Regular Paper

A Hierarchical Extension of the HOG Model Implemented in the Convolution-net for Human Detection

Yasuto Arakaki[1] Hayaru Shouno[1,a)]
Kazuyuki Takahashi[2] Takashi Morie[2]

Received: February 1, 2012, Revised: March 22, 2012,
Accepted: April 25, 2012

Abstract: For the detection of generic objects in the field of image processing, histograms of orientation gradients (HOG) is discussed for these years. The performance of the classification system using HOG shows a good result. However, the performance of using HOG descriptor would be influenced by the detecting object size. In order to overcome this problem, we introduce a kind of hierarchy inspired from the convolution-net, which is a model of our visual processing system in the brain. The hierarchical HOG (H-HOG) integrates several scales of HOG descriptors in its architecture, and represents the input image as the combinatorial of more complex features rather than that of the orientation gradients. We investigate the H-HOG performance and compare with the conventional HOG. In the result, we obtain the better performance rather than the conventional HOG. Especially the size of representation dimension is much smaller than the conventional HOG without reducing the detecting performance.

Keywords: HOG model, convolution-net model, human detection, feature extraction

1. Introduction

The image recognition technology has been applied in many areas such as inspection systems for manufactured products in factories, driving assistance system for automobile, and so on. And more reducing of computational cost for the image recognition system is required for enlarging the application area.

In the field of generic object recognition, the histogram of oriented gradient (HOG) proposed by Dalal has been focused for description of objects in the images because of its simple feature extraction rule [1], [2]. HOG represents features of an object as a histogram of the input image gradient of certain areas that includes the object. Using HOG for the recognition of the image area, the histogram of the image is usually treated as a vector for the input of classification machine, such like a support vector machine (SVM). Even though HOG is a simple model, it has shown several good recognition performances for pedestrian and car detection [3].

However, we consider HOG includes two problems to solve. The first point is the number of feature dimensions, which can become huge number by the value of image dividing parameter. The large number of feature dimensions prevents reducing computational cost and requires a lot of training images for classification, so that we should reduce it for the cost down. The second point is that invariance for both the location and the scale of the object may not be good enough for the robust recognition. In the conventional HOG framework, input image is scanned with several scales cropping window to adjust detecting object size. However, this window size adjusting process might take a computational cost.

On the other hand, we human have a flexible recognition system, for example, we can recognize an object in the image with any locations and with several scale changes. From the physiological viewpoints, our visual processing system in the brain is believed to have a kind of hierarchical structure [4], [5]. HOG can be interpreted as a simple layered structure neural network, so that we consider introducing hierarchical structure such like the brain into the HOG can improve the robustness against the deformation of the objects. Such hierarchical image processing models, which are inspired from the brain, have been proposed. Fukushima proposed “Neocognitron”, and applied it to handwritten character recognition [6], [7]. LeCun also proposed “LeNet” series and showed good performance in the meaning of the recognition accuracy [8]. Serre et al. compared such hierarchical image processing system with the real brain [9]. These kinds of hierarchical neural networks are called “convolution-net”. In the convolution-net, one of the important points is hierarchy, so that we introduce a kind of hierarchical structure represented by the convolution-net into the HOG. Introducing the hierarchy, we expect the robustness for variations of both location and scale can be improved to the conventional HOG model. Moreover, in the convolution-net, the local feature description is gradually integrated through the hierarchical processing. The integrated description can be considered as a representation of the compressed visual information for the input image. Therefore, we also expect reducing the number of dimension of the HOG description by introducing such hierarchy. In order to overcome the conven-

[1] University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[2] Kyushu Institute of Technology, Kitakyushu, Fukuoka 808–0196, Japan
[a)] shouno@uec.ac.jp

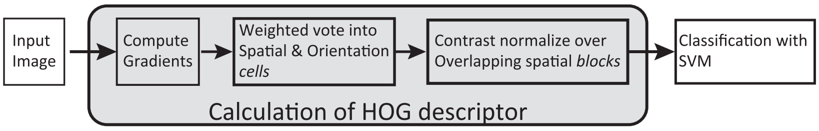
Input Image
Compute Gradients
Weighted vote into Spatial & Orientation *cells*
Contrast normalize over Overlapping spatial *blocks*
Classification with SVM
Calculation of HOG descriptor

Fig. 1 A schematic diagram of a typical image recognition system using HOG descriptor [1].

tional HOG problems, we proposed an improving model of the HOG that introduce a concept of the hierarchical structure of the convolution-net. In this study, we evaluate the performance of our proposing model by use of the INRIA Person Dataset, which is an image database for the pedestrian detection, and we discussed about recognition performance.

2. Conventional Model Formulation

In this study, we introduce the hierarchical structure, which is inspired from visual processing in our brain, into the conventional HOG called Hierarchical HOG (H-HOG). Thus, we explain summaries about conventional visual recognition system using HOG, and the convolution-net proposed by Mutch & Lowe in this section [1], [10].

2.1 Conventional HOG System

The conventional HOG, which is applied to the generic image recognition, is a kind of feature descriptor using histograms of the local image gradients. We can obtain a HOG descriptor, which can represent a rough shape of the object, into the local area of an image, so that the HOG descriptor is often applied to the object detection such like human detection [3].

