Combination Features and Models for Human Detection

Yunsheng Jiang and Jinwen Ma*
Department of Information Science, School of Mathematical Sciences
and LMAM, Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn

Abstract

This paper presents effective combination models with certain combination features for human detection. In the past several years, many existing features/models have achieved impressive progress, but their performances are still limited by the biases rooted in their self-structures, that is, a particular kind of feature/model may work well for some types of human bodies, but not for all the types. To tackle this difficult problem, we combine certain complementary features/models together with effective organization/fusion methods. Specifically, the HOG features, color features and bar-shape features are combined together with a cell-based histogram structure to form the so-called ***HOG-III features****. Moreover, the detections from different models are fused together with the new proposed* ***weighted-NMS algorithm****, which enhances the probable "true" activations as well as suppresses the overlapped detections. The experiments on PASCAL VOC datasets demonstrate that, both the HOG-III features and the weighted-NMS fusion algorithm are effective (obvious improvement for detection performance) and efficient (relatively less computation cost): When applied to human detection task with the Grammar model and Poselet model, they can boost the detection performance significantly; Also, when extended to detection of the whole VOC 20 object categories with the deformable part-based model and deepCNN-based model, they still show competitive improvements.*

1. Introduction

Object detection is an essential task in computer vision, which grants computers the ability to "see" objects in digital images/videos. **Human Detection** is a primary issue among object detection, due to the specificity of human bodies in our daily lives. Unlike pedestrian detection (where almost all the targets are upright persons in distant views), human detection is still a challenging problem because of the large variations in visual appearance, which can be caused by various viewpoints and scales in photo-taking, different clothes and poses on target people, changeful illumination and large intra-class variations. Besides, the possible occlusions and complex backgrounds may create further difficulties.

Generally, an human detector mainly has two components: a feature extraction algorithm that encodes an input image as a feature vector, and a detection model that locates the target human bodies according to the computed vector.

In fact, feature extraction is a fundamental process for human detection. A good feature extraction algorithm should provide robust invariance to the large variations of human bodies while extracting enough information for detection. Dalal & Triggs [7] suggested the Histograms of Oriented Gradients (HOG) features that are robust to significant changes in image illumination and color as well as small changes in image contour locations and directions. The HOG features have proven effective for the detection of human and other shape-based object categories. Zhang *et al.* [32] and Wang *et al.* [30] showed that HOG-LBP features, a combination of HOG and Local Binary Patterns (LBP) [24], under some circumstances, could further improve the detection performance. This fact implies that the HOG features also have self-bias, and thus could be improved by combining with the other kinds of features. Other popular features for detection include the Scale-Invariant Feature Transform (SIFT) [23], Haar-like features [22], Wavelet features [29], Shape-Context features [1], and so on.

As for detection model, it is clear that a monolithic model is not so effective for human detection in consideration of the articulated structure of human bodies. A representation based on serval parts seems more powerful. Felzenszwalb [10, 12] introduced the Deformable Part-based Model (DPM), which described an object as a root block surrounded with several movable parts, and thus could alleviate the problems of appearance variations and occlusion. The DPM and its variants have shown significant progress on many difficult datasets, such as PASCAL VOC datasets.

On the basis of general grammar formalism [13], Girshick *et al.* [15] further proposed the person Grammar Mod-

[*] The corresponding author.

el which extended DPM from simple star-structure to general hierarchical structure. In the person Grammar model, a human body is composed of six parts and a possible occluder, and some parts (like head and torso) even have several subparts. All parts/subparts/occluder are movable and have two subtypes. This Grammar model has more adjustable structure than DPM, and thus has richer representation ability and gains better performance for human detection.

Besides, Bourdev *et al.* [3] proposed the Poselet Model for human detection, which is a two-layer feed-forward network based on the pattern of *poselet* activations. Poselets [4] represent the parts that are tightly clustered in both configuration space and appearance space. Recently, based on the region proposals and deep CNN features, Girshick *et al.* [14] constructed the R-CNN model, which have obtained impressive performances for object detection. Most important of all, though the performances of these models are all competitive, their output detections behave very differently, which means the combination or cross-fertilization of these models is possible or effective.

On the other hand, many efforts have also been made to reduce the detection time. Felzenszwalb *et al.* [11] accelerated the DPM model by more than one order of magnitude with the cascade models, similar to the work in [5, 28]. Kokkinos [20, 21] also speeded up the detection significantly with some well-designed search algorithms based on the bounds of part scores.

