The 4th IWVF, Capri, Italy, May 28-30, 2001.

Pattern Detection Using a Maximal Rejection Classifier

Michael Elad[1], Yacov Hel-Or[2], and Renato Keshet[3]

[1] Jigami LTD Israel, email: elad@jigami.com
[2] The Interdisciplinary Center, Herzlia, Israel, email: toky@idc.ac.il
[3] Hewlett-Packard Laboratories - Israel, email: renato@hpli.hpl.hp.com

Abstract. In this paper we propose a new classifier - the Maximal Rejection Classifier (MRC) - for target detection. Unlike pattern recognition, pattern detection problems require a separation between two classes, *Target* and *Clutter*, where the probability of the former is substantially smaller, compared to that of the latter. The MRC is a linear classifier, based on successive rejection operations. Each rejection is performed using a projection followed by thresholding. The projection vector is designed to maximize the number of rejected *Clutter* inputs. It is shown that it also minimizes the expected number of operations until detection. An application of detecting frontal faces in images is demonstrated using the MRC with encouraging results.

1 Introduction

In target detection applications, the goal is to detect occurrences of a specific *Target* in a given signal. In general, the target is subjected to some particular type of transformation, hence we have a set of target signals to be detected. In this context, the set of non-*Target* samples are referred to as *Clutter*. In practice, the target detection problem can be characterized as designing a classifier $C(z)$, which, given an input vector z, has to decide whether z belongs to the *Target* class $\mathbf{X}$ or the *Clutter* class $\mathbf{Y}$. In example based classification, this classifier is designed using two training sets - $\hat{\mathbf{X}} = \{x_i\}_{i=1..L_x}$ (*Target* samples) and $\hat{\mathbf{Y}} = \{y_i\}_{i=1..L_y}$ (*Clutter* samples), drawn from the above two classes.

Since the classifier $C(z)$ is usually the heart of a detection algorithm, and is applied many times, simplifying it translates immediately to an efficient detection algorithm. Various types of example-based classifiers are suggested in the literature [1,2,3]. The most simple and fast are the linear classifiers, where the projection of z is performed onto a projection vector u, thus, $C(z) = f(u^t z)$ where $f(*)$ is a thresholding operation (or some other decision rule). The Support Vector Machine (SVM) [2] and the Fisher Linear Discriminant (FLD) [1] are two examples of linear classifiers. In both cases the kernel u is chosen in some optimal manner. In the FLD, u is chosen such that the Mahalanobis distance of the two classes after projection will be maximized. In the SVM approach the

[4] This work has been done at HP Labs, Israel.

C. Arcelli et al. (Eds.): IWVF4, LNCS 2059, pp. 514–524, 2001.

motive is similar, but the vector u is chosen such that it maximizes the margin between the two sets.

In both these classifiers, it is assumed that the two classes have equal importance. However, in typical target detection applications the above assumption is not valid since the a-priori probability of z belonging to $\mathbf{X}$ is substantially smaller, compared to that of belonging to $\mathbf{Y}$. Both the FLD and the SVM do not exploit this property. Moreover, in both of these methods, it is assumed that the classes are linearly separable. However, in a typical detection scenario the target class is surrounded by the clutter class, thus the classes are not linearly seperable (see, e.g. Figure 2). In order to be able to treat more complex, and unfortunately, more common scenarios, non-linear extensions of these algorithms are required [1,2]. Such extensions are typically at the expense of much more computationally intensive algorithms.

In this paper we propose the *Maximal Rejection Classifier* (MRC) that overcomes the above two drawbacks. While maintaining the simplicity of a linear classifier, it can also deal with non linearly separable cases. The only requirement is that the *Clutter* class and the convex hull of the *Target* class are disjoint. We define this property as two *convexly-separable* classes, which is a much weaker condition compared to linear-separability. In addition, this classifier exploits the property of high *Clutter* probability. Hence, it attempts to give very fast *Clutter* labeling, even if at the expense of slow *Target* labeling. Thus, the entire input signal is classified very fast.

