Entropic Selection of Histogram Features for Efficient Classification

Ákos Utasi

Computer Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest, Hungary
akos.utasi@sztaki.mta.hu
http://web.eee.sztaki.hu/~ucu

Abstract. This paper addresses the problem of local histogram-based image feature selection for learning binary classifiers. We show a novel technique which efficiently combines histogram feature projection with the conditional mutual information (CMI) based classifier selection scheme. Moreover, we investigate cost-sensitive modifications of the CMI-based selection procedure, which further improves the classification performance. Extensive evaluations show that the proposed methods are suitable for object detection and recognition tasks.

Keywords: classifier selection, mutual information, histogram feature.

1 Introduction

Histogram-based local image features are widely used in many pattern recognition and computer vision applications. In object detection, categorization, or recognition algorithms such features are usually combined with an efficient classification technique. Among the vast variety of histogram features, local binary patters (LBP) [1] or histogram of oriented gradients (HOG) [2] are widely adopted in many applications, because they can be calculated easily, and they are robust against small deformations and varying illumination. In monolithic classification approaches a single feature vector is constructed by concatenating the local features extracted over a dense predefined 2D grid. Finally, the feature vector is combined with a classifier, *e.g.* linear support vector machine (SVM) [3]. However, such monolithic approaches suffer from high computation costs since either (a) features are extracted at a large number of locations, or (b) the combined feature vector has a high dimension, resulting in slow classification.

One possible solution to overcome the above drawbacks is to limit the feature extraction step to grid locations where the extracted feature has a high discriminative power for classification. In each location we can train a weak learner, which usually has moderate classification accuracy. However, from the combination of several weak learners we can construct a strong classifier, which achieves a high classification performance. AdaBoost [4] is one of the most widely used techniques for boosting weak learners, and has been successfully applied *e.g.* for face detection using local Haar-like image features [5].

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 264–272, 2012.

The binary feature selection technique proposed in [6] is based on the fundamental concepts of information theory to quantify the uncertainty of random variables and to measure the information shared between them. The Conditional Mutual Information (CMI) estimates the information shared between the training data and a classifier, given another classifier. This can be utilized to select the feature which best describes the training data, and is the most independent from other features selected previously. One main advantage of this technique is that it is able to cope with overfitting, while AdaBoost is known to be sensitive to this phenomenon. The CMI-based feature selection technique has been successfully applied for facial expression recognition using LBP features [7].

Fisher Linear Discriminant (FLD) [8] analysis is frequently used to find the projection of histogram features which best separates two object classes, *e.g.* [9] embedded the projected features into the AdaBoost learning framework to detect faces. [10] proposed the Weighted Fisher Linear Discriminant (WFLD) as the weak learner in the AdaBoost framework. Thereby the WFLD minimizes the weighted classification error computed from the sample weights, which are updated by the AdaBoost procedure. The main advantage of this technique is that it eliminates the need of re-sampling the training data, and it leads to a more efficient use of the training samples.

In the proposed method we adopt the CMI-based feature selection technique, but we employ weak learner parameter optimization during the feature selection process to further improve classification accuracy, as opposed to previous methods [6] where these parameters are assumed to be already set. In AdaBoost the sample weights are used for WFLD, which are updated in each iteration using the weights of misclassified samples. However, sample misclassification is not defined in the CMI-based feature selection. Therefore, by using the concepts of information theory we introduce a novel method for updating the sample weights. Finally, we introduce cost-sensitive modifications of the CMI feature selection, which improves the classification accuracy of imbalanced datasets, where learning methods usually end up preferring the larger class.

The rest of the paper is organized as follows. In Sec. 2 we briefly overview the AdaBoost classifier learning method [4] and the WFLD weak learner technique [10]. In Sec. 2.1 we discuss the CMI-based feature selection [6] in more detail. The proposed method is presented in Sec. 3. Finally, in Sec. 4 we show our experimental results using two public image databases.

