Real-time Accurate Object Detection using Multiple Resolutions

Wei Zhang[†] Gregory Zelinsky[‡] Dimitris Samaras[†]
Department of Computer Science[†] Department of Psychology[‡]
Stony Brook University, US
{wzhang, samaras}@cs.sunysb.edu[†] Gregory.Zelinsky@stonybrook.edu[‡]

Abstract

We propose a multi-resolution framework inspired by human visual search for general object detection. Different resolutions are represented using a coarse-to-fine feature hierarchy. During detection, the lower resolution features are initially used to reject the majority of negative windows at relatively low cost, leaving a relatively small number of windows to be processed in higher resolutions. This enables the use of computationally more expensive higher resolution features to achieve high detection accuracy. We applied this framework on Histograms of Oriented Gradient (HOG) features for object detection. Our multi-resolution detector produced better performance for pedestrian detection than state-of-the-art methods [7], and was faster during both training and testing. Testing our method on motorbikes and cars from the VOC database revealed similar improvements in both speed and accuracy, suggesting that our approach is suitable for realtime general object detection applications.

1.Introduction

In this paper, we propose a general real-time object detection method with state-of-the-art detection rates. Object detection (or object localization) is the problem of finding the positions of all target objects in an image. More specifically, the goal is to find the bounding box for each target object. One common approach is to use a sliding window to scan the image exhaustively in scale-space, and classify each window individually [25, 22, 30, 28, 31]. This approach reduces the problem to a set of patch or image classification problems, for which only a binary classifier is needed to indicate whether a target is present or absent from a given window. However, due to the large number of possible target locations in an image, a classifier may need to be applied on the order of 10^4 to 10^5 times (depending on the image resolution and the density of the sliding windows), which creates two other problems for this approach. First, how does one keep a good detection rate while maintaining a reasonable false positive rate? Although a single-window classification can be extremely accurate, even small errors can accumulate quickly when the number of windows is large. Second, without sacrificing accuracy, how can objects be detected fast, or even in real-time?

Real-time accurate object detection is necessary in many applications. In computer vision, it can be used in object tracking to recover a lost tracker, or even to do tracking by detection on each frame. However, current methods, such as cascaded classifiers, require significant training time and cannot be used in real time. For interactive or mobile robotics applications, in which target object localization is a central component, real-time object detection is again desirable. Object detection in robotics has been limited to simple, or easily segmented targets: In [21], Mitri *et al*. presented a biologically inspired attentional system (see [15]) for ball recognition. Fasola and Veloso used the segmented color image to hypothesize plausible locations, then used grayscale image to further classify the initial hypotheses in a robotics application [10]. Current object detection methods in computer vision are either not real-time or not easily generalizable. For example, a cascaded classifier requires significant training time, which would require a robot to be continuously retrained for different tasks. The real-time general object detection method in this paper is fast in training and more accurate in detection. The following will be a review of recent object detection methods, upon which our method builds.

State-of-the-art object detection methods recognize specific categories such as faces, pedestrians and cars with high accuracy [25, 22, 17]. In other work inspired by the success of Scale Invariant Feature Transform (SIFT), Dalal and Triggs proposed using Histogram of Oriented Gradients (HOG) features to solve the pedestrian detection problem [7]. HOG features were first extracted from smaller spatial regions, then normalized over larger regions called "blocks". The concatenation of all block features was used as the feature descriptor for the whole detection window. Those feature vectors were classified by SVMs.

Recent object detection approaches have used a variety of features and methods, such as: a model integrating both general and discriminative methods [13], boundary-fragment-model using local contour features [23, 24], chamfer distance matching on edges [29], contour segment network [11] and shared hierarchical codebook [20]. These new approaches are able to detect more general objects, e.g., motorbikes, bicycles and cows.

Despite the advantages of the above methods, they are in general not fast enough for real-time applications. Viola and Jones presented the first real-time system for face detection, which used simple rectangular features combined with AdaBoost for feature selection [30]. They proposed using the integral image structure and the use of cascading classifiers to achieve real-time detection. The method was also applied to pedestrian detection by introducing motion features [31]. More recently, Zhu *et al*. extracted HOG features from the integral image using a cascaded classifier framework [32]. This approach was able to detect pedestrians in real-time with comparable performance as in [7], at the expense of greatly increased training time, typically on the order of days or even weeks [30, 32].

We propose a multi-resolution framework for general object detection that can be applied in both real-time and with minimal training time, comparable to that of single resolution methods. Our approach is modeled after biological vision and human search behavior in that humans guide their eye movements during object detection based on a low-resolution description of the target [27]. Multi-resolution search is necessitated in humans by the highly non-linear distribution of photoreceptors on the retina, which can be modeled by a multi-resolution pyramid in which pixels distant from the fovea are sampled more coarsely than those near the fovea [26]. As a result of this coarse-to-fine search dynamic, a great many target candidates can be excluded based solely on their low-resolution descriptions, thereby saving neural computation time and producing highly efficient search behavior.

