Multi-Class Multi-Instance Boosting for Part-Based Human Detection

Yu-Ting Chen [1,2], Chu-Song Chen [1,3], Yi-Ping Hung [1,2,3], and Kuang-Yu Chang [1,2]
[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan.
[2] Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.
[3] Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan.
song@iis.sinica.edu.tw

Abstract

With the purpose of designing a general learning framework for detecting human parts, we formulate this task as a classification problem over non-aligned training examples of multiple classes. We propose a new multi-class multi-instance boosting method, named MCMIBoost, for effective human parts detection in static images. MCMIBoost has two benefits. First, training examples are represented as a set of non-aligned instances, so that the alignment problem caused by human appearance variation can be handled. Second, instead of learning part detectors individually, MCMIBoost learns a unified detector for efficient detection, and uses the feature-sharing concept to design an efficient multi-class classifier. Experiment results on MIT and INRIA datasets demonstrate the superior performance of the proposed method.

1. Introduction

Human detection is essential for many applications, such as visual surveillance, driver-assistance systems, and content-based image retrieval. However, it is more challenging than other detection problems in visual surveillance, such as faces and vehicles. It is because that humans have a large variation of appearances caused by different factors, e.g. human postures, illumination conditions, and view points (as shown in Figure 1).

The foregoing works mainly focused on detecting full-body humans [22, 28, 4, 2]. This approach may limit its practical use when occlusion occurs. Recently, some researches [20, 19, 30] began to investigate part-based human detection to deal with this problem. They considered humans as an assembly of distinct parts. Detectors are learned independently to identify candidate parts in an image and determine if they define a human together.

However, all of the above methods demand aligned training examples to learn good detectors with high detection accuracy via supervised learning. This problem becomes severer in aligning human images because humans have various appearances. Recently, *Multiple Instance Learning* (MIL) has been proved with its capability of handling this problem. MIL is a variant of supervised learning, and its key idea is to provide a different way in constituting training examples; instead of using singleton training examples, examples are organized into positive and negative bags of instances, and each bag may contain many instances [12, 16]. In a positive bag, at least one instance is positive (i.e. object), while in an negative bag, all instances are negative (i.e. non-object). To obtain positive training examples, we know that objects are in images, but the exact locations are unknown. It is therefore suitable to represent the object by a bag of multiple instances (non-aligned human images). Then, MIL can learn which instances in the positive bags are positive, along with a binary classifier [29, 1, 21].

Figure 1. Some human examples of the INRIA person dataset [4]. Though the examples are aligned according to shoulders, various appearances still exist.

In this paper, MIL is employed for part-based human detection with non-aligned training examples. We propose a new multi-class MIL framework, named *multi-class multi-instance boosting* (MCMIBoost). The MCMIBoost learns a unified classifier instead of individual classifiers [20, 19, 30] for all classes, so that the detection efficiency can be increased without compromising accuracy.

1.1. Related Work

There are two types of approaches for human detection in a single image, the holistic and the part-based approaches. Holistic approaches employ a full-body detector to analyze a detection window. For example, Gavrila and Philomin [10] proposed a method to detect pedestrians

2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops
978-1-4244-4441-0/09/$25.00 ©2009 IEEE

in images by extracting edge images and matching them to a hierarchy of shape templates. In [22], Haar wavelets are used as feature descriptor and *support vector machine* (SVM) is employed to learn a detector. Viola and Jones [27] proposed a boosted cascade of rectangle features for fast face detection, which has been extended for walking person detection [28]. Dalal and Triggs [4] proposed gradient-based HOG (Histograms of Oriented Gradients) descriptors, which are fed into an SVM to learn a detector. In [32], Zhu et al. speeded up Dalal and Triggs' work by combining cascaded AdaBoost with HOG features. In [2], Chen and Chen employed heterogeneous features and inserted meta-stages into the cascaded AdaBoost structure, so that inter-stage information can be exploited to further enhance the performance.

A holistic detector may fail to detect humans when occlusion occurs. It is therefore important to develop part-based human detectors. Typically, part-based methods search for a human by looking for its distinct parts and exploring the relationships between parts. In [15], Lin et al. extends the hierarchical template matching method in [10] by decomposing the global shape models into parts and constructing a part-template tree for matching it to images hierarchically. In [8], body plans are proposed for human representation and detection. However, the part detectors rely on looking for pairs of parallel edges, and thus may fail to detect humans in cluttered backgrounds. Except for template matching, classification-based methods are the mainstream of research in recent years. In [7, 24], color- and gradient-based part detectors are learned, and the detected parts are assembled into body plans by dynamic programming. Mohan et al. [20] employed the detection method proposed in [22] to learn part detectors. Then, the detected parts are combined by a linear SVM fusion classifier. The above three approaches only detect humans in frontal or rear views. In [19], [30], and [18], position-orientation histograms, edgelet, and rectangle features are respectively used to learn part detectors by boosting. After part detection, these methods employed a joint probability model to aggregate the detected results. Nevertheless, they rely on the situation that face [19, 30, 18], head [19], or entire human [30] is visible. In [6] and [5], latent SVM and *multiple component learning* (MCL) algorithms are used to automatically learn human components, respectively. However, the learned components are not correspond to semantic human parts.

