MULTITHREADING ADABOOST FRAMEWORK FOR OBJECT RECOGNITION

Jinhui Chen, Tetsuya Takiguchi, Yasuo Ariki

Graduate School of System Informatics, Kobe University, Kobe, 657-8501, Japan
ianchen@me.cs.scitec.kobe-u.ac.jp, {takigu,ariki}@kobe-u.ac.jp

ABSTRACT

Our research focuses on the study of effective feature description and robust classifier technique, proposing a novel learning framework, which is capable of processing multiclass objects recognition simultaneously and accurately. The framework adopts rotation-invariant histograms of oriented gradients (Ri-HOG) as feature descriptors. Most of the existing HOG techniques are computed on a dense grid of uniformly-spaced cells and use overlapping local contrast of rectangular blocks for normalization. However, we adopt annular spatial bins type cells and apply the radial gradient to attain gradient binning invariance for feature extraction. In this way, it significantly enhances HOG in regard to rotation-invariant ability and feature description accuracy; The classifier is derived from AdaBoost algorithm, but it is ameliorated and implemented through non-interfering boosting channels, which are respectively built to train weak classifiers for each object category. In this way, the boosting cascade can allow the weak classifier to be trained to fit complex distributions. The proposed method is valid on PASCAL VOC 2007 database and it achieves the state-of-the-arts performance.

Index Terms— multithreading AdaBoost, Ri-HOG, AUC

1. INTRODUCTION

Object recognition is one of the most crucial components in computer vision. The need for this technology in various different fields continues to propel related researches forward every year. Therefore, its performance has been improved substantially in recent years [1, 2, 3, 4, 5, 6]. The objective of this research is to propose a novel learning framework including high-quality local feature descriptors and robust classifying algorithm, which are often separately researched for object recognition by the precursors. However, we attempt to consider them as the component of a learning framework and combine them together to develop a novel multi-object recognition method. The proposed framework is a robust and simultaneous system and it experimentally outperforms above cited methods, which we briefly review here first.

In this paper, we adopt rotation-invariant histograms of oriented gradients (Ri-HOG) as feature descriptors. It is well known that HOG [7, 8] is a useful tool for object recognition, but the rotational robustness of many algorithms based on HOG does not reach the mature level. In order to address this problem, we subdivide the local patch into annular spatial bins (see Fig 2(a)) to achieve spatial binning invariance. Besides, we apply the radial gradient transform (RGT) to attain gradient binning invariance for feature extraction. The approach is derived from the theory of polar coordinate, which is quite different from previous HOG features in the way that blocks are constructed and cells' gradients are calculated. In this way, it can significantly enhance HOG in regard to rotation-invariant ability and feature descripting accuracy.

The proposed learning model is derived from AdaBoost [9], but it is a novel, multi-class, simultaneous cascade; *i.e.*, a multithreaded one. It is implemented through configuring the AUC (Area under ROC curve) [10] of the weak classifier for each object category into a real-valued lookup list. These non-interfering lists are built into thread channels for the boosting cascade of each object category. In this way, boosting cascade-based approaches can be trained to fit complex distributions and can simultaneously process multi-class events considerably more robustly.

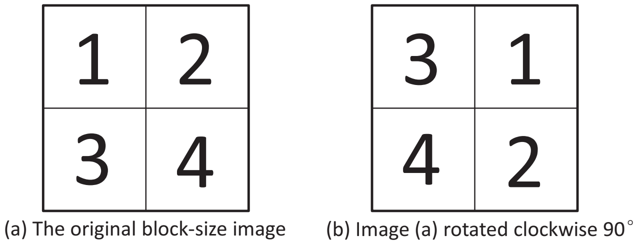
The main contribution of this paper is the novel learning framework for multi-object recognition, by addressing the both above approaches. For evaluation, experiments are carried out on PASCAL VOC 2007 and the proposed method is compared with some well-known methods. The results show our method achieves the state-of-the-arts performance.

2. PROPOSED METHOD

This section describes the proposed framework, which has these ingredients: the Ri-HOG features for local patch description; logistic regression based weak classifiers, which are also combined with AUC as a single criterion for cascade convergence testing; and multithreading cascade for fitting multiplex categories boosting training.