Our multi-resolution approach naturally defines a coarse-to-fine feature hierarchy and strategy for object detection. In comparison, in a general cascaded classifier, this feature hierarchy is derived from the training process, which explains the considerable time needed for training. Moreover, we found through our experiments that the predefined feature hierarchy performed even better than the one automatically selected by AdaBoost. Tree-structured coarse-to-fine feature hierarchies derived from training data are also used in [8, 1] for object detection. Fleuret and Geman adopted a coarse-to-fine scheme for multi-pose face detection [12], and later Amit et al. used similar idea for general multiclass shape detection [2]. Different from these approaches, multi-resolution representation is intrinsic with an image and thus can be applied to any feature type. Schneiderman and Kanade [28] utilized wavelets which are multi-resolution by nature, without further exploring the issue. Multi-resolution is also related to pyramid matching [14, 16], where matching scores from different levels of the pyramid are combined together. In our method, each level is processed hierarchically, in a low-to-high resolution sequence.

The paper is organized as follows: we first present our general multi-resolution framework, then give details of the training and detection algorithms. We then apply it to HOG features for general object detection. In our experiments, we compare the performance in both accuracy and speed with [7] for pedestrian detection. We further perform experiments on the VOC2006 challenge database for motorbikes and cars. For all the cases, our method improved both accuracy and detection speed compared to [7] with almost the same training speed. The detection speed is at $25 \sim 30$ fps for images of resolution 320×240.

2.Methodology

In this section we will introduce the multiple resolution framework and compare it to a general cascaded classifier, then give detailed algorithms for both training and detection. Finally, we will use HOG features [7] to apply our method to general object detection.

2.1.Multiple resolution framework

The general concept of object detection using multi-resolution is shown in Fig. 1. The framework encodes both resolution and scale spaces in 2D coordinate system. Along the vertical axis in scale space, the image gets downsampled in order to use a fixed size detection window to locate objects of larger scales. Detection in different scales are independent and can be processed in parallel for faster speed. Along the horizontal axis in resolution space, any detection window is classified hierarchically from its lowest resolution to full resolution. [1] A window rejected in lower resolution will not be passed to any higher resolutions. In general, there will be a computational advantage to analyzing features and classifiers at low resolutions. This is because the majority of detection windows can be excluded at a low resolution, leaving a fewer number of windows to be classified at the higher resolution. Additionally, finer and more expensive features can be used at the higher resolution, which would not be computationally feasible using a single resolution approach for real-time applications. A well designed multi-resolution object detection system can approach the detection speed of the lowest resolution, often with better recognition rates than a single full resolution classifier.

Figure 2 shows why multi-resolution is useful for object detection. A Canny edge detector was applied to images from the TU Darmstadt database at different resolutions. [2]

[1] Note there is no redundant computation over the two axes. It is possible for two images in the 2D space to be identical (i.e., same downsampling ratio), but the detection windows have different sizes and thus evaluate different features.