1.2. Our Approach

Although the part-based approaches can improve the performance of human detection, they require well-aligned training examples that are difficult to acquire or evaluate. In [21], Pang et al. applied *logistic multiple instance boosting* (LMIB) [31] to learn a holistic human detector. LMIB estimates the bag probability of being positive by using the weighted average probability of the instances in the bag. However, this rule is probably unsuitable for object detection as indicated in [29, 1]. It is because a bag is positive if it contains at least one instance being positive, while the average rule will greatly reduce the bag probability when only few instances are objects and the others are non-objects. In [29], Viola et al. proposed *multiple instance boosting* (MILBoost) that can better model the bag probability via Noise-OR model [16], which has been successfully employed for holistic face detection.

MILBoost can provide good results for object detection. However, it cannot be directly applied to part-based human detection because MILBoost is a binary classifier. In this paper, we propose a multi-class MIL framework, MCMIBoost, by extending MILBoost from a two-class predictor to a multi-class one. Unlike previous approaches that learn part detectors individually [19, 30, 18], the MCMIBoost learns a unified detector by employing the idea of feature sharing [26, 14, 11] for efficient detection. Inspired from Real AdaBoost [25], a real-version MCMIBoost is proposed to select more discriminative weak learners. We also adopt the cascaded structure [27] of MCMIBoost detectors to speed up the detection. Finally, we introduce *Probability Combination Classifier* (PCC) to aggregate part detection results and determine a detection window as either "human" or "non-human". Experiment results show that our approach can detect humans with high efficiency and accuracy.

One closely related to our work is Dollar et al.'s MCL [5], which can automatically learn individual human component classifiers and combine these into an overall classifier. Although both of us have been aware that MIL is benefit for part-based human detection, these two works are different in many aspects. First, Dollar et al.'s work detects human components not corresponding to semantic human parts, but our work can detect semantically meaningful parts that have advantages in many applications. Second, we proposed a multi-class MILBoost algorithm, while Dollar et al.'s method remains a two-class one which simply treats MIL as weak learners in AdaBoost. Third, instead of learning component detectors individually by MILBoost, our approach learns a unified detector and uses the feature-sharing concept to design an efficient multi-class classifier.

The remainder of this paper is organized as follows: In Section 2, the real version MILBoost is described. A new multi-class MIL, MCMIBoost, is proposed in Section 3 for part-based human detection. Experiment results are shown in Section 4. Finally, a conclusion is given in Sectioin 5.

2. Real Version MILBoost

2.1. A Review of AnyBoost

We begin with some notations of the standard supervised learning. Let training data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be n training examples $x_i \in \mathcal{X}$ and n corresponding labels $y_i \in \mathcal{Y}$. Consider a binary classification problems with $\mathcal{Y} = \{-1, +1\}$. Given S, the weak learner computes a weak hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$. Recall that the goal of boosting is to learn a strong classifier $G(x) = sign(H(x))$, where H is the confidence value of G:

$$H_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x). \tag{1}$$

In [17], Mason et al. proposed AnyBoost to sequentially select weak learners by gradient descent in function space. The idea is to optimize a specified cost function $\mathcal{C}(H)$ by performing gradient descent on H. H can be considered as an n-D vector where its i-th element H_i is $H(x_i)$. To select the optimal h_t, we initially compute a weight for each example x_i based on $t-1$ selected weak learners: $w_i = -\frac{\partial \mathcal{C}}{\partial H_i}$. Ideally h_t is selected to satisfy $h_t(x_i) = w_i$ for all i. However, it is, in practice, impossible to select such an h_t because we only have finite choices of h_t. Instead, AnyBoost searches for an h_t with the greatest inner product with w_i, so as to most reduce the cost over training examples:

$$h_t = \arg\max_h \sum_{i=1}^{n} w_i h(x_i). \tag{2}$$

Note that, examples with high absolute weights, $|w_i|$, dominate the weak learner selection. After selecting h_t, its coefficient α_t is chosen via line search to minimize the cost:

$$\alpha_t = \arg\min_{\alpha} \mathcal{C}(H_{t-1} + \alpha_t h_t). \tag{3}$$

2.2. MILBoost

In [29], Viola et al. combined AnyBoost with MIL and proposed MILBoost for learning a classifier with non-aligned training examples. In MIL, examples come into positive and negative bags of instances. Each instance x_{ij} is indexed with two indices: i for the bag and j for the instance within the bag. All instances in a bag share a bag label y_i. In MILBoost, the probability of x_{ij} being positive is estimated by the logistic function: $p_{ij} = \frac{1}{1+\exp(-H(x_{ij}))}$. Given p_{ij}, the probability of bag i being positive is approximated by the Noise-OR model [16]: $p_i = 1 - \prod_{j \in i}(1 - p_{ij})$. Under this model, the cost function is defined as the negative log likelihood:

$$\mathcal{C}(H) = -\sum_{i}^{n} (\mathbf{1}_{(y_i=1)} \ln p_i + \mathbf{1}_{(y_i=-1)} \ln(1-p_i)), \tag{4}$$

where $\mathbf{1}_{(z)}$ is the indicator function that equals 1 when z is true and 0 otherwise. According to AnyBoost, the weight of each instance is set as the negative derivative of the cost function with respect to the score of each instance, H_{ij}:

$$w_{ij} = -\frac{\partial \mathcal{C}}{\partial H_{ij}} = -\frac{\partial \mathcal{C}}{\partial p_i}\frac{\partial p_i}{\partial p_{ij}}\frac{\partial p_{ij}}{\partial H_{ij}} = \begin{cases} \frac{p_{ij}(1-p_i)}{p_i} & \text{if } y_i = +1 \\ -p_{ij} & \text{if } y_i = -1 \end{cases} \tag{5}$$

Note that the weights w_{ij} are signed and the sign interprets the label of the instance x_{ij}. A positive instance x_{ij} is assigned with a high weight if it has a high p_{ij} (i.e. close to the target 100%) or low p_i (i.e. far away from 100%). High p_{ij} depicts that x_{ij} is likely to be a true positive. Low p_i indicates that the bag does not have a good prediction yet, and so the algorithm gives high weights to all instances in the bag. As for negative instances, if p_{ij} is predicted incorrectly (i.e. p_{ij} approaches 100%), a high negative weight is assigned. In selecting the weak learner, MILBoost will pay much attention to important instances that have high absolute weights, $|w_{ij}|$. Eq. 2 can be rewritten as:

$$h_t = \arg\max_h \sum_{i,j} w_{ij} h(x_{ij}). \tag{6}$$

After selecting h_t, α_t is chosen according to Eq. 3.

2.3. Real MILBoost

Assume that we have a set of 1-D features, such as rectangle features [27], in the feature pool $\mathcal{F} = \{f_r\}_{r=1}^{R}$ where each f projects a training instance onto a real feature value, $f(x_{ij}) \in \mathbb{R}$. These features are allowed to be selected as weak learners. Typically, MILBoost [29] selects weak learners of binary-valued outputs obtained by thresholding the feature values. However, a disadvantage is that it is probably too crude to discriminate the complex distributions of the positive and negative instances. In supervised learning, AdaBoost has suffered from the same problem, and Schapire et al. [25] suggested the Real AdaBoost to solve the problem. Inspired by [25], we present a real-version MILBoost, named Real MILBoost, to generate more discriminative weak learners.

First of all, Eq. 6 can be converted into a form of optimizing weighted errors as in most boosting algorithms:

$$h_t = \arg\min_h \sum_{i,j} (2 \cdot \mathbf{1}_{(h(x_{ij}) \neq y_i)} - 1)|w_{ij}|. \tag{7}$$

Since $\sum -|w_{ij}|$ is a constant, minimizing Eq. 7 is equivalent to minimizing Eq. 8 as follows:

$$h_t = \arg\min_h \sum_{i,j} \mathbf{1}_{(h(x_{ij}) \neq y_i)}|w_{ij}| \leq Z. \tag{8}$$

In order to minimize the weighted error, a reasonable method might be to minimize its upper bound Z in each

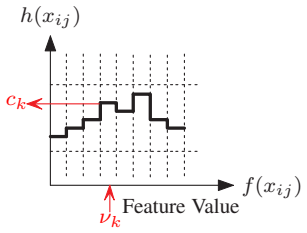


Figure 2. Weak learner used in the real version MILBoost.

round of boosting. Because exponential loss has shown its success in boosting [9], we also employ this loss function in our approach. Eq. 8 can be bounded by:

$$Z = \arg\min_h \sum_{i,j} \exp(-y_i h(x_{ij}))|w_{ij}|. \quad (9)$$

To better represent the distributions of positive and negative instances, the domain space of the feature value is evenly partitioned into K bins, denoted as $\{\nu_k\}_{k=1}^K$. Each bin ν_k has a real-value output c_k. Given an input instance x_{ij} and its feature value $f(x_{ij})$, the weak learner output $h(x_{ij})$ is a mapping $h : x_{ij} \rightarrow \{c_1, \ldots, c_K\}$; if $f(x_{ij})$ is quantized to the bin ν_k, then $h(x_{ij}) = c_k$ as shown in Figure 2. With these notations, the weighted positive and negative histograms of ν_k is evaluated by:

$$W_k^+ = \sum_{x_{ij} \in \nu_k \wedge y_i = +1} |w_{ij}| \quad \text{and} \quad W_k^- = \sum_{x_{ij} \in \nu_k \wedge y_i = -1} |w_{ij}|.$$

Then, Eq. 9 becomes:

$$Z = \arg\min_h \sum_{k=1}^{K} (W_k^+ \exp(-c_k) + W_k^- \exp(c_k)). \quad (10)$$

By setting the derivative of Eq. 10 to be zero, it can be conducted that Z is minimized when:

$$c_k = \frac{1}{2} \ln(\frac{W_k^+}{W_k^-}) \quad \text{and} \quad Z = 2 \sum_{k=1}^{K} \sqrt{W_k^+ W_k^-}. \quad (11)$$

Because the weak learner h can be scaled by any constant, its parameter α can be set as 1 without loss of generality.