2 Classifier Learning with WFLD

We denote by $\mathbf{X} = \{x_1, \ldots, x_N\}$ a set of N training images, where each image x_i has a binary class label $y_i \in \{1, 0\}$ and $\mathbf{Y} = \{y_1, \ldots, y_N\}$. We extract $\mathbf{F} = \{f_1, \ldots, f_K\}$ a set of K features from an image x, where $f_k(x) \in \mathbb{R}^m$ is a histogram feature extracted at a given position. Finally, each feature is projected by the $g_k : \mathbb{R}^m \to \mathbb{R}$ function. In our case g is the WFLD [10], which guarantees optimal classification of the two classes, and is defined as $g = w^\intercal f$, such that

$$w = (\Sigma_1 + \Sigma_0)^{-1} (\mu_1 - \mu_0) \ , \tag{1}$$

where μ denotes the weighted mean and Σ is the weighted covariance matrix of the training set of a given class, *i.e.*

$$\mu = \frac{1}{n\sum_i d_i}\sum_i d_i f(x_i)\,, \quad \Sigma = \frac{1}{(n-1)\sum_i d_i^2}\sum_i d_i^2 (f(x_i)-\mu)\,(f(x_i)-\mu)^{\mathsf{T}}\,, \tag{2}$$

where n denotes the number of samples in the given class, and d_i denotes the weight of a particular sample. Hereafter we use $g_k(\cdot) = g_k(f_k(\cdot))$ as a shorthand. Similarly to [5] our weak classifier h_k at a given position is defined in the form

$$h_k(x) = \begin{cases} 1 & \text{if } p_k g_k(x) < p_k \theta_k \\ 0 & \text{otherwise} \end{cases}\,, \tag{3}$$

where p_k is the parity and θ_k is a threshold. Having a subset of weak learners the strong classifier $H(x)$ is defined as

$$H(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_{\iota(t)}(x) - b\right)\,, \tag{4}$$

where T denotes the number of selected weak learners, $\nu(t)$ returns the index of the t^{th} weak learner, $\{\alpha_t\}$ are the weights and b is the bias. In an iterative boosting scheme the weak learner is selected in each step , which minimizes an error function $\epsilon_k = \epsilon(h_k)$ describing the fitness of the weak learner on the labeled training data $(\mathbf{X}, \mathbf{Y})$. In AdaBoost ϵ_k is the classification error, which is expressed as the sum of the $\mathbf{D} = \{d_1, \ldots, d_N\}$ weights of the misclassified samples, α_t is estimated from ϵ_k, and the bias is expressed as $b = \frac{1}{2}\sum_t \alpha_t$.

Thus in iteration t first the optimal WFLD projection w_k is determined from $\mathbf{D}$ using Eqs. 1–2, then the optimal $p_k^\star$ and $\theta_k^\star$ parameters are determined in a brute force manner, and the $h_{\nu(t)}$ classifier with minimal ϵ_k is selected, *i.e.*

$$(p_k^\star, \theta_k^\star) = \underset{p_k,\theta_k}{\operatorname{argmin}} \{\epsilon_k\}\,, \tag{5}$$

$$\nu(t) = \underset{k}{\operatorname{argmin}} \{\epsilon_k\}\,. \tag{6}$$

2.1 CMI-Based Classifier Selection

[6] proposed an iterative *binary* feature selection method based on CMI. In each iteration the feature is selected, which maximizes the mutual information on training samples $(\mathbf{X}, \mathbf{Y})$, depending on the output of any feature selected in previous iterations. This procedure can be formalized as follows. Let $\hat{x}_i^k \in \{1, 0\}$ denote the response of the k^{th} classifier on the i^{th} sample, *i.e.* $\hat{x}_i^k$ is a *binary* feature, and $\hat{\mathbf{X}}^k = \{\hat{x}_i^k\}$. In the first step the feature which maximizes the $I(\mathbf{Y}; \mathbf{X})$ mutual information (MI) on the samples is selected, *i.e.*

$$\nu(1) = \underset{k}{\operatorname{argmax}} \left\{ I\left(\mathbf{Y}; \hat{\mathbf{X}}^k\right) \right\}\,. \tag{7}$$