Figure 1 shows the schematic diagram of a typical visual recognition system using the HOG descriptors. At first, the image gradient is calculated from the input image $I(u, v)$, where (u, v) indicates the location in the image and $I(u, v)$ means the pixel value at the location. The gradient for each location (u, v) is represented as both intensity $m(u, v)$ and orientation $\theta(u, v)$,

$$m(u, v) = \sqrt{I_u(u, v)^2 + I_v(u, v)^2}, \tag{1}$$

$$\theta(u, v) = \tan^{-1}\left(\frac{I_v(u, v)}{I_u(u, v)}\right), \tag{2}$$

where $I_u(u, v)$ and $I_v(u, v)$ means the difference between the neighbors:

$$I_u(u, v) = I(u + 1, v) - I(u - 1, v) \tag{3}$$

$$I_v(u, v) = I(u, v + 1) - I(u, v - 1). \tag{4}$$

2.1.1 Histogram Representation for the Cell

In the next step for obtaining HOG descriptor, we divide the gradient into several small areas called "cell" and make an orientation histogram for each cell. In the process, the intensity $m(u, v)$ plays a roll of the weight for the voting bin of $\theta(u, v)$. Applying such local orientation histograms, local translation deformation effect of the object in the cell is reduced. The histogram, which describes the distribution of the edge components for the orientation, can be regarded as a vector. Quantizing the orientation $\theta(u, v)$ on the i-th cell into the Q state, we can obtain Q bins histogram whose elements are represented by the vector: $\boldsymbol{f}_i = \{f_{i,1}, f_{i,2}, \cdots, f_{i,Q}\}$:

$$f_{i,q} = \sum_{(u,v)\in i\text{th cell}} m(u, v)\, \delta_q(\theta(u, v)), \tag{5}$$

$$\delta_q(s) = \begin{cases} 1 & (q - 1)\Delta s \le s < q\Delta s \ \text{ where } \Delta s = \pi/Q \\ 0 & \text{else} \end{cases}. \tag{6}$$

The size of cells is an important parameter that makes influence to the number of feature dimension for the classifier. Large cell size conducts small number of the feature dimension, which is treatable property for the classifier; however, feature extraction may become too rough to represent the object. On the contrary, small size cell size, which can represent detail of the object, may make huge number of feature dimensions to describe the object, which makes hard problem for the classifier.

2.1.2 Block Normalization for the Cells Representation

The gradient strength varies over a wide range owing to local variations in illumination and contrast between foreground and background. Thus, effective local contrast normalization may be good for the classification. We adopt L2 normalization in the same manner of Dalal [1]. Considering the neighboring cells for the i-th cell as a group, which is called "block", the i-th block feature can be described as collection of the histogram vectors:

$$\tilde{\boldsymbol{V}}_i = \{\boldsymbol{f}_i, \{\boldsymbol{f}_k\}_{k\in\mathrm{NN}(i)}\}, \tag{7}$$

where NN(i) means the neighbor cells for the i-th cell. Then the block vector $\tilde{\boldsymbol{V}}_i$ is normalized as

$$\boldsymbol{V}_i = \frac{\tilde{\boldsymbol{V}}_i}{\sqrt{\|\tilde{\boldsymbol{V}}_i\|^2 + \epsilon^2}}, \tag{8}$$

where ϵ is a small positive constant to prevent diverging.

HOG descriptor, in the final form, is a collection of these normalized block vectors $\{\boldsymbol{V}_i\}$. For example, let us consider the HOG descriptor for 100×200 [pixels] images. Assuming the cell size as 20×20 [pixels], we obtain 5×10 blocks, however, the border cells does not have enough neighbors to normalize. Thus, we take $3 \times 8 = 24$ blocks as the effective blocks when we take 3×3 cells as one block. When we divide orientation into each $\pi/9$, the histogram quantization parameter becomes $Q = 9$, and a block vector $\boldsymbol{V}_i$ has $9\times(3\times3) = 81$ dimensions. The block size is 24, and dimension of each block vector is 81 dimensions, so that, the number of feature vector dimensions in the HOG descriptor becomes 1,944 in this case.

2.1.3 Conventional HOG Classifying System

HOG descriptor is a robust expression for the local translation deformation and illumination variation, and it can represent rough feature of the object in an image. Thus, HOG is considered as good for the generic object recognition [2]. Dalal proposed to use HOG descriptor as the input for the support vector machine (SVM), which is a kind of classifier, and showed better classification performance rather than those of other features [1].

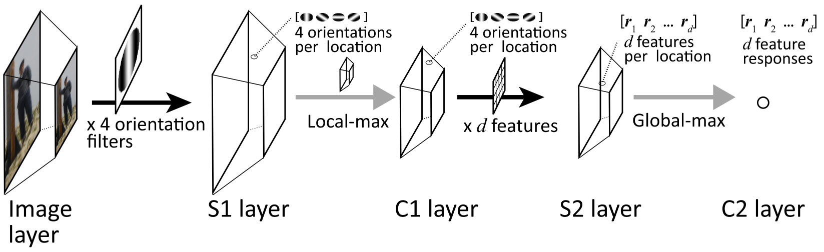
4 orientations per location
4 orientations per location
[r1 r2 ... rd] d features per location
[r1 r2 ... rd] d feature responses
x 4 orientation filters
Local-max
x d features
Global-max
Image layer
S1 layer
C1 layer
S2 layer
C2 layer

Fig. 2 A schematic diagram of the convolution-net proposed by Mutch&Lowe [10]. The input image would be processed through the pathway, that is, S1 → C1 → S2 → C2 layers. The input image is translated the description of d dimensional vector appeared in $C2$ layer.

2.2 Summary of a Convolution-net of Mutch & Lowe

For introducing the brain inspired mechanism into the HOG, we explain a kind of convolution-net proposed by Mutch & Lowe [10]. The convolution-net is a hierarchical artificial neural network model originated from the model proposed by Hubel and Wiesel [5]. Hubel & Wiesel found two types of cells in the early visual processing area of the mammal brain, which are called "simple cell" and "complex cell". Each type of cell has local area for responding in the viewing field, which is called "receptive field", and responds to the specific input stimulus such that line/edge component in the receptive field. The difference between these types of cells is response for the location of the preferred input stimulus in the receptive field. The simple cell only responds to the preferred input stimulus at the specific location. On the other hand, the complex cell responds to the preferred input stimulus at any location in the receptive field. Thus, Hubel & Wiesel proposed a kind of hierarchy between these cells, that is, a complex cell may gather the outputs of simple cells that respond to same preferred stimulus but have slightly different receptive fields. The convolution-net has a hierarchy of these types of cells, and connects this hierarchy alternately [6], [7], [10].

