

An Integrated Face Tracking and Facial Expression Recognition System

Angappan Geetha, Venkatachalam Ramalingam, Sengottaiyan Palanivel

Department of Computer Science & Engineering, Faculty of Engineering & Technology, Annamalai University, Annamalainagar, India.

Email: agst@sify.com

Received October 17th, 2010; revised September 30th, 2011; accepted October 15th, 2011.

ABSTRACT

This article proposes a feature extraction method for an integrated face tracking and facial expression recognition in real time video. The method proposed by Viola and Jones [1] is used to detect the face region in the first frame of the video. A rectangular bounding box is fitted over for the face region and the detected face is tracked in the successive frames using the cascaded Support vector machine (SVM) and cascaded Radial basis function neural network (RBFNN). The haar-like features are extracted from the detected face region and they are used to create a cascaded SVM and RBFNN classifiers. Each stage of the SVM classifier and RBFNN classifier rejects the non-face regions and pass the face regions to the next stage in the cascade thereby efficiently tracking the face. The performance of tracking is evaluated using one hour video data. The performance of the cascaded SVM is compared with the cascaded RBFNN. The experiment results show that the proposed cascaded SVM classifier method gives better performance over the RBFNN and also the methods described in the literature using single SVM classifier [2]. While the face is being tracked, features are extracted from the mouth region for expression recognition. The features are modelled using a multi-class SVM. The SVM finds an optimal hyperplane to distinguish different facial expressions with an accuracy of 96.0%.

Keywords: Face Detection, Face Tracking, Feature Extraction, Facial Expression Recognition, Cascaded Support Vector Machine, Cascaded Radial Basis Function Neural Network

1. Introduction

The proposed work focuses mainly on recognizing the facial expressions of a user interacting with the computer. Many applications such as video conferencing, intelligent tutoring systems, measuring advertisement effectiveness, behavioural science, e-commerce, video games, home/service robotics etc. require efficient facial expression recognition in order to achieve the desired results. Tracking a face in the video is the first step in automated facial expression recognition system. This paper deals with the problem of tracking a face in a video using cascaded SVM and cascaded RBFNN. Tracking requires a method for face detection which determines the image location of a face in the first frame of a video sequence. A number of methods have been proposed for face detection which include neural networks [3,4], wavelet basis functions [5] and Bayesian discriminating features [6]. Also many methods have been proposed for face tracking which include active contours [7], robust appearance filter [8], probabilistic tracking [9], adaptive active appearance model [10] and active appearance mo-

del [11]. In a face tracking application, Boosting and cascading detectors have gained great popularity due to their efficiency in selecting features for face detection. In this work, Adaboost algorithm [1] is used for face detection.

In Adaboost algorithm [1] faces are detected in three steps. First, the image is represented by a new representation called "Integral image" which computes the features quickly. In the second step, a learning algorithm is used which selects critical visual features from a large number of features. This yields a large number of efficient classifiers. In the third step, the classifiers are combined in a cascade that discards background regions from the object-regions of interest. A rectangular bounding box is fitted using these features for the face region. Once the face is detected, the detected face is tracked in consecutive frames in the video sequence.

Face tracking is to follow the face region through the video sequence. Face tracking is done in two methods. The first method tracks the face by detecting the face region in each and every frame. The second method detects

the face region in the first frame and the detected face region is tracked in the consecutive frames by classifying the face region from non face regions. Many models have been proposed for classification. Support vector machines (SVM) and Radial basis function neural network (RBFNN) models are popular due to their good capacity of generalization and possibility to build non-linear classifiers.

In this work, a method is proposed using haar-like features for feature selection and a cascade of SVMs and a cascade of RBFNNs for classification. The detected face and non face regions are modelled using cascaded SVMs and cascaded RBFNNs. The trained models are used to track the face region in the video sequence.