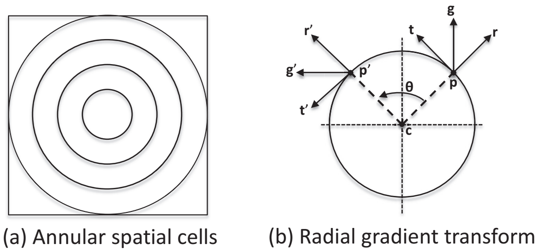
2.1. Feature Description

Background and problems: HOG are feature descriptors, which are computed on a dense grid of uniformly-spaced cells and use overlapping local contrast normalization for improved accuracy. This features set based on *cells* and *blocks*



(a) The original block-size image (b) Image (a) rotated clockwise 90°

Fig. 1. Analyzing the robustness of conventional HOG descriptors in regard to image rotation.



(a) Annular spatial cells (b) Radial gradient transform

Fig. 2. Illustration of rotation-invariant HOG descriptors.

representation system is widely used in object detection, especially human detection. The describing ability of HOG features set outperforms many existing features [8], however, its robustness against image rotation does not reach maturity. Here one direct evidence is that the HOG feature is seldom applied to object tracking or image retrieval successfully. Giving a more scientific reason, see Fig. 1 for an example. Supposing Fig. 1(a) is an image with HOG block size, there are 4 cells in the block. Fig. 1(b) is an image of Fig. 1(a) after making a quarter turn. HOG features are extracted from the two images individually. If the histogram of oriented gradients obtained from the regions 1, 2, 3, and 4 are severally denoted as x_1, x_2, x_3, x_4, then, the HOG features extracted from Fig. 1(a) and Fig. 1(b) are (x_1, x_2, x_3, x_4) and (x_3, x_1, x_4, x_2) respectively. This means that the rotation of image accompanies easily with the change of its HOG descriptors. Hence, we have to substantially enhance the robustness of HOG descriptors. Otherwise applications of HOG features will be limited in some narrow ranges.

Our approach: Now, the question is how to significantly improve the robustness of conventional HOG. In this paper, we use annular spatial cells to replace rectangular ones, furthermore, these cells are computed on a dense radial gradients as feature descriptors to achieve the goal of making HOG be rotation-invariant. How to calculate these descriptors? See Fig. 2(b), $\forall$ point p in circle c, the task is to compute the radial gradient magnitude of point p (x, y). Decompose vector g into its local coordinate system as $(g^T r, g^T t)$, by projecting g into the r and t orientations as shown in Fig. 2(b). Because the component vectors of g in r and t orientations can be obtained by $r = \frac{p-c}{\|p-c\|}$ and $t = R_{\frac{\pi}{2}} r$ quickly, and we can obtain the gradient g easily on the gradient filter. In addition, R_θ is the rotation matrix by angle θ.

The radial gradient is derived from Takacs *et al.*'s approach [11]. Nevertheless, since Takacs *et al.* focus on image tracking applications, the speed is more important, they use Approximate RGT and ROC curve to compute the feature descriptors [11]. However, in so doing, it will decrease the distinctiveness of feature descriptors for recognition applications. In order to keep the distinctiveness of feature descriptors for recognition application, we do not follow Takacs *et al.*'s way to abandon gradient magnitudes, cells, and blocks representation system. Therefore, essentially, the Ri-HOG is an improved HOG feature, but Takacs *et al.*'s method is a novel and excellent feature representation method for image tracking applications, which cannot be considered as a type of HOG feature. Ri-HOG persists and develops the discriminative representation of conventional HOG features. Meanwhile, it also can significantly enhance the descriptors in regard to rotation-invariant ability. Simply, we use the following four steps to extract Ri-HOG descriptors:
1. Subdivide the local patch into annular spatial cells as shown in Fig. 2(a);
2. Calculate RGT $(g^T r, g^T t)$ of each pixel in the cell;
3. Calculate the gradient magnitude and the orientation of RGT using the Eq. 1:

$$\begin{aligned} M_{GRT}(x, y) &= \sqrt{(g^T r)^2 + (g^T t)^2}, \\ \theta(x, y) &= arctan \frac{g^T t}{g^T r}; \end{aligned} \tag{1}$$

4. Accumulating the gradient magnitude of radial gradient for each pixel over the annular spatial cells into 9 bins, which are separated according to the orientation of radial gradient. In this way, we can extract the feature descriptors from a dense annular spatial bin of these uniformly spaced cells.