Figure 2 is a schematic diagram of the convolution-net proposed by Mutch & Lowe [10]. The lowest of the figure shows the input layer. The most specific feature of this model is introducing the multi-scale expression for the input image. An input image is scaled into the several resolutions, which is described as the image pyramid in the bottom of the figure. The S1-layer in the figure is a model for the simple cells that extract line/edge components for each location and resolution. Mathematically, this extracting operation can be described as a convolution with line/edge filtering template. The next C1-layer in the figure is a model for the complex cells. The C1-layer cell calculate local maximum for the corresponding S1-layer cells. This type of operation is called spatial pooling, which is to tolerate the deformation of local translation of the pattern in the image. The spatial pooling operation can also describe as a convolution with non-linear operation. These two layers are corresponds to the model of early visual processing area.

The S2-layer is a higher feature extraction layer, which is implemented as a kind of template-matching mechanism. The S2-layer consists of d types of templates, and each template is also treated as the line/edge extracting filters. These template filters are obtained with sampling from the patterns appeared in C1-layer for training input images.

The final layer, which is called C2-layer, integrates extracted features in the S2-layer pyramids. The unit in the C2-layer detects maximum value for the corresponding S2-layer pyramid. As the result, the unit represents the containing rate of the template pattern for the input pattern. The expression of the C2 layer can be regarded as the d dimensional vector for an input image, so that, we can apply several classifiers such like SVM for this expression.

3. Hierarchical HOG Formulation

The hierarchical architecture of the convolution-net is an important concept for our study. The S1 layer in the convolution-net plays a role of line/edge extractor for in the viewing field. Calculating histogram from the image gradient operation in HOG is the similar function for the line/edge extractor. Focusing to the function of line/edge extraction, the difference between these two models is only representation for the extracted features. The function of the C1 layer and that of the cell/block mechanism in HOG is also similar. **Figure 3** shows the corresponding architectures between the convolution-net and HOG. In the figure, the conventional HOG output corresponds to the C1 layer output; however, the convolution-net has more deep hierarchy such like S2 and C2 layer. Thus, we can introduce hierarchy extension for HOG in the manner of the convolution-net. In the following we call our hierarchical HOG as "H-HOG".

3.1 Input Feature Modulation

Before explaining the higher layers, we introduce several modifications for the calculation of HOG descriptor. In Fig. 3, HOG descriptor, which is conducted from Eq. (6) to Eq. (8), is calculated as the input of the higher stage. In the count up for the histogram by Eq. (6), the intensity $m(u, v)$ is piled up linearly, however, the small value of $m(u, v)$ might be a kind of contamination. Considering the case that all components of i-th block $\tilde{\boldsymbol{V}}_i$ are weak, all the weak components are enhanced in the normalization procedure in Eq. (8). Thus, we introduce a nonlinear modulation in order to emphasize the histogram:

$$f_{i,q} = \sum_{(u,v)\in i\text{th cell}} \psi(m(u, v))\ \delta_q(\theta(u, v)), \tag{9}$$

$$\psi(x) = \begin{cases} |x - h|^d & \text{if } x > h \\ 0 & \text{else} \end{cases}, \tag{10}$$

where h plays a role of threshold and d means an enhance factor. After calculating the modulated histograms $\{\boldsymbol{f}_i\}$, we adopt the same manner of the conventional HOG representation.

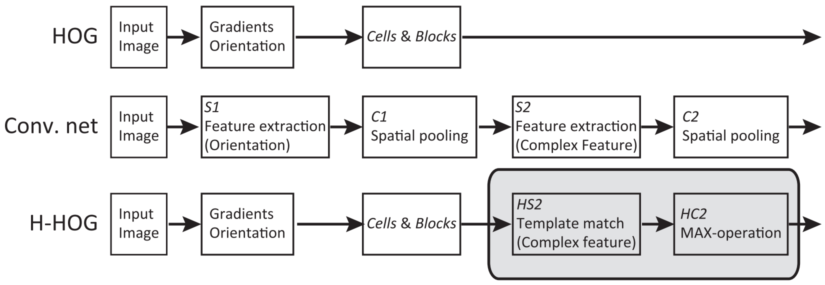

Fig. 3 Comparison among conventional HOG, convolution-net, and our hierarchical HOG. The convolution-net alternate S-layer, which extracts features, and C-layer, which tolerate deformation with spatial pooling. The conventional HOG corresponds to the S1-layer and C1-layer in the convolution-net. Our hierarchical HOG extends HOG description with the manner of convolution-net, which is described as S2 and C2 layers.

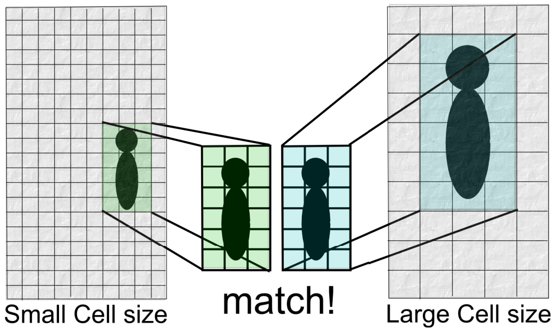

Fig. 4 Applying multi-resolution cell representation, we can treat different scale objects as identical representation.

3.2 Multi-resolution Representation

The convolution-net proposed by Mutch & Lowe introduces a multi-resolution representation. In order to introduce the multi-resolution representation in our system, we prepare several HOG descriptors that have different cell sizes. The advantage of the multi-resolution representation is to the deformation of object magnification and shrinkage. **Figure 4** shows the advantage of introducing the multi-resolution representation. Preparing different scale representations, we can treat different scale objects as identical representation.