Motivated by these extracted features and detection models, we try to construct effective combination models with a group of reorganized features for human detection. Firstly, we extend the first-order gradients in the HOG features to a collection of gradients with three different orders, augmented with zero-order gradients and second-order gradients which correspond to the color information and the bar-shape information, respectively. After re-organizing them in a cell-based structure, we refer to these combination features as the **HOG-III features** (Histograms of Oriented Gradients with Three Orders). Then, we fuse different models with the new proposed ***weighted*-NMS algorithm** (Weighted Non-Maximum Suppression), which makes full use of the overlapped detections between different models to enhance the probable "true" activations as well as eliminate the redundant activations. We apply the HOG-III features and *weighted*-NMS fusion algorithm to (1) the Grammar model and Poselet model for human detection, and to (2) the deformable part-based model and deepCNN-based model for the detection of the whole VOC 20 object categories, and they lead to competitive improvements in both cases, which are indeed demonstrated by the experiments.

The rest of this paper is organized as follows. Section 2 introduces the computation procedure of HOG-III features. Section 3 presents the model fusion method based on the *weighted*-NMS algorithm. The experimental results are given in Section 4, and we conclude briefly in Section 5.

2. HOG-III features

As we know, the HOG features are based on the first-order gradients, then what are about the other k^{th}-order gradients, such as zero-order or second-order gradients? In digital images, the first-order gradients are related to the edge information. In fact, the other k^{th}-order gradients also contain some valuable information for detection.

Note: Here the k^{th}-order gradient means the maximization of k^{th}-order directional derivative, that is, the magnitude of k^{th}-order gradient is the maximum value of k^{th}-order directional derivative over all directions, and the orientation of k^{th}-order gradient is just the direction corresponding to the maximum value.

2.1. First-order gradients — HOG features

The HOG features were originally introduced by Dalal & Triggs [7]. To obtain them, we need to compute the *first-order* gradient at each pixel, aggregate the gradients to the corresponding cell, make a histogram on each cell, normalize the histogram along four directions, and finally concatenate all the normalized histograms to get the feature vector. However, we here use a modified HOG features suggested by Felzenszwal *et al.* [10], which mainly has two improvements from the original HOG: 1. The cell feature normalized along four directions are summed together, instead of concatenation, which reduces the dimensionality of feature vector to one-fourth; 2. A 4-dimensional texture feature vector is added for each cell. See [12] for the detailed description of the modified HOG features.

2.2. Zero-order gradients — Color features

The zero-order gradient of an RGB image is itself. Though the three RGB channels are descriptors of red, green and blue, respectively, their tri-tuple is not a good representation for feature extraction, due to the mixture of pure color information and intensity information. To separate these two kinds of information, we convert RGB to Hue-Saturation-Intensity (HSI) color space. As the intensity information has already been used in HOG features (the computation of the first-order gradient), to avoid redundant information, we only retain the hue and saturation channels in HSI space, skipping the intensity channel.

Figure 1 is the schematic diagram of HSI color space. It can be seen that, without regard to intensity channel, the hue and saturation channels form a disk-shape space, where hue corresponds to angle and saturation corresponds to radius. If we map hue and saturation to the orientation and magnitude of the first-order gradient in the HOG features, respectively, and follow the entire computation process of the HOG features, we can obtain the histograms of saturation over hue bins, which can describe the distribution of

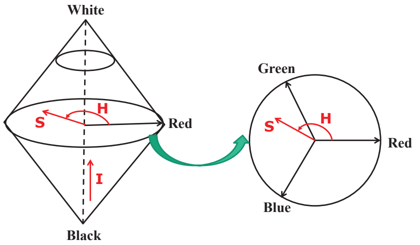


Figure 1. Schematic diagram of HSI color space

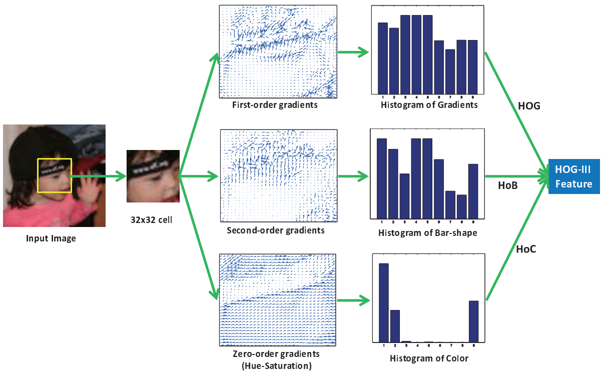


Figure 2. Construction procedure of the HOG-III features. Note that the 4D texture features are omitted in this figure.

color in the image. These **Histograms of Color (HoC)** features are also cell-based, similar to the structure of the HOG features.

2.3. Second-order gradients — Bar-shape features

As the zero-order gradients (with a particular transformation) are related to color information, then what do the second-order gradients mean? According to [6], the second-order gradients are related to bar-shape information. On one hand, the mammalian visual system seems to have bar-like receptive fields [17]. On the other hand, articulated objects like human bodies can also be modelled as connected bar-and-blob structures [18]. Therefore, the second-order gradients may also be helpful for human detection.

