Hindawi
Advances in Multimedia
Volume 2018, Article ID 9204854, 11 pages
https://doi.org/10.1155/2018/9204854

Research Article

Hand Detection Using Cascade of Softmax Classifiers

Yan-Guo Zhao,[1,2] Feng Zheng,[3] and Zhan Song [1,2,4]

[1] *Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*
[2] *Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China*
[3] *Swanson School of Engineering, The University of Pittsburgh, Pittsburgh, PA 15261, USA*
[4] *CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*

Correspondence should be addressed to Zhan Song; zhan.song@siat.ac.cn

Received 24 October 2017; Revised 15 April 2018; Accepted 23 May 2018; Published 10 July 2018

Academic Editor: Andreas Uhl

Sliding-window based multiclass hand posture detections are often performed by detecting postures of each predefined category using an independent detector, which makes it lack efficiency and results in high postures confusion rates in real-time applications. To tackle such problems, in this work, an efficient cascade detector that integrates multiple softmax-based binary (SftB) models and a softmax-based multiclass (SftM) model is investigated to perform multiclass posture detection in parallel. The SftB models are used to distinguish the predefined postures from the background regions, and the SftM model is applied to discriminate among all the predefined hand posture categories. Another usage of the cascade structure is that it could effectively decompose the complexity of background pattern space and therefore improve the detection accuracy. In addition, to balance the detection accuracy and efficiency, the HOG features of increasing resolutions will be adopted by classifiers of increasing stage-levels in the cascade structure. The experiments are implemented under various scenarios with complicated background and challenging lightings. Results show the superiority of the proposed SftB classifiers over the traditional binary classifiers such as logistic regression, as well as the accuracy and efficiency improvements brought by the softmax-based cascade architecture compared with the noncascade multiclass softmax detectors.

1. Introduction

Hand detection refers to determining the hands location and their shapes. It works as a prerequisite step for various hand gesture recognition systems [1, 2] that have been widely studied, due to their potential application in entertainment and virtual reality [3], medical systems, and assistive technologies, as well as in crisis management and disaster relief [4]. However, hand detection is never an easy task due to the hand deformation [5], the sensitivity of skin colors to lighting conditions [6], and the complicated environments for practical applications. As a result, robust and efficient hand detection remains a challenging task in computer vision community.

Multiclass hand posture detection is worthy of investigation for several reasons: different users may be habituated to using different postures for interaction, many application systems require multiple postures to realize different functions, and robust detection of human hand from multiple viewpoints can be achieved through multiclass hand detection by letting different posture categories represent postures captured under different viewpoints. One way to deal with multiclass hand detection is first to locate the human hand and then to determine the hand shape by classification. Such methods are usually of low accuracy. For example, to locate human hand using skin color cues can be easily affected by the lighting condition and the skin-like background, which will lead to high miss and false rates and will degrade the follow-up classification accuracy/speed in detection. Another example is to train binary classifier for sliding-window-based hand localization, in which all predefined postures are treated as a positive class and the background is regarded as negative class. In this method, the difference in posture shapes increases the pattern complexity

of positive space and resultantly leads to low excluding rate for background. Another way for multiclass hand detection is to build independent detector for each predefined posture and perform multiposture detection by sequentially detecting each of the predefined postures with the corresponding posture detectors [7, 8]. The disadvantage of such practice contains several aspects: (a) the computing cost is high, because multiple rounds of detections are required to find the postures of multiple categories; (b) a window image may be predicted into multiple posture categories, which would result in heavy overlapping detection results; and (c) the multiple detectors are trained independently rather than jointly and in collaboration, which causes confusion detection between different postures easily.

To improve the performance of multiclass hand posture detection system, here in this work, we provide a softmax-based cascade detector that integrates several SftB classifiers at early stages and a SftM classifier at the last stage. Advantages of this proposed method include the following: (a) the softmax-based structure makes it possible to perform multiclass posture detection in parallel; (b) the cascade structure helps decompose the complexity of background pattern space and therefore improve the detection accuracy; (c) the pass-rate of postures and the false rates of background can be adjusted easily by using the binary SftB classifiers (adapted from softmax models) in the first few stages; (d) the SftB-based binary classification is actually made based upon the multiple decision surfaces implied by the softmax model and has a stronger background excluding ability than the binary classifiers trained with examples of all defined posture categories as a single positive class; and (e) with cascaded softmax scheme, the prediction probability across multiple stages can be merged to make final decisions, which helps to reduce the confusion rates between posture categories. Moreover, stage-classifiers of increasing stage-levels will take the HOG features of increasing resolutions to balance the detection accuracy and efficiency. To sum up, the major contribution of this work can be concluded as follows:

(1) A softmax-based cascade architecture is proposed to perform multiclass hand postures detection in parallel and meanwhile to decompose the complexity of background pattern space to improve the detection accuracy.

(2) The SftB classifier is proposed to better distinguish the predefined postures from the background regions, since it could decompose the complexity of multiclass posture pattern space by the multiclass decision boundaries that are learned jointly.

(3) The cascade is designed to take low-resolution HOG features at the lower stages and to use HOG features of higher resolutions for stage-classifiers of higher levels, which helps to balance between the detection accuracy and efficiency.

The remainder of this paper is organized as follows. Section 2 briefly reviews the existing work on vision-based human hand detection problem. The proposed softmax-based cascade architecture is described in Section 3 in detail. Experimental results and discussions are provided in Section 4. Conclusions and future work are offered in Section 5.

2. Related Work

The vision-based hand detection methods can generally be separated into two groups: the appearance-based methods and the 3D-model-based methods [2, 7]. The appearance methods carry out the detection by directly comparing the image features with prebuilt appearance models. These methods are usually of high efficiency, but their performance can be easily affected by viewpoint variation and hand deformation. The 3D methods adopt a kinematic model with high degree of freedom [5, 8]. Such methods offer a richer hand description and therefore could deal with more posture categories, but they are usually computationally expensive due to the complex model matching algorithms. Here in this work, an appearance method is explored to perform the multiclass hand posture detection in parallel.

The key of appearance methods is to seek effective features for hand posture representation as well as to develop an efficient and expressive posture classification model. The frequently used appearance features include the Haar-like [2, 7, 9], HOG [10–12], SIFT [13, 14], and BRIEF [14, 15]. However, such features are seriously affected by the cluttered backgrounds that introduce noise to features encodings. For this reason, recently there are trends to adopt the combination of multiple feature descriptions, such as the integration of HOG and skin features in [16] and the association of Haar-like and HOG in [2]. However, the accuracy improvements for such multifeature methods are usually gained at the expense of considerable increase in computing cost. To improve the efficiency, a classifier of two levels is presented in [1], in which the possible presence of hands is determined from a global perspective in the first level, and then hand regions are precisely delineated at pixel level by a probabilistic model in the second level. And, in [17], the saliency map generated by a Bayesian model is firstly thresholded to localize the hand regions, and then shape and texture features are extracted from the saliency map of hand regions for hand posture recognition. More recently, the deep learning (DL) methods are also investigated for hand posture detection, such as the integration of CNN scheme with fast candidate generation [18], the multiscale deep feature approach [19], and the deep architecture with three networks of sharing convolution layers [20]. However, the speeds of DL-based methods are much lower than those of the classical methods if the algorithms are running on a machine without advanced GPUs.