3.3 Template-matching Layer: *HS*2

The template-matching layer in the H-HOG corresponds to the S2 layer in the convolution-net, which is to extract just more complex feature rather than those of the S1 layers. Mutch&Lowe adopt to use templates of partial C1 descriptions for the training patterns chosen by sampling. For convenience, we call this template-matching layer as $HS2$-layer.

3.3.1 Template Selection

In the conventional HOG, the output description is represented by $\{\boldsymbol{V}_i\}$ in Eq. (8), where i means the block location. In the training mode of the H-HOG, at first, we select several block locations in the several resolutions randomly. We treat neighbor blocks of the selected block as a cluster for the template, and the size of cluster is selected from several variations randomly. Thus, when location i' is selected for the k-th template $\boldsymbol{T}_k$, the template vector can be denoted as:

$$\boldsymbol{T}_k = \{\boldsymbol{V}_{i'}, \{\boldsymbol{V}_j\}_{j\in \mathrm{NN}_k(i')}\}, \tag{11}$$

where $\mathrm{NN}_k(i')$ means collection of neighborhood blocks of the i'-th block, and the size of the neighborhood is chosen from several variations randomly.

Moreover, in order to prevent using similar template, we adopt the following discard rule. Selecting new template $\boldsymbol{T}_n$ from the input description, we calculate similarities for whole existing templates by use of direction cosine:

$$u_{n,k} = \frac{\boldsymbol{T}_n \cdot \boldsymbol{T}_k}{\|\boldsymbol{T}_n\|\,\|\boldsymbol{T}_k\|}, \tag{12}$$

where k is the index for the any existing templates. If the similarity $u_{n,k}$ is larger than a threshold U, we regard the $HS2$ layer already has template $\boldsymbol{T}_n$ and the template $\boldsymbol{T}_n$ is not accepted for the $HS2$ layer.

3.3.2 Calculation of *HS*2 Layer Representation

For the representation of the input image in the $HS2$ layer, we adopt direction cosine between a template and input representation in for the feature extraction of the $HS2$ layer. Denoting $\boldsymbol{X}_j$ as the conventional HOG description of the location j for the k-th template, that is,

$$\boldsymbol{X}_j = \{\boldsymbol{V}_j, \{\boldsymbol{V}_l\}_{l\in N_k(j)}\}, \tag{13}$$

we describe the output of the $HS2$ layer $R^{HS2}_{k,j}$ as

$$R^{HS2}_{k,j} = \frac{\boldsymbol{T}_k \cdot \boldsymbol{X}_j}{\|\boldsymbol{T}_k\|\,\|\boldsymbol{X}_j\|}. \tag{14}$$

Thus, the feature description $R^{HS2}_{k,j}$ describes the including degree of k-th template at the location j. Applying this operation to whole input location, we can obtain a feature map for the k-th template.

3.4 Max-operator Layer: *HC2*

In the convolution-net, the C2 layer plays a role of integration of feature by use of maximum operation, which can be considered as a kind of spatial pooling. In the H-HOG model, we also adopt the maximum operator for the location, so that, the output for the k-th template R^{HC2_k} is calculated as

$$R^{\mathrm{HC2}}_k = \max_j R^{HS2}_{k,j}. \tag{15}$$

As the result, when we prepare K templates to describe input data set, we can obtain K dimension vector for an input image. This local template-matching and maximum integration might be effective for the deformation of the image caused by the object translation.

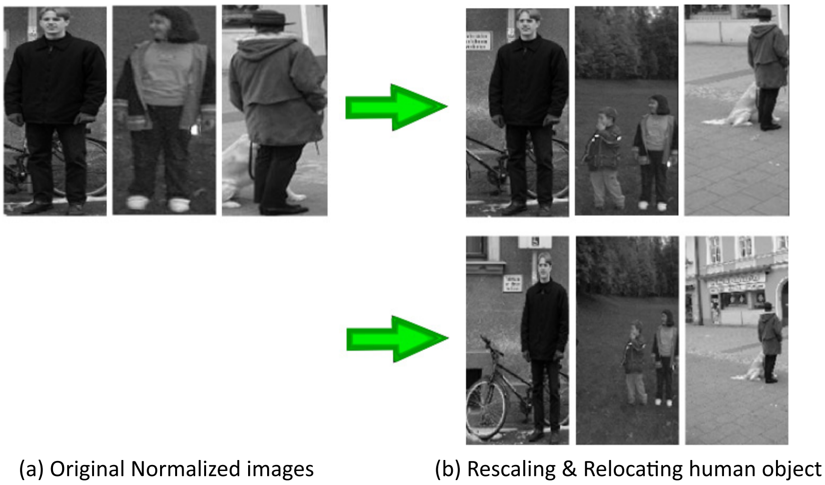
(a) Original Normalized images
(b) Rescaling & Relocating human object

Fig. 5 Examples of the human images. The left shows those in the original INRIA data set provided by Dalal 100×200 [pixels^2] [11]. The right top shows those of the dataset 1, which is re-cropped into 180×400 [pixels^2] and re-scaled into 100×200 [pixels^2]. The right bottom shows those of the dataset 2, which is re-cropped into 300×600 [pixels^2] and re-scaled into 100×200 [pixels^2].

4. Computer Simulation & Results

In the evaluation of the H-HOG using compute simulation, we adopt following parameters. In the experiment using conventional HOG, we prepare several sizes for the cell, which is the unit description for the histogram described as Eq. (6). These sizes are $\{5{\times}5, 10{\times}10, 15{\times}15, 20{\times}20, 25{\times}25\}$ [pixels^2]. The quantization parameter is fixed as $Q = 9$, and the block size is also fixed as 3×3 [cells^2]. For example, when a 100×200 [pixels^2] input image is provided, the total dimensions of conventional HOG descriptors for these cell sizes become {55,404, 11,664, 3,564, 1,944, 972} elements vectors respectively. In the following, we denote the HOG descriptor that have $n \times n$ cells size as HOG$_n$ for convenience.

