Using a sparse learning Relevance Vector Machine in Facial Expression Recognition

W.S. Wong, W. Chan, D. Datcu, L.J.M. Rothkrantz
Man-Machine Interaction Group
Delft University of Technology
2628 CD, Delft,
The Netherlands
E-mail: L.J.M.Rothkrantz@ewi.tudelft.nl

KEYWORDS
Facial expression recognition, face detection, facial feature extraction, facial characteristic point extraction, relevance vector machine, corner detection, AdaBoost, Evolutionary Search, hybrid projection.

ABSTRACT

At TUDelft there is a project aiming at the realization of a fully automatic emotion recognition system on the basis of facial analysis. The exploited approach splits the system into four components. Face detection, facial characteristic point extraction, tracking and classification. The focus in this paper will only be on the first two components. Face detection is employed by boosting simple rectangle Haar-like features that give a decent representation of the face. These features also allow the differentiation between a face and a non-face. The boosting algorithm is combined with an Evolutionary Search to speed up the overall search time. Facial characteristic points (FCP) are extracted from the detected faces. The same technique applied on faces is utilized for this purpose. Additionally, FCP extraction using corner detection methods and brightness distribution has also been considered. Finally, after retrieving the required FCPs the emotion of the facial expression can be determined. The classification of the Haar-like features is done by the Relevance Vector Machine (RVM).

INTRODUCTION

For the past decades, many projects have been started with the purpose of learning the machine to recognize human faces and facial expressions. Computer vision has become one of the most challenging subjects nowadays. The need to extract information from images is enormous. Face detection and extraction as computer-vision tasks have many applications and have direct relevance to the face-recognition and facial expression recognition problem. Potential applications of face detection and extraction are in human-computer interfaces, surveillance systems, psychology and many more. It is not so hard to imagine the importance of face detection in the means of face and emotion recognition. The importance of this subject can be ratified by the recent terrorism bombings in London. Face detection and extraction of biometric features helps in the identification of the terrorists. In London, monitoring of people especially in the public places is done by closed-circuit cameras and televisions, which are linked via cables and other direct means. These can too be found in casinos and banks for instance. They are also used to aid in the prevention of calamities using face detection, emotion recognition and crowd behaviour analysis techniques.

Facial expressions are crucial in human communication. Human communication is a very complex phenomenon as it involves a huge number of factors: we speak with our voice, but also with our hands, eyes, face and body. The interpretation of what is being said does not only depend on the meaning of the spoken words. Our body language i.e. gestures modify, emphasize, and sometimes even contradict what we say. Facial expressions provide sensitive cues about emotional responses and play an important role in human communication. Therefore, it is valuable if this aspect of human communication can also be applied for more effective and friendly methods in man-machine interaction. According to Ekman et al. (Ekman and Friesen 1978) people are born with the ability to generate and interpret only six facial expressions: happiness, anger, disgust, fear, surprise and sadness. All other facial expressions have to be learned from the environment the person grows up. Humans are capable of producing thousands of expressions that vary in complexity, intensity, and meaning. Subtle changes in a facial feature such as tightening of the lips are sufficient to turn the emotion from happy to angry. And to think that the eyes and eye brows can also take on different shapes, one may imagine how complex the problem gets. In the past, Morishima et al. (Morishima and Harashima 1993) implemented a five-layered manual-input neural network which is used for recognition and synthesis of facial expressions. In (Zhao and Kearney 1996) a singular emotional classification of facial expressions is explained using a three-layered manual-input back propagation neural network. Kearney et al. (Kearney and McKenzie 1993) developed a manual-input memory-based learning expert system, which interprets facial expressions in terms of emotion labels given by college students without formal instruction in emotions signals. Rothkrantz et al. (Rothkrantz and Pantic 2000) proposed a point-based face model composed of two 2D facial views, namely the frontal- and the side view. Given a characteristic points based face model, expression-classification rules can be converted straight-forwardly into the rules of an automatic classifier.