According to the definition of the k^{th}-order gradients, the second-order gradients can be computed as follow:

$$r^* = \max_{\theta} \frac{\partial^2 I}{\partial^2 \boldsymbol{u}}, \qquad \theta^* = \arg\max_{\theta} \frac{\partial^2 I}{\partial^2 \boldsymbol{u}} \tag{1}$$

where I is the intensity value of the input image, and $\boldsymbol{u} = (\cos\theta, \sin\theta)$ is the unit direction vector. By zeroing the derivative of the maximization item we can obtain

$$\theta^* = \frac{1}{2} \arctan\left(\frac{2 \cdot I_{xy}}{I_{xx} - I_{yy}}\right) \tag{2}$$

$$r^* = I_{xx} \cos^2\theta^* + 2I_{xy} \cos\theta^* \sin\theta^* + I_{yy} \sin^2\theta^* \tag{3}$$

where I_{xx}, I_{xy}, I_{yy} are the second-order derivatives of I with respect to the corresponding orientations.

After we get the second-order gradient $(r^*_{xy}, \theta^*_{xy})$ at each pixel (x, y), we can follow the entire computation process of the HOG features, just with the first-order gradients replaced by second-order gradients, and then we can obtain the **Histograms of Bar-shape (HoB)** features, which can describe the distribution of bar-shapes in the image and also have similar structure with HOG features.

2.4. Combination of HOG, HoC and HoB

The HoC, HOG and HoB features correspond to the zero-order, first-order and second-order gradients respectively, and they have similar structures (cell-based), so they can be easily concatenated together and thus form our **HOG-III features** (Histograms of Oriented Gradients with Three Orders). Generally, the number of histogram bins is set as 9 (just the same as that in HOG features). Thus for each cell, the HoC and HoB feature vectors are 13D (9 histogram features and 4 texture features), while the HOG vector is 31D due to the augmentation of 18 contrast-sensitive features (*i.e.*, 9 contrast-insensitive features, 18 contrast-sensitive features and 4 texture features). If we combine the HOG, HoC and HoB feature vectors directly, the dimensionality for each cell is $31 + 13 + 13 = 57$. However, experimental results (see Table 1(a)) show that, though the contrast-sensitive features in HOG are helpful for human detection when only the HOG features are used, they are not necessary for the HOG-III features. In fact, if we remove this 18D features from the HOG-III vector, the detection performance even has a slight improvement, and the training/test stage becomes faster. The reason may be that the HoC and HoB features compensate the remove of contrast-sensitive features in a certain way. Therefore, we choose to exclude these contrast-sensitive features from HOG-III features, and the final HOG-III feature vector is 39-dimensional for each cell, consisting of 13 HOG features, 13 HoC features and 13 HoB features. Figure 2 shows the construction procedure of HOG-III features.

Note that, features combined by color information, first and second order derivatives were also used in [26]. However, they were just a simple concatenation of the RGB values, the norm of first and second order derivatives on pixel-level. In this paper we use the gradients instead the original derivatives, and organize all the features in the cell-based histogram structure. These two differences provide us more robust and effective features for human detection.

3. *Weighted*-NMS based model fusion method

It is almost impossible for a single human detection model to detect all types of human bodies precisely. For example, the person Grammar model can not detect all types of human bodies (the recall cannot reach 100%), and it can only detect some particular types that are compatible with its framework. Every model has a bias, which is rooted in its own theoretic limitation. If we do not jump out of a theory framework, it will be difficult to overcome the bias. In the previous section we combine different features and get a satisfactory progress (refer to Table 1(a)). This inspires us that, different models, especially some complementary models, can also be combined together to cross-fertilize the whole detections and suppress their respective biases.

3.1. Calibration of confidence scores

In general, the output of a detection model for an input image can be formulated as $\{(p_i, s_i)\}_{i=1}^{n}$, where n is the number of detections, and (p_i, s_i) is the i-th detection. The position p_i is a bounding box denoted as *(xmin, ymin, xmax, ymax)*, while the score s_i denotes the confidence score of the i-th detection. Larger confidence score means more likely the detection being true positive.

However, the confidence scores from different models may have very different value ranges or magnitude scales, thus the same score may have different confidence levels in different models. For example, the score range of the Grammar model is $(-\infty, +\infty)$ with value 0 representing the half-confidence, while the score range of the Poselet model is $(0, +\infty)$ with value 0.5 representing the half-confidence, thus the value 0.2 is above half-confidence level in the Grammar model while below half-confidence level in the Poselet model. Therefore, to make the scores from different models comparable, we need to calibrate the scores into the same framework before the fusion.