Note that the mutual information $I(\mathbf{Y};\mathbf{X})$ of two random variables $\mathbf{Y}$ and $\mathbf{X}$ can be expressed in terms of entropy as $I(\mathbf{Y};\mathbf{X}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})$, where the conditional entropy $H(\mathbf{Y}|\mathbf{X})$ quantifies the uncertainty of $\mathbf{Y}$ when $\mathbf{X}$ is selected. By minimizing this uncertainty in Eq. 7 we obtain the classifier which best describes the training data. By similar considerations in subsequent iterations the $I(\mathbf{Y};\mathbf{X}|\mathbf{Z})$ CMI is utilized for feature selection, thus for $t = 2, \ldots, T$

$$\nu(t) = \underset{k}{\operatorname{argmax}} \left\{ \min_{s<t} I\left(\mathbf{Y}; \hat{\mathbf{X}}^k | \hat{\mathbf{X}}^{\nu(s)}\right) \right\} \ . \tag{8}$$

3 Proposed Method

We introduce the generalization of the CMI-based technique of Sec. 2.1 for boosting arbitrary features, where the optimal $p_k^\star$ and $\theta_k^\star$ parameters of the weak learners are determined using the

$$\epsilon = 1 - \min I(\mathbf{Y};\mathbf{X}|\mathbf{Z}) \tag{9}$$

error function in Eq. 5, and the weak learner with minimal ϵ_k is selected as in Eq. 6. Finally, the classifier weights $\{\alpha_t\}$, and the bias b of the strong classifier are estimated by a linear SVM [3]. During the SVM learning we also utilize cost factors c_1 and c_0 for the two classes $\mathcal{C}_1$ and $\mathcal{C}_0$, which are chosen to satisfy $c_0/c_1 = n_1/n_0$ [11], where n_1 and n_0 denote the cardinality of the two classes. This re-balancing technique is necessary when the training data is imbalanced, *i.e.* when the size of one of the two classes is significantly larger then the other's. Without re-balancing the resulting classifier will tend to favor the larger class, and the samples of the smaller class will be misclassified with a higher probability.

3.1 Sample Weights for CMI-Based Feature Selection

The generalized CMI feature selection technique uses the error function Eq. 9 to determine the optimal parameters of the weak learners in Eq. 5, and to select the optimal weak learner using Eq. 6. However, in the original CMI procedure no sample weights and no update procedure are available for computing the WLFD projection vector w of Eq. 1. Therefore, we extend this method with sample weights together with an update procedure, and we use information theory concepts to define sample misclassification for the update.

Recall that in AdaBoost the sample weights are updated using the weights of the misclassified samples, *i.e.* by defining the

$$e_i = \mathbb{1}\{h(x_i) \neq y_i\} \in \{0, 1\} \tag{10}$$

indicator function the classifier error ϵ_i is calculated as $\epsilon_i = \sum d_i \cdot e_i$, and the sample weights are updated as $d_{t+1,i} = d_{t,i}\beta^{1-e_i}$, such that $\beta = \frac{\epsilon}{1-\epsilon}$ for $\epsilon < 0.5$. In the proposed CMI-based method we assume that a particular sample x_i is misclassified by the weak learner h_k when changing its response $\hat{x}_i^k$ would imply

an increase of the MI ($t = 1$) or of the CMI ($t = 2, \ldots, T$). We use the following notations to formally define our technique. Let $p(\varphi, \upsilon) = p_{X,Y}(\varphi, \upsilon)$ denote the joint distribution of the two random variables, similarly $p(\varphi) = p_X(\varphi)$, and $p(\upsilon) = p_Y(\upsilon)$, where φ and υ are boolean variables. By definition MI is

$$I\left(\mathbf{Y}; \hat{\mathbf{X}}\right) := \sum_{\varphi,\upsilon} p(\varphi, \upsilon) \log \frac{p(\varphi, \upsilon)}{p(\varphi)\, p(\upsilon)} \ . \tag{11}$$