While the face is being tracked the mouth region is also tracked. Since the mouth plays an eminent role in expressing emotions, the mouth features are used for classifying expressions. Many methods have been proposed in the literature which include Active appearance model [12], Adaboost classifier [1,13], hidden markov models [14], neural networks [15], Bayesian networks [16,17], fuzzy systems [18], Support vector machines [19-21]. In this work a multi-class SVM is used for modelling mouth features and to classify the facial expressions.

Section 2 explains the face detection process. Facial feature extraction and face and non face modelling using cascaded SVM and cascaded RBFNN are presented in Section 3. Section 4 describes the modelling of mouth region for expression classification. Experimental results are given in Section 5. Section 6 concludes the paper.

2. Face Detection

Detecting faces automatically from the intensity or colour image is an essential task for many applications like person authentication and video indexing. Face detection is done in three steps. In step 1 simple rectangle features as shown in **Figure 1** are used.

These features are reminiscent of Haar basis functions [19]. The rectangle features are computed quickly using an intermediate representation by the sum of the pixels above and to the left of x, y inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

where $ii(x, y)$ is the integral image and $i(x', y')$ is the original image. The following pair of recurrences can be used to compute the integral image in one pass over the original image.

$$s(x, y) = s(x, y-1) + i(x, y) \quad (2)$$

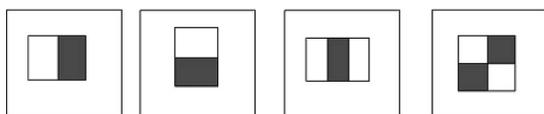


Figure 1. Rectangle features.

$$ii(x, y) = ii(x-1, y) + s(x, y) \quad (3)$$

where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$. The set of rectangle features used provide rich image representation which supports effective learning.

In the second step, Adaboost learning algorithm is used to select a small set of features to train the classifier. About 180,000 rectangle features are associated with each image sub-window. Out of these large numbers of features, a very small number of features are combined to form an effective classifier.

In the third step, a cascade of classifiers as shown in **Figure 2** is constructed which increases the detection performance thereby reducing the computation time.

A positive result from the first classifier triggers the evaluation of the second classifier and a positive result from the second classifier triggers the third classifier and so on. A negative result from any stage is rejected. The stages in the cascade are constructed using Adaboost algorithm and the threshold is adjusted to minimize the false negatives. Stages are added until the target for false positive and detection rate is achieved. A rectangle bounding box is placed over the face region. The detected face region is shown in **Figure 3**.

3. Modelling of Face and Nonface Regions for Face Tracking

3.1. Facial Feature Extraction

One of the main issues in constructing a face tracker is to extract the facial features that are invariant to the size of the face. The three types of haar-like features shown in **Figure 4** are used to extract the facial features.

The haar-like features are calculated as the sum of gray values in the black area subtracted from the sum of

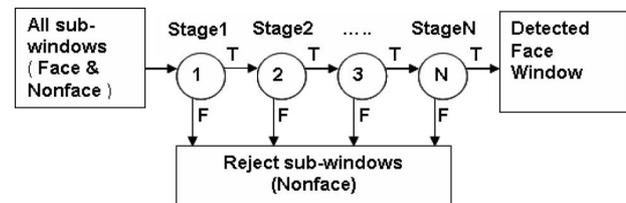


Figure 2. Detection cascade.



Figure 3. Detected face region.

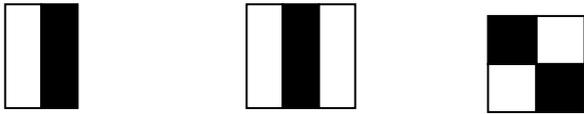


Figure 4. Types of haar-like features for feature selection.

gray values in the white area. The gray values are obtained by converting the RGB image into gray level image I , given by

$$I(i, j) = 0.299R(i, j) + 0.587G(i, j) + 0.114B(i, j) \quad (4)$$

To extract the facial features, each haar-like feature is moved over the face region which is marked as a rectangular window as shown in **Figure 5**. Non facial features from the non face regions are also extracted in the same way by moving each haar-like feature over the non face regions. The features extracted as shown in **Figure 5** also shows the scaling of features. Scaling is done by increasing the size of the feature window along the vertical and horizontal directions while moving them over face and non face regions.