If we test a detection model on the *validation* set, we can plot the **threshold-precision** curve by tuning the threshold score to output the detections with different confidence scores, and then collect a set of $(score, precision)$ tuples from the curve. It's reasonable to measure the actual confidence of a threshold $score$ according to its corresponding $precision$ [16, 8], that is, different scores from different models have the same confidence level, provided that they correspond to the same precision value. Thus we can obtain the score-calibration function from model B to model A as follows:

1. Test model B on the *validation* set, plot its threshold-precision curve, and collect a set of threshold *scores* $\{x_i\}_{i=0}^{10}$ whose corresponding precision values are $\{0.0,\ 0.1,\ 0.2,\ 0.3,\ \ldots,\ 0.9,\ 1.0\}$;

2. Do the same as above for model A and collect $\{y_i\}_{i=0}^{10}$;

3. Fit a transfer function $y = f(x)$ using $\{(x_i, y_i)\}_{i=0}^{10}$.

Once we get the transfer function $y = f(x)$, we can calibrate the score of model B with $\hat{s} = f(s)$, where s is the original score, and $\hat{s}$ is the calibrated score. After such calibration, the scores from model B and model A will have identical range and scale, and they are comparable in the framework of the model A.

3.2. Fusion of detections

Now let us discuss how to fuse model A and model B. For each image, we can obtain a set of detections $\{(p_i, s_i)\}$ from model A, and a set of detections $\{(p_j, s_j)\}$ from model B. Here we assume the scores $\{s_j\}$ from model B have already been calibrated into the framework of model A.

It is foreseeable that model A and model B may output many overlapped detections. Overlaps cause redundancies. If a group of overlapped detections correspond to the same "true object" (in fact this often happens), all these overlapped detections, except one, are redundant. If we merge $\{(p_i, s_i)\}$ and $\{(p_j, s_j)\}$ directly, we will get a high recall, but with a low precision. The high recall can be ascribed to the complementary differences between these two models, while the low precision may result from the redundancies between the two models.

In this way, the elimination of redundant detections is necessary. Non-Maximum Suppression (NMS) method is often used for redundancy elimination. However, general NMS algorithms are not suitable for this case. Assume that (p_{i_0}, s_{i_0}) is a detection from model A, $\{(p_{j_0}, s_{j_0})\}$ is detection from model B, and they have a big overlap. General NMS methods will simply delete the lower-scored detection, and retain the higher-scored detection with its score unchanged. As these two detections have a big overlap, they probably correspond to the same "hypothesized object". We believe that, if a "hypothesized object" can be detected by two different/complementary models, it is more likely to be a "true object". Therefore, it is reasonable to enhance the score of the retained detection, instead of keeping it unchanged.

Based on the above idea, we propose a modified NMS algorithm, denoted as ***weighted*-NMS**, to fuse these two models. The detailed fusion procedure based on *weighted*-NMS is shown in Algorithm 1. We firstly merge the detections from these two models and normalize their calibrated scores to the interval $[0, 1]$, corresponding to the range of confidence degree. After that, we take into account each detection greedily, from high score to low score. If $(p_h, \tilde{s}_h)$ is a high-scored detection, and there exists a lower-scored detection $(p_l, \tilde{s}_l)$ which has *enough* overlap with $(p_h, \tilde{s}_h)$, then $(p_l, \tilde{s}_l)$ will be deleted, **AND**, the score of p_l will be partially added to the score of p_h with a decay weight w_{hl} :

$$\tilde{s}_h \leftarrow \tilde{s}_h + w_{hl} \cdot \tilde{s}_l. \tag{4}$$

In other words, the detection p_l is merged into p_h, and its weighted score $w_{hl} \times \tilde{s}_l$ is absorbed into $\tilde{s}_h$ at the same time. By this way we can enhance the probable "true" detections as well as eliminate the redundant overlaps.

```
Input  : Detections of model $A$: $\{(p_i, s_i)\}$,
         Detections of model $B$: $\{(p_j, s_j)\}$ ;          // $s_j$ has already been calibrated
Output: Fused detections, i.e., the updated $U$
Merge $\{(p_i, s_i)\}_{i=1}^{M}$ and $\{(p_j, s_j)\}_{j=1}^{N}$ to a union set $U : \{p_k, s_k\}_{k=1}^{M+N}$;
Normalize the scores $s_k$ to interval $(0, 1)$ with the sigmoid function
$\tilde{s}_k = 1/\left(1 + \exp\{-\alpha \cdot (s_k - \beta)\}\right)$ ;          // $\alpha, \beta$ are fixed hyper-parameters
Sort the tuples $\{(p_k, \tilde{s}_k)\}_{k=1}^{M+N}$ in $U$ by descending order of $\tilde{s}_k$;
for $h \leftarrow 1$ to $end(U)$ do
    for $l \leftarrow h + 1$ to $end(U)$ do
        Compute $overlap(p_h, p_l) = \frac{area(p_h \bigcap p_l)}{area(p_h \bigcup p_l)}$;
        if $overlap(p_h, p_l) > T$ then          // $T$: threshold for overlapped detections
            $w_{hl} \leftarrow overlap(p_h, p_l)$ ;          // $w$: decay weight for score absorption
            $\tilde{s}_h \leftarrow \tilde{s}_h + w_{hl} \cdot \tilde{s}_l$;
            Delete $(p_l, \tilde{s}_l)$ from $U$;
        end
    end
end
return The updated detection set $U$
```

Algorithm 1: *Weighted*-NMS based fusion procedure for model A and model B.