The MRC is an iterative rejection based classification algorithm. The main idea is to apply at each iteration a linear projection followed by a thresholding, similar to the SVM and the FLD. However, as opposed to these two methods, the projection vector and the corresponding thresholds are chosen such that at each iteration the MRC attempts to maximize the number of rejected *Clutter* samples. This means that following the first classification iteration, many of the *Clutter* samples are already classified as such, and discarded from further consideration. The process is continued with the remaining *Clutter* samples, again searching for a linear projection vector and thresholds that maximizes the rejection of *Clutter* samples from the remaining set. This process is repeated iteratively until a small number or non of the *Clutter* samples remain. The remaining samples at the final stage are considered as *Targets*. The idea of rejection-based classifier was already introduced by [3]. However, in this work we extend the idea by using the concept of maximal rejection.

2 The MRC in Theory

Assume two classes are given in $\Re^n$, $\mathbf{X}$ (the *Target* class) and $\mathbf{Y}$ (the *Clutter* class). It is required to discriminate between these two classes, i.e., given an input z drawn from one of these classes, we would like to be able to label it correctly as either *Target* or *Clutter*. One important point, however, is that we know a-priori that for a typical input stream the vast majority of the inputs are

Clutters, i.e.:

$$P\{\mathbf{X}\} \ll P\{\mathbf{Y}\} \tag{1}$$

where $P\{\mathbf{X}\}$ is the a-priori probability that an input signal will be a *Target*, and $P\{\mathbf{Y}\}$ is defined similarly. Based on this knowledge, we would like the classifier to give a decision as fast as possible (i.e., with as few operations as possible). Thus, *Clutter* labeling should be performed fast, even if at the expense of slow *Target* labeling.

Similar to other linear classifiers [1,2], we suggest to first project the sample z onto a vector u, and label it based on the projected value $\alpha = u^T z$. Projecting the *Target* class and the *Clutter* class onto u results with a Probability Density Functions (PDF) $P\{\alpha|\mathbf{X}\}$ and $P\{\alpha|\mathbf{Y}\}$ respectively. We define the following intervals based on $P\{\alpha|\mathbf{X}\}$ and $P\{\alpha|\mathbf{Y}\}$:

$$\begin{aligned} C_t &= \{\alpha | P\{\alpha|\mathbf{X}\} > 0, P\{\alpha|\mathbf{Y}\} = 0\} \\ C_c &= \{\alpha | P\{\alpha|\mathbf{X}\} = 0, P\{\alpha|\mathbf{Y}\} > 0\} \\ C_u &= \{\alpha | P\{\alpha|\mathbf{X}\} > 0, P\{\alpha|\mathbf{Y}\} > 0\} \end{aligned} \tag{2}$$

(t-Target, c-Clutter and u-Unknown). After projection, z is labeled either as a *Target*, *Clutter*, or *Unknown*, based on the interval at which α belongs to.

Unknown classifications are obtained only in the C_u interval, where a decision cannot be made. Figure 1 presents an example for the construction of the intervals C_t, C_c and C_u and their appropriate decisions. The probability of the *Unknown* decision is given by:

$$P\{Unknown\} = \int_{\alpha \in C_u} P\{\mathbf{Y}\} P\{\alpha|\mathbf{Y}\} d\alpha + \int_{\alpha \in C_u} P\{\mathbf{X}\} P\{\alpha|\mathbf{X}\} d\alpha \tag{3}$$

The above term is a function of the projection vector u. We would like to find the vector u which minimizes the "Unknown" probability. However, since this is a complex minimization problem, an alternative minimization is developed here, using a proximity measure between the two PDF's.

If $P\{\alpha|\mathbf{Y}\}$ and $P\{\alpha|\mathbf{X}\}$ are far apart and separated from each other $P\{Unknown\}$ will be small. Therefore, an alternative requirement is to minimize the overlap between these two PDF's. We will define this requirement using the following expected distance between a point α_0 and a distribution $P\{\alpha\}$:

$$D(\alpha_0 \,||\, P\{\alpha\}) = \int_\alpha \frac{(\alpha_0 - \alpha)^2 P\{\alpha\}}{\sigma^2} d\alpha = \frac{(\alpha_0 - \mu)^2 + \sigma^2}{\sigma^2}$$

where μ is the mean of $P\{\alpha\}$ and σ is the variance of $P\{\alpha\}$. The division by σ is performed in order to make the distance scale-invariant (or unit-invariant). Using this distance definition, the distance of $P\{\alpha|\mathbf{X}\}$ from $P\{\alpha|\mathbf{Y}\}$ can be defined as the expected distance of $P(\alpha|\mathbf{Y})$ from $P\{\alpha|\mathbf{X}\}$:

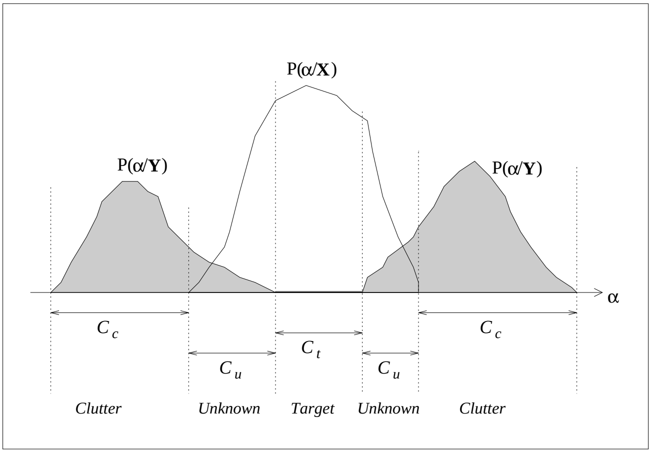

Fig. 1. The intervals C_t, C_c and C_u, for specific PDFs $P\{\alpha|\mathbf{X}\}$ and $P\{\alpha|\mathbf{Y}\}$.

$$D(P\{\alpha|\mathbf{Y}\} \,||\, P\{\alpha|\mathbf{X}\}) = \int_{\alpha'} D(\alpha' \,||\, P\{\alpha|\mathbf{X}\})\; P\{\alpha'|\mathbf{Y}\}d\alpha' = \tag{4}$$
$$= \int_{\alpha'} \frac{(\alpha' - \mu_x)^2 + \sigma_x^2}{\sigma_x^2}\; P\{\alpha'|\mathbf{Y}\}d\alpha' = \frac{(\mu_y - \mu_x)^2 + \sigma_x^2 + \sigma_y^2}{\sigma_x^2}$$

where $[\mu_x, \sigma_x]$ and $[\mu_y, \sigma_y]$ are the mean-variance pairs of $P\{\alpha|\mathbf{X}\}$ and $P\{\alpha|\mathbf{Y}\}$, respectively. Since we want the two distributions to have as small an overlap as possible, we would like to maximize this distance or minimize the *proximity* between $P\{\alpha|\mathbf{Y}\}$ and $P\{\alpha|\mathbf{X}\}$, which can be defined as the inverse of their mutual distance. Note, that this measure is asymmetric with respect to the two distributions, i.e the proximity defines the closeness of $P\{\alpha|\mathbf{Y}\}$ to $P\{\alpha|\mathbf{X}\}$, but not vice versa. Therefore, we define the overall proximity between the two distributions as follows:

$$Prox\;(P\{\alpha|\mathbf{Y}\}, P\{\alpha|\mathbf{X}\}) = \tag{5}$$
$$= P\{\mathbf{X}\}\frac{\sigma_y^2}{\sigma_x^2 + \sigma_y^2 + (\mu_y - \mu_x)^2} + P\{\mathbf{Y}\}\frac{\sigma_x^2}{\sigma_x^2 + \sigma_y^2 + (\mu_y - \mu_x)^2}$$

Compared to the original expression in Equation 3, the minimization of this term with respect to u is easier. If $P\{\mathbf{X}\} = P\{\mathbf{Y}\}$, i.e. if there is an even chance to obtain *Target* or *Clutter* inputs, the proximity becomes:

$$Prox(P\{\alpha|\mathbf{Y}\}\;,\; P\{\alpha|\mathbf{X}\}) = \frac{\sigma_x^2 + \sigma_y^2}{\sigma_x^2 + \sigma_y^2 + (\mu_y - \mu_x)^2} \tag{6}$$

which is associated with the cost function minimized by the Fisher Linear Discriminant (FLD)[1]. In our case $P\{\mathbf{X}\} \ll P\{\mathbf{Y}\}$ (Equation 1), thus, the first term is negligible in Equation 5 and can be omitted. Therefore, the optimal u should minimize the resulting term:

$$d(u) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_y^2 + (\mu_y - \mu_x)^2} \tag{7}$$

where $\sigma_y^2, \sigma_x^2, \mu_y$ and μ_x are all a function of the projection vector u.