3. Multi-Class Multi-Instance Boosting

The Real MILBoost serves as a binary classification problem. It cannot be directly employed for part-based human detection. However, extending Real MILBoost from a two-class to a multi-class problem is not trivial. The most straightforward way is to train different detectors individually for each parts as most previous works did [20, 19, 30]. The work in [30] has proved that this strategy has good performance for part-based human detection. Nevertheless, the method is probably not efficient enough because the computation time is proportional to the number of parts.

Recently, [26, 14, 11] proposed multi-class boosting algorithms named JointBoost, MBHBoost, and VectorBoost, respectively. The key idea of these approaches is to share weak learners among classes. JointBoost shares weak learners directly, where the decision boundaries and the output values are the same for a group of classes, and thus the discriminability of the weak learners is restricted. MBHBoost and VectorBoost are essentially similar. They extended real-valued weak learners to vector-valued ones. Unlike JointBoost, these two methods take advantage of the diverse distributions from different classes of data. By sharing the same feature, a weak learner with vector-valued output is introduced to simultaneously classify each class, so that each class has its own decision boundary and output value.

3.1. MCMIBoost: Confidence Value Evaluation

We extend the Real MILBoost to a new multi-class multi-instance boosting, MCMIBoost, by employing the idea of feature sharing. Assume that a human has M distinct parts denoted as $\mathcal{P} = \{P_m\}_{m=1}^M$. For each part $Q \in \mathcal{P}$, a set of training bags is denoted by $B^Q = \{(X_i^Q, y_i^Q)\}_{i=1}^{|B^Q|} = B^{Q+} \cup B^{Q-}$, where each bag contains instances $X_i^Q = \{x_{ij}^Q\}$ and its label $y_i^Q \in \{+1, -1\}$. B^{Q+} and B^{Q-} are the sets of positive bags (i.e. $y^Q = +1$) and negative bags (i.e. $y^Q = -1$), respectively. Similar to MILBoost, a weight w_{ij}^Q of each instance x_{ij}^Q can be computed initially in each MCMIBoost iteration by Eq. 5.

For each feature $f \in \mathcal{F}$ and its feature value $f(x_{ij})$, a weak learner with vector-valued output is defined as $\mathbf{h}(x_{ij}) = [h^{P_1}(x_{ij}), h^{P_2}(x_{ij}), \ldots, h^{P_M}(x_{ij})]$, where $\mathbf{h}$ is an M-D vector. Each element $h^Q(x_{ij})$ is the output value of part Q computed from its own B^Q and w_{ij}^Q by using Eq. 11: $h^Q(x_{ij}) = \frac{1}{2} \ln(\frac{W_k^+}{W_k^-})$, if $f(x_{ij}) \in \nu_k$. The error upper bound in classifying part Q is $Z^Q = 2 \sum_{k=1}^K \sqrt{W_k^+ W_k^-}$. To evaluate a weak learner $\mathbf{h}$, it is reasonable to use a measurement based on the classification errors of all classes. In MCMIBoost, the overall classification error is defined as: $\mathcal{Z} = \sum_{Q \in \mathcal{P}} Z^Q$. At each iteration t, MCMIBoost searches for an $\mathbf{h}_t$ with the lowest $\mathcal{Z}$ to most reduce the errors among all classes: $\mathbf{h}_t = \arg\min_{\mathbf{h}} \mathcal{Z}$. After selecting T weak learners, MCMIBoost can be expressed as $\mathbf{H}(x_{ij}) = \sum_{t=1}^T \mathbf{h}_t(x_{ij})$, where each element in $\mathbf{H}$, $\mathbf{H}^Q = \sum_{t=1}^T h_t^Q$, is a confidence value corresponding to the part Q prediction, as analogous to Eq. 1.

The MCMIBoost algorithm can provide efficient multi-class detection because all classes share the same feature f and use the same bin ν_k, $f(x_{ij}) \in \nu_k$; thus, only one feature value $f(x_{ij})$ and its bin ν_k are necessary to be computed. Since the output values of all $h^Q(x_{ij})$ have been obtained during training, all $h^Q(x_{ij})$ can be quickly fetched by ν_k. Therefore, the major computation cost in classifica-

tion is the calculation of $f(x_{ij})$ and its bin ν_k. In addition, MCMIBoost typically needs fewer weak classifiers to accomplish multi-class classification than those required by learning each class independently.