Feature	AP %
HOG	45.8
HOG+HoC	48.0
HOG+HoB	48.7
HOG+HoC+HoB	50.1
HOG-III Feature	**51.3**

(a) Feature Combination

Model	AP %
Grammar	45.8
Poselet	47.0
G+P (None)	38.8
G+P (NMS)	46.7
G-P Model	**52.3**

(b) Model Combination

Table 1. Human detection results on PASCAL VOC2007 testset.

The decay weight should belong to $[0, 1]$, and in this paper we simply set it as the *overlap* between the two corresponding detections:

$$w_{hl} = overlap(p_h, p_l) = \frac{area(p_h \bigcap p_l)}{area(p_h \bigcup p_l)}. \tag{5}$$

Note that if we fix this weight as $w_{hl} \equiv 0$, then the *weighted*-NMS algorithm degenerates to general NMS algorithms.

So far, we have presented the detailed procedure of model combination, including the calibration step and the fusion step.

4. Experimental results

To test the performance of the HOG-III features and *weighted*-NMS based model fusion method, we conduct a series of experiments on the PASCAL VOC datasets. All the detectors are trained on the *train-val* set, and we use the *Average Precisions* (AP) on *test* set as the measurement of the detection performance.

4.1. Evaluation of HOG-III and weighted-NMS

A. Single test for feature combination

To test the effect of the HOG-III features described in Section 2, we conduct a set of comparative experiments on VOC2007 dataset, to show the performances for different feature combination methods. We apply all the features to the *person Grammar model* [15]. The results are shown in Table 1(a), where "HOG+HoC" is the direct mergence of HOG and HoC, "HOG+HoB" is the direct mergence of HOG and HoB, "HOG+HoC+HoB" is the direct mergence of HOG, HoC and HoB, while "HOG-III" denotes our proposed combination features.

From Table 1(a) we can see that, in comparison with the original HOG, the HOG+HoC and HOG+HoB have improved detection performance by 2.2% and 2.9%, respectively, and the direct combination HOG+HoC+HoB obtains an improvement of 4.3%. However, our combination features HOG-III, which exclude the contrast-sensitive features, can gain an improvement of 5.5%, which is even greater than the sum of the respective improvements for HOG+HoC and HOG+HoB. Considering that the dimensionality of HOG-III (39D) is less than that of HOG+HoC (44D), HOG+HoB (44D) and HOG+HoC+HoB (57D), it is clear that our combination features HOG-III are efficient and effective.

Note that the combinations without HOG features (like HoC+HoB or single use of HoC/HoB) have unsatisfactory performance, so we do not show them in the table. Certainly this also proves the importance of HOG features.

B. Single test for model combination

To test the effect of the *weighted*-NMS based model fusion method described in Section 3, we also conduct comparative experiments on VOC2007 dataset. We combine the person Grammar model [15] and Poselet model [3] with different fusion methods, to show the effects for different fusion methods. Both Grammar model and Poselet model use the *HOG features* [12].

Here the person Grammar model and Poselet model are selected due to their outstanding performances for human detection. Most important of all, as described in Section 1, the theoretical frameworks of Grammar model and Poselet model have a very big difference: the Poselet model is a two-layer network based on novelly defined poselets, while the Grammar model is based on deformable parts and occluder. Opportunity comes from the difference, which makes the combination and cross-fertilization possible for these two models.

In the calibration step, we use the piece-wise linear function to calibrate the scores of Poselet model into the framework of Grammar model. The hyper-parameters α and β in the fusion step are fixed to $\alpha = 2, \beta = 0$, which are optimized on the *validation* set. The threshold for overlap is set as $T = 0.5$ by convention (just the same as general NMS methods). Both the Grammar model and the Poselet model are non-maximum suppressed individually before they are combined. Note that, usually the general NMS is used to suppress the overlapped detections from the same model, while our *weighted*-NMS aims to enhance probable true activations from different models.

The results for model combination are shown in Table 1(b), where **"G-P Model"** denotes the combination model of Grammar model and Poslet model, obtained by our *weighted*-NMS fusion algorithm. "G+P (None)" is the direct mergence of these two models, while "G+P (NMS)" uses the general NMS method as the post-process.

From Table 1(b) we can see that, the naive combination (without any post-process) has a low AP, due to the large redundancies between two models. If we use the general NMS method as a post-process, the resultant model can not gain any substantial improvement either. In fact, the AP of the general NMS method is just the average of the APs for Grammar model and Poselet model. As for our *weighted*-NMS based G-P model, it gains an obvious outperformance, improving AP by 6.5% over the Grammar model and by 5.3% over the Poselet model. These great improvements show the effectiveness of our proposed *weighted*-NMS fusion algorithm.

