Human Detection for Multiple Pose by Boosted Randomized Trees

Takayoshi Yamashita
OMRON Corpration
SHIGA, Japan
Email: takayosi@omm.ncl.omron.co.jp

Yuji Yamauchi
Chubu University
Aichi, Japan
Email: yuu@vision.cs.chubu.ac.jp

Hironobu Fujiyoshi
Chubu University
Aichi, Japan
Email: hf@cs.chubu.ac.jp

***Abstract*—In this paper we propose a robust pose invariant human detection framework. Most of the existing human detection frameworks assume a standing posture and needing a separate detectors for supporting other human postures. We propose a single framework with a hierarchical tree structure that can detect various poses. The proposed method is based on Randomized trees. Candidate features are selected as shown below, to learn high performing decision trees, 1)each node of the decision tree is constrained with classes based on class likelihood, 2)effective features are pre-selected with Joint Boosting for the above classes, 3)the candidate features are randomly generated based on these effective features. From 1) and 2), the root nodes can be trained for discriminating the human from the background, and leaf nodes can be trained for specific poses. Performance comparison was performed for various poses that arise for a "shopping scenario", and the proposed method outperformed other multi-class classifiers based on Joint boosting, Randomized trees and Adatree.**

I. Introduction

Detecting objects in a scene is the key technology to computer vision, and detecting humans finds application in surveillance, human computer interaction, and human behavior analysis like the customer behavior analysis in a shopping scenario. Recently, detection based on HOG like features and SVM [7], and detection based on statistical learning like, boosting, hierarchical detection framework [9][10][12][13][14], parts based detection [11], methods that use spatial relationship of local features [8] and many more methods have been proposed. Dalal and Triggs[7] proposed a method based on Histogram of Oriented Gradients (HOG) features with SVM for human detection and achieved near perfect performances for standing postures. Zhu et al.[9] realized a fast human detection with a cascade of HOG. Hou et al.[10], proposed a multi-pose framework with vector boosting, which is a hierarchical tree of detector cascade. Duan and Hang [12] studied features for finding human poses. On the other hand, Liebe et al.[8] proposed a bottom-up framework for human detection based on the spatial relationship of local features. Most of the human detection framework constrain the detection to a limited number of poses. Human detection is possible in cluttered background [8], and with occlusion [11] but the number of applications are limited owing to the limitations. For example, detecting human in a shopping scenario for customer behavior analysis would require a real time detection of various poses. Multi-pose detection can be treated as a multi-class classification problem. To solve the multi-pose problem, methods with a hierarchical structure that share features like [6], AdaTree [5], and Randomized Trees [1] are effective.

Joint boosting [6] train weak classifiers that are shared between classes, and hence training classifiers for specific classes is unnecessary, enabling the creation of a multi-class classifier with less number of weak classifiers. To classify for a particular class, using the weak classifiers the entire class class set is an computational overhead, which can be overcome with a hierarchical structure such as a decision-tree. A tree structure that uses branches at each stage for classification and processing time for a particular class depends only on the tree depth. AdaTree [5] also train classifiers in a hierarchical structure, but, they may overfit as each node is created with many weak classifiers. While each node performs binary classification, likelihood of a specific class is not known. Randomized Tree is a tree structure for multi-class recognition [1][2][3][4], it is made up of an ensemble of decision trees in which each decision tree outputs the likelihood for each of the classes. Classification is based on the sum of the likelihoods for all the classes from all the decision trees. But, since the features for each of the node is selected at random, the curse of randomness might affect the feature performance.

In this paper, we propose a human detection based on a hierarchical tree structure that can detect multiple poses, called Boosted Randomized Trees. By introducing a hierarchical structure, the root nodes differentiates background from humans, while the lower nodes detects specific poses. By introducing Joint Boosting each node is trained with features with high classification performance for the associated pose.

II. Related Work and Proposed Method

In the proposed method, candidate feature set is pre-selected with Joint Boosting, and features for each of the nodes are selected at random from the set. Thus a candidate pool of effective features are pre-selected and retaining a degree of randomness. During training, background is trained into a class that discriminates object from the background. After explaining Randomized Trees and Joint Boosting, we will explain Boosted Randomized Trees.