[2] Available at: http://www.mis.informatik.tu-darmstadt.de/leibe

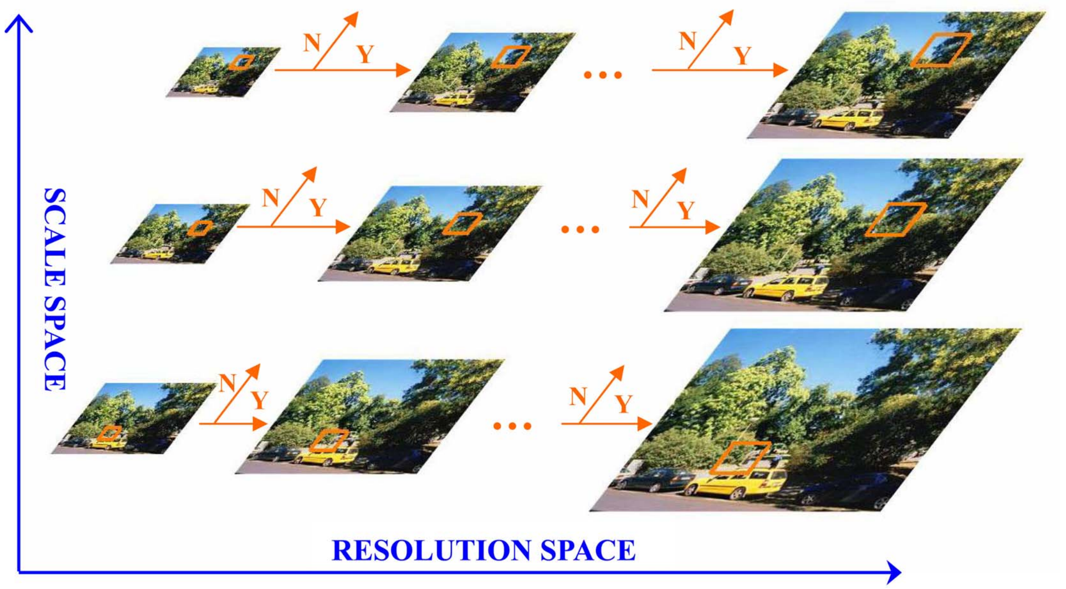

Figure 1. Multiple resolution framework for object detection. 'N' indicates rejected windows that will not be checked further. 'Y' indicates windows that are accepted by the classifier and will be passed to the classifier at the next higher resolution. The method works in both scale and resolution spaces. Detection windows at different scales are independently processed from lowest resolution to full resolution in the hierarchy, with windows being classified at a higher resolution only after they have been accepted by a lower resolution classifier.

Images in lower resolutions were created by downsampling and then upsampling to the original size for easier comparison. For the motorbikes and cows categories, we picked two samples from each and computed the shape context matching distances [3] shown in the figure. In general, the similarity between pairs of images was increased (the matching distance decreased) at the lower resolutions (although the highest similarity score for the two cows was not found at the lowest resolution due to excessive downsampling). Given this resolution, it is expected that an object detection method using edge features can benefit from the multi-resolution framework. For example, Opelt *et al*. [23] used a boundary fragment model to create a visual shape alphabet for object detection. Introducing a hierarchical multi-resolution shape alphabet might further improve the results.

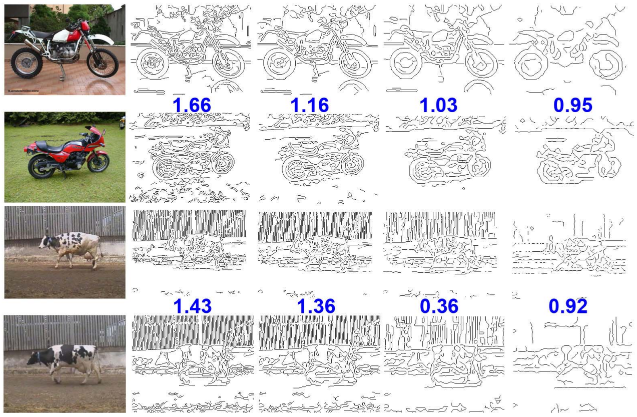

Figure 2. Sample images from TUD database, with edge detected images at different resolutions. From left to right in each row, resolution decreases by a ratio of 0.5 starting from the first edge image from full resolution. The numbers between pair of images are shape context matching distances.

2.2.Training/detection algorithms

We now formulate the training and detection algorithms of our framework. For a typical object detection system, the training set is composed of normalized image patches. Negative patches were scanned from images that do not contain any target object instances. Negative training patches in higher resolutions are bootstrapped using hard samples that can't be classified correctly in any lower resolutions. Suppose we use R resolutions, where $r = 1, ..., R$ from the lowest resolution to the full resolution. Assume the step of downsampling ratio is α. For each resolution r, we define the training set as $T_r = \{(I_r(i), l_r(i)), i = 1, ..., N_r\}$, where $I_r(i)$ is the image patch, $l_r(i) \in \{1, -1\}$ is the associated label and N_r is the number of patches. Each patch is described by a set of features $F_r(i)$, noting that feature representations in different resolutions are different. Table 1 shows the training algorithm. The training starts from the lowest resolution, then loops between bootstrap and training towards higher resolutions until the full resolution. The output of the training is a hierarchical classifier with components from each resolution. The optimal setting of learning goals (detection rate or false positive rate) should further improve performance [19, 4], and will be explored in future work.

As described in Section 1, object detection in an image is essentially a window classification problem at each possible location in multi-scale space. The number of scales depends

```
• Create the lowest resolution training set $T_1$ by downsampling the full resolution image patches with a ratio of $\alpha^{R-1}$. Train a classifier $C_1$ at the lowest resolution.
• For each resolution $r = 2, ..., R$,
    – Depending on the application, adjust the threshold $\theta_{r-1}$ of the classifier $C_{r-1}$ s.t. false positive rate $E_{r-1} < \mu_{r-1}$ or detection rate $D_{r-1} > \nu_{r-1}$, where $\mu_{r-1}$ and $\nu_{r-1}$ are user specified learning goals.
    – Bootstrap: scan the collection of negative images with resolution $r$, use the detection algorithm in Table 2 to find the hard negative patches s.t. $H_{r-1}(P) > \theta_{r-1}$.
    – Train a classifier $C_r$ on the training set $T_r$. Image patches are downsampled with a ratio of $\alpha^{R-r}$.
• Output the final hierarchial multiple resolution classifier $H_R = \{(C_t, \theta_t), t = 1, ..., R\}$.
```