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

RELATED WORK

The online Facial Expression Dictionary (FED) is an ongoing project at the Man-Machine Interaction group of the TU Delft (de Jongh 2002). The goal of the project is to develop a non-verbal dictionary which would contain information about non-verbal communication of people. Resembling a verbal dictionary, instead of words the FED contains facial expressions. This online non-verbal dictionary allows people to issue a query using description of the expression in terms of the expression classes (happiness, sadness, jealousy, etc.) and in terms of characteristics of the expression. Another interesting possibility of FED is the labelling of a picture containing a face. In other words an appropriate facial expression will be matched with the face. In the current state of the system the latter requires the user to select the region of the face and select the characteristic points of the face. These points are predefined consistent with the chosen face model. The face model used is that of Kobayashi and Hara (Kobayashi and Hara 1997). The facial expression recognition model is based on a three-tier framework. The chosen approach splits the FED system in three components, i.e. face detection, facial characteristic point (FCP) detection, tracking and classification. To fully automate the labelling process when inputting a picture, a project has been started on automating the face detection and the FCP detection part. This paper discusses the face detection and the FCP detection module.

METHODOLOGY

In this section we describe the theoretical background and the methodology of our research. The detection process is based on the detectors described in (Viola and Jones 2001) and (Treptow and Zell 2003).

Face detection

Haar-like features

In order to classify a face, some characteristic features need to be extracted. For this purpose, we used Haar-like features. These features have a rectangular shape and are fairly simple. The processing of this kind of features is computationally very efficient. In our face detection algorithm, five types of rectangular features are used (see Figure 1). Type 1, 2 and 5 are calculated as the sum of all pixels in the dark area minus the sum of all pixels in the light area. Type 3 and 4 are calculated as half the sum of all pixels in both dark areas minus the sum of all the pixels in the light area in the middle.

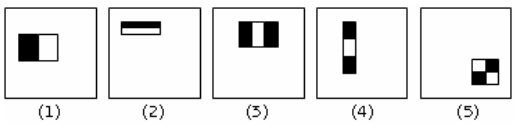

Figure 1: The five basic types of Haar-like features used in our approach

Each of the five basic features is scanned on every possible scale and every possible position within a training sample. Given that the sample's dimension is 24x24, the complete set of features that can be constructed is quite large, namely 162336. From this set of features, we want the most relevant ones that best characterize the face. The best features are chosen using the AdaBoost learning algorithm (Figure 2).