Block normalization: We tried all of 4 normalization approaches listed by Dalal *et al* in [7]. In practice, $L_2 - Hys$, L_2 normalization followed by clipping is shown working best. The recognition template is 100 × 100 with 10 cells and each cell includes 9 bins. It allows the feature patch size ranging from 50 × 50 pixels to 100 × 100 pixels. We slide the patch over the recognition template with 5 pixels forward to ensure enough feature-level difference. We further allow different aspect ratio for each patch (the ratio of width and height). The descriptors are extracted according to the order from the inside to the outside of cells. Hence, concatenating features in 10 cells together yield a 90-dimensional feature vector.

Now, we still have the question why this feature is rotation-invariant. As shown in Fig. 2 (b), assuming the local patch has been rotated by an any angle θ. It generates a new gradient system: point $p \rightarrow p'$, $R_\theta p = p'$; $R_\theta r =$

r'; $R_\theta t = t'$; $R_\theta g = g'$. We can verify the coordinates of the gradient in point p' can be expressed by $(g^T r, g^T t)$:

$$\begin{aligned}(g'^T r', g'^T t') &= ((R_\theta g)^T R_\theta r, (R_\theta g)^T R_\theta t) \\ &= (g^T R_\theta^T R_\theta r, g^T R_\theta^T R_\theta t) \\ &= (g^T r, g^T t).\end{aligned} \tag{2}$$

All rotated points in the local patch also can obtain their coordinates of the gradient from the corresponding original points, because all gradients are rotated by the same angle θ, they are one-to-one mapping. Thus, the set of gradients on any given circle or annular spatial bin centered around the patch is invariant.

2.2. Weak Classifier Construction

In our previous work [12], we have proposed a look up table model to make AdaBoost be able to train multi-class classifiers simultaneously. But the probability model for the weak classifier was simply calculated on Gaussian function based on Haar-like feature distribution. These lead to low boosting convergence speed and accuracy. In this paper, we build a weak classifier over each local patch described by the Ri-HOG descriptor, and pick optimum patches in each boosting iteration from the patch pool. Meanwhile, we construct the weak classifier for each local patch by logistic regression to fit our classifying framework, due to its linear classifier with probability. Given a HOG feature $\mathbb{F}$ over local patch, logistic regression defines a probability model:

$$P(q|\mathbb{F}, \mathbf{w}) = \frac{1}{1 + exp(-q(\mathbf{w}^T \mathbb{F} + b))}, \tag{3}$$

when $q = 1$ means the trained sample is the positive sample of current class, $q = -1$ means negative samples, $\mathbf{w}$ is a weight vector for the model, and b is a bias term. We will train the classifiers on local patches from large-scale dataset. Assuming in each boosting iteration stage, there are K possible local patches, which are represented by Ri-HOG feature $\mathbb{F}$, each stage is a boosting training procedure with logistic regression as weak classifiers. In that way, the parameters can be found via minimizing the objective,

$$\sum_{k=1}^{K} log(1 + exp(-q_k(\mathbf{w}^T \mathbb{F}_k + b))) + \lambda \left\|\mathbf{w}\right\|_p. \tag{4}$$

λ denotes tunable parameter for the regularzation term, and $\left\|\mathbf{w}\right\|_p$ means L_p norm of the weight vector, $p = 1$ or 2 [13]. We can solve this question on open source LIBLINEAR [13].