However, in our case the above probabilities are determined from a limited training set having N elements. Therefore, we can re-write it using a fast look-up-table (LUT) solution as follows. We express the above distributions in terms of frequencies of the random variables' occurrences as $p(\varphi, \upsilon) = \frac{1}{N} n(\varphi, \upsilon)$, $p(\varphi) = \frac{1}{N} n(\varphi)$, and $p(\upsilon) = \frac{1}{N} n(\upsilon)$, where $n(\cdot)$ denotes the cardinality. Note that $n(\upsilon{=}0) = n_0$, $n(\upsilon{=}1) = n_1$ denote the cardinality of the two classes $\mathcal{C}_0$ and $\mathcal{C}_1$ of the training set. We create a LUT $\mathcal{L}$ on the $0 \leq n \leq N$ integer range as $\mathcal{L}[n] = n \log n$, and we rewrite Eq. 11 as

$$I\left(\mathbf{Y}; \hat{\mathbf{X}}\right) = \frac{1}{N} \left(\sum_{\varphi,\upsilon} \mathcal{L}[n(\varphi, \upsilon)] - \sum_{\varphi} \mathcal{L}[n(\varphi)] - \sum_{\upsilon} \mathcal{L}[n(\upsilon)] + \mathcal{L}[N] \right) \ . \tag{12}$$

We can see that the terms $\sum \mathcal{L}[n(\upsilon)]$ and $\mathcal{L}[N]$ in the above equation are constants during feature selection. Furthermore, the $1/N$ normalizing constant can be neglected, and we refer to this unnormalized MI as $\tilde{I}(\mathbf{Y}; \hat{\mathbf{X}})$. According to our original assumption, changing the value of response $\hat{x}_i$ will affect $n(\varphi)$ and $n(\varphi, \upsilon)$ only, since $n(\upsilon)$ depends solely on the training set. For example assuming class label $y_i = 0$ and changing the response value $\hat{x}_i = 0$ to 1 will increase $n(0, 1)$ and $n(1)$ but will decrease $n(0, 0)$ and $n(0)$. Using this property we can express the change of the unnormalized MI denoted by $\tilde{I}_\Delta(y_i = \upsilon; \hat{x}_i = \varphi)$ as

$$\begin{aligned} \tilde{I}_\Delta(\upsilon; \varphi) = {} & \mathcal{L}[n(\upsilon, \varphi){-}1] + \mathcal{L}[n(\upsilon, 1{-}\varphi){+}1] - \mathcal{L}[n(\upsilon, \varphi)] - \mathcal{L}[n(\upsilon, 1{-}\varphi)] \\ & + \mathcal{L}[n(\varphi)] + \mathcal{L}[n(1{-}\varphi)] - \mathcal{L}[n(\varphi) - 1] - \mathcal{L}[n(1{-}\varphi) + 1] \ . \end{aligned} \tag{13}$$

Similarly to [5] in our method the sample weights are initialized to $d_i = \frac{1}{2n_0}, \frac{1}{2n_1}$ for $y_i = 0, 1$ respectively, but for updating their value we utilize Eq. 13. First, we define the indicator function

$$e_i = \mathbb{1}\{\tilde{I}_\Delta(y_i; \hat{x}_i) > 0\} \in \{0, 1\} \tag{14}$$

to indicate whether sample x_i is misclassified or not. Then we define the classification error γ of the selected weak learner as the sum of the weights of the misclassified samples, *i.e.* $\gamma = \sum d_i \cdot e_i$. Finally, sample weights are updated as

$$d_{t+1,i} = d_{t,i} \gamma^{1-e_i} \ . \tag{15}$$

Thus the above update rule decreases the weights of the samples which were classified correctly by the selected weak learner. Note that in the case of CMI we can define the rules similarly to Eqs. 12–13 in a straightforward way, but these were omitted in the present paper due to space limitations. In the following we refer to this method as *CMISVM*.