Table 1. The training algorithm of a multiple resolution approach.

```
• Initialize status of all windows in resolution 1, set $\gamma_{1,s}(k)$ to be 1.
• For each resolution $r = 1, ..., R$,
    – For each scale $s = 1, ..., S$,
        ∗ If all windows are marked as rejected, continue to next scale.
        ∗ Downsample the image with a ratio of $\alpha^{R-r}\beta^{S-s}$. Apply the classifier $C_r$ to each window $P$ where $\gamma_{r,s}(k) = 1$, set the status to $-1$ (rejected) if $C_r(P) < \theta_r$.
        ∗ If $r \neq R$, set status for $r + 1$ resolution $\gamma_{r+1,s}(k) = \gamma_{r,s}(k)$.
• Output all windows where $\gamma_{R,s}(k) = 1, s = 1, ..., S$.
• Postprocess detection results using mode seeking methods (e.g. mean shift [5]).
```

Table 2. The detection algorithm of a multiple resolution approach.

on the sizes of the image and the detection window with the downsampling ratio. Suppose that the image is downsampled with a ratio of β for each scale up, and that we have S scales $s = 1, ..., S$, for larger to smaller objects (Note: a larger scale object means we need to downsample the image more.). Suppose the window size at each resolution is $(w_r, h_r) = (w, h)/\alpha^{R-r}$, and that the size is fixed across all scales. For each resolution r and scale s, a list of status of detection windows $\gamma_{r,s}(k) \in \{1, -1\}$ is kept. The status is initialized to be 1, and set to -1 if the window is rejected by a classifier. A detection window in lower resolution can correspond to one or multiple windows in higher resolution, depending on the detection window density. Table 2 shows the object detection algorithm on a test image.

2.3.Application to HOG

Dalal and Triggs proposed the Histogram of oriented Gradients (HOG) feature for pedestrian detection (HOG method) [7]. In their approach, pixels are first grouped into smaller spatial units called "cells". For each cell, a histogram feature on gradients orientations is extracted. The magnitude of the gradient is used as the weight for voting into the histogram. Multiple cells form larger spatial units called "blocks". The descriptor of each block is the concatenation of all cell features. A number of strategies are crucial for the final performance: when computing the HOG for each cell, a Gaussian weighting window is applied to each block; each pixel's gradient votes into the histogram using trilinear interpolation in both spatial and orientation dimensions; the block feature is normalized (L2-hys: Lowe style clipped L2 norm [18]) for invariance to illumination. Inside each detection window, densely sampled and overlapping blocks produce redundant descriptors, which is important for better performance. The descriptor of a window is again the concatenation of all block features. Finally, a linear Support Vector Machine (SVM) is used to classify individual detection windows. The system also has a retraining process that uses hard negative samples from the initial classifier as part of the new training set. This was shown to significantly improve the classifier's performance [7].

In our multi-resolution approach based on the above method, we used blocks of varying sizes to capture features in different spatial frequencies. Low frequency features were extracted in low resolutions for efficiency; at higher resolutions, we used smaller blocks but higher sampling density and more orientation bins (see Section 3 for our detailed parameters settings). An important consequence of the feature hierarchy is that, both more-global (low resolution) and more-local (high resolution) features are encoded.

3.Experimental results

In this section we discuss applications of our multiresolution method (using HOG features) to the detection of pedestrians and other object categories. Our first comparison used the INRIA pedestrian database, with both HOG and cascaded-HOG methods. The detector speed was greatly improved relative to the HOG method, and approached the speed of cascaded-HOG with significantly less training time.

3.1. Pedestrians

We experimented on the INRIA pedestrian database. [3] The database contains 1208 pedestrians for training and 563 for testing. Positive samples are cropped image patches of size 64 × 128 plus margins on each side. Left-right reflections of these images were also added to the database. We used the negative images that are supplied with the database. For comparison purposes, we used the same training/testing sets as in [7, 32]. Similar to [7], we selected the parameters that optimize detection. We used the same parameters in 3 additional experiments using different random training/testing splits of the whole dataset to avoid the potential problem of "overtraining". At the False Positive Per Window (FPPW) level of 10^{-4}, average detection rate of the HOG method and multi-resolution was 88.3%, 88.1% respectively. The performance of multi-resolution approach can be improved by optimal setting of leaning goals in each resolution using a validation set [19, 4].

