# Weighted Hough Forest for object detection

Yusuke Murai, Yuji Yamauchi, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University
Aichi, Japan
{ym@vision., hf@}cs.chubu.ac.jp

## Abstract

*Hough Forest is an object detection method based on voting from patch images. In the Hough Forest training, some negative patches are trained as a positive sample because the patches are truncated from the background region in a positive image. This makes a reason to occur false positives. To overcome this problem, we introduce weight updating of training sample to the Hough Forest. In the training of the proposed method, if there is a positive sample with a high value of similarity with a negative sample, sample weight are updated to be smaller at each layer of the decision tree. This makes it possible to suppress the vote to the background area. Experimental results show that the detection performance of the proposed method is 11% better than that of conventional method, and is same as the conventional method with masked images.*

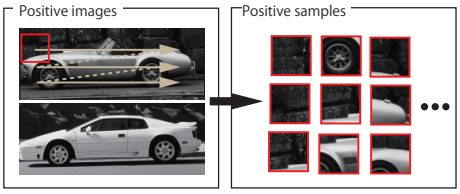


Figure 1. Image patches in positive sample

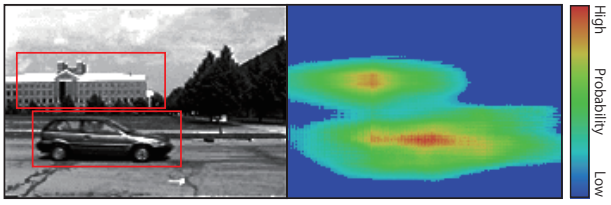


Figure 2. Likelihood map voted by Hough Forest

## 1 Introduction

There are two approaches in object detection, called region-based methods and local patch-based methods. A region-based method performs feature extraction from entire images of the detection window, and discriminates this image is a target object or not by classifiers[1][2][3]. The method performs an exhaustive raster scan on the input image and uses the trained classifiers to detect target objects. During this time, there is a problem in that the detection precision can drop if part of an object is occluded or changes in shape.

With a local patch-based method, on the other hand, the voting is done using patches, so robust detection is possible with respect to partially occluded objects. Local patch-based methods are based on voteing by Hough transform. Leibe et al., proposed an implicit shape model (ISM) using a codebook [4]. With ISM, the system performs vector quantization on a patch image centered on a feature point extracted from an input image, with reference to a codebook, and votes for the centroid position of the object using training sample information contained in the same codebook. Gall et al., proposed a Hough Forest [6] that uses the Random Forest[5] to construct decision tree to classify patch images into two classes. At detection, the system traverses the Hough Forest during the raster scan and votes for the centroid position from patches that have been identified as the target object. In the Hough Forest training, some negative patches are trained as a positive sample, because the patches are cut out from the background region in a positive image, as shown in Figure 1. This makes a reason to occur false positive as shown in Figure 2(a). To overcome this problem, [7] reported a method for designing training sample. This method [7] uses positive samples that are not mixed with negative samples using mask image of the object. However, the creation of masks human cost is high. To resolve this problem, it is necessary to reduce the influence of negative sample automatically, which are similar to positive sample.

The objective of this work is to suppress the vote to the background area which as shown in Figure 2(b) becomes false positives, for object detection based on Hough Forest. This paper presents a weighted Hough Forest for object detection. In the proposed method, we introduce weight updating of training samples to the Hough Forest. In the training of the proposed method, a weight of each training sample is updated in each layer of decision tree. This makes it possible to suppress the vote to the background area.

## 2 Proposed method

This section describes an algorithm of weighted Hough Forest for object detection. In the training of the proposed method, if there is a positive sample with a high value of similarity with a negative sample, sample weight is updated to be smaller at each layer of the decision tree.

### 2.1 Training samples

Using a patch $x_i$ by grid sampling from the positive images and negative images, we generate training samples $\{(x_1, c_1, \mathbf{r}_1, w_1^{(d)}), \cdots, (x_N, c_N, \mathbf{r}_N, w_N^{(d)})\}$. In this case, $x_i$ is the patch, $c_i$ is the class label $\{0, 1\}$ of the patch from image, $\mathbf{r}_i$ is the offset vector, and $w_i^{(d)}$ is the sample weight at layer $d$ of the decision tree. The patches sampled from the background are assigned the class label $c_i = 0$, while the patches sampled from the object are assigned $c_i = 1$. The subsets are generated by random sampling from the training sample collection.