3.2 Balanced Feature Selection and Weight Update

The method presented in Sec. 3 uses cost factors in the final step of constructing the strong classifier. In our second method first we incorporate re-balancing into the weak learner selection by utilizing the weighted mutual information (wMI), which is defined as

$$I_w\left(\mathbf{Y};\hat{\mathbf{X}}\right) = \sum_{\varphi,\upsilon} w(\varphi,\upsilon)\, p(\varphi,\upsilon) \log \frac{p(\varphi,\upsilon)}{p(\varphi)\, p(\upsilon)} \ , \tag{16}$$

where we use the cost factors of Sec. 3 to define the weights as $w(\varphi,1) = 1$ and $w(\varphi,0) = n_1/n_0$. The weighted conditional mutual information (wCMI) is defined similarly. Finally, we incorporate a re-balancing technique into the weight update rule defined in Eq. 15 by taking into account the distribution of $\tilde{I}_{\Delta}(y_i;\hat{x}_i)$. Our goal is to achieve a more aggressive change in the weight of the correctly classified samples (where $\tilde{I}_{\Delta}(\cdot;\cdot) \leq 0$), which do not change the MI significantly, *i.e.* if $|\tilde{I}_{\Delta}(y_i;\hat{x}_i)| < |\tilde{I}_{\Delta}(y_j;\hat{x}_j)|$ then d_i can be decreased more aggressively. Therefore, we modify Eq. 15 as

$$d_{t+1,i} = d_{t,i}\gamma^{1-e_i} \cdot \frac{I(\mathbf{Y};\hat{\mathbf{X}}) + \min_j\{\tilde{I}_{\Delta}(y_j;\hat{x}_j)\}}{I(\mathbf{Y};\hat{\mathbf{X}}) + \tilde{I}_{\Delta}(y_i;\hat{x}_i)} \ . \tag{17}$$

In the rest of the paper this method will be referred as *wCMISVM*.

4 Experiments

In our experiments we used two public datasets. From the FERET face database [12,13] we used the annotations to align the heads into the same eye positions. For detection the smaller class contains faces cropped from the aligned images and are resized to 112×128 pixels. Moreover, the other class contains randomly cropped parts from background images. For recognition we used a slightly larger part of the head and a 128×128 pixels size. From the available annotations we defined three classification problems for recognition: a) *race*: Asian or White, b) *glasses*: wearing or not, and c) *gender*: female or male. The second dataset we used is the MIT CBCL Car [14] database, which contains front and rear view of cars, and the size of the images were 128×128 pixels. Again, the samples of the other class are random background images not containing any cars. We extended the datasets by adding the mirrored version of each sample in the set. Note that there is a significant difference between the two experiments. In case of recognition the samples of a class are similar, while in the detection experiment the larger class contains very different samples as they are random parts of backgrounds.

Our features are HOG blocks[2] which are computed in a single cell of 8×8 pixels, and a 9-bin histogram ($0°-180°$) is calculated using linear gradient voting and L2-Hys normalization. In our evaluation we selected *AdaBoost* with WFLD weak learners [10] (see Sec. 2) as baseline, and the number of weak learners T was limited to the $\{2,4,\ldots,20\}$ range.

4.1 Recognition

For the race recognition experiment the $\mathcal{C}_1$ class contains faces of Asian people, the size of *training* data is $n_1 = 512$ and for *testing* 98 samples were used. The $\mathcal{C}_0$ class contains 3080 faces of White people, from which we used $n_0 = 2628$ for *training* and 452 for *testing*. For recognizing people wearing glasses we *trained* the classifiers with $n_1 = 194$ faces with glasses, and $n_0 = 1698$ without glasses. For *testing* 68 and 446 samples were used. Finally, in the gender recognition experiment the $\mathcal{C}_1$ class contains 1828 female faces, from which we used $n_1 = 1532$ samples for *training* and 296 for *testing*. $\mathcal{C}_0$ contains $n_0 = 2574$ male faces for *training* and 428 for *testing*. All these datasets are imbalanced, in order to present the advantages of the proposed approach, and we can also see that the *glasses* dataset is the most imbalanced (approx. 1:9 ratio). Fig. 1 shows sample images from recognition experiment.

Fig. 1. Example images from the recognition experiment. Left: Asian vs White; Center: glasses vs no glasses; Right: female vs male.

4.2 Detection

In the face detection experiment the $\mathcal{C}_1$ class contained 4818 faces, from which we used $n_1 = 3170$ samples for *training* and 1648 for *testing*. $\mathcal{C}_0$ contained 18880 non-faces, from which $n_0 = 12208$ samples were used for *training* and 6672 for *testing*. For the car detection the *training* set of $\mathcal{C}_1$ contained $n_1 = 828$ car images, and the size of the *testing* set was 204. The $\mathcal{C}_0$ class contained $n_0 = 4990$ *training* samples, and 1282 *test* samples. Examples from the datasets are shown in Fig. 2.