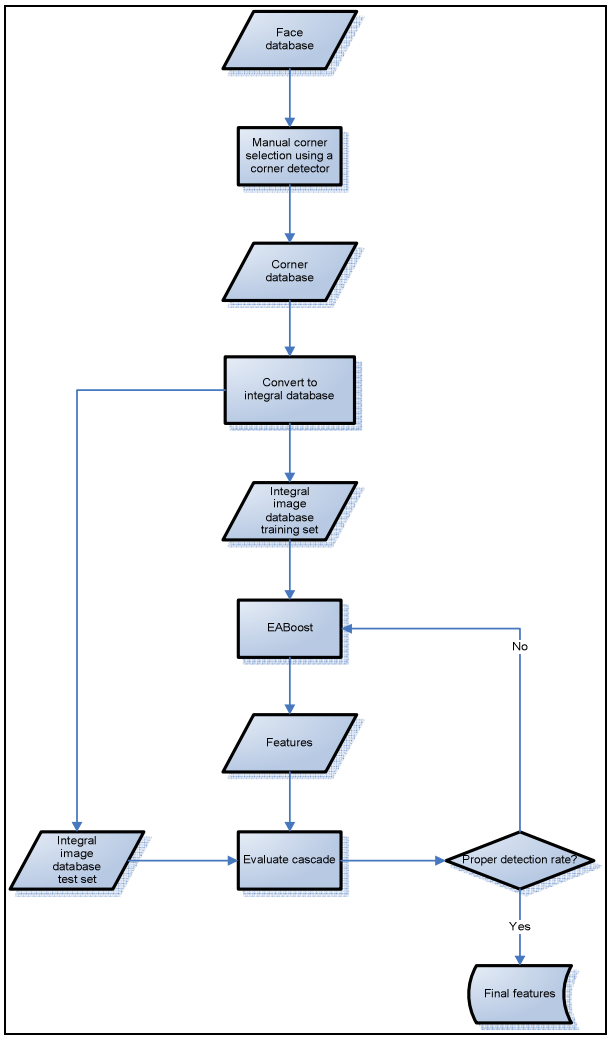

Figure 2: Scheme representing the training of weak classifiers

AdaBoost

The AdaBoost algorithm (Freund and Schapire 1995) aims at boosting the classification performance. It is an aggressive and effective algorithm used to select a low number of good classification functions, so called 'weak classifiers', to form a stronger classifier. The final strong classifier is actually a linear combination of the weak classifiers. Each weak classifier is restricted to the set of single feature functions.
In the algorithm of (Viola and Jones 2001) the training stage for a single weak classifier involves the computation of a threshold for the feature value to discriminate between positive and negative examples. In our approach, the latter is slightly different. Instead of using a threshold, the chosen weak classifier is the Relevance Vector Machine - RVM for discriminating between the positive and negative examples. This means that for each feature, the weak RVM classifier determines the optimal classification function such that a minimum number of examples is misclassified.

The input of Adaboost is a predefined set of positive and negative training examples. In our case the positive examples are face images and the negative examples are non-face images. At the testing stage of face detection process, a set of scanning windows also called subwindows is extracted from the original image. Each element from the set is used as input for the cascaded classifier. The cascade generated at the training step has the form of a generate decision tree.
The structure of the cascade reflects the fact that within any single image on overwhelming majority of sub-windows are negative. As such, the cascade attempts to reject as many negatives as possible at the earliest stage possible (Figure 3). Every layer consists of only a small number of features. While a positive instance will trigger the evaluation of every classifier in the cascade, this is an exceedingly rare event.

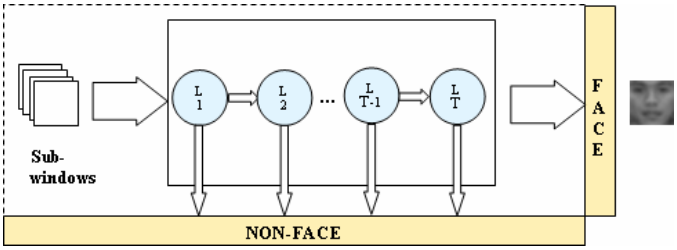

Figure 3: Cascaded classifier with T layers

Evolutionary Search

AdaBoost implies a brute force search on the whole space of rectangular Haar-like features. The process of training 162336 features would be time-consuming. Therefore it is beneficial to use GA in combination with AdaBoost (Figure 2).
The purpose of Genetic Algorithms (GA) in our research is to speed up the AdaBoost algorithm. This is done by replacing the exhaustive search of AdaBoost by a genetic search algorithm called Evolutionary Search (ES). The crossover and mutation genetic operators that drive the ES process are used for selecting features from the feature space. The fitness operator measures the performance associated to the use of a certain feature, for the all classification process.
The population consists of 250 features. At each stage, a feature is selected so that it satisfies the fitness function for minimum error for all the generated features. The process is similar to the criterion AdaBoost uses to select weak features.

Relevance Vector Machine

Tipping (Tipping 2000) proposed the Relevance Vector Machine (RVM) to recast the main ideas behind SVMs in a Bayesian context. A prior is introduced over the weights controlled by a set of hyperparameters, one associated with each weight, whose most probable values are iteratively estimated from the data.
The results have been shown to be as accurate and sparse as SVMs yet fit naturally into a regression framework and yield full probability distributions as their output.
The results in the case of face detection given some kernel functions are presented in Table 1. In the case of three features, the most efficient kernel function is chosen by using ROC curves. By analyzing Figure 4, it can be concluded that the best classification is obtained by using Laplace 4.0. This kernel function is further used in EABoost and in the process of constructing the final strong cascaded classifier.