3.2. Cascaded MCMIBoost Architecture

To make the detection efficient, we also combine the MCMIBoost with the cascaded structure of Viola and Jones [27] to fast discard blocks not containing human parts. A number of S stages are cascaded, where each stage is realized by an MCMIBoost consisting of several weak learners as depicted in Section 3.1.

In the cascade, the s-th stage classifier is denoted as $\mathbf{H}_s$ ($s = 1, \ldots, S$). Two goals are set to learn each stage $\mathbf{H}_s^Q$: 1) the minimum detection rate θ_s^Q of positive bags B^{Q+}; and 2) the maximum false-positive rate ϕ_s^Q of negative instances $x_{ij} \in B^{Q-}$. Typically, θ_s^Q is set very large to ensure positive instances passing the stage (e.g. higher than 99.9% in our case), and ϕ_s^Q is relatively low to allow a portion of negative instances to be rejected early (e.g. lower than 70% in our case). MCMIBoost described in Section 3.1 is used to select weak learners for each stage, so that the part goals (θ_s^Q, ϕ_s^Q) are satisfied for all $Q \in \mathcal{P}$, and the decision of each part is made by a proper threshold T_s^Q via line search.

We also employ a bootstrap set Λ^Q that contains a huge number of negative instances for part Q. To train stage s, B^{Q+} remains the same, but B_s^{Q-} contains approximately many instances randomly selected from those not rejected by the previous stages in Λ^Q. After learning $\mathbf{H}_s$, the negative instances correctly rejected by $\mathbf{H}_s^Q$ are removed from Λ^Q, and another negative training bag set B_{s+1}^{Q-} is randomly selected from Λ^Q to learn the next stage. The above steps will keep going until the instances remained in Λ^Q are too few to compose a negative training bag set with the same instance cardinality of B^{Q+}. Then, we remove the part Q from $\mathcal{P}$, i.e. $\mathcal{P} = \mathcal{P} - Q$, and restart to learn next stage of $|\mathcal{P}|$ classes until $\mathcal{P}$ is an empty set. In this way, the number of parts in learning $\mathbf{H}_s$ is not fixed and is non-increasing as s becomes larger, reflecting that the difficulties of classifying distinct parts is usually different.

3.3. Probability Combination Classifier

We design a two-stage algorithm for human detection as shown in Figure 3. In the first stage, a block is employed to scan the input image. In each scanned position, the cascaded MCMIBoost depicted in Section 3.2 is used to compute the confidence value of the block with respect to human parts. For each scanned block, if it is accepted by $\mathbf{H}_s^Q$ for all the stages s, we call this block a candidate block of part Q, denoted by b^Q, and its confidence value is defined as that of the last-stage, $\mathbf{H}_s^Q(b^Q)$.

After the first-stage has been done, each position has been with a confidence value of some part if the position is associated with a candidate block of this part. In the second-stage, we propose a *Probability Combination Classifier* (PCC) to explore the geometric relationships among parts and aggregate part candidates for final decision. In this stage, a detection window of a full-body human is defined (see Figure 4 for an example). Within the detection window, a scan range R^Q is specified as a permissible region in which part Q could appear. These ranges define rough geometric constraints of parts for a standing human. To combine the candidate parts, instead of collecting the highest confidence value of each part in its R^Q to learn a post classifier, such as the way suggested in [20], we employ information of all candidate parts in R^Q to better estimate the human part occurrence.

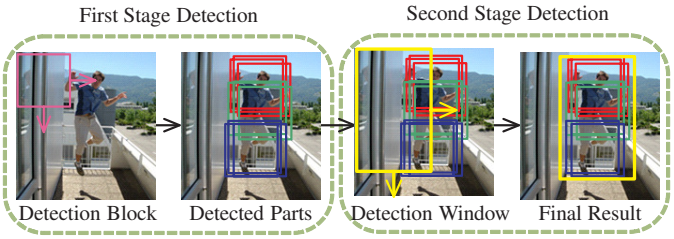


Figure 3. The two-stage detection approach. Assume that a human has three parts: head-shoulder, torso, and legs. In the first stage, a detection block (shown by pink rectangle) is used to scan an image for human parts detection. Red, green, and blue rectangles are the detected candidates of head-shoulder, torso, and legs, respectively. In the second stage, a detection window (shown by yellow rectangle) is used to scan an image and aggregate candidate parts in its R^Q for final decision.

We employ a sliding detection window to scan an image. For each scanned site, if a candidate part b^Q found in the first stage is contained in R^Q, its probability to be part Q is measured by the logistic function: $p_b^Q = \frac{1}{1+exp(-\mathbf{H}_S^Q(b^Q))}$. Based on the Noise-OR model, the probability of R^Q that contains part Q is approximated over all of the detected candidates b^Q in R^Q: $P^Q = 1 - \prod_b (1 - p_b^Q)$. If no candidates are found in R^Q, $P^Q = 0$. Then, PCC adopts the probabilities P^Q of all parts to form an M-dimensional feature vector, and a linear SVM is learned to determine the detection window as either human or non-human.