For training the cascaded SVM, a set of 1700 (120 from face and 1580 from non face) such feature vectors each of dimension 60 are extracted from the face and non face regions using the first type of haar-like features, 1700 (120 from face and 1580 from non face) feature vectors using the second type and 700 (120 from face and 580 from non face) feature vectors using the third type.

For training the cascaded RBFNN, a set of 700 (120 from face and 580 from non face) such feature vectors each of dimension 60 are extracted from the face and non face regions using the first type of haar-like features, 700 (120 from face and 580 from nonface) feature vectors using the second type and 700 (120 from face and 580 from non face) feature vectors using the third type.



Figure 5. Extraction of scaled type of haar-like features.

3.2. Support Vector Machines Machine

Support vector machine (SVM) is a learning machine [22,2,19]. It is based on structural risk minimization (SRM). For a two-class linearly separable data, SVM finds a decision boundary which is a hyperplane. This hyperplane defined by the support vectors separates the data by maximizing the margin of separation. For linearly inseparable data, it maps the input pattern space \mathbf{X} into a high-dimensional feature space \mathbf{Z} using a nonlinear function $\phi(\mathbf{x})$. Then the SVM finds an optimal hyperplane as the decision surface to separate the examples of two classes in the feature space. Cover's theorem states that, "A complex pattern classification problem cast in a high dimensional space nonlinearly is more likely to be linearly separable than in a low dimensional space." An example of mapping two-dimensional data into three-dimensional space using the function $\phi(\mathbf{x}) = \{x_1^2, x_2^2, \sqrt{2}x_1x_2\}$ is shown in **Figure 6**. This shows that the data is linearly inseparable in two-dimensional space and it is linearly separable in three-dimensional space.

The support vector machine can be used for modeling the face and non face regions. The feature vectors are extracted from face and non face regions and given as input to the SVM to find out the optimal hyperplane for the face and non face regions. The proposed method uses three SVM classifiers for constructing the cascade as shown in **Figure 7**. The three classifiers are trained using the three types of Haar-like features as shown in **Figure 4** respectively.

A set of 1700 first type of feature vectors each of dimension 60 is extracted and trained thereby creating the first SVM model (SVM1). Second SVM model (SVM2) is created with 1700 second type of feature vectors each of dimension 60 and the third SVM model (SVM3) with 700 third type of feature vectors of dimension 60.

While testing, the first SVM classifier in the cascade is tested with the first type of haar-like features which discriminates possible face regions from the background regions

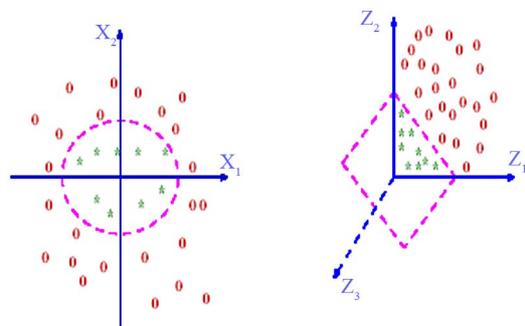


Figure 6. Mapping of two dimensional data into three dimensional data.

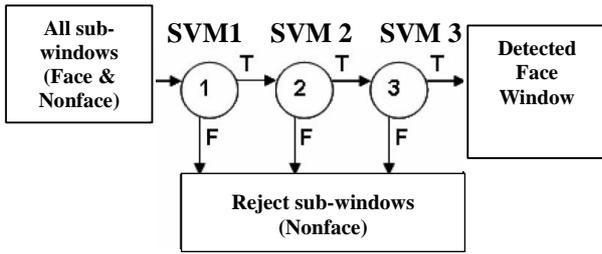


Figure 7. Cascaded SVM.

and passes them to the second classifier. The second classifier is tested with the second type which in turn discriminates most possible regions and passes them to the third one. Finally, the third classifier tested with the third type, detects the exact face region in the image. Since the cascade of classifiers rejects the non face regions in each stage so that the face region is efficiently detected at the final stage, the proposed method performs better than the single SVM classifier.