Table 1: 2-fold cross validation results on three week classifiers for face detection based on Haar-like features

Kernel	Error rate		
	Feature 1	Feature 2	Feature 3
Gauss 2.0	26.30% ± 0.85	35.45% ± 7.71	38.60% ± 1.84
Gauss 5.0	25.35% ± 5.30	32.40% ± 2.83	36.55% ± 3.61
Laplace 0.5	35.20% ± 14.42	26.70% ± 0.99	42.70% ± 9.62
Laplace 2.0	29.25% ± 9.83	32.00% ± 7.50	41.60% ± 7.78
Laplace 5.0	26.20% ± 2.12	25.90% ± 1.84	37.45% ± 7.57

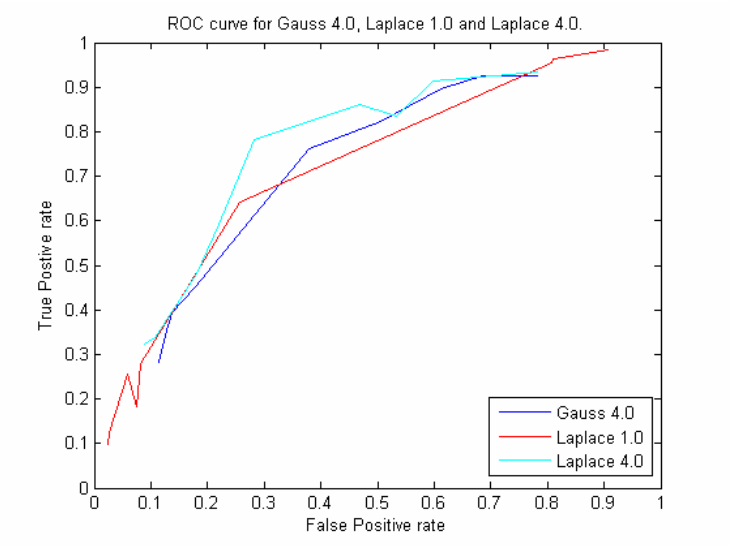

Figure 4: ROC curves of three kernels, obtained by adjusting each classifier's threshold

Facial characteristic point detection

Facial characteristic points (FCP) consist of 30 facial points. The detection of FCPs is based on a set of techniques that include corner detection, RVM and hybrid projection methods. The scanning of the whole picture is avoided by the use of a corner detector as a primary step for mouth, eyes and eyebrow FCP detection.

Fused corner detector

The corner detector of (Harris and Stephens 1988) takes into account the edge information. It also considers the neighborhood for corner decision, since the gradient swings sharply around the corners. The algorithm of (Sojka 2003) determines what neighborhoods are relevant for deciding whether or not a point is a corner by using a probability function. Where other corner detectors implicitly take into account the corner angle, the Sojka corner detection algorithm explicitly computes the corner angle. This helps to reduce erroneous detection of corners on contrast edges.
A mixed Harris-Stephens and Sojka corner detection algorithm is tuned to select enough corners so that the FCPs are also included.

This is because neither of the two detection algorithms is effective enough in detecting corners, which also includes all corner points of the facial features (mouth corners, eye corners, and eyebrow corners). Tests show that the efficiency of detecting the corner FCPs is increased by using such a combination. Given the selected set of corners, the next step is to identify the FCPs.

Classification of candidate corners

To classify the candidate corners selected by the fused detection algorithms, a set of RVMs is trained. For every corner point type a different RVM classifier is trained to distinguish the point from other points detected with the corner detector. The training model for corners employs the boosting of simple rectangle features. The set of features is limited to five basic feature types.

Hybrid projection

Using the combination of corner detectors along with RVM does not enable us to detect all facial characteristic points. To detect the remaining FCPs, a projection method called the hybrid projection (Zhou and Geng 2002) is used. The FCPs can be located at the corresponding boundaries of a face feature. To locate the horizontal boundaries of the features we analyze the horizontal intensity variations in the image containing only the face feature.

The final step aims at deriving the FCPs that were not identified at the previous stages, by using a hybrid projection method. This can be done by calculating a parabolic curve through the detected FCPs.