Following Dalal and Trigg's experiments using HOG features, we: (1) took the maximum gradients of RGB channels, (2) applied a Gaussian weighting window centered at each block, (3) trilinear interpolation of gradient magnitudes, and (4) used L2-hys normalization for each block. Other more specific parameters of our multi-resolution approach are provided in Table 3. We used 4 resolutions for pedestrian detection, the lowest resolution image was only $1/8$ of the original size on each side. From lower to higher resolution, the blocks became more spatially local - the sizes were 48 × 48, 32 × 32, 24 × 24 and 16 × 16 image patches when projecting to the full resolution. Moreover, we increased the sampling densities of blocks, and the number of bins in HOG when the resolution increases. To compute such a detailed feature is expensive, but in practice we only need to detect about 0.1% of the total windows in the highest resolution. Our detailed performance analysis for both accuracy and speed follows.

3.1.1. Detection accuracy

In Fig. 3 we plot the Detection Error Trade-off (DET) curves shown . A DET curve reveals how detection rate (miss rate) changes with the rate of FPPW. Given an image of 320×240, a sparse scan (an 8 pixel spacing between windows) will generate about 10^3 windows. The results from HOG were obtained by running the original binaries provided by the authors. [4] For our multi-resolution approach, an FPPW level less than 10^{-3} was achieved with detection rate of 94% without using any full resolution information. The multi-resolution method had worse performance when the FPPW level was above 10^{-3} with the absence of full resolution features, but few applications could tolerate such a high FPPW rate. The multi-resolution approach always performed better than the HOG methods when full resolution features were used. For example, at the level of 10^{-4} FPPW, our detection rate was 91.0% (miss rate 9%) compared to 88.7% obtained using the HOG method, a reduction of more than 20% of the miss rate. By setting the detection rate at the same level, the FPPW was reduced to 5.0×10^{-5} for the multi-resolution method: a 50% reduction in the total number of false positive windows. This is especially important for applications like robotics, where the FPPW rate should be limited to as low as possible given that false positive are usually more costly than misses.

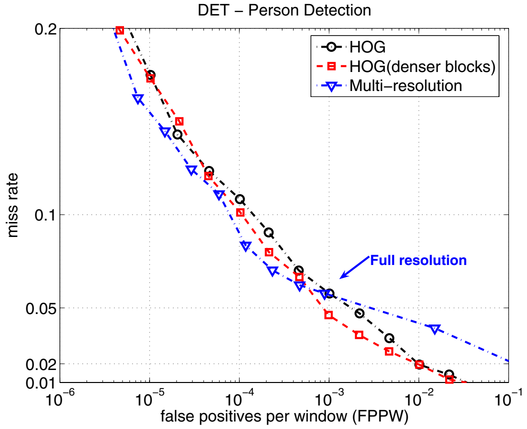

Figure 3. Comparison between HOG and multi-resolution methods using Detection Error Trade-off curves. 'HOG': results using original settings; 'HOG(denser blocks)': results using our settings in full resolution with more densely sampled blocks; 'Multi-resolution': results from multi-resolution. The blue arrow shows the point where we start using full resolution.

	1	2	3	4
Detection rate (%)	98.6	96.1	94.1	
FPPW	0.239	0.015	0.0009	
Detection time (%)	24.1	51.6	77.3	100.0

Table 4. Cumulative detection accuracy and speed at each resolution level.

Figure 4 shows some of our detection results, which were obtained with only window classification using the multi-resolution classifier. We did not any apply further postprocessing, thus leaving multiple detections for some of the pedestrians in the figure. A mean-shift based method can be used to effectively suppress duplicate detections by clustering [5].

Table 4 shows the cumulative performance at each resolution. The lowest resolution classifier was able to reject almost 80% of the windows with a detection rate of 98.6%. When combined, the first 3 low resolutions rejected more than 99.9% of the windows, with a detection rate of almost 94%. Given this, the method can afford finer but more expensive features in the full resolution. We sampled 195 blocks compared with 105 in the HOG method, and

[3] Available at: http://pascal.inrialpes.fr/data/human/

[4] Available at: http://pascal.inrialpes.fr/soft/olt/OLT.tar.gz

Resolution	cell size	block size	detector size	#orientations	block stride	detector stride
1	(3, 3)	(6, 6)	(8, 16)	9	(2, 2)	(1, 1)
2	(4, 4)	(8, 8)	(16, 32)	9	(2, 4)	(2, 2)
3	(6, 6)	(12, 12)	(32, 64)	9	(4, 4)	(4, 4)
4	(8, 8)	(16, 16)	(64, 128)	18	(4, 8)	(8, 8)

Table 3. Parameters in each resolution level of our pedestrian detection system. All parameters except the number of orientation bins were expressed as (width, height) pair in pixels.