Finally, we rescale the input image into different sizes, and apply the two-stage procedure to all the scaled images, so that human of unknown size can be handled.

4. Experiment Results

To evaluate the proposed detection approach, two human datasets, MIT [20] and INRIA [4], are adopted in our experiments. The MIT dataset contains 1848 standing human images (924 humans and its reflections) of 64×128 resolution in frontal and rear views. As to the INRIA dataset, the training and testing sets are well designed. The training

Figure 4. The pink and the yellow rectangles in the left image are a detection block and a detection window, respectively. Red, green, and blue rectangles in the other images depict the scan ranges R^Q, $Q \in \{\mathcal{HS}, \mathcal{T}, \mathcal{L}\}$, of human parts: head-shoulder ($R^{\mathcal{HS}}$), torso ($R^{\mathcal{T}}$), and legs ($R^{\mathcal{L}}$).

Figure 5. The i-th positive training image (left) can construct three bags X_i^{Q+} with nine instances in each bag, for the parts $Q \in \{\mathcal{HS}, \mathcal{T}, \mathcal{L}\}$.

set contains 2416 human images of 64×128 resolution and 1218 non-human images. The testing set contains 1132 human images and 453 non-human images. Compared to the MIT dataset, the INRIA dataset is more challenging since it contains people standing in different positions with various orientations and poses.

We assume that a human has three parts: head-shoulder ($\mathcal{HS}$), torso ($\mathcal{T}$), and legs ($\mathcal{L}$), i.e. $\mathcal{P} = \{\mathcal{HS}, \mathcal{T}, \mathcal{L}\}$. The sizes of these three parts are all 48×48 pixels. The defined search ranges R^Q ($Q \in \{\mathcal{HS}, \mathcal{T}, \mathcal{L}\}$) of three parts are specified in Figure 4, and the sizes of the detection block and the detection window are 48×48 and 64×128 pixels, respectively. Each positive bag X_i^{Q+} contains evenly sampled instances of 48×48 pixels in the R^Q of i-th positive training image. More instances in a bag can take care of larger human variations, and thus the detection accuracy can be improved; nevertheless, the disadvantage is that more training time is required. In our experiments, each positive bag contains nine instances as shown in Figure 5.

Because heterogeneous features have shown its capability to detect holistic humans in [2], we employ both the intensity-based rectangle features [27], the gradient-based *Edge Orientation Histogram* (EOH) [13], and *Edge Density* (ED) [23] features to construct the feature pool $\mathcal{F}$. All settings of features follow that of [2]. Within a 48×48 detection block, $\mathcal{F}$ contains 19920 features (9960 rectangle features, 8964 EOH features, and 996 ED features) for learning the MCMIBoost stages, and the domain space of the feature value is evenly divided into 10 disjoint bins (i.e. $K = 10$) for each feature. A bootstrap set, Λ^Q, with over seven million negative instances is collected by sampling sub-images from the 1218 INRIA negative training images.

4.1. Results on the MIT Dataset

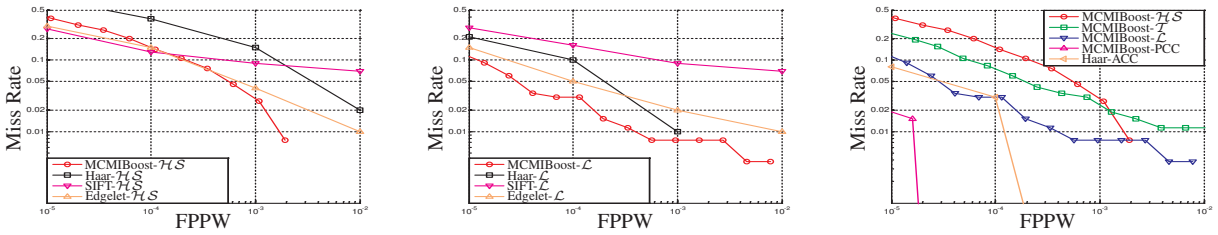
To evaluate our approach on the MIT dataset, we choose the images in 6/7 MIT dataset (1584 images) as positive training images, and the remaining images (264 images) are positive testing images as suggested by Wu [30]. Because MIT dataset does not provide negative images, we adopt INRIA negative images for training and testing.

We begin by investigating the detection performance of human parts detection. The proposed cascaded MCMIBoost is referred to as MCMIBoost-$\mathcal{P}$. In this experiment, θ_s^Q and ϕ_s^Q are set to 99.9% and 50%, respectively, for all $Q \in \mathcal{P}$. Three previous works [20, 19, 30] are compared with our approach. These works are named Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, Edgelet-$\mathcal{P}$, because Haar, SIFT-like and Edgelet features were used for training part detectors, respectively.