Multiclass posture detection problem is often addressed by two-stage methods [20–23], in which hand region proposals are firstly obtained by techniques like skin, motion, or saliency detection which are robust to hand deformation and viewpoint variation, and then these regions are classified by multiple binary models or single multiclass model to achieve the final posture recognition. For such methods, precise region proposals are prerequisite to achieve satisfactory recognition rates, while obtaining precise proposals is never an easy job in itself if no specific posture models are

utilized. As a result, the misdetection is often relatively high for such methods. The sliding-window-based methods usually perform the multiclass posture detection with multiple posture-specific detectors [9, 24]. Such methods may have relatively high recall rates. But they lack efficiency since each window needs to be classified by multiple detectors and suffer from heavy confusion detections because the detectors for different categories are trained independently rather than in a coordinated manner. Besides, there are works that adopt tree-type structure [7], but practical experiments show that there is no significant improvement in accuracy or efficiency. Here in this work, we propose a softmax-based cascade detector to perform multiclass hand posture detection simultaneously rather than category by category. Moreover, owing to the multiclass objective function, the decision boundaries are essentially obtained by seeking a balance among all categories and therefore can help reduce the confusion rates among different posture categories.

3. The Proposed Methodology

In this section, the softmax model is firstly presented for multiclass classification. Then, the softmax-based cascade architecture is introduced for multiclass hand posture detection. And, finally, we will show how to apply multiresolution features to the cascade architecture to balance the detection accuracy and efficiency.

3.1. Multiclass Hand Posture Classification by Softmax Regression. Instead of utilizing multiple independent binary classifiers, here in our method, the softmax model [25] is applied to discriminate among the background category and multiple hand posture categories. To be specific, given the feature vector x_z of image z, the distribution of class label $l(z) \in \{p\}_{p=0}^{P}$ can be modeled as

$$p\left(l\left(z\right) \mid x_z; \Theta\right) = y_p\left(x_z; \Theta\right) = \frac{\exp\left(F_p\left(x_z; \Theta\right)\right)}{\sum_{i=0}^{P} \exp\left(F_p\left(x_z; \Theta\right)\right)}$$

$$F_p\left(x; \Theta\right) = \sum_{m=1}^{M} \theta_m^{(p)} \varphi_m\left(x\right) = \left\langle \theta^{(p)}, \varphi\left(x\right)\right\rangle \tag{1}$$

$$\theta^{(p)} = \left(\theta_1^{(p)}, \cdots, \theta_M^{(p)}\right),$$

$$\varphi\left(x\right) = \left(\varphi_1\left(x\right), \cdots, \varphi_M\left(x\right)\right)$$

where $\Theta = \{\{\theta_m^{(p)}\}_{m=1}^{M}\}_{p=0}^{P}$ are model parameters and $\{\varphi_m(\cdot)\}_{m=1}^{M}$ represent basis functions used for feature transformation. $l(z) \in \{1, \cdots, P\}$ means that z is an image of the pth posture category, and $l(z) = 0$ indicates that z is an image of background or undefined postures. In this work, the identity basis functions are adopted; that is, there is $\varphi(x) = x$. For kernelized softmax model, there is $\varphi(x) = (k(x, x_1), \cdots, k(x, x_N))$, where $k(\cdot, \cdot)$ is the kernel function and $\{x_n\}_{n=1}^{N}$ are the features for the training examples. To facilitate the subsequent discussions, the ground-truth label of z is reformulated into a $(P+1)$-dimensional vector as $t = t(z) \in \{0, 1\}^{P+1}$, where its pth element t_p $(0 \le p \le P)$ is equal to 1 if $l(z) = p$ and $t_p = 0$ otherwise. Moreover, we use $y(\cdot; \Theta)$ to denote the softmax model with parameter Θ and use $y(x; \Theta)$ to denote the vector $(y_0(x; \Theta), \cdots, y_P(x; \Theta))$ for simplicity. With these notations, the distribution for label vector t can be formulated as

$$p\left(t \mid x_z; \Theta\right) = \prod_{p=0}^{P} \left\{y_p\left(x_z; \Theta\right)\right\}^{t_p} \tag{2}$$

The model parameter Θ can be obtained by maximal likelihood estimation (MLE) [25, 26]. To be specific, given the training set $\{\mathbf{z} = \{z_n\}_{n=1}^{N}, \mathbf{t} = \{t_n\}_{n=1}^{N}\}$, under the assumption of identical and independent distributions, the likelihood for parameters Θ can be formulated as

$$L\left(\Theta\right) = p\left(\mathbf{t} \mid \mathbf{z}; \Theta\right) = \prod_{n=1}^{N} \prod_{p=0}^{P} \left\{y_p\left(x_n; \Theta\right)\right\}^{t_{np}} \tag{3}$$

where x_n is the feature representation for z_n, t_n is $(P+1)$-dimensional label for example z_n, and t_{np} is the pth component of t_n. In implementation, Θ is acquired by minimizing the negative log-likelihood as follows:

$$-\ln L\left(\Theta\right) = -\sum_{n=1}^{N} \sum_{p=0}^{P} \left\{t_{np} \ln y_p\left(x_n; \Theta\right)\right\}. \tag{4}$$

Since the loss function in (4) remains unchanged as all elements in Θ change in the same proportion, the penalization on Θ should be added to the objective function to suppress the magnitude of model parameters. Therefore, in practice, we take the loss function with regularization term as follows:

$$\mathscr{L}\left(\Theta\right) = -\sum_{n=1}^{N} \sum_{p=0}^{P} t_{np} \ln y_p\left(x_n; \Theta\right) + \lambda \left\|\Theta\right\|_F^2 \tag{5}$$

where $\|\Theta\|_F^2 = \sum_{p=0}^{P} \sum_{m=1}^{M} \|\theta_m^{(p)}\|_2^2$ and λ is the regularization coefficient. Finally, we take the efficient iterative BFGS algorithm [27, 28] to find the solution of (5). Once the model parameters Θ are obtained, the prediction of $l(z)$ can be made based upon the softmax model by

$$\hat{l}\left(z; \Theta\right) = \arg\max_{p} \left\{y_p\left(x_z; \Theta\right); p = 0, \cdots, P\right\} \tag{6}$$

This prediction formula will be slightly modified in the next subsection to carry out two-class classification.