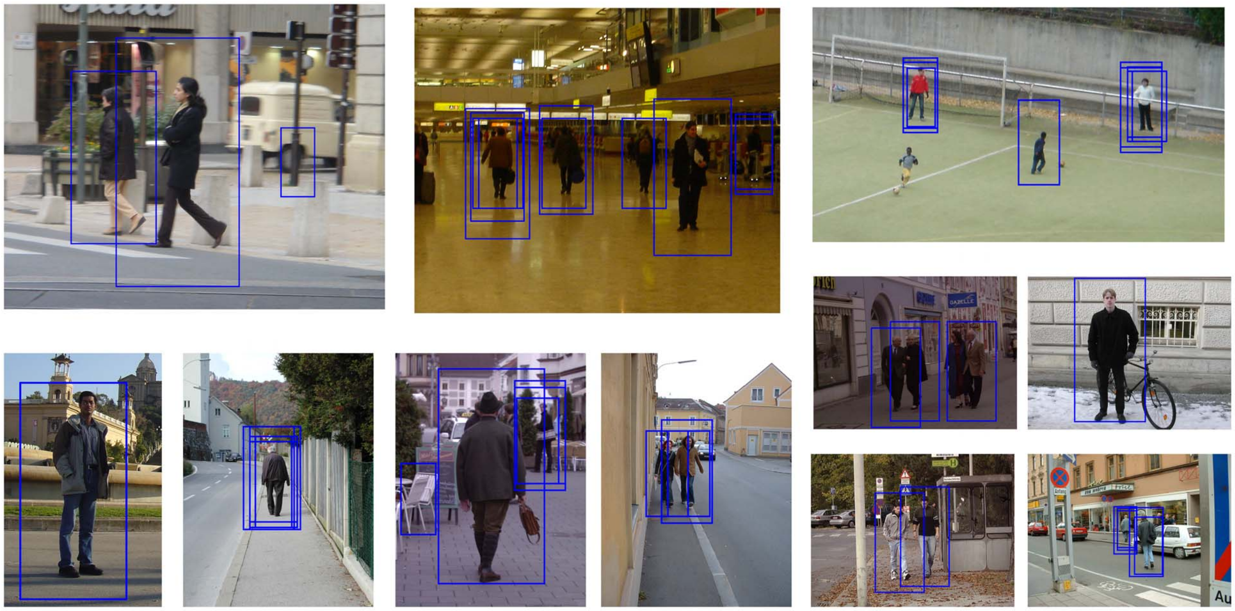

Figure 4. Detection results on some sample images. No postprocessing was applied. The FPPW level was set to 10^{-5}.

we used 18 orientation bins versus 9. The more redundant information encoded in the feature vector was useful for recognition, since the full resolution classifier was dealing with much harder cases. Two experiments were conducted to verify this claim. First, for the multi-resolution classifier, we kept the settings of the first 3 resolutions, and used the original settings [7] in full resolution, the detection dropped to 89.3% when FPPW rate was 10^{-4}. Second, we used our finer feature settings with the HOG method (see Fig. 3). The multi-resolution still outperformed HOG method, if only slightly. Moreover, for the case of HOG method, both training and testing times were more than doubled, with 4 times the memory cost (more than 2.5G in retraining). Because fewer training samples are processed using our multi-resolution method, this approach yields affordable time/memory costs.

3.1.2. Training/testing speed

The training time of multi-resolution was about the same as the HOG method. Both took less than one hour for training, while the cascaded-HOG took a couple of days [32]. Compared with the HOG method, multi-resolution needs to train 3 more classifiers, but the bootstrap (retraining for the case of HOG) was faster. Compared with the cascaded-HOG, both methods introduced more features, but in the multi-resolution approach features were defined as a hierarchy - meaning only the corresponding feature set needs to be evaluated at any level. In a cascaded classifier, all features must be evaluated for each level of the cascade, which is a very time consuming process.

The HOG method executed at about $3 \sim 4$ fps; the multi-resolution classifier executed at 25fps for 320×240 images with a sparse scan (detector window was spaced by 8 pixels in each direction). Both methods ran on machines with the same configuration: Xeon Dual-processor 3.6GHZ CPU with 4G memory. Also, the multi-resolution framework can flexibly trade-off between detection time and accuracy. When we allow low resolutions to reject more windows, we can detect at 30fps with 89.6% detection rate when FPPW was 10^{-4}. A cascaded classifier could not easily make such an adjustment due to the training time. A detailed analysis of detection time distribution w.r.t resolutions is shown in Table 4. The time spent at each resolution was about 25% of the total time (we removed the overheads of loading images in this analysis). The classifier achieved a good balance between speed and accuracy. Although a hierarchical classifier, the multi-resolution method does not gain detection time at the expense of greatly increased training time.