2.3. Multithreading Cascade Implementation

In this paper, we have ameliorated our previous approach [12] using the AUC of the weak classifier and construct multi-threaded type training channels to train multi-class classifiers simultaneously. Indeed, our multithreading cascade algorithm is a good way to implement Real AdaBoost [14, 15]. Generally, the variants of Real AdaBoost are the same with Gentle AdaBoost [16], namely, the base learner fits a regression function over training data, and outputs a real value. Assuming there are M object categories in the training sample set, given weak classifiers $h_i^{(n)}$ for category i object, the strong classifier is defined as $H_i^{(N)}(\mathbb{F}) = \frac{1}{N}\sum_{n=1}^{N} h_i^{(n)}(\mathbb{F})$.

Assuming there are total N boosting iteration rounds, in the round n, we will build K weak classifiers $[h_i^{(n)}(\mathbb{F}_k)]_{k=1}^{K}$ for each local patch in parallel from the boosting sample subset. Meanwhile, we also test each model $h_i^{(n)}(\mathbb{F}_k)$ in combination with previous $n-1$ boosting rounds. In other words, we test $H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)$ for $H_i^{(n)}(\mathbb{F})$ on the all training samples, and each test model will produce a highest AUC score [10, 17] $J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k))$. *i.e.*,

$$S_i^{(n)} = \max_{k=1,\cdots K} J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)). \tag{5}$$

This procedure is repeated until the AUC score is converged, or the designed number of iterations N is reached. Then, the selected S_i is set as a threshold to generate an AUC score pool, which contains the values of $J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)) \geq 0.8 \times S_i$. In this way, it will build an AUC score pool for each one class of object.

In order to learn multi-class classifiers simultaneously, we adopt these AUC data to construct independent channels for boosting learning. The details are summarized as follows:
1. Assuming AUC score pools have been normalized to $[0, 1]$, we divide the range into M sub-range bins. Each bin corresponds to a channel ID. In this way, we can obtain a channel ID set $\mathbf{C} = \{bin_j = [\frac{(j-1)}{M}, \frac{j}{M}] |\, j = 1, \cdots, M\}$. In each channel, we will build an independent boosting model for training the classifiers of a corresponding object category;
2. Set $u = S_i(\mathbb{F}, x)$ and define the weak classifier $h_i(x)$ as follows:

$$\begin{aligned}&\textbf{if}\ u \in\ \mathbf{C}\ and\ x \in \{category\ i\ samples\}, \\ &\textbf{then}\ h_i(x) = 2P(q|\mathbb{F}, \mathbf{w}) - 1.\end{aligned} \tag{6}$$

These will guarantee the precision of h is more than 0.5;
3. Given the characteristic function

$$B^{(j,i)}(u, \mathbf{Y}) = \left\{ \begin{array}{ll} 1 & u \wedge \mathbf{Y} = i \\ 0 & otherwise \end{array} \right., \tag{7}$$

where $i \in \mathbf{Y}$, and $\mathbf{Y}$ is defined as the label set of those categories that the classifier h can recognize. This function is used to check and ensure the categories among the channel, classifier and sample are consistent;
4. Covering the characteristic function, finally, we can formally express the weak classifier as:

$$h(\mathbb{F}) = \sum_{j=1}^{M}\sum_{i=1}^{M} (2P(q|\mathbb{F}, \mathbf{w}) - 1)\, B^{(j,i)}(u, \mathbf{Y}). \tag{8}$$

Table 1. Comparison with state-of-the-art methods on PASCAL VOC 2007 database.