3.2. Softmax-Based Cascade Architecture for Human Hand Detection. For multiscale sliding-window-based hand detection, the background pattern space is highly complicated because of the varied background window images. To decompose the complexity of background space, a softmax-based cascade architecture is introduced, which comprises a set of softmax-based binary (SftB) classifiers $\{B_k(\cdot)\}_{k=1}^{K}$ and a softmax-based multiclass (SftM) classifier $B_{K+1}(\cdot)$. These classifiers are obtained based on the $(K+1)$ softmax regression models $\{y(\cdot; \Theta_k)\}_{k=1}^{K+1}$ which are learned with a cascade training procedure. The classifiers $\{B_k(\cdot)\}_{k=1}^{K}$ with

outputs in $\{0, 1\}$ are mainly used to distinguish the defined hand postures from the background window images, where SftB $B_k(\cdot)$ is formulated as

$$B_k(x) = \begin{cases} 1, & H_k(x) > \xi_k \\ 0, & H_k(x) \le \xi_k, \end{cases} \quad 1 \le k \le K,$$
$$H_k(x) = \max\left(\left\{y_p(x;\Theta_k)\right\}_{p=1}^{P}\right) - y_0(x;\Theta_k) \tag{7}$$

That is to say, for stage k, the window z can be accepted if and only if the maximal probability of posture categories is larger than the probability of background category by at least ξ_k. The parameters $\{\xi_k\}_{k=1}^{K}$ are set to the values so that most windows that properly contain the defined postures can get through, and they are determined at the training stage based upon the settings for posture example pass-rates (for $H_k(\cdot)$, $\Upsilon_{ps}^{(k)} = \{H_k(x_i^{ps})\}_{i=1}^{N_{ps}}$ could be computed based upon the posture examples set $\{x_i^{ps}\}_{i=1}^{N_{ps}}$ which is used for learning Θ_k. Sort $\Upsilon_{ps}^{(k)}$ in ascending order to produce vector $\chi \in \mathrm{R}^{N_{ps}}$, and take the value $\xi_k = \chi(\mathrm{floor}((1-\beta_k)N_{ps}))$ as threshold, where β_k are the preset posture examples pass-rates for the kth stage SftB during training period). The SftM classifier $B_{K+1}(\cdot)$ with output in $\{p\}_{p=0}^{P}$ is of the formulation as described in (6), and it is mainly used to discriminate among the $(P+1)$ categories including the p classes of defined posture and the difficult backgrounds. To speed up the classification, the classifier $\{B_k(\cdot)\}_{k=1}^{K+1}$ can be replaced by the classifiers $\{C_k(\cdot)\}_{k=1}^{K+1}$ defined as follows:

$$C_k(x) = \begin{cases} 1, & \mathscr{R}_k(x) > \zeta_k \\ 0, & \mathscr{R}_k(x) \le \zeta_k, \end{cases} \quad 1 \le k \le K,$$
$$C_{K+1}(x) = \arg\max_p \left\{F_p(x;\Theta_{K+1}); p = 0, 1, \cdots, P\right\} \tag{8}$$
$$\mathscr{R}_k(x) = \max\left(\left\{F_p(x;\Theta_k)\right\}_{p=1}^{P}\right) - F_0(x;\Theta_k)$$

The threshold $\{\zeta_k\}_{k=1}^{K}$ can be determined in a similar way to that in which the threshold $\{\xi_k\}_{k=1}^{K}$ is determined (for $H_k(\cdot)$, $\Upsilon_{ps}^{(k)} = \{H_k(x_i^{ps})\}_{i=1}^{N_{ps}}$ could be computed based upon the posture examples set $\{x_i^{ps}\}_{i=1}^{N_{ps}}$ which is used for learning Θ_k. Sort $\Upsilon_{ps}^{(k)}$ in ascending order to produce vector $\chi \in \mathrm{R}^{N_{ps}}$, and take the value $\xi_k = \chi(\mathrm{floor}((1-\beta_k)N_{ps}))$ as threshold, where β_k are the preset posture examples pass-rates for the kth stage SftB during training period).

The classification of window image z is achieved by a two-step decision process. In the first step, the class label of z is predicted as

$$\widetilde{l}(z) = C_{K+1}\left(x_z^{(K+1)}\right)\prod_{k=1}^{K} C_k\left(x_z^{(k)}\right) \tag{9}$$

where $x_z^{(k)}$ represents the feature representation used by $B_k(\cdot)$. The range of $\widetilde{l}(z)$ is $\{0, 1, \cdots, P\}$. When $\widetilde{l}(z)$ is 0, the window z will be directly excluded, and the second step will not be carried out any more. In the second step, the class label of window image z accepted by (9) is reidentified as

$$\check{l}(z) = \arg\max_p \left\{\gamma_p(z); p = 0, \cdots, P\right\} \tag{10}$$

where $\gamma(z)$ is the $(P+1)$-dimensional score vector calculated using the softmax models at the high-level stages:

$$\gamma_p(z) = \prod_{k=k_0+1}^{K+1} y_p\left(x_z^{(k)};\Theta_k\right) \tag{11}$$

In the experimental part, k_0 is set at 2. For ease of understanding, the flowchart for the window image classification is provided in Figure 1.

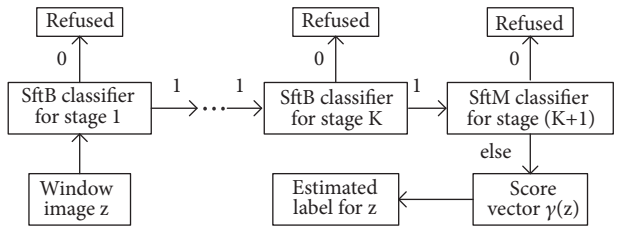


Figure 1: The flowchart of window image classification using softmax-based cascade classifier.

3.3. Multiresolution HOG Feature for Different Stage-Classifiers. For sliding-window-based hand detection, there are tens of thousands window images to be classified in single frame, which makes the detection system lack efficiency. To improve the efficiency, here in this work, the multiresolution HOG features are adopted for posture representations [24]. The cascade is designed so that the HOG features with low resolutions are utilized by classifiers of lower stage-levels, and HOG features with high resolutions are utilized by classifiers of higher stage-levels. The varied feature resolutions can be achieved by adjusting the density of cell splits in window images as discussed in [24]. With such multiresolution scheme, a large number of background windows can be excluded by the classifiers using low-resolution HOG features. And only few difficult background windows need to be further classified by the HOG features of high resolutions which are more discriminative and more computationally costly. In this way, the detection speed can be greatly improved without sacrificing the detection accuracy. Concretely, let J_k denote the time consumption for single window classification with $B_k(\cdot)$, and denote the percentage of windows through the kth stage as follows: ρ_k = number of windows through the first k stage-classifiers/number of all windows generated from the full-sized image. Then, based upon the proposed multiresolution and cascade scheme, the average time expense for classifying one window image is $E_1 = J_1 + \sum_{k=1}^{K} \rho_k J_{k+1}$. However, if the detection system adopts a single softmax with HOG features of the highest resolution, the time expense would be J_{K+1}, which is usually several times as much as E_1.

