed pipeline, by using the detections of an existing HOG detector one obtain significantly less false positives while preserving the amount of true positives. As Fig. 6 shows, it turned out that a significant amount of fused detections was constructed within the fusion branch. This in turn posed a bottleneck for the pipeline's applicability, since grouping times would reach values up $\approx$ 200s (right plots in Fig. 6). Yet, by using the concept for massively parallelized mean shift computation [7] this time could be reduced to a maximum of $\approx$ 10ms, which in turn led to the pipelines complete processing times as depicted in Fig. 7. The pipelines maximal detection time was $\approx$ 88ms, which still enables one to process $\approx$ 11 frames per second. An additional speed-up could be achieved by using the tile image approach from [8] since the actual SVM-based detection process makes up about 1/3 of the total processing time (see the red line in Fig 7).

One has to note that the detection quality is determined to a large extend by the initial HOG detector. An improvement of the underlying detections would further boost the pipelines results, such an improvement may include scene spe-

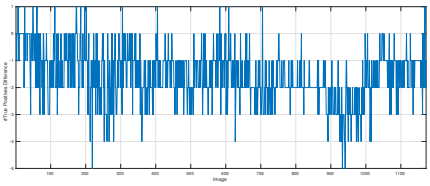
cific SVM training or boosting approaches from machine learning. Very similar results were obtained on the *ThreePastShop1Cor* image set.



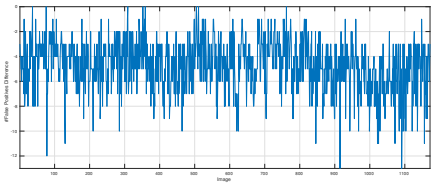
(a) HOG vs metric true positive count



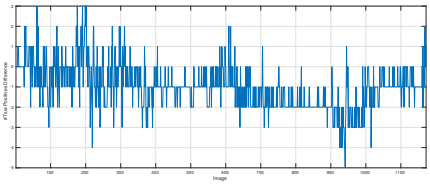
(b) HOG vs metric false positive count



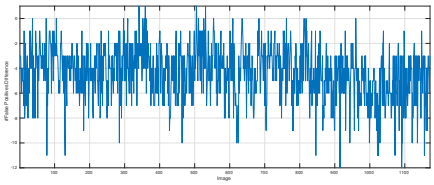
(c) HOG vs fusion true positive count



(d) HOG vs fusion false positive count



(e) HOG vs streak true positive count



(f) HOG vs streak false positive count

Figure 5. Comparison of recall statistics for the canonical HOG body detector and pipeline segments on the *ThreePastShop1Cor* image set. Image a) shows the difference: #true positives fusion branch - #true positives classic HOG, image b) the corresponding false positive difference.

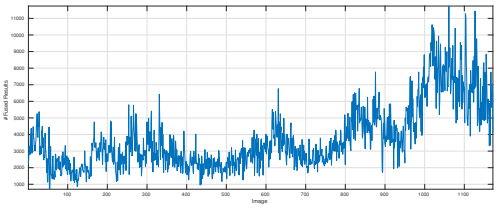


Figure 6. Statistics for the multidimensional mean-shift grouping on the *ThreePastShop1Cor* image set. The graph depicts the amount of computed detection combinations for the multidimensional mean-shift grouping.

6 Conclusion

A very common application for object detection is that of video surveillance, in which a system must process many parallel video streams. Section 3 shows how an existing HOG based system can be augmented for this application; using the described pipeline it becomes possible to increase the systems reliability, the induced complexity increase is negligable and can easily be compensated by techniques from [8] and [7]. The pipeline incorporates two developed concepts; metric scaling of SVM weights and ROI fusion, both being introduced in section

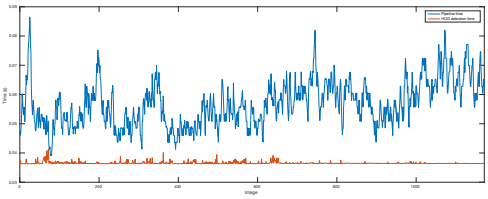


Figure 7. Complete processing time (blue line) of the pipeline for the *ThreePastShop1Cor* image set. The red line depicts the detection time, note that these time values are invariant to the amount of detections since the grouping has been left out.

2.2 and 2.1, respectively. The results indicate a stabilizing effect onto the initial HOG detector, i.e. the amount of false positive detections can be reduced while retaining the amount of true positives. As shown in section 5 one can expect, considering an adequate choice of parameters, a reduction of false positives by a factor of ≈ 4 while reducing the amount of true positives only marginally with a factor of ≈ 0.2. Increasing the detectors initial reliability will most likely eliminate the reduction of false positives, yet this remains a question for future research. The developed concepts can be applied to any object detector, yet the resulting gain in detection quality might be different, e.g. it may have the same stabilizing effect or even provide an increase of true positives. Furthermore it should be studied how the boosting parameters can be infered from the camera perspective itself, this may provide a simple applicable "black box" boost solution for existing detectors, no redesign or retraining is required.

References

1. Caviar: Context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, accessed: 2016-03-01
2. Benenson, R., Omran, M., Hosang, J., Schiele, B.: Ten years of pedestrian detection, what have we learned? In: Computer Vision-ECCV 2014 Workshops. pp. 613–627. Springer (2014)
3. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(5), 603–619 (May 2002)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
5. Fukui, H., Yamashita, T., Yamauchi, Y., Fujiyoshi, H., Murase, H.: Pedestrian detection based on deep convolutional neural network with ensemble inference network. In: Intelligent Vehicles Symposium (IV), 2015 IEEE. pp. 223–228 (June 2015)
6. Luo, P., Tian, Y., Wang, X., Tang, X.: Switchable deep network for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 899–906 (2014)
7. Malysiak, D., Handmann, U.: An algorithmic skeleton for massively parallelized mean shift computation with applications to gpu architectures. In: Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on. pp. 109–116. IEEE (2014)
8. Malysiak, D., Markard, M.: Increasing the Efficiency of GPU-Based HOG Algorithms Through Tile-Images, pp. 708–720. Springer Berlin Heidelberg, Berlin, Heidelberg (2016), http://dx.doi.org/10.1007/978-3-662-49381-6_68
9. Zou, W.Y., Wang, X., Sun, M., Lin, Y.: Generic object detection with dense neural patterns and regionlets. arXiv preprint arXiv:1404.4316 (2014)