978-1-4577-0121-4/11/$26.00 ©2011 IEEE

A. Randomized Trees

Randomized Trees [1] is a method of multi-class recognition learning that is used in keypoint detection [2] and image classification[4]. It is robust against noise in the training samples, and computational parallelization is possible as all decision trees are independent. It consists of multiple decision trees, T, with branch nodes and terminating leaves. When recognizing C individual classes, each leaf has a probability distribution for each of the classes, $c = 1, 2, ..., C$, and branching at each node is based on a split function. The split function determines if feature $I(x)$ on the left child node is less than the threshold, θ, or if that on the right child node is larger, as shown in Eqn.(1)

$$I(x) = \begin{cases} < \theta & branch\ to\ the\ left\ child\ node \\ \theta & branch\ to\ the\ right\ child\ node \end{cases} \quad (1)$$

The training consists of three processes: creating subsets, generating nodes, and partitioning the subsets. First, subset X_s of the training sample, $X = x_i, c_j; i \in [1, N], j \in [1, C]$, is created to train the decision trees. The subset is a randomly selected set of S sample images. Nodes are made of a split function, a feature and a threshold. For prepared features, $f_m; m \in [1, M]$ and thresholds, $\theta_{m,k}; k \in [1, K]$, the best combination is selected based on information entropy as in Eqn. (2).

$$\triangle E = \frac{|I_l|}{|I|} E(I_l) \quad \frac{|I_r|}{|I|} E(I_r) \quad (2)$$

Note that $E(I_l)$ and $E(I_r)$ are the Shannon entropy for the samples in each class when taking the left or right branch for a given combination of features and thresholds. The Shannon entropy is computed as in Eqn.(3).

$$E(I) = \sum_{j=1}^{C} P(c_j) \log P(c_j) \quad (3)$$

$P(c_j)$ is the probability distribution for class c_j at the node.

Subsets are partitioned by using the features that were selected as was described above. Feature values less than the threshold form the subset for the left child node, and values larger than the threshold form the subset for the right child node. This process is repeated on each child node using the new subsets.

Node generation is terminated when the number of training samples is less than a pre-determined depth, or when the training samples comprise only a single class, or when the nodes have reached a certain depth. Terminating leaves have a probability distribution, $P(c)$, for each class. The probability distribution for class c_j can be computed as in Eqn.(4).

$$P(c_j|l) = \frac{|I_{c_j}|}{|I|} \quad (4)$$

$|I|$ is the number of samples for all classes, and $|I_{c_j}|$ is the number of samples for class c_j.

The input image reaches a single leaf node in each of the decision trees. Then, the probability distributions, $P(C|L_t)$, for each of these leaf nodes, $L = L_t; t \in [1, T]$, are accumulated for each class as in Eqn.(5) and the average is obtained. The class with the highest average probability in Eqn.(5) is output as the recognition class.

$$P(C|L) = \frac{1}{T} \sum_{t=1}^{T} P(C|L_t) \quad (5)$$

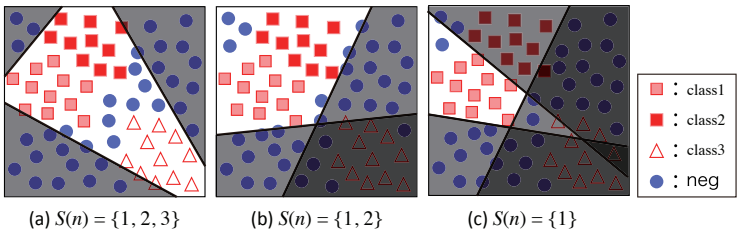

Fig. 1. Learning example of Joint Boosting

B. Joint Boosting

Joint Boosting [6] is a multi-class learning algorithm that enables features to be selected shared between classes. As shown in Eqn.(6), Joint Boosting trains strong classifiers for the partial sets, $S(n)$, of all classes.