(1) Prepare multiclass posture example set $\mathcal{X}$ and the full-sized background images set $\mathcal{Z}$. Specify the control factors $\{N_t, N_T\}$, the stage number $K+1$, the HOG resolutions for different stages, the posture samples pass-rates for the first k stages $\{\beta_k\}_{k=1}^{K}$, and the size of train samples $w \times h$. Set the current stage level as $k = 1$, the set of stage-classifiers as $\mathcal{Q} = \{\}$. Note that, all sub-images cropped from full-sized background images are of size $w \times h$ in training process.

(2) Train the first stage classifier as follows:

(2.1) Set $\mathbf{X} = \mathcal{X}$, $\mathbf{Z} = \{N_t$ sub-images randomly cropped from images in $\mathcal{Z}\}$, and $\mathbf{S} = \varnothing$.

(2.2) Train a softmax model with sample sets $\mathbf{X}$, $\mathbf{Z}$ and HOG of specified resolution, and modify the model into two SftB classifiers $B_1(\cdot)$ (Eq. (7)) and $C_1(\cdot)$ (Eq. (8)) based upon the pass-rate β_1.

(2.3) Add $B_1(\cdot)$ and $C_1(\cdot)$ to $\mathcal{Q}$. If $|\mathbf{Z}| > N_T$, go to step (3). Otherwise, go to step (2.4). Here $|\mathbf{Z}|$ represents the number of examples in $\mathbf{Z}$.

(2.4) Randomly crop sub-image z from an image I queried from $\mathcal{Z}$. Add z to $\mathbf{S}$ if $B_1(x_z^{(1)}) = 1$. Repeat this process until $|\mathbf{S}|$ reaches to N_t.

(2.5) Reset $\mathbf{Z} = \mathbf{Z} \cup \mathbf{S}$ and $\mathbf{S} = \varnothing$. And go to step (2.2).

(3) Train the remaining k stage classifiers:

(3.1) Set example sets $\mathbf{X}$ and $\mathbf{Z}$ as: $\mathbf{X} = \{z \in \mathcal{X}; (\prod_{i=1}^{k} B_i(x_z^{(i)})) \neq 0\}$, $\mathbf{Z} = \varnothing$.

(3.2) Randomly crop z from image $I \in \mathcal{Z}$. Add z to $\mathbf{S}$ if $(\prod_{i=1}^{k} B_i(x_z^{(i)})) = 1$. Repeat this process until $|\mathbf{Z}|$ reaches to the predefined N_T.

(3.3) Train a softmax model with sample sets $\mathbf{X}$, $\mathbf{Z}$ and HOG of specified resolution. Then, modify this model into SftB classifiers $B_{k+1}(\cdot)$ and $C_{k+1}(\cdot)$ based on pass-rate β_{k+1}, if $k+1 \leq K$.

(3.4) Add $B_{k+1}(\cdot)$ and $C_{k+1}(\cdot)$ to $\mathcal{Q}$, and let $k = k+1$.

(3.5) If $k \leq K$, go to (3.1). Otherwise, cascade training has been finished and the procedure could be stopped.

Algorithm 1: The procedure of training softmax-based cascade.

To promote the understanding, details of the training process for the proposed method are described in Algorithm 1. In Step (1), the training data is prepared and some hyperparameters are defined to control the training process. In Step (2), the first stage-classifier is trained, while the rest of stage-classifiers are trained one by one in Step (3). During training of the first stage, the initial N_t negatives are randomly cropped, and all the rest are acquired using hard example mining techniques (Step (2.4)). Such strategy could enhance the discriminative ability of the first stage-classifier. For stage larger than 1, all the N_T negatives are directly mined based upon the previous stage-classifier (Step (3.2)). In the kth stage, the multiclass softmax model is firstly learned, and then based upon the predefined pass-rate hyperparameters β_k, the modified SftB classifiers $B_k(\cdot)$ and $C_k(\cdot)$ can be generated. Once the stage reaches the predefined $K+1$, the procedure could stop and return the set of cascade components $\mathcal{Q}$.

4. Experimental Results and Discussions

The proposed method is evaluated on a dataset that is collected under various scenarios with complex background and challenging light conditions. In this section, we firstly describe the dataset and experimental settings. Then, performances of the proposed SftB classifier and softmax-based cascade are evaluated. And, finally, influences of the settings for posture example pass-rates are discussed.

4.1. Datasets and Experimental Settings

4.1.1. Datasets. The experimental dataset comprises four predefined posture categories. For each category, there are around 2000 positive examples with normalized size of 80×80 pixels. The samples are obtained by cropping hand regions from the full images that are collected from ten subjects under various backgrounds and lighting conditions. The negative samples are generated during training process by randomly cropping image regions from 500 extra complicated pictures of full size. These full-sized images comprise various undefined hand postures but contain no hand posture of predefined categories. Except for the training samples, we also prepare 4000 full-sized images to evaluate the performance of the proposed method, and each image contains at least one predefined posture instance. Examples for the defined posture categories are presented in Figure 2.

4.1.2. Experimental Settings. In the experiment, training samples are normalized into the resolution of 80×80 pixels. HOG features of various resolutions are utilized for classification, where different resolutions are achieved by adopting different cell splits. Cell splits for the adopted 3 resolutions are illustrated in Figure 3. Parameters for HOG features of all resolutions are fixed as unsigned gradient orientation, 9 equally distributed angle bins, 2×2 cells per block, and block steps equaling to cell size. Totally four stages-classifiers are incorporated into the softmax structure. The first three are SftB classifiers, and the last one is SftM classifier. Feature configuration for each stage-classifier is presented in Table 1. In addition, to improve the detection efficiency, changing window size is employed for multiscale search rather than resizing the image itself (e.g., we could take window size of 64×64, 80×80, 96×96, and 120×120 to detect hands of different scales in the frame. For window size of $s \times s$, the region of cell $c(i, j)$ will be taken as $[x + x1, x + x2, y + y1, y + y2]$, where (x, y) are the top left coordinates of this window image, $x1 = \text{floor}((i-1) * s/rc) + 1$, $x2 = \text{ceil}(i * s/rc)$, $y1 =$

Table 1: The feature configuration for each cascade stage.