There are two factors that control $d(u)$. The first factor is the distance between the two means μ_y and μ_x. Maximizing this distance will minimize $d(u)$. However, this factor is negligible when the two means are close to each other. This scenario is typical in detection cases when the target class in surrounded by the clutter class (see Figure 2). The other factor is the ratio between σ_x and σ_y. Our aim is to find a projection direction which results in a small σ_x and large σ_y. This means that the projection of *Target* inputs tend to concentrate in a narrow interval, whereas the *Clutter* inputs will spread with a large variance (see e.g. Fig 2).

For the optimal u, most of the *Clutter* inputs will be projected onto C_c, while C_t might even be an empty set. Subsequently, after projection, many of the *Clutter* inputs are usually classified, whereas *Target* labeling may not be immediately possible. This serves our purpose because there is a high probability that a decision will be made when a *Clutter* input is given. Since these inputs are more frequent, this means a faster decision for the vast majority of the inputs.

The method which we suggest follows this scheme: The classifier works in an *iterative* manner, projecting and thresholding with different parameters at each iteration sequentially. Since the classifier is asymmetric, the classification is based on *rejections*; *Clutter* inputs are classified and removed from further consideration while the remaining inputs are kept as suspected *Targets*. The iterations and the *rejection* approaches are both key concepts of the proposed scheme.

3 The MRC in Practice

Let us return to Equation 7 and find the optimal projection vector u. In order to do so, we have to express $\sigma_y^2, \sigma_x^2, \mu_y$ and μ_x as functions of u. It is easy to see that:

$$\mu_x = u^T \mathbf{M}_x \quad \text{and} \quad \sigma_x^2 = u^T \mathbf{R}_{xx} u \tag{8}$$

where we define:

$$\mathbf{M}_x = \int_z z P\{z|\mathbf{X}\}\, dz \quad ; \quad \mathbf{R}_{xx} = \int_z (z - \mathbf{M}_x)(z - \mathbf{M}_x)^T P\{z|\mathbf{X}\}\, dz \tag{9}$$

In a similar manner we express μ_y and σ_y^2. As can be seen, only the first and second moments of the classes play a role in the choice of the projection vector u.

In practice we usually do not have the probabilities $P\{z|\mathbf{X}\}, P\{z|\mathbf{Y}\}$, and inference on the *Target* or *Clutter* class is achieved through examples. For the two example-sets $\hat{\mathbf{X}} = \{x_k\}_{k=1}^{L_x}$ and $\hat{\mathbf{Y}} = \{y_k\}_{k=1}^{L_y}$, the mean-covariance pairs ($\mathbf{M}_x$, $\mathbf{R}_{xx}$, $\mathbf{M}_y$, and $\mathbf{R}_{yy}$) are replaced with empirical approximations:

$$\hat{\mathbf{M}}_x = \frac{1}{L_x}\sum_{k=1}^{L_x} x_k \quad ; \quad \hat{\mathbf{R}}_{xx} = \frac{1}{L_x}\sum_{k=1}^{L_x}(x_k - \hat{\mathbf{M}}_x)(x_k - \hat{\mathbf{M}}_x)^T \tag{10}$$

and similarly for $\hat{\mathbf{M}}_y$ and $\hat{\mathbf{R}}_{yy}$. The function we aim to minimize is therefore:

$$d(u) = \frac{u^T \hat{\mathbf{R}}_{xx} u}{u^T \left[\hat{\mathbf{R}}_{xx} + \hat{\mathbf{R}}_{yy} + \left(\hat{\mathbf{M}}_y - \hat{\mathbf{M}}_x\right)\left(\hat{\mathbf{M}}_y - \hat{\mathbf{M}}_x\right)^T\right] u} \tag{11}$$

Similarly to [1,4,5], it is easy to show that u that minimizes the above expression satisfies:

$$\hat{\mathbf{R}}_{xx} u = \lambda \left[\hat{\mathbf{R}}_{xx} + \hat{\mathbf{R}}_{yy} + \left(\hat{\mathbf{M}}_y - \hat{\mathbf{M}}_x\right)\left(\hat{\mathbf{M}}_y - \hat{\mathbf{M}}_x\right)^T\right] u \tag{12}$$

and should correspond to the smallest possible λ. A problem of the form $Au = \lambda Bu$, as in Equation 12, is known as the generalized eigenvalue problem [1,4,5], and has a closed form solution. Notice that given any solution u for this equation, βu is also a solution with the same λ. Therefore, without loss of generality, the normalized solution $\|u\| = 1$ is used.