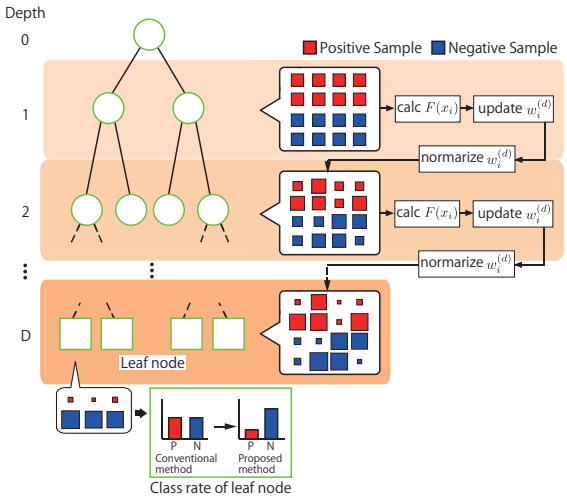


Figure 3. Weighted Hough Forest training process

## 2.2 Training of decision tree

The weighted Hough Forest training process is shown in Figure 3, with each processing step being described below.

**Step1 : Weight initialization** Initialize the weights $w_i^{(d)}$ of the training sample $i$ to $w_i^{(0)} = 1/N$. In this case, $d$ is a hierarchy of decision tree.

**Step2 : Split function candidates** In this research, we use the similarity according to a dominant orientation template (DOT) [8, 9] in the split function. The DOT is obtained from a oriented histogram for each local region, and, is represented as binary feature. Template is choose from positive samples in the split node. If the similarity of a training patch and the template is less than the threshold value, it is branched to the left, otherwise it is branched to the right. Split function candidates are created by repeating the process of random selecting the template and the threshold.

**Step3 : Split function decisions** Select the optimal one from among the split function candidates. To do this, we use two measures $U_1$ and $U_2$ to evaluate the uncertainty for a set of cuboids $A = \{x_i, c_i, \mathbf{r}_i, w_i^{(d)}\}$. We switch between the two benchmarks described below in each layer in the evaluation function $U_\star$. We define the $U_1$ by Equation (1) for a sample collection A, using entropy:

$$U_1(A) = |A|(-a \cdot \log a - (1-a) \cdot \log(1-a)), \qquad (1)$$

where a is the positive sample rate, which we calculate from Equation (2) using the sample weight $w_i$:

$$a = \frac{\sum_{w_i^{(d)} \in A \wedge c_i = 1} w_i^{(d)}}{\sum_{w_i^{(d)} \in A} w_i^{(d)}}, \qquad (2)$$

The second $U_2$ is a function that evaluates variations in the offset vector $\mathbf{r}_i$, as defined by Equation (3):

$$U_2(A) = \sum_{\mathbf{r}_i \in A} (\mathbf{r}_i - \mathbf{r}_A)^2, \qquad (3)$$

where $\mathbf{r}_A$ is the average value of the offset vector $\mathbf{r}_i$, so that $U_2$ is equivalent to scattering of the offset vector.

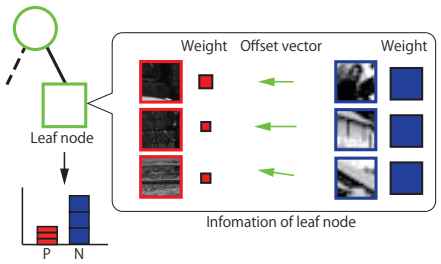


Figure 4. Leaf node information

In each decision tree, we repeat Steps 3 and 4 until all the nodes in layer $d$ have been generated.