Cascade stage	Cell split	Block layout	Feature dimension	Output domain
Stage 1	5×5	4×4	576	{0, 1}
Stage 2	8×8	7×7	1764	{0, 1}
Stage 3	10×10	9×9	2916	{0, 1}
Stage 4	10×10	9×9	2916	$\{0, \cdots, P\}$

Figure 2: Examples for the four hand posture categories used in our experiments. From the first to the fourth row, the four posture categories are, respectively, denoted as **vict**, **close**, **open**, and **fist**.

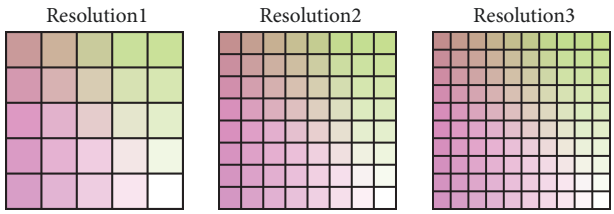


Figure 3: The three adopted HOG resolutions.

floor$((j-1)*s/rc)+1$, $y2 = \text{ceil}(j*s/rc)$, and rc is the cell number at horizontal or vertical direction (totally $rc*rc$ cells as shown in Figure 3). To sum up, the cell size changes with the window size. Although such calculation for cell location is not so accurate when s is not divisible by rc, the feature is still effective. In video-based detection, if the application scenario requires the users to be near the camera, the window sizes should be larger, while if the users are required to stay far away from the camera, the window sizes should be smaller) and the window step is set as 0.05 times of the window size. For live hand detection, the web-camera is set so that image with 320×240 resolution could be captured. All experiments are conducted on a PC equipped with Intel(R) Pentium(R) G3220 @3.00GHz CPU, 4.00GB RAM, and under the visual studio 2013 platform.

4.2. Effectiveness of the Proposed SftB Classifiers. To evaluate the proposed SftB classifier, we, respectively, use the softmax and logistic regression (LR) techniques to train the first three binary stage-classifiers to produce the final four-stage cascade. During the SftB cascade training period, all samples prepared for the pth stage-classifiers $C_{(p,\text{SftB})}(\cdot)$ are divided into the training set $S_{\text{train}}^{(p,\text{SftB})}$ and the testing set $S_{\text{test}}^{(p,\text{SftB})}$. $C_{(p,\text{SftB})}(\cdot)$ is learned from dataset $S_{\text{train}}^{(p,\text{SftB})}$, and ROC curve for $C_{(p,\text{SftB})}(\cdot)$ is calculated based on the testing set $S_{\text{test}}^{(p,\text{SftB})}$ (the ROC describes the variation relation between false positive rates (FPR) and true positive rates (TPR). Different TPR of $S_{\text{test}}^{(k,\text{SftB})}$ are achieved by adjusting the value of threshold ζ_k. And varying ζ_k can in return produce varying FPR on $S_{\text{test}}^{(k,\text{SftB})}$. In this way, the ROC curve for SftB can be produced). Similarly, we can train $C_{(p,\text{LR})}(1 \le p \le 3)$. In this way, totally six ROC curves are produced based upon $\{C_{(p,\text{SftB})}, C_{(p,\text{LR})}\}_{p=1}^{3}$. In addition, an extra ROC curve is also generated for a SftB classifier based upon $\{S_{\text{train}}^{(2,\text{SftB})}, S_{\text{test}}^{(2,\text{SftB})}\}$ and using HOG features of the first resolution. All the seven ROC curves are displayed in Figure 4, where the notations "stage2&Reso2&LR" and "stage2&Reso2&SftB," respectively, represent the LR and SftB classifiers trained with HOG features of the second resolution. Other notations can be explained in a similar way.

From Figure 4, we can see that, with the same HOG resolution and for fixed **TPR** (Table 4), the **FPR** (Table 4) under SftB classifier is much smaller than that calculated with LR classifier. This is because that the SftB is modified from a multiclass classifier, which essentially provides the decision boundaries among different posture categories and therefore can decompose the complex space formed by multiclass posture examples. Moreover, we find that the classifier "Stage2&Reso1&SftB" seriously underperforms the others, which indicates that increasing the resolution of HOG features is crucial to guarantee the classification accuracy.

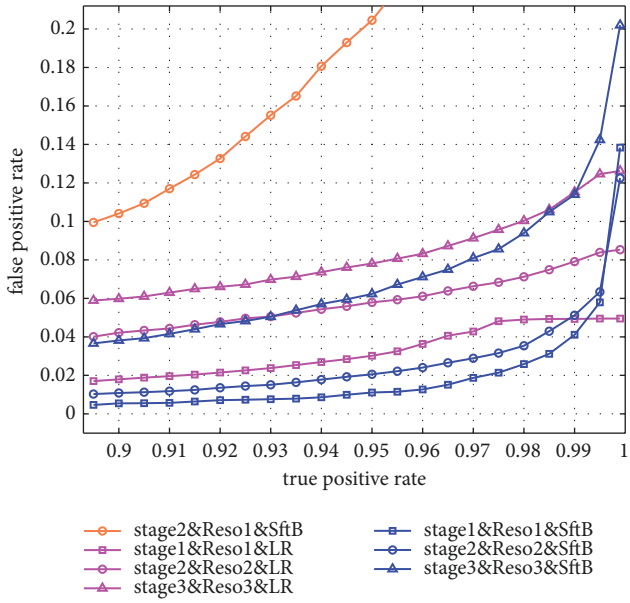


Figure 4: ROC curves for different stage-classifiers which are calculated from the test set generated during the training period.

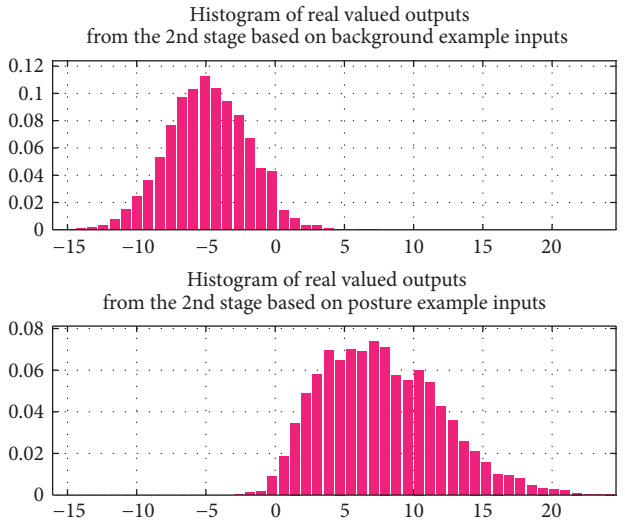


Figure 5: Histograms of output values from $\mathcal{R}_2(\cdot)$ in the second stage of SftB classifier. The upper one is calculated from the background examples and the bottom one is from the hand posture examples.