After finding the optimal projection vector u, the intervals C_t, C_c, and C_u can be determined according to Equation 2. An input z is labeled as a *Target* or *Clutter* if its projected value $u^T z$ is in C_t or C_c, respectively. Figure 2 (left) presents this stage for the case where C_t is empty, i.e. there are no inputs which can be classified as *Target*.

Input vectors whose projected values are in C_u are not labeled. For these inputs we apply another step of classification, where the design of the optimal projection vector in this step is performed according to the following new distributions:

$$P\{z|\mathbf{Y} \;\&\; u_1^T z \in C_u\} \quad \text{and} \quad P\{z|\mathbf{X} \;\&\; u_1^T z \in C_u\}$$

We define the next projection vector u_2 as the vector which minimizes the proximity measure between the above two distributions. This minimization is performed in the same manner as described for the first step. Figure 2-right presents the second rejection stage, which follows the first stage shown in Figure 2-left.

Following the second step, the process continues similarly with projection vectors $u_3, u_4, \cdots$, etc. Due to the optimality of the projection vector at each

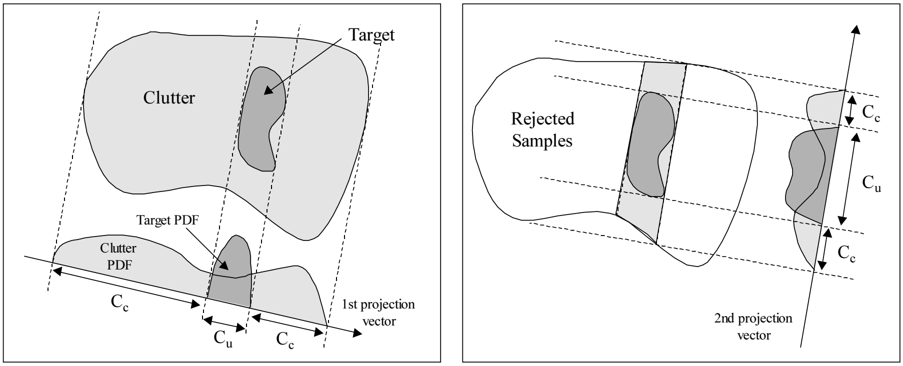

Fig. 2. Left: The first rejection stage for a 2D example. Right: The second rejection stage.

step, it is expected that a large portion of the input vectors will be labeled as *Clutter* at each step, while following steps will deal with the remaining input vectors. Applying the cascade of classifiers in such an iterative manner ensures a good performance of the classification with respect to an accurate labeling and a fast classification rate.

Since we exchanged the class probabilities with sets of points, it is impractical to define the intervals C_t, C_c, and C_u using Equation 2. This is because the intervals will be composed of many fragments each of which results from a particular example. Moreover, the domain of α cannot be covered by a finite set of examples. Therefore, it is more natural to define for each set, two thresholds bounding its projection values. As explained above, due to the functional that we are minimizing, in typical detection cases the *Target* thresholds define a small interval located inside the *Clutter* interval (see Figure 2). Therefore for simplicity, we define, for each projection vector, only a single interval $\Gamma = [T_1, T_2]$, which is the interval bounding the *Target* set. After projection we classify points projected outside Γ as *Clutter* and points projected inside Γ as *Unknown*.