3.3. Radial Basis Function Neural Networks

Radial basis function neural network is a variant of Artificial Neural Network (ANN). It has a feedforward architecture [23-25] with an input layer, a hidden layer, and an output layer. It is applied to the problems of supervised learning and associated with radial basis functions. RBFNN trains faster than multilayer perceptron. It can be applied to the fields such as control engineering, time-series prediction, electronic device parameter modelling, speech recognition, image restoration, motion estimation, data fusion etc. The architecture of RBFNN is shown in **Figure 8**. Radial basis functions are embedded into a two-layer feed forward neural network. Such a network is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basis function. The input layer of this network has n_i units for a n_i dimensional input vector. The input units are fully connected to the n_h hidden layer units, which are in turn fully connected to the n_c output layer

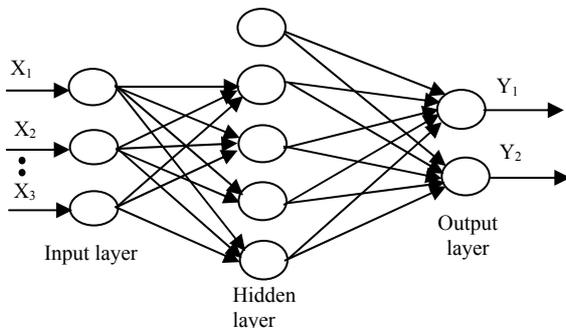


Figure 8. Structure of Radial basis function neural network.

units, where n_c is the number of output classes.

The activation functions of the hidden layer were chosen to be Gaussians, and are characterized by their mean vectors (centers) μ_i , and covariance matrices C_i , $i = 1, 2, \dots, n_h$. For simplicity, it is assumed that the covariance matrices are of the form $C_i = \sigma_i^2 I$, $i = 1, 2, \dots, n_h$. Then the activation function of the i^{th} hidden unit for an input vector x_j is given by

$$g_i(x_j) = \exp\left(\frac{-\|x_j - \mu_i\|^2}{2\sigma_i^2}\right) \tag{5}$$

The μ_i and σ_i^2 are calculated by using suitable clustering algorithm. Here the k -means clustering algorithm is employed to determine the centers. The algorithm is composed of the following steps:

- 1) Randomly initialize the samples to k means (clusters), μ_1, \dots, μ_k .
- 2) Classify n samples according to nearest μ_k .
- 3) Recompute μ_k .
- 4) Repeat the steps 2 and 3 until no change in μ_k .

A number of activation functions in the network and their spread influence the smoothness of the mapping. The assumption $\sigma_i^2 = \sigma^2$ is made and σ^2 is given in (6) to ensure that the activation functions are not too peaked or too flat.

$$\sigma^2 = \eta d^2 / 2 \tag{6}$$

In the above equation d is the maximum distance between the chosen centers, and η is an empirical scale factor which serves to control the smoothness of the mapping function. Therefore, The Equation (5) is written as

$$g_i(x_j) = \exp\left(\frac{-\|x_j - \mu_i\|^2}{\eta d^2}\right) \tag{7}$$

The hidden layer units are fully connected to the n_c output layer units through weights w_{ik} . The output units are linear, and the response of the k^{th} output unit for an input x_j is given by

$$y_k(x_j) = \sum_{i=0}^{n_h} w_{ik} g_i(x_j), k = 1, 2, \dots, n_c \tag{8}$$

where $g_0(x_j) = 1$. Given n_i feature vectors from n_c classes, training the RBFNN involves estimating μ_i , $i = 1, 2, \dots, n_h$, η , d^2 , and w_{ik} , $i = 0, 1, 2, \dots, n_h$, $k = 1, 2, \dots, n_c$. The procedure for training RBFNN is given below:

Determination of μ_i and d^2 : Conventionally, the unsupervised k -means clustering algorithm, can be applied to find n_h clusters from n_i training vectors. However, the training vectors of a class may not fall into a single cluster. In order to obtain clusters only according to class, the k -means clustering may be used in a supervised manner. Training feature vectors belonging to the same class are clustered to n_h/n_c clusters using the k -means clustering

algorithm. This is repeated for each class yielding n_h cluster for n_c classes. These cluster means are used as the centers μ_i of the Gaussian activation functions in the RBFNN. The parameter d was then computed by finding the maximum distance between n_h cluster means.

Determining the weights w_{ik} between the hidden and output layer: Given that the Gaussian function centers and widths are computed from n_t training vectors, (7) may be written in matrix form as

$$Y = GW \tag{9}$$

where Y is a $n_t \times n_c$ matrix with elements $Y_{ij} = y_j(x_i)$, G is a $n_t \times (n_h + 1)$ matrix with elements $G_{ij} = g_j(x_i)$, and W is a $(n_h + 1) \times n_c$ matrix of unknown weights. W is obtained from the standard least squares solution as given by

$$W = (G^T G)^{-1} G^T Y \tag{10}$$

To solve W from (8), G is completely specified by the clustering results, and the elements of Y are specified as

$$Y_{ij} = \begin{cases} 1, & \text{if } x_i \in \text{class } j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The Radial basis function neural networks can be used for modelling the face and non face regions. The feature vectors are extracted from face and non face regions and given as input to the RBFNN to find out the face and non face regions. The proposed method uses three RBFNN classifiers for constructing the cascade as shown in **Figure 9**. The three classifiers are trained using the three different types of Haar-like features as shown in **Figure 4** respectively.

A set of 700×60 (120×60 from face region and 580×60 from non-face region) first type of feature vectors is extracted and trained thereby creating the first RBFNN model (RBFNN1). Second RBFNN model (RBFNN2) is created with 700×60 (120×60 from face region and 580×60 from non-face region) second type of feature vectors. Third RBFNN model (RBFNN3) is created with 700×60 (120×60 from face region and 580×60 from non-face region) second type of feature vectors.

While testing, the first RBFNN classifier (RBFNN1) in the cascade is tested with the first type of haar-like

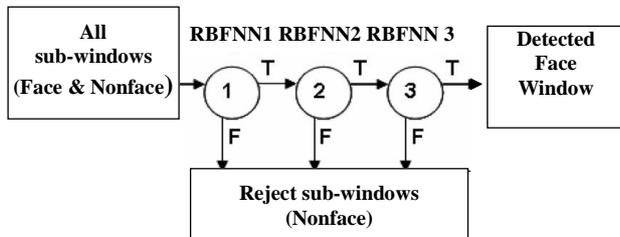


Figure 9. Cascaded RBFNN.

features which discriminates possible face regions from the background regions and passes them to the second classifier. The second classifier (RBFNN2) is tested with the second type which in turn discriminates most possible regions and passes them to the third one. Finally, the third classifier (RBFNN3) which is tested with the third type, detects the exact face region in the image. Since the cascade of classifiers rejects the non face regions in each stage so that the face region is efficiently detected at the final stage, the proposed method with cascaded RBFNN performs better than the single RBFNN classifier.

The hidden layer nodes are assigned weights of size 10×2 where 10 clusters (2 clusters for face and 8 clusters for non face) are formed. In this work, the output layer consists of 2 nodes where the patterns are classified as belonging to face or non face.

4. Modelling of Mouth Region for Expression Classification

The mouth region in a face plays an efficient role in classifying the basic facial expressions. The basic facial expressions are shown in **Figure 10**. To extract the features from the mouth region, it is first located in the face region. Mouth location is determined relative to the coordinates of the face detection window. This is shown in **Figure 11**. Once the mouth is located, it is divided into sixteen regions. The average of the gray values in each of the sixteen regions is extracted to model the facial expressions.