	Accuracy of different object-category items (%)																				
	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV	mAP
UCI [1]	28.8	56.2	3.2	14.2	**29.4**	38.7	48.7	12.4	16.0	17.7	24.0	11.7	45.0	39.4	35.5	15.2	16.1	20.1	34.2	35.4	27.1
DPM [6]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
LEO [2]	29.4	55.8	9.4	14.3	28.6	44.0	51.3	21.3	20.0	19.3	25.2	12.5	50.4	38.4	36.6	15.1	19.7	25.1	36.8	39.3	29.6
DSO [4]	32.5	60.1	11.1	16.0	31.0	50.9	59.0	26.1	21.2	**26.5**	25.4	**16.4**	61.7	48.3	42.2	16.1	**28.2**	30.1	44.6	46.3	34.7
CA [3]	34.5	61.1	**11.5**	**19.0**	22.2	46.5	58.9	24.7	**21.7**	25.1	27.1	13.0	59.7	51.6	44.0	**19.2**	24.4	33.1	48.4	49.7	34.8
HoPS [5]	**37.0**	60.7	11.2	18.6	27.8	**54.5**	59.1	**26.9**	20.5	25.8	**29.0**	15.3	**59.9**	49.8	43.0	13.4	23.2	**38.4**	48.8	45.1	35.4
Ours	32.6	**61.5**	5.6	11.3	26.1	**54.5**	**61.7**	15.2	20.0	16.2	25.8	12.5	59.5	**52.2**	**47.1**	10.6	19.7	20.1	**52.0**	**50.1**	**37.9**

Using the above approaches, M independent channels can be constructed. Meanwhile, the classifier category is able to be judged and auto-selected into the related channel. In this way, we can train the classifiers of each expression simultaneously in its training channel via boosting cascade. In order to improve boosting convergence speed and accuracy, we do not use the source code of Open CV, but using the released codes of Li *et al*'s cascade model to adopt Ri-HOG and implement our boosting cascade in each channel.

3. EXPERIMENTS

Databases and Implementation Details: Our framework is implemented in C++ on the PC with Core i7-2600 3.40 GHz CPU and 8 GB RAM. In this paper, experiments are evaluated on PASCAL VOC 2007 dataset [18], which includes 9,963 images of 20 different object classes, containing 5,011 training images and 4,952 testing images. It is the most popular dataset for object detection/recognition and many evaluation experiments of state-of-the-art methods are carried out on it. But note that it is very difficult to improve the results on this dataset. The latest top results are only slightly different.

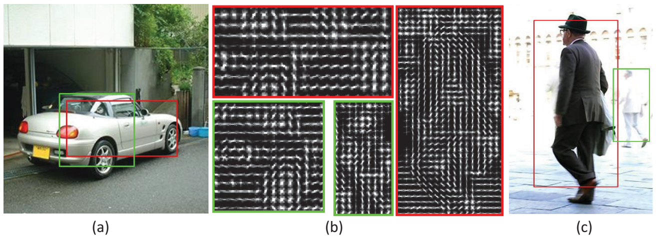


Fig. 3. Top-2 patches picked by training procedure in the red-green order: (a) the example on car object (c) the example on people task (b) the picked image regions of (a) and (b) described by our Ri-HOG descriptors.

Experimental results The proposed method used 265 minutes to converge at the $12th$ boosting iteration stage. The cascade detector generated $5,994$ classifiers of all 20 categories. Only needing to evaluate average 1.3 patches per window (the example is illustrated in Fig 3), the classifier can recognize one object category. On the contrary, 8-bin T2 SURF feature [19, 20] needs average 3 patches; Our previous work [12] requires average 37.3 Haar-like feature patches per window. Hence, the description ability of Ri-HOG efficiently outperforms the other local features on our framework. But the reason why we have to develop Ri-HOG as the feature of our framework is not only the describing efficiency, but also that it dominates others on the accuracy: the mean average precision (mAP) of Ri-HOG, the conventional HOG, SURF, SIFT [21], Haar-like on PASCAL VOC 2007 dataset are 37.9%, 28.1%, 24.6%, 29.4% and 19.7% respectively. Indeed, the proposed framework with SIFT features also can obtain quite good recognition results. However the SIFT's version is not the ideal one, because its recognition speed is only 16 frames per second (FPS), which would limit some real-scene applications.

In Table 1, experimental results are carried out on comparisons of our approach and six state-of-the-art methods, *i.e.*, UCI [1], DPM [6], LEO [2], DSO [4], CA [3], HoPS [5]. Our method achieves the mAP of 37.9%, which is highly competitive to these methods 27.1% [1], 33.7% [6], 29.6% [2], 34.7% [4], 34.8% [3], 35.4% [5]. Therefore, the proposed framework reaches the state-of-the-art performance. In addition, the average recognition speed of the proposed framework can reach 42.8 FPS.