In the case where the *Target* class forms a convex set, and the two classes are disjoint, it is theoretically possible to completely discriminate between them. This property is easily shown by noticing that we are actually extracting the *Target* set from the *Clutter* set by a sequence of two parallel hyper-planes, corresponding to the two thresholding operations. This constructs a parallelogram that bounds the *Target* set from outside. Since any convex set can be constructed by a set of parallel hyper-planes, exact classification is possible. However, if the *Target* set is non-convex, or the two classes are non-convexly separable (as defined in the Introduction), it is impossible to achieve a classification with zero errors; *Clutters* inputs which are inside the convex hull of the *Target* set cannot be rejected. Overcoming this limitation can be accomplished by a non-linear extension of the MRC, which is outside the scope of this paper. In practice, even if we deal with a convex *Target* set, false-alarms may exist due to the sub-optimal approach we are using, which neglects multi-dimensional moments higher than the second. However, simulations demonstrate that the number of false-alarms is typically small.

4 Face Detection Using the MRC

The face detection problem can be specified as the need to detect all instances of faces in a given image, at all spatial positions, all scales, all facial expressions, all poses, of all people, and under all lighting conditions. All these requirements should be met, while having few or no false alarms and mis-detections, and with as fast an algorithm as possible. This description reveals the complexity of the detection problem at hand. As opposed to other pattern detection problems, faces are expected to appear with considerable variations, even for the detection of frontal and vertical faces only. Variations are expected because of changes in skin color, facial hair, glasses, face shape, and more.

Several papers already addressed the face detection problem using various methods, such as SVM [2,6], Neural Networks [7,8,9], and other methods [10,11, 12,13]. In all of these studies, the above complete list of requirements is relaxed in order to obtain practical detection algorithms. Following [6,7,9,10,13], we deal with the detection of frontal and vertical faces only.

In all these algorithms, spatial position and scale are treated through the same method, in which the given image is decomposed into a Gaussian pyramid with near-unity (e.g., 1.2) resolution ratio. The search for faces is performed in each resolution layer independently, thus enabling the treatment of different scales. In order to be able to detect faces at all spatial positions, fixed sized blocks of pixels are extracted from the image at all positions (with full or partial overlap) for testing. In addition to the pyramid part, which treats varying scales and spatial positions, the core part of the detection algorithm is essentially a classifier which provides a *Face/Non-Face* decision for each input block.

We demonstrate the application of the MRC for this task. In the face-detection application, *Faces* take the role of targets, and *Non-Faces* are the clutter. The MRC produces very fast *Non-Face* labeling at the expense of slow *Face* labeling. Thus, on the average, it has a short decision time per input block.

The first stage in the MRC is to gather two example-sets, *Faces* and *Non-Faces*. As mentioned earlier, large enough sets are needed in order to guarantee good generalization for the faces and the non-faces that may be encountered in images. As to the *Face* set, the ORL data-base [1] was used. This database contains 400 frontal and vertical face images of 40 different individuals. By extracting the face portion from each of these images and scaling to 15×15 pixels, we obtained the set $\hat{\mathbf{X}} = \{x_k\}_{k=1}^{L_x}$ (with $L_x = 400$). The *Non-Face* set is required to be much larger, in order to represent the variability of *Non-Face* patterns in images. For this purpose we have collected from images with no faces more than 20 million *Non-Face* examples.

[1] http://www.cam-orl.co.uk/facedatabase.html: ORL database web-site

5 Results

We trained the MRC for detecting faces by computing 50 sets of kernels $\{u_k\}_{k=1}^{50}$ and associated thresholds $\{[T_1^k, T_2^k]\}_{k=1}^{50}$, using the above described databases of *Faces* and *Non-Faces*. Figures 3 and 4 show three examples of the obtained results. In these examples, the first stage rejected close to 90% of the candidates.

Fig. 3. An example for face detection with the MRC.