In the H-HOG experiment, the model has multi-resolution representation in the pre-$HS2$ layer. We prepare several cell sizes for the multi-resolution representation, that are $\{5 \times 5, 10 \times 10, 15 \times 15, 20 \times 20, 25 \times 25\}$ cells. For template choosing in the $HS2$-layer denoted as Eq. (11), template block size are randomly chosen from following candidates: $\{1 \times 2, 2 \times 4, 3 \times 6\}$ [blocks^2]. For example, when we segmented 2×4 [blocks^2] as a template, the template become 648 elements vector. Each location for the template is also chosen randomly from one cell size representation. We determine to choose 4 templates for each cell size representation, so that 20 templates are selected from one training input image.

For the H-HOG experiments, we focus to the feature extraction ability in $HS2$ layer, so that, we prepare the following three types of H-HOGs. One is H-HOG with applying template selection described in Section 3.3.1, and we denote it as "H-HOG$_{sel}$". We choose $U = 0.8$ for the threshold in Eq. (12) to discard similar templates, which is determined experimentally.

The second type is applying the non-linear modulation denoted as Eq. (10) into the H-HOG$_{sel}$. We described this type as "H-HOG$_{sel}^{non}$". The modulation parameters, which are threshold h and emphasize factor d, are experimentally determined as $h = 50$ and $d = 1.1$ respectively.

The last one is not applying these modifications, and we denote it without any suffixes as "H-HOG".

4.1 Dataset: INRIA Person Detection

We evaluate the performance of the H-HOG for a person detection problem using a modified INRIA person dataset [11]. The conventional INRIA person dataset is for the evaluation of the classification accuracy of the conventional HOG. In the conventional dataset provided by Dalal [1], human objects are segmented and normalized in the 64×128 [pixels^2]. **Figure 5** left shows several examples of the normalized human image in the same manner of Dalal except image size, which is 100×200 [pixels^2]. Roughly speaking, the parts of the human such that head, body, and legs in the normalized images looks located similar position, and each size of human looks same size even though the target is child or adult.

In this study, we re-crop the human images from the original image database for evaluation of the scale and location invariance. The dataset 1 is cropped as 180×400 [pixels^2] from the original image, and normalized in 100×200 [pixels^2], which includes human object at random position for positive samples. The dataset 2 is also cropped as 300×600 [pixels^2] from the original image, and normalized in 100×200 [pixels^2]. The locations and sizes of human objects are assigned more random rather than those of the dataset 1 for positive samples. Figure 5 right top shows corresponding examples in the dataset 1, and the bottom shows the dataset 2. The difference between dataset 1 and 2 is human objects sizes and locations. The larger cropping images, which is included in the dataset 2, have more flexibility rather that that of the smaller set. Thus, the dataset 2 is considered to be the most difficult for human detection in our prepared datasets, since the size and location is further from the normalized images provided by Dalal [1].

In order to compare performances among the H-HOG and several conventional HOGs, we prepare following 3 image groups. One is for creating templates in the $HS2$ layer of the H-HOG. For creating proper templates, we use original segmented positive images in the INRIA person dataset [11]. The other 2 groups,

Table 1 Result for the dataset 1, which has small variations of human object locations in normalized 100 × 200 [pixels^2]. Each column shows, detecting accuracy, input dimension for classifier, time for classifier learning, time for classifier testing, and required memory size respectively. The memory size is indicated as required page size, which has 4,096 [KBytes/pages].

	Accuracy[%]	# Dimension	time [sec]		memory [pages]
			Learn	Test	
HOG$_5$	88.2	55,404	140.80	2.92	7.02×10^6
HOG$_{10}$	88.2	11,664	27.88	0.55	1.49×10^6
HOG$_{15}$	78.3	3,565	8.42	0.16	5.33×10^5
HOG$_{20}$	82.2	1,944	4.58	0.07	5.48×10^5
HOG$_{25}$	78.3	972	2.31	0.04	5.52×10^5
H-HOG	86.3	4,000	8.73	0.12	5.63×10^5
H-HOG$_{sel}$	86.8	2,431	5.34	0.07	5.56×10^5
H-HOG$^{non}_{sel}$	88.7	3,306	7.04	0.08	5.55×10^5

Table 2 Result for the dataset 2, which has large variations of human object locations in 300 × 600 [pixels^2]. Each column shows same as Table 1.

	Accuracy[%]	# Dimension	time [sec]		memory [pages]
			Learn	Test	
HOG$_5$	71.5	55,404	140.82	2.89	7.01×10^6
HOG$_{10}$	75.3	11,664	27.77	0.55	1.48×10^6
HOG$_{15}$	77.5	3,565	8.42	0.14	5.39×10^5
HOG$_{20}$	76.7	1,944	4.59	0.08	5.44×10^5
HOG$_{25}$	80.5	972	2.28	0.03	5.53×10^5
H-HOG	83.3	4,000	8.54	0.11	5.70×10^5
H-HOG$_{sel}$	83.0	2,431	5.25	0.06	5.56×10^5
H-HOG$^{non}_{sel}$	86.5	3,306	7.01	0.10	5.55×10^5

which are prepare from dataset 1 and 2, are used for SVM classifier learning and evaluation. These 3 groups do not share any images.

For obtaining templates in the $HS2$ layer of the H-HOG, we prepare 200 inputs images for template creation. The contents of these images are 100 positive examples that involve human object, and 100 negatives that does not involve. For creating proper templates, we use scale and location normalized images for positive samples in the same manner with Dalal such like images in Fig. 5 left. In our simulation, our algorithm select 20 templates for each training image, so that $HS2$ layer would have 4,000 templates at a maximum, and similar templates would be discarded in the H-HOG$_{sel}$ and H-HOG$^{non}_{sel}$ by applying templates selection described in Section 3.3.1.