4. CONCLUSIONS

In this paper, we have proposed a novel cascade framework for multiclass objects recognition. The main contribution of this paper is that we present a novel variant of HOG, which has a simple feature extraction system and robust feature descriptors. This is important to those with closely related research interests. Meanwhile, we have developed AdaBoost for simultaneously and robustly computing, which can extend the application range of cascade. About the further work, we will try to apply Ri-HOG to image retrieval; and we also attempt to study the question of further reducing cascade stages and descriptors extracting speed, which would help to improve the boosting convergence speed.

5. REFERENCES

[1] C. Desai, D. Ramanan, and C. Fowlkes, “Discriminative models for multi-class object layout,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sept. 2009, pp. 229–236.

[2] Long Zhu, Yuanhao Chen, A. Yuille, and W. Freeman, “Latent hierarchical structural learning for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1062–1069.

[3] F. Shahbaz Khan, R.M. Anwer, J. van de Weijer, A.D. Bagdanov, M. Vanrell, and A.M. Lopez, “Color Attributes for Object Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3306–3313.

[4] Xiaolong Wang, Liang Lin, Lichao Huang, and Shuicheng Yan, “Incorporating Structural Alternatives and Sharing into Hierarchy for Multiclass Object Recognition and Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3334–3341.

[5] W. Voravuthikunchai, B. Cremilleux, and F. Jurie, “Histograms of Pattern Sets for Image Classification and Object Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 224–231.

[6] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” in *IEEE Transactions on PAMI*, Sept. 2010, vol. 32, pp. 1627–1645.

[7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[8] Qiang Zhu, M.-C. Yeh, Kwang-Ting Cheng, and S. Avidan, “Fast Human Detection Using a Cascade of Histograms of Oriented Gradients,” in *CVPR*, 2006, vol. 2, pp. 1491–1498.

[9] Paul Viola and MichaelJ. Jones, “Robust Real-Time Face Detection,” in *Int. J. Comput. Vis. (IJCV)*, 2004, vol. 57, pp. 137–154.

[10] César Ferri, Peter A. Flach, and José Hernández-Orallo, “Learning Decision Trees Using the Area Under the ROC Curve,” in *Proc. Int. Conf. Machine Learn. (ICML)*, San Francisco, CA, USA, 2002, pp. 139–146.

[11] G. Takacs, V. Chandrasekhar, S.S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, “Fast computation of rotation-invariant image features by an approximate radial gradient transform,” *IEEE Trans. Image Proc.*, vol. 22, no. 8, pp. 2970–2982, Aug. 2013.

[12] Jinhui Chen, Yasuo Ariki, and Tetsuya Takiguchi, “Robust Facial Expressions Recognition Using 3D Average Face and Ameliorated Adaboost,” in *ACM MM*, 2013, pp. 661–664.

[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, “LIBLINEAR: A Library for Large Linear Classification,” in *J. Mach. Learn. Res.*, June 2008, vol. 9, pp. 1871–1874.

[14] Robert E Schapire and Yoram Singer, “Improved boosting algorithms using confidence-rated predictions,” in *Machine learning*. 1999, vol. 37, pp. 297–336, Springer.

[15] Bo Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct 2005, vol. 1, pp. 90–97 Vol. 1.

[16] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, “Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors),” in *The Annals of Statistics*, 04 2000, vol. 28, pp. 337–407.

[17] Phil Long and Rocco Servedio, “Boosting the Area under the ROC Curve,” in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2007, pp. 945–952.

[18] Mark Everingham, Luc Van Gool, ChristopherK.I. Williams, John Winn, and Andrew Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *Int. J. Comput. Vis. (IJCV)*, vol. 88, no. 2, pp. 303–338, 2010.

[19] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “SURF: Speeded Up Robust Features,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, vol. 3951, pp. 404–417.

[20] Jianguo Li, Tao Wang, and Yimin Zhang, “Face Detection Using SURF Cascade,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, Nov 2011, pp. 2183–2190.

[21] DavidG. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” in *Int. J. Comput. Vis. (IJCV)*, 2004, vol. 60, pp. 91–110.