With regard to the detection accuracy, we use the *detection error tradeoff* (DET) curves on a log-log scale, i.e., miss rate versus *false positives per windows* (FPPW). To draw the DET curve, the FPPW is varied by applying different thresholds to the confidence value of the last stage of the cascaded detector, $\mathbf{H}_S^Q(x_{ij})$. In this way, the maximum FPPW is restricted since many negative blocks have already been successfully rejected in the previous stages. In our experiments, the maximum FPPW values are about 10^{-3}. We use 10^{-4} as a reference point for comparison, as suggested in [4, 32, 2].

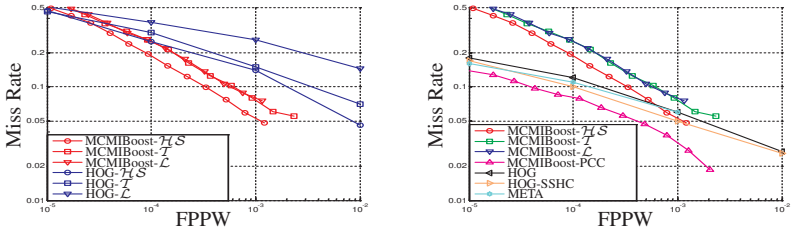
Figures 6(a) and 6(b) show the DET curves of applying MCMIBoost-$\mathcal{P}$, Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, and Edgelet-$\mathcal{P}$ on different human parts, where the curves of Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, and Edgelet-$\mathcal{P}$ are copied from the original papers. However, it shall be noted that direct comparisons based on the MIT dataset is difficult, because different negative images are collected and different ratios of positive examples for training and testing are used in these approaches. Though the examples are not exactly the same, the positive examples are all from the MIT dataset. The results can thus provide a rough comparative evaluation.

Because [20] and [19] did not apply their methods on human torso and [30] did not show their torso detection result, the comparison is made on the head-shoulder and legs parts. The DET curves of MCMIBoost-$\mathcal{P}$, Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, and Edgelet-$\mathcal{P}$ on head-shoulder and legs parts are shown in Figures 6(a) and 6(b), respectively. It can be seen that for the head-shoulder part, the MCMIBoost-$\mathcal{HS}$ performs approximately the same to the other approaches, such as the Edgelet-$\mathcal{HS}$ approach. However, for the legs part, MCMIBoost-$\mathcal{L}$ outperforms the other approaches apparently. This is because that humans in MIT dataset have larger variations in their legs than that in head-shoulder part and MCMIBoost has the capability of handling non-aligned examples.



(a) (b) (c)

Figure 6. The performance of selected detectors on the MIT dataset. (a) and (b) show the DET curves of MCMIBoost-$\mathcal{P}$, Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, and Edgelet-$\mathcal{P}$ on head-shoulder and legs parts, respectively. (c) shows the DET curves of two aggregation results, MCMIBoost-PCC and Haar-ACC. DET curves of MCMIBoost-$\mathcal{P}$ on three human parts are also shown. All curves of Haar-$\mathcal{P}$, SIFT-$\mathcal{P}$, Edgelet-$\mathcal{P}$, and Haar-ACC are copied from the original papers.



(a) (b)

Figure 7. The performance of selected detectors on the INRIA dataset. (a) shows the DET curves of MCMIBoost-$\mathcal{P}$ and HOG-$\mathcal{P}$ on all human parts. (b) shows the DET curves of MCMIBoost-$\mathcal{HS}$, MCMIBoost-$\mathcal{T}$, MCMIBoost-$\mathcal{L}$, MCMIBoost-PCC, HOG, HOG-SSHC, and META. All curves of HOG-$\mathcal{P}$, HOG, HOG-SSHC, and META are copied from the original papers.

Although different results are obtained for parts, the aggregation results are typically better than all of its individual parts. Figure 6(c) shows some aggregation results obtained by our PCC, named MCMIBoost-PCC, and Mohan et al.'s ACC [20], named Haar-ACC. It also shows the three part detection results of MCMIBoost-$\mathcal{HS}$, MCMIBoost-$\mathcal{T}$, and MCMIBoost-$\mathcal{L}$. From this figure, we can see that the proposed MCMIBoost-PCC significantly outperforms Haar-ACC. It can also be seen that legs part has more discrimination than the other parts, and head-shoulder has lowest detection accuracy among all parts, which is consistent with the observations in [20, 30].

After training, there are six stages with only 176 features being selected. For a 320×240 image (containing 6150 detection blocks), the averaged processing speeds of the MCMIBoost-PCC is 8.62 fps (frames per second), by using a PC with a 2.4 GHz CPU. It reveals that incorporating the feature-sharing ability into MIL can result in a highly efficient human detector.