$$G^{S(n)}(v) = \sum_{m=1}^{M} h_m^{S(n)}(v, c) \quad (6)$$

Here, $h_m^{S(n)}(v, c)$ is the mth weak classifier, and v is the feature vector. The training process consists of changing the combinations of positive classes and selecting the best weak classifiers as shown in Algorithm 1. Weak classifier $h_m^{S(n)}(v, c)$ with minimum error from all $2^C \quad 1$ is selected. For $S(n) = \{1, 2, 3\}$, a weak classifier for all positive classes is trained as shown in Fig.1(a). Similarly, for $S(n) = \{1, 2\}$, a weak classifier for classes 1 and 2 is trained as shown in Fig.1(b). For $S(n) = \{1\}$, a weak classifier that classifies class 1 is trained as seen in Fig.1(c). The weight of samples in $S(n)$ are updated with Eqn.(7).

$$w_i^c = w_i^c e^{\ z_i^c h_m^{S(n)}(v,c)} \quad (7)$$

Note that $z_i^c \in \{+1, \quad 1\}$ represents the labels of class c. The response of $h_m^{S(n)}(v, c)$ for classes not included in S(n) is 0; hence, the weight is updated for samples in these classes.

C. Boosted Randomized Trees

The flow for generating the nodes of Randomized Trees and Boosted Randomized Trees is outlined in Fig.2. As shown in Fig.2(a), the generating the nodes for Randomized Trees consists of three steps: preparing random features and the threshold, selecting the best combination of features and thresholds, and evaluating sample images to generate child node subsets. Child nodes are generated in four steps with the proposed method, first by defining training classes, second by preselecting features through Joint Boosting, third by optimizing features, and fourth by generating child node subsets. Node

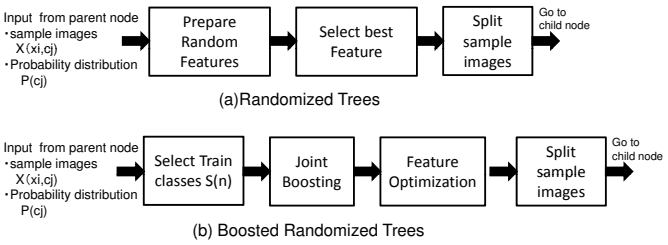

Fig. 2. Training of node of decision tree.

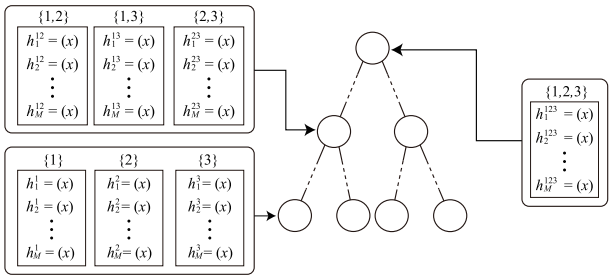

Fig. 3. Feature Selection by Joint Boosting

generation for the proposed method is shown in Algorithm 1. Each of the steps is discussed in the following sections.

1) Defining the class set: Joint Boosting selects a weak classifier for a specific class subset, hence it is not possible to pre-define class sets. The decision trees in the proposed method have a hierarchical structure, with upper nodes handling multiple classes and lower nodes handling specific classes. Thus, weak classifiers for class sets with multiple classes are selected for the upper nodes, and class sets with a specific class are selected for the lower nodes. The best class sets for each node are selected based on the class likelihood as shown in step-2 of Algorithm 1. The class likelihoods for all classes are computed from the probability for each class as shown in Eqn.(4). For a given node, we define the class set, $S(n)$, as a combination of classes n in all class combinations with a total of class likelihoods L, as in Eqn.(8), greater than threshold τ.

$$S(n) \in \{n_1, n_2, ..., n_i : L(n_i) > \tau, i \in I\} \tag{8}$$

Note that n_i is a class element, I is the number of total class elements, and τ is the threshold. The combination with the least number of classes is selected. Upper nodes have more classes with the likelihood for each class being lower and lower nodes tend toward a specific class with the likelihood for the specific class being higher. This removes the need to consider classes with low likelihood for recognition, and class sets only consist of specific classes.