4.2. Integrated frameworks for human detection

A. Performance of (G-P, HOG-III) framework

To obtain the best performance for human detection, we apply both the HOG-III features and the *weighted*-NMS fusion algorithm to the Grammar model and Poselet model, and conduct a group of comparative experiments with different *(feature, model)* arrangements. The *precision-recall* curves on VOC2007 testset are shown in Figure 3, where (Grammar, HOG-III) denotes the Grammar model with HOG-III features, (G-P, HOG) denotes the G-P model with HOG features, while **(G-P, HOG-III)**, the G-P model with HOG-III features, is an integrated framework which uses both the HOG-III features and the *weighted*-NMS fusion algorithm. The framework of (G-P, HOG-III) is shown in Figure 4, where our work is emphasized with *blue* color.

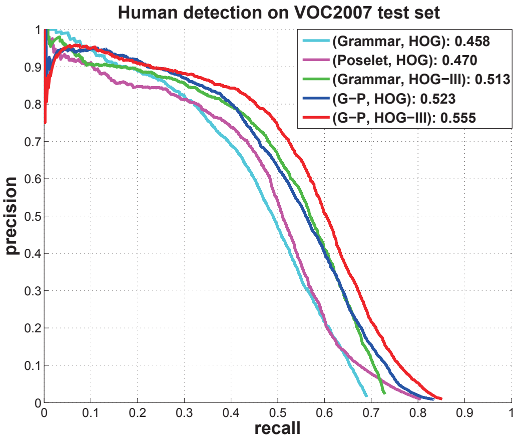
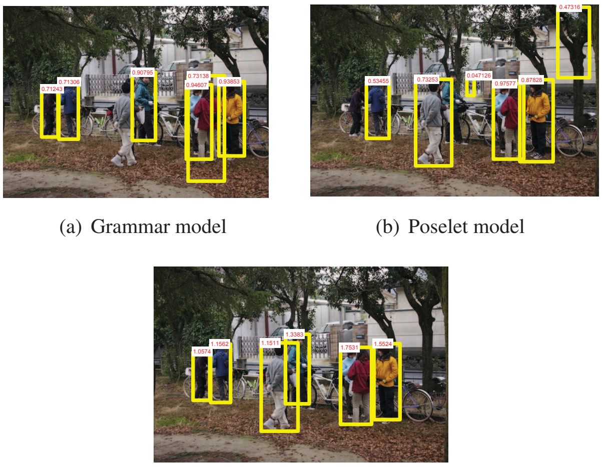


Figure 3. The precision-recall curves for different *(feature, model)* arrangements on VOC2007 testset.



(a) Grammar model

(b) Poselet model

(c) (G-P, HOG-III) model

Figure 5. The top *six* detections for Grammar model, Poselet model and (G-P, HOG-III) model. Note the confidence scores of Grammar/Poselet model have been calibrated and normalized.

From Figure 3 we can see that, compared with the original (Grammar, HOG) model, the single use of the HOG-III features improves AP by 5.5%, while the single use of the G-P model improves AP by 6.5%. However, if both the

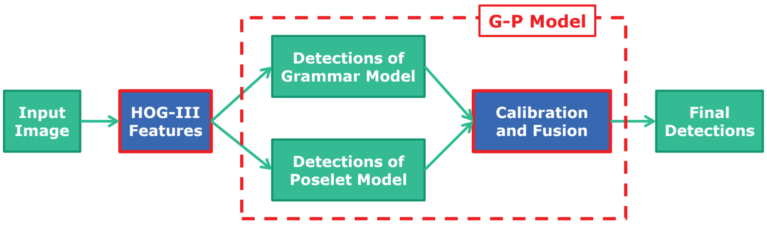


Figure 4. The integrated framework: G-P model with HOG-III features.

HOG-III features and G-P model are applied (*i.e.*, the integrated framework), the AP can be improved by 9.7%, which is a significant progress. Figure 5 is an example to show the top detections from different models.

To further verify the performance of our integrated framework (G-P, HOG-III), we also conduct some comparative experiments on VOC2010/2012 dataset. Table 2 shows the full results of our models as well as some related models which are competitive for *human detection*. All the results here are obtained without using large auxiliary datasets or contextual information about other object categories. From this table we can see that our models, including (Grammar, HOG-III), (G-P, HOG) and (G-P, HOG-III), outperform all the other related models. Especially, the integrated framework (G-P, HOG-III) has gained a substantial advantage over the best model excluding ours, *i.e.*, an improvement of 8.5% over Poselet model on VOC2007 testset, 8.0% over DDSSM on VOC2010 testset, and 8.9% over Poselet model on VOC2012 testset.

B. Analysis of detection time

The detection time of the integrate framework (G-P, HOG-III) is just the sum of the time for Grammar model and Poselet model with HOG-III features. In fact, the computation of HOG-III features costs nearly the same time as that for the original HOG features. It takes very little time to compute the extra features in HOG-III (*i.e.*, HoC and HoB), due to their similar structures and computing processes with HOG. Besides, the HOG-III features remove 18D contrast-sensitive features. In our experiments on VOC datasets, it takes about 2s per image for (Grammar, HOG-III) model.