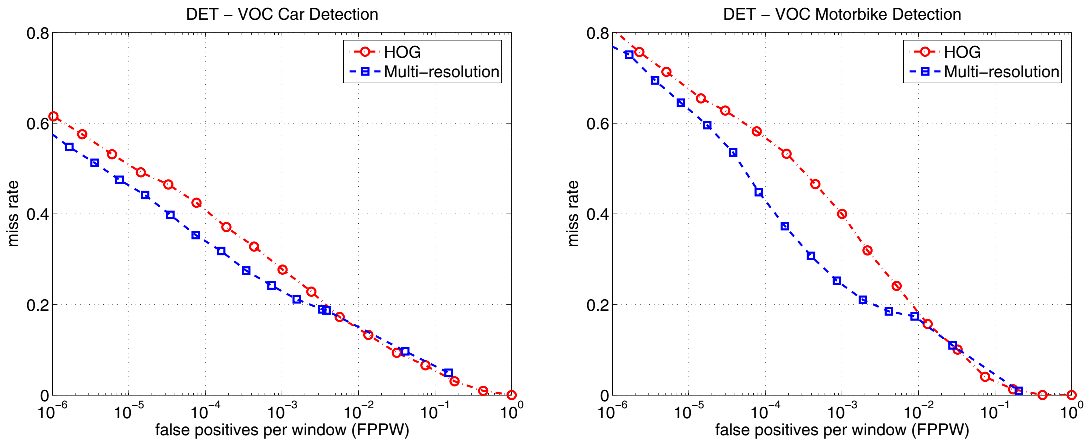

Figure 5. Comparison between HOG and multi-resolution methods using Detection Error Trade-off curves on car and motorbike categories in VOC2006 database.

Resolution	cell size	block size	detector size	#orientations	block stride	detector stride
			VOC Car			
1	(3, 2)	(6, 4)	(13, 7)	9	(2, 1)	(1, 1)
2	(4, 3)	(8, 6)	(26, 14)	18	(3, 2)	(2, 2)
3	(6, 4)	(12, 8)	(52, 28)	18	(4, 4)	(4, 4)
4	(8, 6)	(16, 12)	(104, 56)	18	(8, 4)	(8, 8)
			VOC Motorbike			
1	(3, 3)	(6, 6)	(15, 10)	9	(3, 2)	(1, 1)
2	(5, 4)	(10, 8)	(30, 20)	15	(4, 3)	(2, 2)
3	(6, 6)	(12, 12)	(60, 40)	18	(6, 4)	(4, 4)
4	(8, 8)	(16, 16)	(120, 80)	18	(8, 4)	(8, 8)

Table 5. Parameter settings in each resolution level on VOC2006 car and motorbike categories.

3.2. VOC challenge database

We further experimented on the PASCAL Visual Object Classes challenge 2006 database [9]. The database contains 10 object categories, e.g., cars, motorbikes, cows, and sheep. Objects in the same class typically have big variations in views and significant occlusions. We applied the multi-resolution approach for the car and motorbike categories. As reported in [9], the HOG method obtained the best results among all participants on these two categories. For better demonstration of how a multi-resolution approach can improve the performance of an object detector, we did not apply any postprocessing methods on our detection results and thus still report our results using DET curves (instead of recall-precision curves used in the VOC2006 challenge). The same training/testing sets as in VOC2006 were used in our experiments. We adopted the parameter settings suggested in [6] for the HOG method. Table 5 shows parameters of our multi-resolution approach. Similar to the pedestrian detection in Section 3.1, we varied block sizes, block sampling density and the number of orientations in different resolutions.

Fig. 5 shows the comparison of two methods using DET curves. Whenever the full resolution was included, for FPPW level less than 10^{-2}, the multi-resolution approach outperformed the HOG method. This was more obvious on the motorbike category. At the FPPW level of 10^{-4}, our method achieved 66.0% detection rate compared with 58.9% of the HOG method on the car category, and 57.2% compared with 43.1% on the motorbike category. In comparison with the results of pedestrian detection, the improvement in detection accuracy of the multi-resolution approach was larger on the harder VOC database. Fig. 6 shows sample detection results on these two categories with no further postprocessing. To produce these results, we set the FPPW level to 10^{-5} for the car category, and 10^{-4} for the motorbike category. The detector was able to detect objects in multiple views. On both categories, side view objects were detected more accurately given that there were more side view training images.