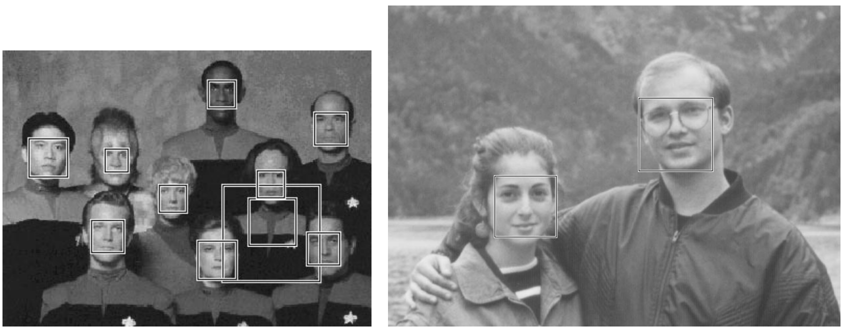

Fig. 4. Two examples for face detection with the MRC

This stage is merely a convolution of the input image (at every scale) with the first kernel, followed by thresholding. Successive kernels yield further rejection at about 50% for each projection. Thus, the complete MRC classification required an effective number of close to two convolutions per each pixel in each resolution layer. As can be seen from the examples, the MRC approach performed very well and was able to detect most of the existing faces. There are few false alarms, which typically correspond to blocks of pixels having a pattern which may resemble a face. In addition mis-detection occurs when a face is partially occluded or rotated too much. Generally speaking, the algorithm performs very well in terms of detection rate, false alarm rate, and most important of all, computational complexity. Due to space limitation we do not include more technical details in this paper. Comprehensive description of the results as well as comparative study with other face detection algorithms can be found in [14].

6 Conclusion

In this paper we presented a new classifier for target detection, which discriminates between *Target* and *Clutter* classes. The proposed classifier exploits the fact that the probability of a given input to belong to the *Target* class is much lower, as compared to its probability to belong to the *Clutter* class. This assumption, which is valid in many pattern detection applications, is exploited in designing an optimal classifier that detects *Target* signals as fast as possible. Moreover, exact classification is possible when the *Target* and the *Clutter* classes are convexly separable. The Fisher Linear Discriminant (FLD) is a special case of the proposed framework when the *Target* and *Clutter* probabilities are equal. In addition, the proposed scheme overcomes the instabilities arising in the FLD in cases where the mean of the two classes are close to each other. An improvement of the proposed technique is possible by rejecting *Target* patterns instead of *Clutter* patterns in advanced stages, when the probability of *Clutter* is not larger anymore. The performance of the MRC is demonstrated in the face detection problem. The obtained face detection algorithm is shown to be both computationally very efficient and accurate. Further details on the theory of the MRC and its application to face detection can be found in [15,14].

References

1. R. O. Duda and P. E. Hart, *Pattern Classification And Scene Analysis.* Wiley-Interscience Publication, 1973. 1st Edition.
2. V. N. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995. 1st Edition.
3. S. Baker and S. Nayar, "Pattern rejection," in *Proceedings 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - San Francisco, CA, USA, 18-20 June*, 1996.
4. G. H. Golub and C. F. V. Loan, *Matrix Computations.* Johns Hopkins University Press, 1996. 3rd Edition.
5. J. W. Demmel, *Applied Numerical Linear Algebra.* SIAM - Society for Industrial and Applied Mathematics, 1997. 1st Edition.

6. E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” in *IEEE Conference on Computer Vision and Pattern Recognition, Puerto-Rico*, 1997.
7. H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–38, 1998.
8. H. Rowley, S. Baluja, and T. Kanade, “Rotation invariant neural network-based face detection,” in *Calnegie Mellon University Internal Report - CMU-CS-97-201*, 1997.
9. P. Juell and R. March, “A hirarchial neural network for human face detection,” *Pattern Recognition*, vol. 29, pp. 781–787, 1996.
10. K. Sung and T. Poggio, “Example-based learning for view-based human face detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39–51, 1998.
11. A. Mirhosseini and H. Yan, “Symmetry detection using attributed graph matching for a rotation-invariant face recognition system,” in *Conference on Digital Image Computing: Techniques and Applications - Australia*, 1995.
12. A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, and M. Patil, “Finding faces in photographs,” in *International Conference of Computer Vision (ICCV) 97, Delhi, India*, 1997.
13. A. Tankus, H. Yeshurun, and N. Intrator, “Face detection by direct convexity estimation,” in *Tel-Aviv University Technical Report*, 1998.
14. M. Elad, R. Keshet, and Y. Hel-Or, “Frontal and vertical face detection in images using the approximated invariant method (aim),” in *Hewlett-Packard Technical Report - HPL-99-5*, 1999.
15. M. Elad, Y. Hel-Or, and R. Keshet, “Approximated invariant method (AIM): A rejection based classifier,” in *Hewlett-Packard Technical Report - HPL-98-160*, 1998.