A set of 100×16 feature vectors from 100 frames in the video are extracted for each of the expressions separately. In this proposed work, four facial expressions such as normal, smile, anger and fear are considered and hence a total of 400×16 feature vectors are extracted



Figure 10. Different facial expressions.



Figure 11. Location of mouth region.

from the facial region while the face is being tracked. A multiclass SVM is used to model these features. It is trained initially with 400×16 feature vectors. While testing, the mouth region is tracked in a video and the features are extracted and given as input to the multiclass SVM models for classification.

5. Experimental Results

The images from which faces are to be tracked are captured at the rate of 10 frames/sec using Logitech Quickcam Pro5000 web camera in a PC with 2.00 GHz Intel Core 2 Duo processor. The performance is evaluated by tracking the face in a video of one hour duration using SVM and RBFNN. Once the face is detected in the first frame, it is tracked in consecutive frames. The tracked face region in consecutive frames is shown by means of drawing a rectangle over the face region as shown in **Figures 12** and **13**. While tracking the face region, the mouth region is also tracked to classify the facial expressions.



Figure 12. Face tracking using polynomial kernel.



Figure 13. Face tracking using RBFNN.

Table 1 shows the performance of proposed face tracking method using the measures, Precision and Recall. The terms Precision and Recall are defined as

$$\text{Precision} = \frac{\text{No..of True Positives}}{\text{No..of True Positives} + \text{False Positives}} \quad (12)$$

$$\text{Recall} = \frac{\text{No..of True Positives}}{\text{No..of True Positives} + \text{False Negatives}} \quad (13)$$

The percentage of Precision and Recall are measured with different SVM kernel functions and RBFNN. It is understood that the SVM with polynomial kernel performs better than the Linear kernel, Radial basis function kernel and RBFNN. While the face is being tracked, mouth features are extracted from about 100 frames for each of the four expressions separately. Hence a total of 400×16 feature vectors are trained using a multiclass SVM. While testing, the SVM is tested for 400 frames (100 consecutive frames for each of the four expressions separately).

Table 2 shows the performance of the proposed method in recognizing facial expressions. **Table 3** shows the confusion matrix for the test examples. The accuracy rates of facial expression recognition measured with different number of consecutive frames is shown in **Figure 14**. From the **Table 2**, it is observed that the Fear expression is recognized well than the other expressions.

Table 1. Face tracking performance using cascaded SVM and cascaded RBFNN.

Models	Precision	Recall
SVM with Linear kernel	72.0%	80.5%
SVM with Polynomial kernel	91.4%	95.0%
SVM with Radial basis function kernel	85.0%	99.0%
RBFNN	89.5%	87.6%

Table 2. Recognition accuracy of facial expression recognition.

Facial Expression	Recognition accuracy
Smile	90.0%
Fear	96.0%
Anger	82.0%
Normal	72.0%

Table 3. Confusion matrix for the test examples.

Expression	Smile	Fear	Anger	Normal
Smile	90	5	0	5
Fear	0	96	0	4
Anger	8	7	82	3
Normal	13	5	10	72

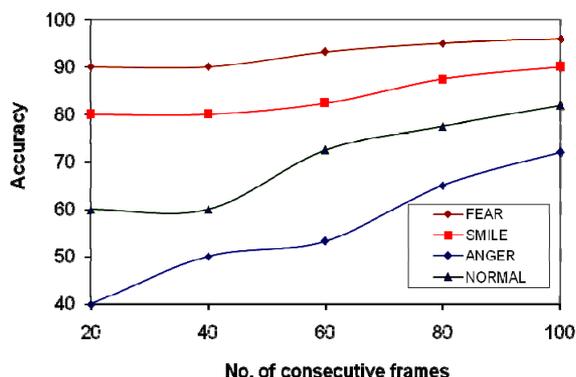


Figure 14. Accuracy rates for facial expression recognition using multi-class SVM.