2) Feature pre-selection using Joint Boosting: The best features in Joint Boosting are trained from all combinations of given classes as described in Section 3.2. Because of this, features specific to a particular class might be selected for lower nodes rather than features that are common to multiple classes. In step-3 of Algorithm 1 with the proposed method, feature candidates are trained for limited class sets pre-defined

Algorithm 1 Node Generation Process

Initialization:
1. Inizialize training sample weight w_i^c
$For\ i = 1..N$ //No. of samples
  $For\ c = 1..C$ //No. of classes
    initialize training sample weight w_i^c

Training:
2. Defining a set of classes
  $S(n) \in \{n_1, n_2, ..., n_i : L(n_i) > \tau, i \in I\}$
3. Preliminary feature selection
  3.1. $m = 1, 2, .., M$ //No. of weak classifier selected
  (a) combination of classes : S(n)
    (i)Compute error for all weak classifier candidates
  (b)Select the weak classifier candidate of S(n) with minimal error
  (c)Update the weight of samples w_i^c
  Repeat to obtain weak classifiers, M,
4. Feature Optimization
  Optimize the size and position of weak classifier,
  and select best one, $h_m^{S(n)}(v, c)$, with random threshold as the node

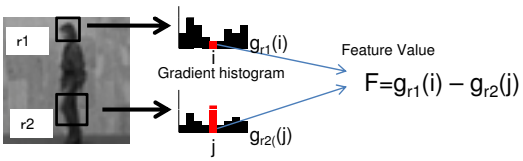

Fig. 4. Local feature extraction.

based on class likelihoods as described in previous section. The case of three classes in Fig.3 in the upper nodes as the training dataset includes many classes, because each class has a high class likelihood. Therefore, features related with many classes are selected, such as $S(n) = \{1, 2, 3\}$. Lower nodes, on the other hand, are trained for a specific class and features specific to a class are selected such as $S(n) = \{1\}$. Thus, by constraining the method of feature selection in Joint Boosting using class sets, hierarchical features can be efficiently selected.

3) Local feature: We extract features based on a histogram of gradients, which is effective for detecting human bodies. There is an outline of the features in Fig.**??**. As seen in Eqn.(9), the feature is the difference between the values, $g_{r_1,t_1}(i)$ and $g_{r_2,t_2}(j)$, which are bins in the gradient histograms of two localized regions, i.e., r_1 and r_2.

$$F = g_{r1}(i) \quad g_{r2}(j) \tag{9}$$

We can capture both changes in gradient differences in a image.

4) Feature optimization: Let us focus on the differences between two local regions, and there are a very large number of combinations of regions. Features are selected in the two steps shown in Fig.5 by pre-selecting feature candidates trained by Joint Boosting, and optimizing features for a node. First, candidate features are trained in step-3 of Algorithm 1 with features generated by grid sampling. Then, the size and position of the features pre-selected with Joint Boosting

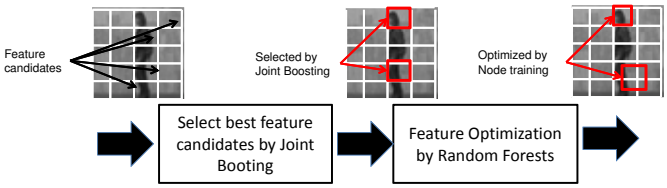

Fig. 5. Feature Optimization by Joint Boosting and Randomized Trees

Fig. 6. Example of our database;a)walk, b)pick up from low, c)pick up from high

are randomly adjusted. The threshold for the split function of nodes is also randomly set. The best combination of features with random adjustment and threshold are selected by using Eqn.(3) in step-4 of Algorithm 1.

5) Human detection by BRTs: During training of Boosted Randomized Trees (BRTs), the training samples along with the class labels from each of the poses have a separate class label for background. Hence each one of the decision trees calculates the posterior for poses and background. A high probability for background class would be classified as a background image, and a high probability for a particular pose would indicate human presence which indicates the proposed method detects humans indirectly.

When Boosted Randomized Trees are trained, the training samples have a separate class label for the background class apart from the classes for each of the poses. Each one of the decision trees output likelihoods for each pose and also for the background. This indicates that a high likelihood for the background class is recognized as a background region, and a high likelihood for a particular pose is detected as human position. Hence, the proposed method not only detects human position but also recognizes human pose. The BRTs search window is made to slide over the image to detect humans at any position in the image.

III. Experiments

A. Experiment overview

Experiment to compare the performance of human detection was carried out. Most of the publicly available database for human detection contains only standing, frontal and profile pose images. In most of the use cases the applications limit the number of poses that can be detected. In this paper the poses involved in a shopping scenario is used for performance comparison. The dataset consisted of walking, taking objects down from an upper shelf, and picking objects up from a lower shelf in the cluttered background shown in Fig.6.