In addition, the histograms for outputs from $\mathcal{R}_2(\cdot)$ (see (8)) are calculated and presented in Figure 5, so that more knowledge can be gained about the proposed softmax-based binary classification. In the illustration, the upper histogram is calculated based upon the background examples and the bottom one is calculated based upon the predefined hand posture examples.

4.3. Effectiveness of the Proposed Softmax-Based Cascade Detector. To fully evaluate the proposed method, we compare the performance of softmax-based cascade and noncascade detectors based on their confusion matrices. The three compared noncascade softmax detectors are trained, respectively, with each of the three HOG feature resolutions as illustrated in Figure 3. For the cascade detector, posture pass-rates for the first three stage-classifiers are set to 98.0%, 98.5%, and 99.0%, respectively. In practice, the multiclass posture detection is carried out on the full-sized testing images with each of the four detectors (one cascade and three noncascade) and based on the multiscale sliding window scheme. For each detector, all rectangular regions that are classified into a same category will be postprocessed by the nonmaxima suppression techniques to determine the final locations for posture instances. The $(P+1)\times(P+1)$ confusion matrix W for a detector D is computed from the final results produced by detector D. With zero-based indexes, the elements of W are defined as follows:

$$
\begin{aligned}
W(0,0) &:= \frac{\left|\mathbf{S}_{bk}^{(bk)}\right|}{\left|\mathbf{S}_{total}^{(bk)}\right|} \\
W(0,p) &:= \frac{\left|\mathbf{S}_{p}^{(bk)}\right|}{\left|\mathbf{S}_{total}^{(bk)}\right|}, \quad 1 \le p \le P \\
W(p,0) &:= \frac{\left|\mathbf{S}_{bk}^{(p)}\right|}{\mathbf{S}_{total}^{(p)}}, \quad 1 \le p \le P \\
W(p,j) &:= \frac{\left|\mathbf{S}_{j}^{(p)}\right|}{\left|\mathbf{S}_{total}^{(p)}\right|}, \quad 1 \le p,\ j \le P
\end{aligned}
\tag{12}
$$

where $\mathbf{S}_{j}^{(p)}$:= {posture instances that belong to the pth posture category but are predicted into the jth category}, $\mathbf{S}_{bk}^{(p)}$:= {posture instances from the pth posture category but they are not predicted into any of the defined categories}, $\mathbf{S}_{total}^{(p)}$:= {all posture instances from the pth posture category}, $\mathbf{S}_{p}^{(bk)}$:= {all background regions that are predicted into the pth posture category}, $\mathbf{S}_{total}^{(bk)}$:= {all full-sized **pure background images** used for evaluation}, and $\mathbf{S}_{bk}^{(bk)}$:= {all full-sized **pure background images** that do not contain false detections}. The **pure background image** refers to the image that does not contain instances of the predefined posture categories. And a detected region R is the correct detection to an instance $\mathcal{O}$ if and only if the following exist: (a) the predicted class of R is just equal to the ground-truth class of $\mathcal{O}$ and (b) the overlap ratio between R and the ground-truth region of $\mathcal{O}$ is larger than 0.6.

The four confusion matrices corresponding to the four detectors are presented in Table 2, where the Softmax+Resolution1, Softmax+Resolution2, and Softmax+Resolution3, respectively, represent the confusion matrix computed from the three noncascade detectors. Note that the confusion matrix here is different from that for classification problem. In fact, for sliding-window-based detection, one target instance may be covered by many windows, and the postprocessing is only applied to windows that are classified into the same category. As a result, one region can be finally predicted into more than one posture category. For this

Table 2: The confusion matrices for detection results computed with single-resolution-based softmax detectors and multiresolution-based cascade detector. Note that row elements of matrix do not need to sum to 1 for confusion matrix of detection problem.

	Softmax+Resolution1					Softmax+Resolution2				
	BK	vict	close	open	fist	BK	vict	close	open	fist
BK	0.7104	0.1480	0.1211	0.0628	0.1469	0.8205	0.0843	0.0291	0.0326	0.1145
vict	0.0420	0.9136	0.2006	0.2157	0.2585	0.0706	0.9001	0.0492	0.0856	0.1776
close	0.0272	0.5438	0.9251	0.5447	0.4443	0.0017	0.4477	0.9626	0.4077	0.4911
open	0.0532	0.3291	0.2181	0.9070	0.3531	0.0630	0.1987	0.0532	0.9055	0.2234
fist	0.0183	0.7114	0.4862	0.5146	0.9233	0.0342	0.4370	0.1935	0.2494	0.9391
	Softmax+Resolution3					The Proposed Softmax Cascade				
	BK	vict	close	open	fist	BK	vict	close	open	fist
BK	0.8369	0.0889	0.0090	0.0287	0.0983	0.9884	0.0061	0.0022	0.0033	0.0053
vict	0.0761	0.8993	0.0214	0.0928	0.2292	0.0547	0.9294	0.0032	0.0389	0.0151
close	0.0026	0.4323	0.9574	0.3872	0.4655	0.0153	0.0383	0.9838	0.1319	0.0502
open	0.0862	0.1717	0.0187	0.9033	0.2219	0.0300	0.0682	0.0090	0.9700	0.0787
fist	0.0350	0.3361	0.0984	0.1435	0.9299	0.0609	0.1093	0.0459	0.0300	0.8957

Table 3: Performance comparison between the noncascade (rows 1 to 3) softmax and cascade softmax (row 4) detectors.

	Mean recall rate	Mean confusion rate	FPPI	Mean correct rate	Time consumption for 240×320 picture
Softmax+Resolution1	0.9172	0.4017	0.4788	0.3548	25.16ms
Softmax+Resolution2	0.9268	0.2512	0.2605	0.4812	63.39ms
Softmax+Resolution3	0.9225	0.2182	0.2248	0.5155	99.45ms
The proposed softmax cascade	0.9448	0.0515	0.0169	0.8475	27.17ms

Table 4: List of acronyms, definitions, and terminology interpretation.