**Step4 : Sample weight update** Update the training sample weights in each child node, after generating all the nodes in layer $d$ in each decision tree. We update the weight $w_i^{(d)}$ using the evaluation function $F$ based on similarity. We define the $F$ by Equation (4), and update the weight $w_{(d)}$ by Equation (5):

$$F(x_i) = \frac{1}{2} \ln \frac{a+\epsilon}{(1-a)+\epsilon} \quad \epsilon = 0.000001, \qquad (4)$$

$$w_i^{(d)} = w_i^{(d-1)} \exp(F(x_i)), \qquad (5)$$

If the sample weight $w_i^{(d)}$ has a high value, this indicates likely to be a target. If the sample weight $w_i^{(d)}$ is low, this indicates likely to be non-target. Finally, we normalize the weights of the positive samples within that node by Equation (6):

$$w_i^{(d)} = \frac{w_i^{(d)}}{\sum_{w_i^{(d)} \in \mathbf{S}^{(d)}} w_i^{(d)}}, \qquad (6)$$

where $\mathbf{S}^d$ represents all of the sample sets in the layer $d$.

**Step5 : Repeat Steps 2 to 4** Repeat the node generation and weight update of Steps 2 to 4 until either the number of samples is less than a given number or a specified depth $D$ has been reached. As the layers become deeper, a decision tree is constructed having a branch structure where the influence of samples with low weights is reduced.

**Step6 : Leaf node generation** Store weighted sample rates for positive and negative, and a list of offset vectors to the object center in a leaf node, as shown in Figure 4. By constructing a decision tree by the above processing, we are able to automatically lower the weight of positive samples that are similar to non-target objects.

## 2.3 Voting process

During the voting, a likelihood map of the same size as the input image is prepared. A patch image that has been cut out from a position of the input image is input to each decision tree, to obtain a positive sample rate for each decision tree. Votes for each decision tree are cast in a vote space using offset vectors stored in the eventual leaf nodes. The system detects the object with respect to the generated likelihood map by performing mean shift analysis.

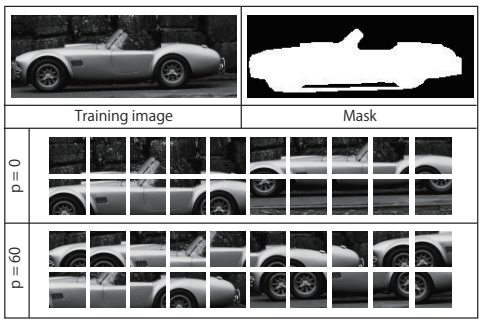


Figure 5. Creating training samples using the mask

## 3 Experiments

We conducted evaluation experiments to confirm the validity of the proposed method. We used cars, pedestrian, and rotated objects as the detection targets in each round of experiments and compared the results with those of the Hough Forest using mask [7], which is the conventional method.

### 3.1 Comparison with Hough Forest using a mask

Training samples are created by using a mask. Patch to be used for learning is truncated based on the content of a fraction containing the positive region. Create a learning sample by changing the content of $p$. In evaluation, we used the F-measure.

**Car detection :** Training is performed using the Car Side Training Set [10] in car detection. The training data is composed 550 positive images and 500 negative images. The evaluation data of car is performed using the UIUC cars dataset of 5667 images.

**Pedestrian detection :** Training is performed using the TUD pedestrian data sets in pedestrian detection. The training data is composed 100 positive images and 190 negative images. The evaluation data of pedestrian is performed using the TUD pedestrian data sets of 250 images.

Mask indicating a positive region is provided in either data set. The car and pedestrian detection accuracy are shown in Table 1. $p = 0$ is a state that does not use a mask. In comparison with the Hough Forest($p = 0$), we see that recall-precision is improved with the proposed method. F value of the proposed method in the car detection 0.13 higher than the conventional method. F value of the proposed method in the pedestrian detection 0.10 higher than the conventional method.

Table 1. Comparison of the F-measure

| | p=0 | p=20 | p=40 | p=60 | proposed |
|---|---|---|---|---|---|
| Car | 0.70 | 0.78 | 0.84 | 0.81 | 0.83 |
| Pedestrian | 0.67 | 0.76 | 0.79 | 0.77 | 0.77 |

### 3.2 Evaluation using the Pascal VOC dataset and INRIA Person dataset

We show the experimental results using the other dataset. In evaluation, we used the recall-precition curve.

**Pascal VOC dataset :** We use the dataset of car in the Pascal VOC. The training data is composed 590 positive images and 5096 negative images. Evaluation data is composed 4487 images.