Both the detection time and detection performance of the Poselet model vary greatly along with the *fineness* of the model (*e.g.*, the length of scanning pace, min/max size of scanning object, scale ratio of feature pyramid). In fact, the implementation of the Poselet model in [2] can obtain an AP of 47.0% (on VOC2007 testset) with 10s per image under a high fineness, or an AP of 37.2% with only 1s per image under a low fineness. However, when we combine the Poselet model and Grammar model together with our *weighted*-NMS based fusion method, the fineness of the Poselet model has negligible effect on the performance of our G-P model. For example, on VOC2007 testset, our G-P model can obtain an AP of 55.5% with about 12s per image (high fineness for the Poselet model), or an AP of 54.5% with about 3s per image (low fineness for the Poselet model). Therefor, in the framework of the G-P model, we usually use a low fineness for the Poselet model in practice.

Currently, the speed of our integrated framework is mainly limited by the models we used. The integrated framework can be speed up by running these two models in parallel. Besides, the Grammar model or Poselet model can also be accelerated by some well-designed implementations, like the cascaded structure [5, 11], the Branch-and-Bound search algorithm [20, 21], and so on.

Note that the detection time is obtained in our personal computer, with 3.2GHz 4-core Intel Core CPU, 8G Memory, Linux-3.5 OS. Besides, we make use of the parallel programming in MATLAB R2012b, and 4 workers are opened in parallel when detecting on test set.

C. Comparison & application to deep learning models

Note that recently Girshick *et al.* [14] proposed the R-CNN model, which has obtained impressive performances for object detection based on region proposals and deep CNN features. For example, on VOC2010 testset, their AP is 53.6% for person class. Further, after utilizing the bounding-box regression, their AP increases to 58.1%, which is then slightly higher than ours 57.2%. Though some improvements can also be expected if we apply similar bounding-box regression to our model, we need to state that, their results can not be compared with ours directly. Due to the usage of deep learning, the R-CNN model needs a large *auxiliary* dataset (*i.e.*, ImageNet) in its pre-training stage. Besides, the training of R-CNN requires high-level hardware conditions (GPU, large memory/disk space, etc) and large amount of time (more than ten times longer than our model). As for prediction time, our model needs only 3s per image (see the previous section), while their model needs more than one minute per image (in CPU mode).

However, without regard to the large auxiliary datasets, high-level hardware conditions or the long training & prediction time, we can also fuse the R-CNN model with the Grammar model by using our *weighted*-NMS algorithm. The resultant fusion model shows very good performance. For example, on VOC2007 testset, the AP of person class

Model	VOC2007	VOC2010	VOC2012
Grammar [15]	45.8	47.6	47.9
Poselet [3]	47.0	48.5	48.1
LSVM-MDPM(V5) [12]	43.2	45.2	44.5
Boosted-HOG-LBP [32]	44.6	46.5	–
HSC [25]	41.4	–	–
DDSMM [33]	44.8	49.2	–
CN-HOG [19]	44.0	43.3	–
Regionlet [31]	43.4	43.5	–
(Grammar, HOG-III)	51.3	52.2	52.1
(G-P, HOG)	52.3	54.1	53.7
(G-P, HOG-III)	**55.5**	**57.2**	**57.0**

Table 2. Full results (AP%) on PASCAL VOC dataset for person-class. All the results here are obtained without using large auxiliary datasets or contextual information about other categories. "LSVM-MDPM(V5)" [12] is the popular MDPMs proposed by Felzenszwalb. "Boosted-HOG-LBP" [32] is a boosted local structured HOG-LBP based object detector, which got the highest mean-AP for the whole twenty object categories in VOC2010 competition [9]. "HSC" [25] denotes the Histogram of Sparse Codes. "DDSMM" [33] is a variation of part based models with data decomposition and spatial mixture modeling method. "CN-HOG" [19] is a HOG variant combined with color attributes. "Regionlet" [31] is a cascaded boosting model with *regionlet* features. Note that the results for some models remain unknown (marked with "–" in the table) due to the lack of their implementation codes.

is 51.3% for Grammar(HOG-III), 58.7% for R-CNN, and 65.2% for the fusion model of Grammar and R-CNN, which is indeed a significant improvement.

4.3. Extension to the whole VOC 20 classes

Our initial aim is just to find good features or detection models for the *human* detection task, so the proposed HOG-III features and *weighted*-NMS based fusion method are initially designed and evaluated only for *person* class. The Grammar model and Poselet model are chosen due to their outstanding performances for person class.

Certainly, these methods may also be valuable for other object detection tasks. To investigate this generalization problem, we extend the HOG-III features and *weighted*-NMS fusion algorithm to the detection of the whole VOC 20 object categories. We use DPM [12] and R-CNN [14] for experiments, as they are representatives of non-deep detection models and deep detection models, and both of them are applicable to the whole object categories.