The performance of human detection with the proposed method was compared with Joint Boosting, Randomized Trees, and AdaTree. The each classifier were trained by 1200 images of each pose for 10 different people and 3000 the background images. The region normalized to 48x48 pixels was used for training each of the classes. 10 decision trees were trained for Randomized Trees and Boosted Randomized Trees until the training samples were exhausted or the tree depth reached 15. Randomized Trees prepared 100 candidate features randomly in the whole human region and 100 thresholds were

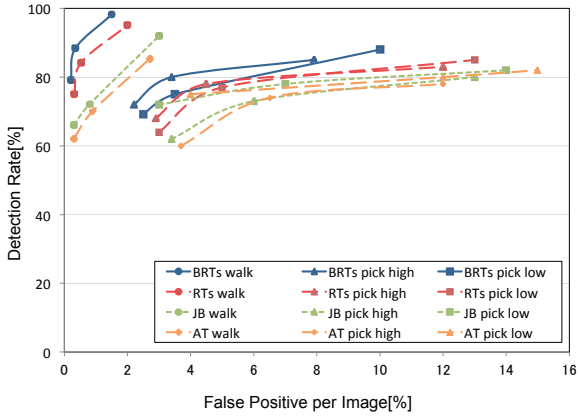

Fig. 7. ROC curve of human detection results.

also prepared randomly for all candidate features. Boosted Randomized Trees prepared 100 candidate features from 10 pre-selected features trained by Joint Boosting with random shift and scaling.

The test dataset has 300 images for each pose with 20 different people. False positive rates included false detection of backgrounds and misclassifications of poses.

B. Experimental result in our database

The proposed method, which was a combination of Randomized Trees and Joint Boosting, efficiently selected features and trained classifiers. The performance of human detection was compared with Joint Boosting and Randomized Trees based approaches for various human poses to demonstrate improvements in detection. The rates for human detection by all the methods are plotted in Fig.7. Compared to Joint Boosting (JB), Randomized Trees (RTs) and AdaTree (AT), Boosted Randomized Trees (BRTs) achieved better detection rates for all poses.

The walking data set contained frontal and side view poses, and even though there were differences in appearance, there were few variations in poses, hence detection rates were high for all the methods. However, subjects picking up objects from the lower shelf had a variety of body postures, creating greater variations in poses. Similarly, subjects taking objects down from the upper shelf also had wider variations in arm inclinations. Joint Boosting selected features that were biased towards a particular class since combinations of classes could not be pre-defined; hence, detection rates for these poses were lower. Randomized Trees selected features at random and

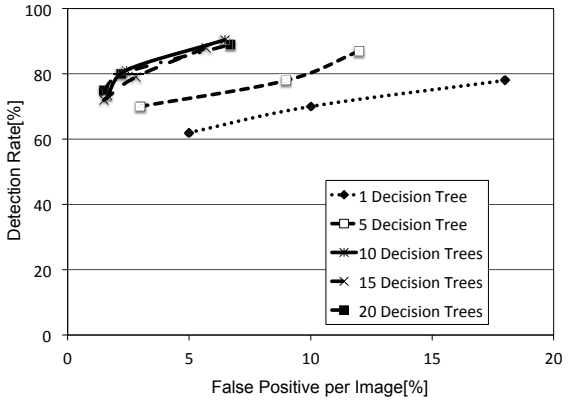

Fig. 8. Performance of Number of trees

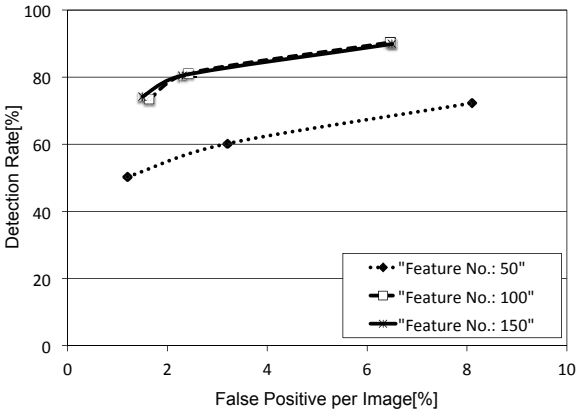

Fig. 9. Feature count and performance

features that could best detect these pose variations might have been ignored. However, as class combinations for the nodes were pre-defined with the proposed method, effective features common between classes were preselected. Also, since it had a tree structure, there could have been multiple leaf nodes for a given class thus allowing variations within a class. The proposed approach was able to improve detection rates for walking, taking down, and picking up poses using Boosted Randomized Trees in this way.