For evaluation, we use both the dataset 1 and 2. In each dataset, 80 images are used for training of the classifier SVM for both the conventional HOG and the H-HOG descriptions. The number of positive images and negatives are 40 images equivalently. We apply the SVM provided from the OpenCV with default parameters [12]. We also prepare another 600 patterns for each dataset in order to evaluate the classification accuracy of the conventional HOG and H-HOG. The positives and negatives are also equivalently included.

4.2 Detection Performance for INRIA Person Dataset

Table 1 shows the result of detection performance for the dataset 1. The dataset 1 has small variations of human object locations in normalized 100 × 200 [pixels^2]. Each column shows detecting accuracy, input dimension size for classifier, spending time for learning and for testing, and required memory size respectively. We investigate these performances over the computer which has following specification: OS: Ubuntu 10.04 LTS, CPU: Xeon E5530 2.4 [GHz], Memory: 24 [GBytes]. In the result Table 1 and **Table 2**, the calculation time for both learning and testing stands for the consuming time for only the SVM classifier. Of course, our H-HOG requires extra time cost for template creation and extraction compared with the conventional HOG system. However, in the conventional HOG, adjusting the object location and scale variances are carried out in the pre-process. Thus, comparing these time costs for feature extractions are difficult, so that, we only evaluate the consuming time for SVM classifier in these tables. The consuming memory size is shown as required page size, which is 4,096 [KBytes/pages]. In the conventional HOG, the small size of cell denoted as HOG$_5$ shows better performance on the accuracy rather than the other HOGs, while the size of input dimension for classifier, required memory and spending time is larger than those of others. Our H-HOG shows also good performance, while the input dimensions is only 4,000 at a maximum. Moreover, H-HOG introduced non-linear modulation and template selection, which is denoted as H-HOG$^{non}_{sel}$, show the best accuracy result for the dataset 1.

Table 2 shows the result of the detection performance for the dataset 2, which has large variations for human object locations and sizes. Each column indicates same as shown in Table 1. The HOG$_5$, which shows the best detecting accuracy in the dataset 1, indicates the worst accuracy in this dataset. So that, the conventional HOG requires designing for adjusting to the detecting object scales. On the contrary, H-HOG also shows the good performance against to the conventional HOGs.

Figures 6 and **7** show the detection error trade-off (DET) curves for dataset 1 and 2 respectively. In each figure, the horizontal axis shows the false positive rate per window (FPPW), and vertical one shows the miss rate [1]. FPPW is calculated as the rate such that (the number of false alarms)/(the total number of testing negative examples). The miss rate is calculated as the (1 − recall rate). Thus the lower DET curve

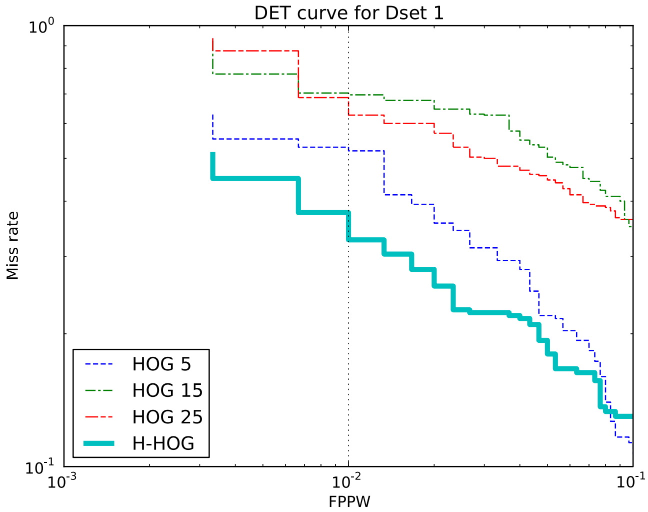

Fig. 6 Detection error trade off (DET) curve for dataset 1. The horizontal axis shows false positive per window (FPPW), and the vertical shows the miss rate. The H-HOG curve shows the lowest miss rate for almost all the FPPW area. The only HOG$_5$ becomes lowest miss-rate with over around 0.08 FPPW.

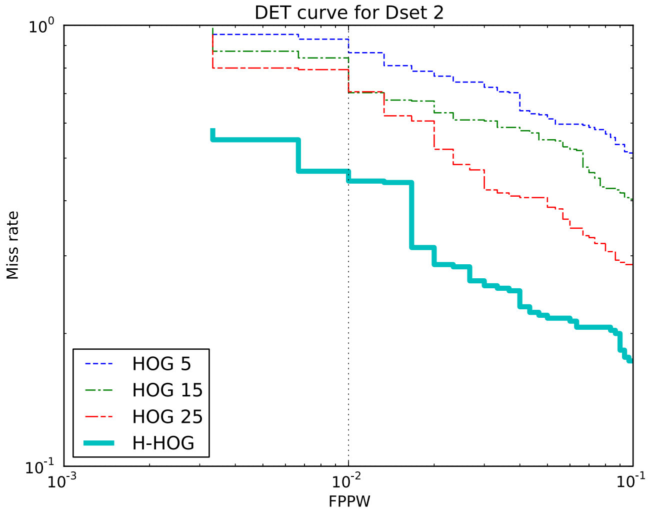

Fig. 7 DET curve for dataset 2. The horizontal and vertical axis is identical to Fig. 6. The H-HOG curve shows the lowest miss rate.

means better performance.

From the DET curves in Fig. 6, we can see the H-HOG shows the best performance in the compared systems under the 0.08 FPPW. Over the value HOG$_5$ shows the better performance rather than the H-HOG.

On the contrary, in Fig. 7 that shows the performance for the dataset 2, HOG$_5$ becomes the worst performance, and the large cell size HOG$_{25}$ becomes better in the conventional HOG. Thus the proper scaling and assignment of the object location is important for the conventional HOG. Our H-HOG keeps better performance in this environment.