The confusion matrix given in **Table 3** shows that out of 100 frames with fear expressions, 96 frames have been correctly classified and 4 frames have been misclassified.

6. Conclusions and Future Work

This paper proposes a method for face tracking and facial expression recognition using cascaded SVM and cascaded RBFNN classifiers. A face is detected in the first frame of the video using face detection process proposed in [12]. The face is then tracked in consecutive frames by classifying the face and non face regions using the cascaded SVM and cascaded RBFNN. Since the non face regions are rejected in each stage, the face region is effectively tracked in the final stage even under varying illumination, scale and pose. The RBFNN classifier is trained with the maximum of 700 feature vectors whereas SVM is trained with the maximum of 1700 feature vectors. It is the drawback of RBFNN that its performance gets reduced with large number of data. Hence it is found that the system tracks the face with greater performance using SVM polynomial kernel than the RBFNN. The system using a cascaded SVM is found to be better with the precision performance of 91.4% than the system using a single SVM classifier [2] in the literature with the maximum Precision performance of 72.8% using the same polynomial kernel.

The proposed method is also compared with a multi-stage face tracker [26] where ratio template algorithm had been used. The ratio template algorithm was modified with the inclusion of better spatial template for facial features. This hybrid face tracker was able to locate only 89% of images in the sequence which performs lower than the proposed method.

Among the four facial expressions, the proposed system is able to recognize the fear expression with the highest accuracy of 96.0%. The system can be extended to track multiple faces in the video sequence with more number of expressions.

REFERENCES

- [1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, 8-14 December 2001, Vol. 1, p. 511.
- [2] L. Carminati, J. Benois-Pineau and C. Jennewein, "Knowledge-Based Super Vised Learning Methods in a Classical Problem of Video Object Tracking," *Proceedings of IEEE International Conference on Image Processing*, Atlanta, 8-11 October 2006, pp. 2385-2389.
- [3] H. A. Rowley, S. Baluj and T. Kanade, "Neural Network-Based Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No.1, 1998, pp. 23-38. [doi:10.1109/34.655647](https://doi.org/10.1109/34.655647)
- [4] R. Feraud, O. J. Bernier, J. Vialle and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 1, 2001, pp. 42-53. [doi:10.1109/34.899945](https://doi.org/10.1109/34.899945)
- [5] C. Papageorgiou, M. Oren and T. Poggio, "A General Framework for Object Detection," *Proceedings of International Conference on Computer Vision*, Bombay, 4-7 January 1998, pp. 555-562.
- [6] C. Liu, "A Bayesian Discriminating Features Method for Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 6, 2003, pp. 725-740. [doi:10.1109/TPAMI.2003.1201822](https://doi.org/10.1109/TPAMI.2003.1201822)
- [7] D. Freedman, "Active Contours for Tracking Distributions," *IEEE Transactions on Image Processing*, Vol. 13, No. 4, 2004, pp. 518-526. [doi:10.1109/TIP.2003.821445](https://doi.org/10.1109/TIP.2003.821445)
- [8] H. T. Nguyen and A. W. M. Smeulders, "Fast Occluded Object Tracking by a Robust Appearance Filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, 2004, pp. 1099-1104.
- [9] H.-T. Chen, T.-L. Liu and C.-S. Fuh, "Probabilistic Tracking with Adaptive Feature Selection," *Proceedings of International Conference on Pattern Recognition*, Washington DC, Vol. 2, 2004, pp. 736-739. [doi:10.1109/TPAMI.2004.45](https://doi.org/10.1109/TPAMI.2004.45)
- [10] A. U. Batur and M. H. Hayes, "Adaptive Active Appearance Models," *IEEE Transactions on Image Processing*, Vol. 14, No. 11, 2005, pp. 1707-1721. [doi:10.1109/TIP.2005.854473](https://doi.org/10.1109/TIP.2005.854473)
- [11] P. Corcoran, M. C. Ionita and I. Bacivarov, "Next Generation Face Tracking Technology Using AAM Techniques," *Proceedings of International Symposium on Signals, Systems and Circuits*, Vol. 1, 2007, pp. 1-4. [doi:10.1109/ISSCS.2007.4292639](https://doi.org/10.1109/ISSCS.2007.4292639)
- [12] K.-S. Cho, Y.-G. Kim and Y.-B. Lee, "Real-Time Expression Recognition System Using Active Appearance Model and EFM," *Proceedings of International Conference on Computational Intelligence and Security*, Guangzhou, Vol. 1, November 2006, pp. 747-750.
- [13] P. Yang, Q.-S. Liu and D. N. Metaxas, "Boosting Coded Dynamic Features for Facial Action Units and Facial Expression Recognition," *Proceedings of IEEE Conference*