IV. Discussion

A. Relation to Number of Trees

We investigated what effect the number of decision trees would have on performance, and the ROC curve for the number of decision trees is plotted in Fig.8. The rate of human detection in our database with a 2% false positive rate. This indicated that the detection rate improved with increased numbers of decision trees. As the detection rate saturated at around a tree count of 10, this indicated that the optimal tree count was around 10 for the present scenario.

B. Relation to Number of Features

Features in the proposed method were randomly pre-selected with Joint Boosting and candidate features were generated from them. The thresholds for all generated candidate features were also randomly prepared. Performance with our database was based on the number of candidates features and thresholds. Candidate features were generated from 10 pre-selected features from Joint Boosting. The results of human detection for 50, 100 and 150 candidate features are plotted in Fig.9. The thresholds were changed to 50, 100, and 150 and the number of decision trees was set to 10. As a result, the number of candidate features and thresholds were directly proportional to performance, but there were no marked improvements in performance between 100 and 150 features.

V. Conclusion

We proposed a pose invariant human detection framework based on Boosted Randomized Trees. Detection classes were defined based on the likelihood of each class when the nodes of a decision tree were generated. By pre-selecting the effective features of these classes by Joint Boosting, shared features were selected for upper nodes, and specific class features were selected for lower nodes. As a result, the nodes were trained such that the upper nodes detected humans from the background, while the lower nodes detected specific poses. We achieved better performance when we compared our approach with similar methods such as AdaTree and Randomized Trees.

References

[1] L. Breiman, “Random forests”, Machine Learning, No.45(1), pp. 5-32, 2001.
[2] V. Lepetit, P. Lagger and P. Fua, “Randomized Trees for real-time keypoint recognition”, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 775-781, 2005.
[3] P. Geurts, D. Ernst and L. Wehenkel, “Extremely Randomized Trees”, Machine Learning, No.36, Vol.1, pp. 3–42, 2006.
[4] J. Shotton, M. Johnson, R. Cipolla, “Semantic Texton Forests for Image Categorization and Segmentation”, IEEE Conf. on Computer Vision and Pattern Recognition, 2008.
[5] E. Grossmann,“AdaTree: Boosting a Weak Classifier into a Decision Tree”, Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’04) Volume 6, pp.105, 2004.
[6] A. Torralba, K. P. Murphy and W. T. Freeman, “Sharing features: efficient boosting procedures for multiclass object detection”, IEEE Conf. on Computer Vision and Pattern Recognition, 2004.
[7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, IEEE Conf. on Computer Vision and Pattern Recognition, 2005.
[8] B. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes ”, IEEE Conf. on Computer Vision and Pattern Recognition, pp.878?885, 2005.
[9] Q. Zhu, S. Avidan, M. C. Yeh and K. T. Cheng, “Fast human detection using a cascade of histograms of oriented gradients, IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, pp.1491-1498, 2006.
[10] C. Hou, H. Ai and S. Lao, “Multiview Pedestrian Detection Based on Vector Boosting”, Asian Conf. of Computer Vision, pp. 210-219, 2007.
[11] B. Wu and R. Nevatia, “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors”, International Journal of Computer Vision, vol.75, pp.247–266, 2007.
[12] G. Duan, H.Ai and S. Lao, “Boosting associated pairing comparison features for pedestrian detection”, Workshop on Visual Surveillance, 2009.
[13] G. Duan, H.Ai and S. Lao, “A structural filter approach to human detection”, European Conference on Computer Vision, 2010.
[14] C. Huang and R. Nevatia, “Hight performance object detection by collaborative learning of joint ranking of granule features”, IEEE Conf. on Computer Vision and Pattern Recognition, 2010.