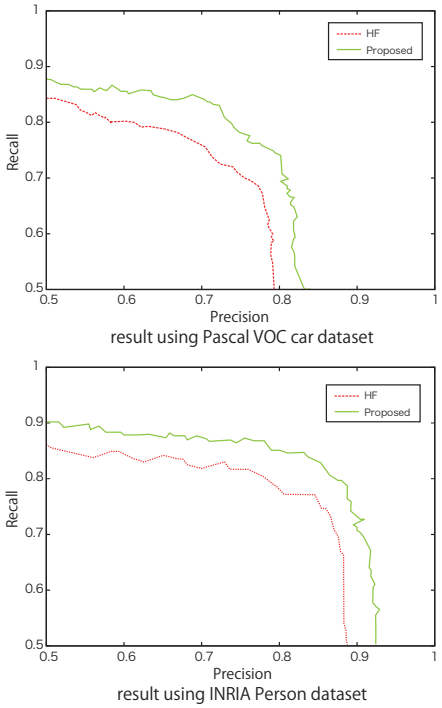


Figure 6. detection result using Pascal VOC car dataset and INRIA Person dataset

**INRIA Person dataset :** The training data is composed 2388 positive images and 12180 negative images. Evaluation data is composed 1126 images.
The detection accuracy are shown in Fig 6. In comparison with the Hough Forest, we see that recall-precision is improved with the proposed method. Proposed method compared to the conventional method was improved 8% accuracy at Pascal VOC dataset, and 7% accuracy at INRIA Person dataset.

### 3.3 Discussion

We discuss the detection accuracy of the proposed method. Examples of the visualization of the weights of training samples are shown in Figure 7. As can be seen from Figs.7, we see that the weights of the car and person regions are higher. Detection examples and likelihood maps for each of the experiments are shown in Figure8. Figure 8(a) is a Hough Forest that was constructed using a content rate $p = 0$ of the training samples, Figure 8(b) is a detection example by the proposed method. False positives occur in the conventional methods, because vote value to the background area is large. The proposed method can suppress false positives, because vote value to the background area is small. From the above results, the weighting of the learning sample can be validated. From the above results, the validity of the weighting of the learning sample was confirmed.

## 4 Conclusions

We have proposed an weighted Hough Forest which can suppress false positive of background area by lowering the weight of training samples that are similar to negative samples. The proposed method has demonstrated it can suppress votes for background region