4.3 Evaluation

For evaluation we selected the *G-mean* from the available metrics [15], which is accepted as a good metric for imbalanced classification problems. After obtaining the *G-mean* values for the three classifiers containing $T \in \{2, 4, \ldots, 20\}$ weak learners we selected the classifier with maximal *G-mean* then we compared the other classifiers to this value and computed the difference which was considered as the error score of the classifier. Summing these differences for all T configurations we obtained a total error score for each method in a particular classification problem. Table 1 shows the error scores of the three methods both

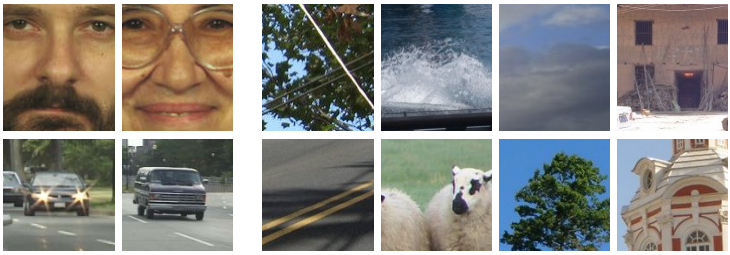

Fig. 2. Example images from the detection experiment. Top: face vs non-face; Bottom: car vs non-car.

for recognition and detection tasks. We can see that the re-balancing techniques of Sec. 3.2 are beneficial for the feature selection, as the *wCMISVM* classifier clearly outperformed the other methods. However, this method is slightly less effective for detection tasks. This may be due to the nature of the data since in this experiment the samples of the larger class contain very different images, and a single cost-factor may not suitable to represent such a large variation.

Table 1. Error scores of the three methods in different classification tasks

	AdaBoost[10]	*CMISVM*	*wCMISVM*
Race	0.1822	0.1001	**0.0527**
Glasses	0.2049	0.2129	**0.0251**
Gender	**0.0889**	0.2389	0.1240
Recognition	0.4760	0.5519	**0.2018**
Car	0.0388	**0.0112**	0.0453
Face	**0.0334**	0.0606	0.0593
Detection	0.0722	**0.0718**	0.1046

5 Conclusions

In this paper we investigated the difficulties of CMI-based classifier selection using WFLD as weak learners. We proposed a novel technique for updating the sample weights of the training data. To improve the efficiency of the CMI-based method on imbalanced datasets we proposed re-balancing techniques for both the feature selection and the weight update procedures. We performed extensive evaluations on two public datasets. The experiments confirmed that the proposed methods improve the efficiency of CMI-based boosting in case of imbalanced datasets. As a part of our future work we plan to extend our experiments with additional datasets and with more tests with various degree of data imbalance.

Acknowledgement. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office. This work was supported by the Hungarian Scientific Research Fund under grant number 80352.

References

1. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29(1), 51–59 (1996)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: International Conference on Computer Vision and Pattern Recognition (2005)
3. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York Inc. (1995)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory (1995)
5. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
6. Fleuret, F.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)
7. Shan, C., Gong, S., McOwan, P.W.: Conditional mutual information based boosting for facial expression recognition. In: British Machine Vision Conference (2005)
8. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2), 179–188 (1936)
9. Wang, H., Li, P., Zhang, T.: Histogram feature-based Fisher linear discriminant for face detection. Neural Computing and Applications 17(1), 49–58 (2008)
10. Laptev, I.: Improving object detection with boosted histograms. Image and Vision Computing 27(5), 535–544 (2009)
11. Morik, K., Brockhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach – A case study in intensive care monitoring. In: International Conference on Machine Learning (1999)
12. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
13. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.J.: The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing 16(5), 295–306 (1998)
14. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International Journal of Computer Vision 38(1), 15–33 (2000)
15. García, V., Mollineda, R.A., Sánchez, J.: Theoretical analysis of a performance measure for imbalanced data. In: International Conference on Pattern Recognition (2010)