5. Conclusion & Discussion

In this study, we propose a hierarchical extension of HOG model, and evaluate the performance about classification. In the conventional HOG models, the HOG$_5$, HOG$_{10}$ shows a good result for the small image dataset, however, the dimension of input vector for the classifiers become over 10,000 dimensions. Generally, the large dimension classification brings several difficulties for classification, that is called 'the curse of dimension'. On the contrary, our H-HOG model can control the dimension, which is the number of the templates selected from multi-resolution representation of the HOG description. We also demonstrate that the performance of our model is as good as that of the HOG$_5$ even in the small dimensions, which is under 4,000 dimensions, of the input vector for classifier. Our H-HOG model shows the best performance for the dataset 2. The dataset 2 have larger variances for the location and size of the human object. The performances of the classification of conventional HOG models are just affected to these variances. The smaller cell size becomes, the worse the classification result becomes. On the contrary, our H-HOG model integrates several resolution size and show robustness for these flexibility. Thus, we consider these hierarchical extension is effective for the generic object recognition for the real world.

Table 3 Performance comparison between raw HOG descriptors and PCA-HOGs for dataset 1.

	Accuracy[%]		# Dimension	
	Raw	PCA	Raw	PCA
HOG$_{10}$	88.2	88.7	11,664	297
HOG$_{25}$	78.3	80.3	972	67
H-HOG$^{non}_{sel}$	88.7	88.3	3,306	130

Table 4 Performance comparison between raw HOG descriptors and PCA-HOGs for dataset 2.

	Accuracy[%]		# Dimension	
	Raw	PCA	Raw	PCA
HOG$_{10}$	75.3	73.7	11,664	285
HOG$_{25}$	80.5	77.7	972	63
H-HOG$^{non}_{sel}$	86.5	87	3,306	121

From the view point of the reducing the input dimension, we can apply to project HOG descriptor into the subspace obtained by principal component analysis (PCA) for the input feature. Lu & Little applied PCA for the HOG descriptor, which is called PCA-HOG, for the video tracking and show the good performance [13]. Thus, we also apply projecting the $HOG_{10,25}$ and the H-HOG descriptors into each PCA subspace and evaluate the performances. **Tables 3** and **4** shows the accuracy and feature descriptor dimensions for dataset 1 and 2 respectively. The general relationship between raw HOG descriptors and PCA projected is not so much differ. For the dataset 1, H-HOG is slightly worse performance rather than that of the HOG$_{10}$ in the condition of low FPPW ($< 9.0^{-3}$), however, HOG$_{10}$ becomes worst in the dataset 2. Thus, we can confirm the robustness of the H-HOG for the scale and location variances. **Figures 8** and **9** show the DET curves for the dataset 1 and 2 respectively. We can see HOG$_{10}$ shows good performance in the small variances of object location and size, however, in the large variances the HOG becomes worst. On the contrary, H-HOG keeps good performance for both environments.

Felzenszwalb et al. proposes a part-based object detection model [14], [15]. This model represents input image in multi-scale resolution by the HOGs. The lower-resolution HOG descriptor plays a roll of rough detection and higher-resolution HOG plays detecting of part of the objects. These descriptors are combined into a feature map in order to find root object locations. The concept of our H-HOG model is similar to this part

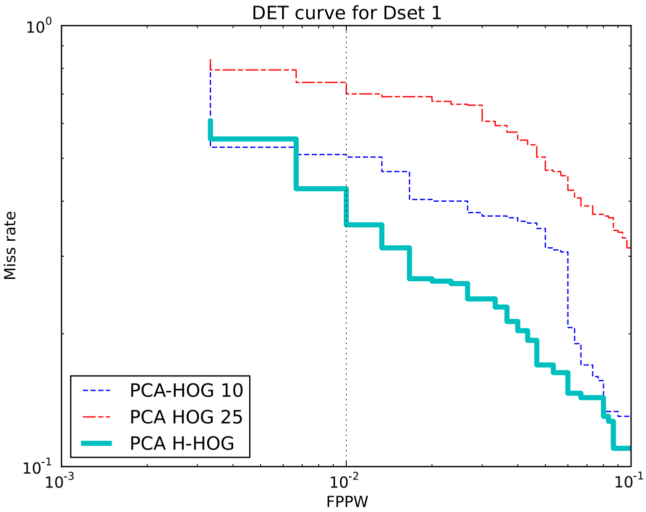

Fig. 8 Detection error trade off (DET) curve for dataset 1 using PCA projected descriptors. The horizontal and vertical axis is identical to Fig. 6.

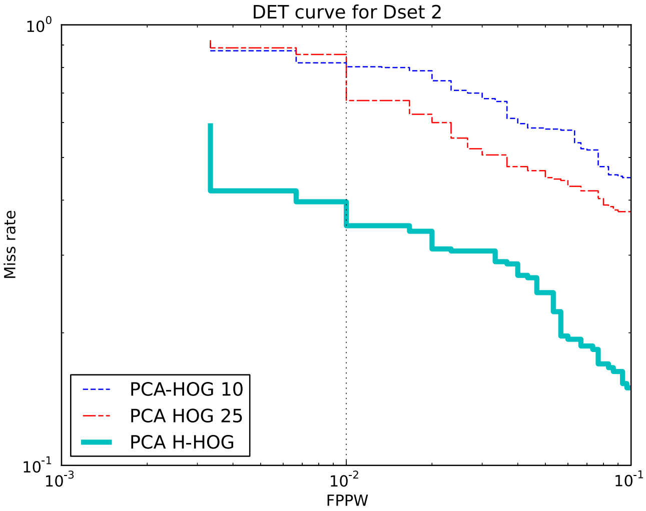

Fig. 9 Detection error trade off (DET) curve for dataset 2 using PCA projected descriptors. The horizontal and vertical axis is identical to Fig. 6.

model except combining these feature into a single map.