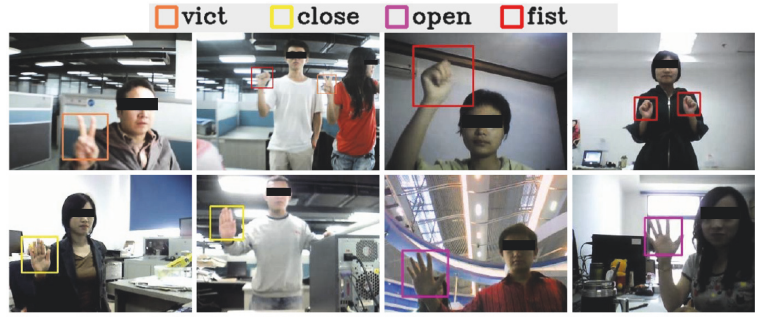
FPPI	$\frac{\text{the number of mis-detected regions from pure background images}}{\text{the number of all full-sized pure background images used for evaluation}}$
FPPW	$\frac{\text{the number of misclassified window images}}{\text{the total number of window images being classified}}$
Mean correct rate	$\frac{\text{the number of full-sized images which comprise no false detections}}{\text{the number of all full-sized images used for evaluation}}$
Detection rate	$\frac{\text{the number of defined posture instances that are predicted into any of the defined posture categories}}{\text{the total number of posture instances that belong to the defined posture categories}}$
Case1	the case in which pass-rate of higher stage is larger than the pass-rate of lower stage
Case2	the case in which the pass-rate of higher stage is smaller than the pass-rate of lower stage
TPR	the abbreviation of *true positive rate*
FPR	the abbreviation of *false positive rate*
LR	the abbreviation of *logistic regression*

reason, the sum of elements in each row does not necessarily equal to one.

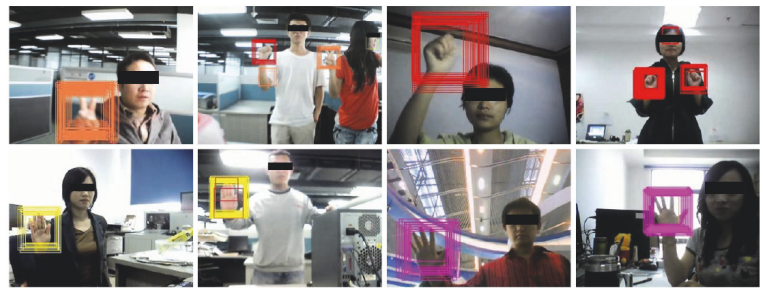
From Table 2, we can see that the hand detection with noncascade softmax detectors may cause high false detection rates at the background areas and high confusion rates among different posture categories. By contrast, the proposed softmax-based cascade could significantly suppress all kinds of false detections without sacrificing the recall rates. This is because the complexity of background space can be effectively decomposed by the usage of multiple stage-classifiers, and therefore it becomes much easier for the final multiclass softmax model to discriminate among the predefined postures and the minorities of remaining backgrounds.

To make more direct and intuitional comparisons, multiple performance values based on summary measures are also computed and provided in Table 3. The measures **mean recall rate** and **mean correct rate**, respectively, represent the averaged recall rates and the averaged confusion rates among the four predefined posture categories. For the definition of **FPPI** and **mean correct rate**, please refer to Table 4.

From Table 3, we can see that the detection accuracy with Softmax+Resolution3 is the highest among the three noncascade classifiers. However, by comparison, the proposed



(a) Detection results after postprocessing



(b) Detection results before postprocessing

FIGURE 6: Detection results based upon the proposed softmax-based cascade detector. Different predictions are marked with rectangles of different colors.

multiclass cascade detector further improves the **mean recall rate** from 0.9225 to 0.9448 and boosts the **mean correct rate** from 0.5155 to 0.8475. Meanwhile, the **mean confusion rate** is reduced from 0.2182 to 0.0515, and the **FPPI** is reduced from 0.2248 to 0.0169. In addition, the proposed detector is faster than Softmax+Resolution3 by almost 4 times.

Figure 6 shows some hand posture detection result based on a normal web-camera. From the results, we can see that the proposed method can detect the defined hand postures under various environments. And the system can reach a real-time running speed of 27 FPS under our experimental setup.

4.4. The Influences of the Settings for Posture Example Pass-Rates. Performance of the proposed cascade is directly affected by the thresholds ζ_k of its stage-classifiers as shown in (8). The thresholds affect not only the detection results but also the training process, since the background samples for the pth stage are acquired by the previous (p-1) stage-classifiers. These thresholds are determined based upon the settings for pass-rates of posture samples (for $H_k(\cdot)$, $\Upsilon_{ps}^{(k)} = \{H_k(x_i^{ps})\}_{i=1}^{N_{ps}}$ could be computed based upon the posture examples set $\{x_i^{ps}\}_{i=1}^{N_{ps}}$ which is used for learning Θ_k. Sort $\Upsilon_{ps}^{(k)}$ in ascending order to produce vector $\chi \in \mathrm{R}^{N_{ps}}$, and take the value $\xi_k = \chi(\mathrm{floor}((1-\beta_k)N_{ps}))$ as threshold, where β_k are the preset posture examples pass-rates for the kth stage SftB during training period) which are set at the training stage to control the training process. To acquire better cascade detector, we prepare multiple groups of settings for the pass-rates and then train the four-stage cascade classifier with each group of settings. After that, the **FPPW** and **detection rate** (Table 4) are computed based upon each of these cascade detectors, and the best group of settings is selected by comparing the values of all **FPPW** and **detection rates**. Note that the **detection rate** does not necessarily equal to the **mean correct rate**, since confusion detections may exist among different posture categories.

The six groups of pass-rates being compared are, respectively, [97%#98%#99%], [99%#98%#97%], [94%#96%#98%], [98%#96%#94%], [95%#97%#99%], and [99%#97%#95%]. The notation [97%#98%#99%] means that, for the first three stage-classifiers, the pass-rate of posture examples is successively set to 97%, 98%, and 99%. Each group contains exact three pass-rate settings because there are exact four stage-classifiers in each cascade detector, while the fourth stage is a multiclass softmax model that will not be modified. The curves for variation relations of **FPPW** with the stage-level are presented in Figure 7, and the **detection rates** are illustrated in Figure 8. Except that **FPPW** and **detection rate** are both increasing with the product of three pass-rates, we have another important observation. That is, when the product of the three pass-rates is fixed, the **detection rate** in **Case1** (Table 4) is significantly higher than that in **Case2** (Table 4), while the **FPPW** in both cases are very close. This indicates that the detectors trained in **Case1** are more discriminative than those trained in **Case2**. This observation suggests that, to achieve good performance, it is better to set low pass-rates for classifiers at low stages and set higher pass-rates for classifiers at higher stages.