4. Conclusions and future work

We proposed a multiple resolution framework for object detection. The framework was compared to the Dalal and Trigg's pedestrian system and was shown to improve both detection rates and running speed. The framework differed from a general cascaded classifier, since it required much less training time, and the feature was organized from coarse to fine. We performed experiments on the INRIA database and other object classes, the system got similar or

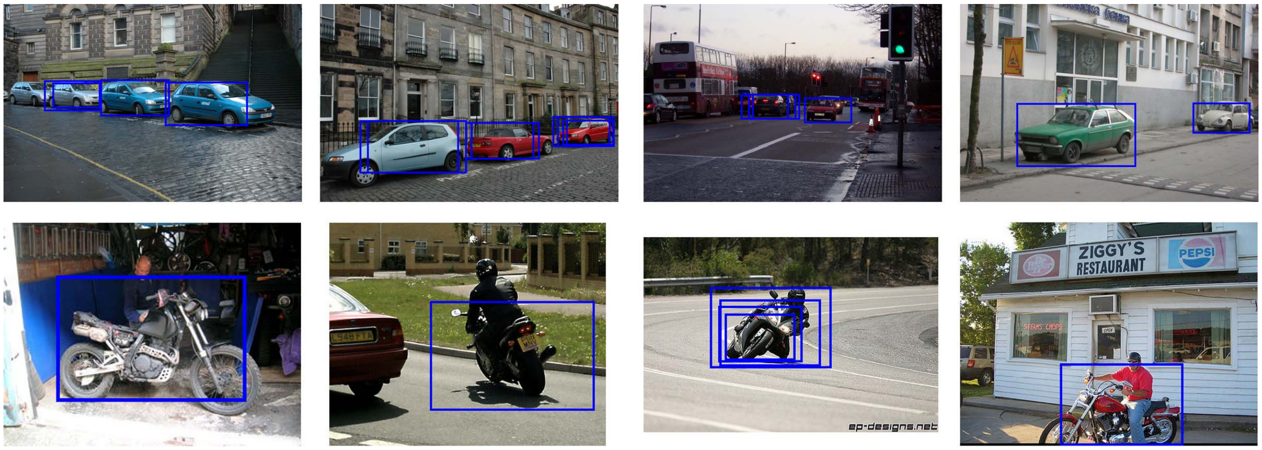

Figure 6. Detection results on some sample images in VOC2006 database. No postprocessing was applied.

better results compared with state-of-the-art methods with a faster detection system.

Future work will apply the multi-resolution approach to other object detection applications, particularly those requiring real-time performance. We also plan to explore our multi-resolution approach as a model for human object detection, particularly with respect to the gaze shifts that occur as one searches for objects.

Acknowledgements

This research was supported by NSF grants, IIS-0527585, CNS-0627645 and ACI-0313184.

References

[1] A. Agarwal and B. Triggs. Hyperfeatures: Multilevel local coding for visual recognition. In *ECCV06*, pages I: 30–43.

[2] Y. Amit, D. Geman, and X. Fan. A coarse-to-fine strategy for multi-class shape detection. *PAMI*, 26(12):1606–1621, Dec. 2004.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[4] S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Towards optimal training of cascaded detectors. In *ECCV06*, pages I: 325–337.

[5] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV99*, pages 1197–1203.

[6] N. Dalal. *Finding people in Images and Videos*. PhD thesis, Institut National Polytechnique De Grenoble, 2006.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR05*, pages I: 886–893.

[8] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *ICCV05*, pages I: 220–227.

[9] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The pascal visual objec classes challenge 2006 results. Technical report.

[10] J. Fasola and M. Veloso. Real-time object detection using segmented and grayscale images. In *ICRA06*, pages 4088–4093.

[11] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Object detection by contour segment networks. In *ECCV06*, pages III: 14–28.

[12] F. Fleuret and D. Geman. Coarse-to-fine face detection. *IJCV*, 41(1-2):85–107, Jan. 2001.

[13] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminative models for object category detection. In *ICCV05*, pages II: 1363–1370.

[14] K. Grauman and T. J. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV05*, pages II: 1458–1465.

[15] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR06*, pages II: 2169–2178.

[17] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR05*, pages I: 878–885.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.

[19] H. Luo. Optimization design of cascaded classifiers. In *CVPR05*, pages I: 480–485.

[20] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR06*, pages I: 26–36.

[21] S. Mitri, A. Nuchter, K. Pervolz, and H. Surmann. Robust object detection in regions of interest with an application in ball recognition. In *ICRA05*, pages 125–130.

[22] A. Mohan, C. P. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, Apr. 2001.

[23] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV06*, pages II: 575–588.

[24] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR06*, pages 3–10.

[25] C. P. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, June 2000.

[26] J. Perry and W. Geisler. Gaze-contigient real-time simulation of arbitrary visual fields. In *Human Vision and Electronic Imaging, SPIE Proceedings*, 2002.

[27] R. Rao, G. Zelinsky, M. Hayhoe, and D. Ballard. Eye movements in iconic visual search. *Vision Research*, 42:1447–1463, 2002.

[28] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *IJCV*, 56(3):151–177, Feb. 2004.

[29] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV05*, pages I: 503–510.

[30] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR01*, pages I:511–518.

[31] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005.

[32] Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR06*, pages II: 1491–1498.