- on *Computer Vision and Image Understanding*, Minneapolis, 18-23 June 2007.
- [14] S Petar, S. Aleksic and A. K. Katsaggelos, "Automatic Facial Expression Recognition Using facial Animation Parameters and Multi-Stream HMMs," *IEEE Transactions on Information Forensics and Security*, Vol. 1, 2006, pp. 3-11. [doi:10.1109/TIFS.2005.863510](https://doi.org/10.1109/TIFS.2005.863510)
- [15] L. Ma and K. Khorasani, "Facial Expression Recognition Using Constructive Feed forward Neural Networks," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 34, No. 3, 2004, pp. 1588-1595. [doi:10.1109/TSMCB.2004.825930](https://doi.org/10.1109/TSMCB.2004.825930)
- [16] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, Vol. 29, No. 2, 1997, pp. 131-163. [doi:10.1023/A:1007465528199](https://doi.org/10.1023/A:1007465528199)
- [17] N. Sebe, I. Cohen, A. Garg, M. Lew and T. Huang, "Emotion Recognition Using a Cauchy Naïve Bayes Classifier," *Proceedings of International Conference on Pattern Recognition*, Quebec City, Vol. 1, August 2002, pp. 17-20.
- [18] N. Esau, E. Wetzel, L. Kleinjohann and B. Kleinjohann, "Real-Time Facial Expression Recognition Using a Fuzzy Emotion Model," *IEEE Proceedings of Fuzzy Systems Conference*, London, 2007.
- [19] A. Geetha, V. Ramalingam, S. Palanivel and B. Palaniappan, "Facial Expression Recognition—A Real Time Approach," *International Journal of Expert Systems with Applications*, Vol. 36, No. 1, 2009, pp. 303-308. [doi:10.1016/j.eswa.2007.09.002](https://doi.org/10.1016/j.eswa.2007.09.002)
- [20] I. Kotsia, N. Nikolaidis and I. Pitas, "Facial Expression Recognition in Videos Using a Novel Multi-Class Support Vector Machines Variant," *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Honolulu, 15-20 April 2007, pp. 585-588.
- [21] S. Avidan, "Support Vector Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, 2004, pp. 1064-1072.
- [22] J. C. B. Christopher, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, 1998, pp. 121-167.
- [23] F. Yang and M. Paindavoine, "Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification," *IEEE Transactions on Neural Networks*, Vol. 14, No. 5, 2003, pp. 1162-1175. [doi:10.1109/TNN.2003.816035](https://doi.org/10.1109/TNN.2003.816035)
- [24] S. Haykins, "Neural Networks: A Comprehensive Foundation," Pearson Publication 2001, Asia.
- [25] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris and D. M. Hummels, "On the Training of Radial Basis Function Classifiers," *IEEE Transactions on Neural Networks*, Vol. 5, No. 4, 1992, pp. 595-603.
- [26] K. Anderson and P. W. McOwan, "Robust Real-Time Face Tracker for Cluttered Environments," *Computer Vision and Image Understanding*, Vol. 95, No. 2, 2004, pp. 184-200. [doi:10.1016/j.cviu.2004.01.001](https://doi.org/10.1016/j.cviu.2004.01.001)