IMPLEMENTATION AND RESULTS

Face detection

We designed a learning model to boost the performance of RVM. This model consists of different techniques and algorithms described in the current paper.
The learning procedure is based on the AdaBoost learning algorithm. This algorithm is perfectly suited for the selection of the best features that boost up the performance of the classifier. As known for AdaBoost training, it is slow since it contains a brute force search. In addition, training of the RVM itself is relatively slow. And since they are combined, there is a continuous feedback from RVM to AdaBoost and the other way around.
A genetic search algorithm is added to improve the learning speed. Instead of a training time in the order of weeks/months, this is reduced to hours/days (on an AMD Athlon™ XP 2200+ 1.80 GHz processor with 512 MB RAM). Note that the size of the chosen training dataset is also significant for the speed of the training. After the learning procedure, faces can be distinguished from non-faces using the trained RVMs.
A cascade consists of several layers of classifiers. Each classifier is a combination of a number of RVMs. A practical problem that we encounter incorporating the cascade technique is that a lot of RVMs need to be trained.

Table 2: RVM test results, both training and testing are performed on MIT CBCL database

Kernel	Nr. of test samples	Detection rate %	Nr. of false negatives	Nr. of false positives
Gauss 5.0	2500	93.92	103	49
Gauss 7.0		95.08	58	49
Laplace 2.0		83.88	339	64
Laplace 5.0		95.04	60	64

Given a few kernel functions, the results of RVM classifier for face detection are presented in Table 2 for the same testing set as the training set, and in Table 3 for different data set for testing stage.

Table 3: RVM test results, the training is done using MIT CBCL, the testing is done on CMU database

Kernel	CMU database consisting of faces only		CMU database consisting of non-faces only	
	Number of test samples	Detection rate %	Number of test samples	Detection rate %
Laplace 2.0	472	22.03	5036	100
Laplace 5.0		51.91		97.34
Gauss 5.0		38.77		96.68
Gauss 7.0		30.30		97.86

However, the test results show that improvement needs to be made. In the current state, the face detector consists of only five layers of classifiers. Recall that in (Viola and Jones 2001) a cascade of 32 layers with over 4000 features is used. Better results are expected by involving more classifiers to the face detector.

Facial characteristic point detection

The same learning model for training the face detection classifier is used for the FCP detection component. Unlike in the case of face detection, no databases of FCPs exist which we can use as our dataset. These databases are extracted manually by us from the BioID and Carnegie Mellon face database. For the detection of the FCPs, a corner detection algorithm is used to filter out the non-FCPs. We have chosen for a combination of the Harris corner detection algorithm and the Sojka corner detection algorithm. Not all of the non-FCPs can be filtered out by these corner detectors. For this, we rely on the corresponding RVMs. The performance of the RVM in the final system is actually determined by that of the corner detectors.
For the FCPs that cannot be detected by the corner detectors, we use the Hybrid Projection technique. This technique is applied on the corresponding facial feature (eye, eye brow and mouth) on which the FCP is localized. Therefore, RVMs are trained to extract these facial features before applying the projection method. The test results of the FCP detector (see Table 4) show that some of the FCPs can be detected better

than others. The explanation for the relative poor performance of some FCPs is probably that the FCP pattern itself is non-stable from the recognition point of view. For instance, the mouth can have different shapes and some associated parameters could exceed the value ranges of the samples used at the training stage, at different expressions.

To detect the FCPs we need to account that noise is very probable at corner regions. Taking this into account it means that at the training of the RVM noise is included in the training samples. This affects the final performance of the RVM. It is a trade-off that needs to be made. In the case of invoking the projection method, finding the boundaries is proven to be very robust, except if the feature boundary is distorted.