4.2. Results on the INRIA Dataset

In this section, we evaluate the detection performance on the INRIA dataset. Currently, there are few researches employing this challenging dataset for part-based human detection. In [4], Dalal and Triggs proposed a holistic human detection approach with promising detection results. We name this method HOG because HOG descriptors have been employed for human representation. With the success in holistic human detection, HOG has been extended for part-based human detection [3], named HOG-$\mathcal{P}$. To our knowledge, HOG-$\mathcal{P}$ is the only work that has results on this difficult dataset for part-based human detection. Unlike the MIT dataset, results based on the INRIA dataset are more comparable since it has provided common training and testing data for both positive and negative classes.

We start by comparing the detection performance of human parts detection. The DET curves of MCMIBoost-$\mathcal{P}$ and HOG-$\mathcal{P}$ on all parts are shown in Figure 7(a). In training MCMIBoost-$\mathcal{P}$, θ_s^Q and ϕ_s^Q are set to 99.95% and 70%, respectively. In this figure, MCMIBoost-$\mathcal{P}$ has higher detection accuracy than HOG-$\mathcal{P}$ for all parts. Besides, a limitation of HOG-$\mathcal{P}$ is that a high-dimensional feature vector is used to describe each detection block, which needs a somewhat high computation cost.

In [3], the *sparse spatial histograms of classifiers* (SSHC) is proposed to aggregate candidate parts of HOG-$\mathcal{P}$. The SSHC creates a 2-D spatial histogram to encode the spatial locations of the candidate parts, and an SVM is employed as the fusion classifier. This method is named HOG-SSHC. The DET curves of two aggregation approaches, MCMIBoost-PCC and HOG-SSHC, are shown in Figure 7(b). The three part detection results of MCMIBoost-$\mathcal{P}$ are also shown in Figure 7(b), and two holistic-based approaches, HOG [4] and the boosted cas-

cading structure with meta-stages [2] (named META) are also shown for comparison. As can be seen, MCMIBoost-PCC has better performance than HOG-SSHC, and also outperforms the individual part detectors or holistic approaches.

After training, 13 stages and 932 weak learners are obtained in the MCMIBoost-PCC. The averaged processing speed of the MCMIBoost-PCC is 7.63 fps. This result demonstrates that the proposed approach can achieve high detection accuracy with satisfactory efficiency.

5. Conclusion

We have proposed a new multi-class multi-instance boosting, MCMIBoost for effective part-based human detection. In our approach, each training example can be represented as a set of non-aligned instances, and thus the alignment problem caused by human variation can be appropriately handled. We introduced the real-version MILBoost, and proposed a new MIL learning algorithm having the feature-sharing ability in a cascaded structure. We also designed a combination method to cope with the part-integration problem based on the Noise-OR model. Experiment results have shown that effective part-based human detectors can be learned.

Acknowledgment. This work was supported in part by Ministry of Economic Affairs, Taiwan, under Grant No. 97-EC-17-A-02-S1-032.

References

[1] B. Babenko, P. Dollar, Z. Tu, and S. Belongie. Simultaneous learning and alignment: Multi-instance and multi-pose learning. In *Workshop on Faces in Real-Life Images*, 2008.

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

[3] N. Dalal. *Finding People in Images and Videos*. PhD thesis, Institut National Polytechnique de Grenoble, 2006.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] P. Dollar, B. Babenko, S. Belongie, P. Perona, and Z. Tu. Multiple component learning for object detection. In *ECCV*, 2008.

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multi-scale, deformable part model. In *CVPR*, 2008.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.

[8] D. Forsyth and M. Fleck. Body plans. In *CVPR*, 1997.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:337–407, 2000.

[10] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *ICCV*, 1999.

[11] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *IEEE TPAMI*, 29(4):671–686, 2007.

[12] J. D. Keeler, D. E. Rumelhart, and W.-K. Leow. Integrated segmentation and recognition of hand-printed numerals. In *NIPS*, 1990.

[13] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. In *CVPR*, 2004.

[14] Y.-Y. Lin and T.-L. Liu. Robust face detection with multi-class boosting. In *CVPR*, 2005.

[15] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part-template matching for human detection and segmentation. In *ICCV*, 2007.

[16] O. Marson and T. Lozano-Perez. A framework for multiple-instance learning. In *NIPS*, 1998.

[17] L. Mason, J. Baxter, P. L. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. *Technical Report, RSISE, ANU*, 1999.

[18] A. S. Micilotta, E. J. Ong, and R. Bowden. Detection and tracking of humans by probabilistic body part assembly. In *BMVC*, 2005.

[19] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

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

[21] J. Pang, Q. Huang, S. Jiang, and W. Gao. Pedestrian detection via logistic multiple instance boosting. In *ICIP*, 2008.

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

[23] S. L. Phung and A. Bouzerdoum. A new image feature for fast detection of people in images. *IJISS*, 3(3):383–391, 2007.

[24] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *ECCV*, 2002.

[25] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *ML*, 37(3):297–336, 1999.

[26] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE TPAMI*, 29(5):854–869, 2007.

[27] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[28] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.

[29] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005.

[30] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007.

[31] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *PAKDD*, 2004.

[32] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, 2006.