First, the HOG-III features still show good performances on the whole 20 classes, though not as impressive as that for person class. For example, in the framework of DPM, the mean AP on VOC2007 testset is 33.7% for HOG [12], 34.3% for HOG-LBP [32], 34.3% for HSC [25], 34.8% for CN-HOG [19], while 35.0% for the proposed HOG-III features.

Second, we fuse the DPM and R-CNN with the *weighted*-NMS algorithm and test it on the whole 20 classes. The fusion model also gains competitive improvements. Specifically, the mean AP on VOC2007 testset is 33.7% for DPM, 58.4% for R-CNN, while 60.5% for the fusion model of DPM and R-CNN.

5. Conclusion

We have investigated the combination features/models for human detection and made two contributions. First, we introduce the HOG-III features which consist of the HOG features, color features (HoC) and bar-shape features (HoB). Second, we propose the new and effective *weighted*-NMS algorithm, leading to the construction of several fusion models, including the G-P (Grammar+Poselet) and G-R (Grammar+RCNN) for person class, and the D-R (DPM+RCNN) for the whole 20 classes. The experiments on PASCAL VOC datasets have demonstrated that, both the HOG-III features and the *weighted*-NMS fusion algorithm can boost the performance significantly for human detection, as well as gain competitive improvements for the detection of the whole VOC classes.

In the future we will try to fuse the HOG, HoC and HoB with more effective methods like Boosting [32], Multiple Kernel Learning [27], *etc*. As for the *weighted*-NMS based fusion method, not limited to the Grammar model, Poselet model, R-CNN or DPM, it can be extended to the fusion other complementary models or multiple models (more than two). Our future work will focus on these issues.

Acknowledgments

This work was supported by the Natural Science Foundation of China for Grant 61171138.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, volume 1, pages 454–461. IEEE, 2001. 1

[2] L. Bourdev. Poselets and their applications in high-level computer vision. http://www.cs.berkeley.edu/~lbourdev/poselets. 7

[3] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *European Conf. on Computer Vision (ECCV)*, pages 168–181. Springer, 2010. 2, 6, 8

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, pages 1365–1372. IEEE, 2009. 2

[5] Y.-T. Chen and C.-S. Chen. Fast human detection using a novel boosted cascading structure with meta stages. *IEEE Trans. on Image Processing (TIP)*, 17(8):1452–1464, 2008. 2, 7

[6] N. Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006. 3

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005. 1, 2

[8] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem. Learning collections of part models for object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013. 4

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge workshop 2010. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/workshop/index.html. 8

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 1, 2

[11] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248. IEEE, 2010. 2, 7

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010. 1, 2, 6, 8

[13] P. F. Felzenszwalb and D. McAllester. Object detection grammars. In *IEEE Int'l Conf. on Computer Vision (ICCV) Workshops*, page 691. IEEE, 2011. 1

[14] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 7, 8

[15] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 442–450, 2011. 1, 5, 6, 8

[16] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014. 4

[17] D. H. Hubel. *Eye, brain, and vision.* Scientific American Library/Scientific American Books, 1995. 3

[18] S. Ioffe and D. Forsyth. Mixtures of trees for object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 180–185. IEEE, 2001. 3

[19] F. S. Khan, R. M. Anwer, J. v. d. Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012. 8

[20] I. Kokkinos. Rapid deformable object detection using dual-tree branch-and-bound. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2681–2689, 2011. 2, 7

[21] I. Kokkinos. Bounding part scores for rapid detection with deformable part models. In *European Conf. on Computer Vision (ECCV) Workshops and Demonstrations*, pages 41–50. Springer, 2012. 2, 7

[22] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE Int'l Conf. on Image Processing (ICIP)*, volume 1, pages 900–903. IEEE, 2002. 1

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 1

[24] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *IEEE Int'l Conf. on Pattern Recognition (ICPR)*, volume 1, pages 582–585. IEEE, 1994. 1

[25] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013. 8

[26] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conf. on Computer Vision (ECCV) Workshops and Demonstrations*. Springer, 2006. 3

[27] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *IEEE Int'l Conf. on Computer Vision (ICCV)*. IEEE, 2009. 8

[28] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004. 2

[29] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, pages 734–741. IEEE, 2003. 1

[30] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *IEEE Int'l Conf. on Computer Vision (ICCV)*, pages 32–39. IEEE, 2009. 1

[31] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *IEEE Int'l Conf. on Computer Vision (ICCV)*. IEEE, 2013. 8

[32] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured HOG-LBP for object localization. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1393–1400. IEEE, 2011. 1, 8

[33] J. Zhang, Y. Huang, K. Huang, Z. Wu, and T. Tan. Data decomposition and spatial mixture modeling for part based model. In *Asian Conf. on Computer Vision (ACCV)*, pages 123–137. Springer, 2013. 8