Table 4: FCP Detection Results

FCP	True positive rate (%)	False positive rate (%)
Right eye inner corner	81.82	6.75
Right eye outer corner	81.82	16.67
Right eye upper corner	88.64	11.63
Right eye lower corner	88.64	11.63
Left eye inner corner	81.82	3.49
Left eye outer corner	63.64	5.94
Left eye upper corner	82.95	17.05
Left eye lower corner	82.95	17.05
Mouth left corner	86.36	3.24
Mouth right corner	90.91	4.71
Mouth upper corner	90.91	9.08
Mouth lower corner	90.91	9.08

CONCLUSION

We have presented an approach using a sparse learning model as the first step towards a fully automatic facial expression recognition system. This learning model is applied on face detection and FCP detection. The test results reveal that some improvements are still to be made.

In the current situation, a detected face cannot be further processed by the FCP detection module if the face is slightly rotated. Some of the FCPs can be occluded by other parts of the face. The face detection module is trained on a database with unaligned faces. Some of them are slightly rotated to the left, some to the right, some looking up, etc. For the two modules to work together perfectly, the face detector should be trained strictly on full frontal aligned faces. This is because the FCP detection module is designed to work with these faces.

The model may be improved by considering a faster implementation of the training application. Other variants of the AdaBoost may also be considered. They differ in the updating schemes for the weights. In the face detection module, the scanning process can be speed up by other techniques. Using edge detectors, plain backgrounds might be filtered out and pruned from being scanned. This reduces the overall scanning time on different resolutions. The performance of the system can also be improved by using an extended set of the Haar-like features. In our training model, we used only 5 simple features. The detection rate during training may be increased by incorporating the bootstrapping method. This method uses misclassified samples as training input in the next iteration. This way we can force the learning algorithm to adapt the output results from previous training rounds. We have not implemented this procedure in the current training model because this would certainly affect the training time negatively.

REFERENCES

Ekman, P. and W. Friesen. 1978. “Facial Action Coding System.” Consulting Psychologists Press, Inc., Palo Alto California, USA.

Freund, Y. and R.E. Schapire. 1995. “A decision-theoretic generalization of on-line learning and an application to boosting.” In *2nd European Conference on Computational Learning Theory*.

Harris, C.G. and M. Stephens. 1988. “A combined corner and edge detector”. Proceedings 4th Alvey Vision Conference, Manchester, 189-192.

Jongh de, E.J. 2002. “FED: An online facial expression dictionary as a first step in the creation of a complete nonverbal dictionary.” TU Delft.

Kearney, G.D. and S. McKenzie. 1993. “Machine interpretation of emotion: design of a memory-based expert system for interpreting facial expressions in terms of signaled emotions.” (JANUS). Cognitive Science 17, Vol. 4, 589–622.

Kobayashi, H. and F. Hara. 1997. “Facial Interaction Between Animated 3D Face Robot and Human Beings”. IEEE Computer Society Press, 3732-3737.

Morishima, S. and H. Harashima. 1993. “Emotion Space for Analysis and Synthesis of Facial Expression”. IEEE International Workshop on Robot and Human Communication, 674-680.

Rothkrantz, L.J.M. and M. Pantic. 2000. “Expert Systems for Automatic Analysis of Facial Expressions”. Elsevier, Image and Computing, Vol. 18, 881-905.

Sojka E. 2003. “A New Approach to Detecting Corners in Digital Images”. Accepted for Publication in IEEE ICIP.

Tipping, M.E. 2000. “The Relevance Vector Machine. Advances in Neural Information Processing Systems”. Vol. 12, 652-658.

Treptow, A. and A. Zell. 2003. “Combining Adaboost Learning and Evolutionary Search to Select Features for Real-time Object Detection”. University of Tuebingen, Department of Computer Science, Germany.

Viola, P. and M. Jones. 2001. “Robust Real-time Object Detection.” Second International Workshop on Statistical and Computational Theories of Vision-Modeling, Learning, Computing, and Sampling.

Zhao J. and G. Kearney. 1996. “Classifying facial emotions by back-propagation neural networks with fuzzy inputs”. International Conference on Neural Information Processing, Vol. 1, 454–457.

Zhou, Z.-H. and X. Geng. 2002. “Projection Functions for Eye Detection”. State Key Labaratory for Novel Software Technology, NU, China.