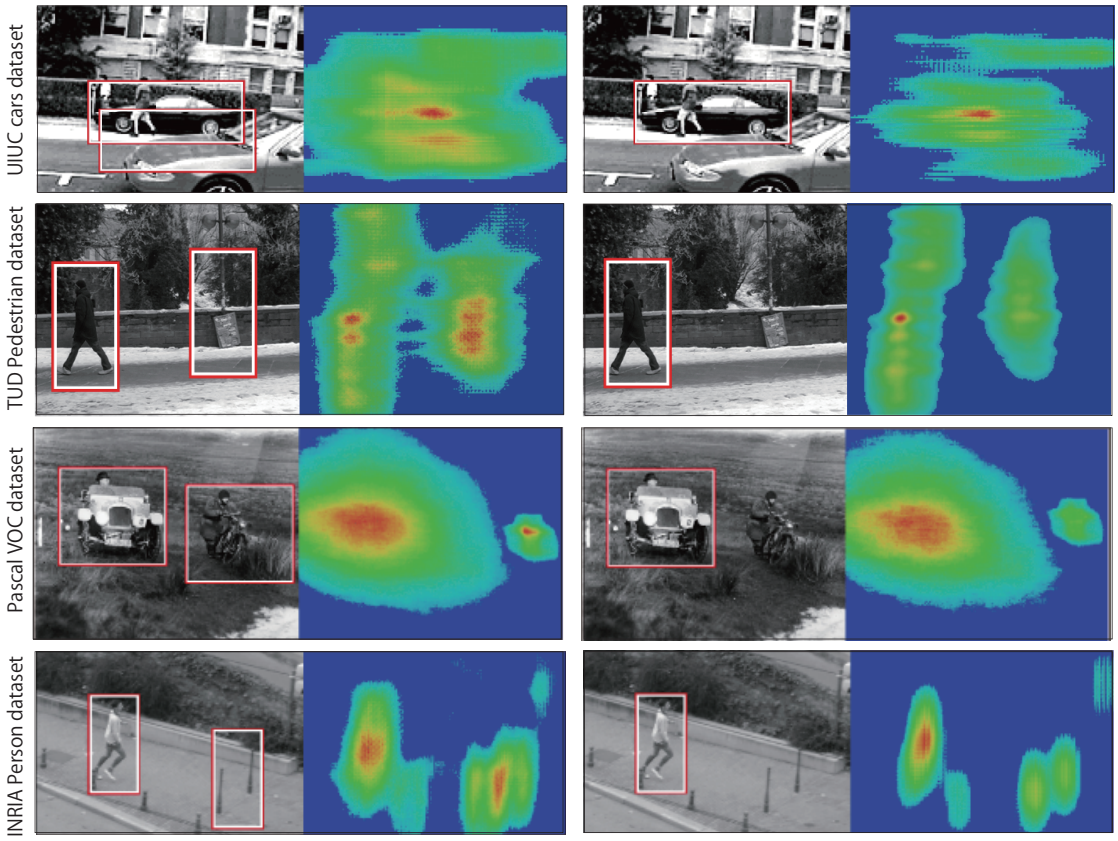


Figure 8. Car detection examples and their likelihood maps

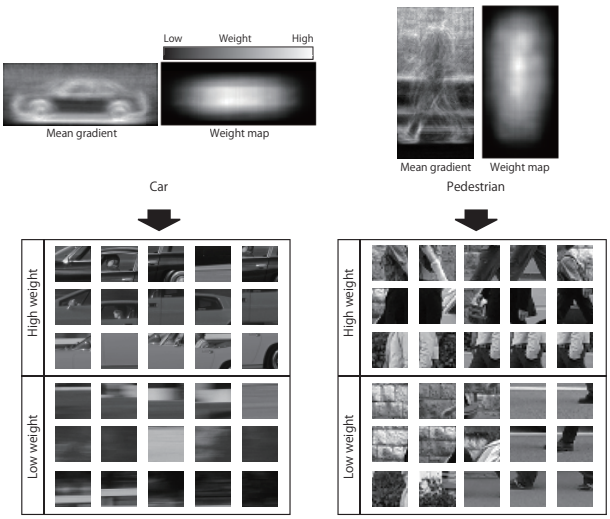


Figure 7. Visualization of teaining sample weights

with respect to conventional Hough Forest and enables the detection of target objects more stably. In the future, we intend to investigate the design of local image feature in order to increase the precision of the proposed method.

## References


[1] N. Dalal, and B. Triggs.: "Histograms of Oriented Gradients for Human Detection", *Computer Vision and Pattern Recognition*, vol.1, pp.886-893, 2005.

[2] C. Cortes, and V. Vapnik.: "Support-vector networks", *Machine Learning*, pp.273-297,1995.

[3] Y. Freund, and R. E. Schapire.: "Experiments with a new boosting algorithm", *International Conference on Machine Learning*, pp.148-156, 1996.

[4] B. Leibe, A. Leonardis, and B. Schiele.: "Robust object detection with interleaved categorization and segmentation", *International Journal of Computer Vision*, vol.77, no.1-3, pp.259-289, 2008.

[5] L. B. Statistics, and L. Breiman.: "Random forests", *Machine Learning*, pp.5-32, 2001.

[6] J. Gall, and V. Lempitsky.: "Class-specific hough forests for object detection", *Computer Vision and Pattern Recognition*, 2009.

[7] S. Shimizu, N. Watanabe, Y. Miwa, M. Seki, T. Hirai and H. Fujiyoshi.: "Training Method for an Object Detection Using Hough Forests", *Vision Engineering Workshop*, 2012.

[8] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab.: "Dominant orientation templates for real-time detection of texture-less objects", *Conference Computer Vision and Pattern Recognition*, pp.2257-2264, 2010.

[9] D. Tang, Y. Liu, and T.-K. Kim.: "Fast pedestrian detection by cascaded random forest with dominant orientation templates", *British Machine Vision Conference*, 2012.

[10] "Car Side Training Set", http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html