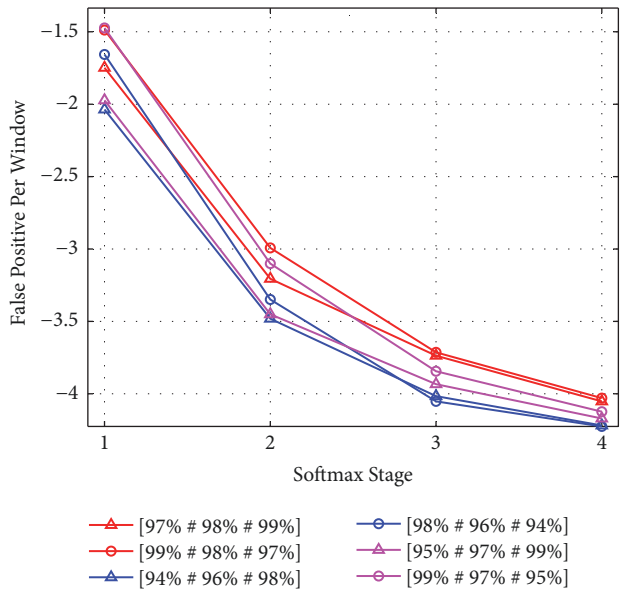


Figure 7: FPPW for cascade with different settings.

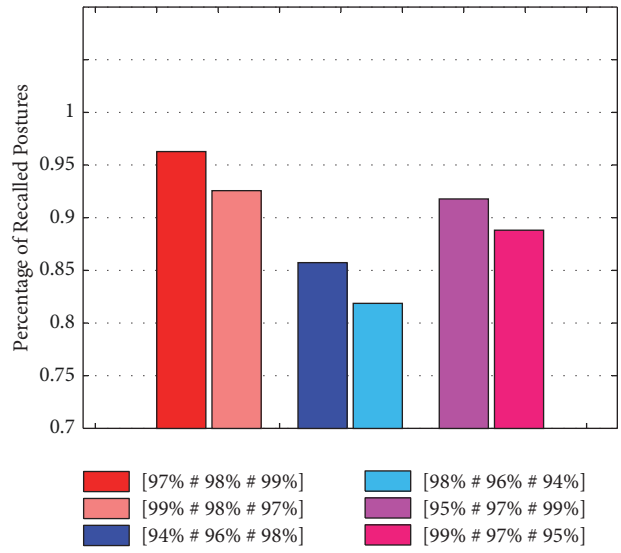


Figure 8: The percentage of detected posture instances.

5. Conclusion and Future Work

In this work, a softmax-based cascade detector is proposed to perform the multiclass hand posture detection in parallel. The cascade contains several SftB classifiers used for distinguishing all predefined postures from the backgrounds and a SftM classifier mainly used to discriminate among all predefined hand postures. Moreover, the HOG features of increasing resolutions are adopted by stage-classifiers with increasing stage-levels so as to further reduce the efficiency without sacrificing the detection accuracy. Experimental comparison of ROC curves demonstrates the superiority of the proposed SftB classifier. And evaluation results on a challenging dataset indicate that the proposed model structure could improve both the accuracy and efficiency as compared with the non-cascade multiclass posture detection methods. In the future work, we will replace the softmax-based stage-classifiers in the cascade with more expressive classification model, such as the convolutional neural networks, to further improve the accuracy of single-stage classification.

Appendix

Acronyms, Definitions, and Terminology

See Table 4.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2017YFB1103602), the National Natural Science Foundation of China (nos. 51705513, U1613213, and U1713213), Shenzhen Science Plan (KQJSCX20170731165108047 and JCYJ20170413152535587), and Shenzhen Engineering Laboratory for 3D Content Generating Technologies (no. [2017]476).

References

[1] A. Betancourt, M. M. López, C. S. Regazzoni, and M. Rauterberg, "A sequential classifier for hand detection in the framework of egocentric vision," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2014*, pp. 586–591, June 2014.

[2] K. Mei, L. Xu, B. Li, B. Lin, and F. Wang, "A real-time hand detection system based on multi-feature," *Neurocomputing*, vol. 158, pp. 184–193, 2015.

[3] J. J. LaViola Jr, "Context aware 3D gesture recognition for games and virtual reality," in *Proceedings of the ACM SIGGRAPH 2015 Courses*, p. 10, ACM, August 2015.

[4] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: a review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[6] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2012.

[7] Q. Chen, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using Haar-like features and a stochastic context-free grammar," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, pp. 1562–1571, 2008.

[8] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[9] M. Kölsch and M. Turk, "Robust hand detection," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 614–619, May 2004.

[10] H. Zhou, D. J. Lin, and T. S. Huang, "Static hand gesture recognition based on local orientation histogram feature distribution model," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2004*, pp. 161-161, IEEE, July 2004.

[11] J. Guo, J. Cheng, J. Pang, and Y. Guo, "Real-time hand detection based on multi-stage HOG-SVM classifier," in *Proceedings of the 2013 20th IEEE International Conference on Image Processing, ICIP 2013*, pp. 4108–4111, IEEE, September 2013.

[12] L. Prasuhn, Y. Oyamada, Y. Mochizuki, and H. Ishikawa, "A HOG-based hand gesture recognition system on a mobile device," in *Proceedings of the IEEE International Conference on Image*, pp. 3973–3977, IEEE, 2014.

[13] C.-C. Wang and K.-C. Wang, "Hand posture recognition using adaboost with sift for human robot interaction," in *Recent Progress in Robotics: Viable Robotic Service to Human*, pp. 317–329, Springer, 2007.

[14] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 3570–3577, June 2013.

[15] C. F. Liew and T. Yairi, "Generalized BRIEF: A novel fast feature extraction method for robust hand detection," in *Proceedings of the 22nd International Conference on Pattern Recognition, ICPR 2014*, pp. 3014–3019, IEEE, August 2014.

[16] A. Mittal, A. Zisserman, and P. H. S. Torr, "Hand detection using multiple proposals," in *Proceedings of the 2011 22nd British Machine Vision Conference, BMVC 2011*, pp. 1–11, September 2011.

[17] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 403–419, 2013.

[18] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the 15th IEEE International Conference on Computer Vision, (ICCV '15)*, pp. 1949–1957, December 2015.

[19] T. H. N. Le, C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Robust hand detection in Vehicles," in *Proceedings of the 23rd International Conference on Pattern Recognition, ICPR 2016*, pp. 573–578, IEEE, December 2016.

[20] T. Chen, M. Wu, Y. Hsieh, and L. Fu, "Deep learning for integrated hand detection and pose estimation," in *Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 615–620, IEEE, December 2016.

[21] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[22] Y. Chuang, L. Chen, and G. Chen, "Saliency-guided improvement for hand posture detection and recognition," *Neurocomputing*, vol. 133, pp. 404–415, 2014.

[23] S. Li, Z. Ni, and N. Sang, "Multiple-classifiers based hand gesture recognition," in *Chinese Conference on Pattern Recognition*, pp. 155–163, Springer, 2016.

[24] Y. Zhao, Z. Song, and X. Wu, "Hand detection using multi-resolution HOG features," in *Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012*, pp. 1715–1720, IEEE, December 2012.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[26] C. K. Enders, "Maximum likelihood estimation," *Encyclopedia of Statistics in Behavioral Science*, 2005.

[27] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.

[28] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 2006.