Comparison with the conventional convolution-net, such like Neocognitron, Le-Net and Lowe model [6], [7], [8], [10], the most difference points is description manner. The convolution-net often apply Gabor function like filter in order to extract line/edge segment in the image, and sub-sampling, which is called blurring, is carried out for local pattern deformation. In this subsampling process, peripheral information is sum up into a scalar value. In the manner of HOG descriptor, the line/edge segment is represented as an image gradients histogram, and this function is similar to the Gabor filter extraction. However, in the block description of the HOG, peripheral information is not sum up but preserving as a vector description as Eq. (8). The vector representation includes more detail information rather than that of the scalar description, so that, the HOG descriptor might be suitable description for object detection/recognition.

Acknowledgments We thank to Mr. Yuki Noda, graduated student of the Kyushu Institute of Technology. He provided a lot of good advises for this study. This work is supported by Grant-in-Aids for Scientific Research (C) 21500214, and Innovative Areas 21103008, MEXT, Japan.

References

[1] Dalal, N. and Triggs, B.: Histograms of Oriented Gradients for Human Detection, *International Conference on Computer Vision & Pattern Recognition*, Vol.2, pp.886–893 (2005) (online), available from ⟨http://lear.inrialpes.fr/pubs/2005/DT05⟩.

[2] Yanai, K.: The Current State and Future Directions on Generic Object Recognition, *IPSJ Trans. Computer Vision and Image Media*, Vol.48, pp.1–24 (2007) (online), available from ⟨http://ci.nii.ac.jp/naid/110006530765⟩ (in Japanese).

[3] Fujiyoshi, H.: Gradient-Based Feature Extraction : SIFT and HOG, Technical Report 2007-CVIM-160, IPSJ (2007). (in Janese).

[4] Felleman, D.J. and van Essen, D.C.: Distributed hierarchical processing in primate cerebral cortex, *Cerebral Cortex*, Vol.1, pp.1–47 (1991) (online), available from ⟨http://cercor.oxfordjournals.org/cgi/content/abstract/1/1/1-a⟩.

[5] Hubel, D.H. and Wiesel, T.: Sequence Regularity and Geometry of Orientation Columns in Monkey Striate Cortex, *J. Comp. Neurol.*, Vol.158, pp.267–293 (1974).

[6] Fukushima, K.: Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*, Vol.36, No.4, pp.193–202 (1980).

[7] Shouno, H.: Recent Studies around the Neocognitron, *Neural Information Processing, 14th International Conference, ICONIP 2007, Kitakyushu, Japan, November 13-16, 2007, Revised Selected Papers, Part I*, Ishikawa, M., Doya, K., Miyamoto, H. and Yamakawa, T. (Eds.), Lecture Notes in Computer Science, Vol.4984, pp.1061–1070, Springer (2007).

[8] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-Based Learning Applied to Document Recognition, *Proc. IEEE*, Vol.86, No.11, pp.2278–2324 (1998) (online), available from ⟨http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=726791⟩.

[9] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M. and Poggio, T.: Robust Ob ject Recognition with Cortex-Like Mechanisms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.29, No.3, pp.411–426 (2007) (online), available from ⟨http://cbcl.mit.edu/publications/ps/serre-wolf-poggio-PAMI-07.pdf⟩.

[10] Mutch, J. and Lowe, D.: Multiclass Object Recognition with Sparse, Localized Features, *IEEE Computer Sciety Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vol.1, pp.11–18 (online), DOI: 10.1109/CVPR.2006.200 (2006).

[11] Dalal, N.: INRIA Person Data set, available from ⟨http://pascal.inrialpes.fr/data/human/⟩.

[12] Bradski, G.: OpenCV: Free Open Source Computer Vision (2000), available from ⟨http://opencv.willowgarage.com/⟩.

[13] Lu, W. and Little, J.: Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor, *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, p.6 (online), DOI: 10.1109/CRV.2006.66 (2006).

[14] Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A. and Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.32, No.9, pp.1627–1645 (2010).

[15] Dollár, P., Wojek, C., Schiele, B. and Perona, P.: Pedestrian Detection: An Evaluation of the State of the Art, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.34, No.4, pp.743–761 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.155 (2011).

Yasuto Arakaki was born in 1986. He received his B.S. from Ryukyu University in 2009, and M.E. in 2012 from University of Electro-Communications. His research interests are the computer vision and visual procession in the human brain.

Hayaru Shouno was born in 1968. He received M.E. and Ph.D. from Osaka University in 1994 and 1999 respectively. He became a research associate at Osaka University in 1994, and moved to Nara Women's University as a research associate in 2000. He became an associate professor at Yamaguchi University in 2001, and moved to University of Electro-Communications in 2008 as an associate professor. His current research interests are the visual information processing, artificial neural network model, and medical imaging. He is a member of IPSJ, JNNS, and IEICE.

Kazuyuki Takahashi was born in 1982. He received his B.A. from Tokyo Denki University in 2005 and his M.S. and Ph.D. from Kyushu Institute of Technology in 2007 and 2012 respectively. He has been working in Axiohelix Co. Ltd. since 2011. His research interests are the computer vision and the emulation of visual processing of the human brain.

Takashi Morie was born in 1956. He received his B.S. and M.S. from Osaka University and Dr.Eng. from Hokkaido University in 1979, 1981 and 1996, respectively. From 1981 to 1997, he was a member of the Research Staff at NTT Corp. From 1997 to 2002, he was an associate professor at Hiroshima University. Since 2002 he has been a professor at Kyushu Institute of Technology. His main interest is in the area of VLSI implementation of neural networks and brain-inspired image processing. He is a member of IEEE, IEICE, IEEJ, the Japan Society of Applied Physics and the Japanese Neural Network Society.