Non-Linear Approaches for the Classification of Facial Expressions at Varying Degrees of Intensity

Jane Reilly, John Ghent, John McDonald
Computer Vision and Imaging Laboratory
Department of Computer Science
National University of Ireland Maynooth
{jreilly, jghent, johnmcd} @cs.nuim.ie

Abstract

The research discussed in this paper documents a comparative analysis of two nonlinear dimensionality reduction techniques for the classification of facial expressions at varying degrees of intensity. These nonlinear dimensionality reduction techniques are Kernel Principal Component Analysis (KPCA) and Locally Linear Embedding (LLE). The approaches presented in this paper employ psychological tools, computer vision techniques and machine learning algorithms. In this paper we concentrate on comparing the performance of these two techniques when combined with Support Vector Machines (SVMs) at the task of classifying facial expressions across the full expression intensity range from near-neutral to extreme facial expression. Receiver Operating Characteristic (ROC) curve analysis is employed as a means of comprehensively comparing the results of these techniques.

1. Introduction

Facial expressions play a major role in how people communicate. Therefore a computer that could interpret and react to facial expression would advance human-computer interfaces, providing a basis for communication that could be compared to human-human interaction. As the complexity of computer applications increase in tandem with user expectations, the development of emotionally intelligent interfaces has become both necessary and possible.

This paper builds on preliminary results reported in [12] and [21], where two nonlinear dimensionality reduction techniques were examined in the context of facial expression classification. In those initial investigations we looked at the performance of *Kernel Principal Component Analysis* (KPCA) and *Locally Linear Embedding* (LLE) in isolation on a small dataset, while in this paper we perform a comprehensive comparative analysis of these techniques on a more substantial dataset. While previously we concentrated on the number of correct classifications, in this paper we extend this work to examine the performance of our techniques at classifying facial expressions at varying degrees of intensity using *Receiver Operating Characteristic* (ROC) analysis. By performing ROC analysis we not only look at the number of correct classifications, but we also consider the number of incorrect classifications when deciding which technique performs best (see section 2.5 for more details on ROC analysis).

Since the importance of facial expressions was first established in 1872 [6], many studies have been carried out attempting to interpret their meaning. In more recent years a considerable amount of research has been performed investigating various methodologies for the classification of facial expressions. Techniques range from template based methods [18], to neural network based methods [8], or a combination of the two [20, 11]. Perhaps the most substantial work in this area has been done by Bartlett *et al* [1]. Bartlett *et al.* proposed a technique which combines Gabor wavelets and SVMs to classify AUs with 93.3% accuracy [1]. Again in [16], Littlewort and Bartlett propose a similar technique which classifies AUs with 97% accuracy. In [17], Bartlett used SVMs again to successfully distinguish between genuine and fake smiles.

One of the techniques which we appraise in this paper, KPCA is a non-linear dimensionality reduction technique which is closely related to methods applied to SVMs [28, 27]. The principal idea behind KPCA is to transform the input data into a potentially infinite high dimensional space. This is achieved by utilizing the kernel trick to perform feature space operations using dot-products between data points in the feature space (for more information see [10]).

KPCA and more prominently kernel methods have been used extensively in recent years [2]. Several methods have

0-7695-2887-2/07 $25.00 © 2007 IEEE
DOI 10.1109/IMVIP.2007.11

been proposed for facial expression analysis [31, 30]. More specifically KPCA has been applied successfully for the purpose of de-noising of images of hand-written digits [19], and as a pre-processing step in regression problems [26]. On a related note to this paper, KPCA has also been used to analyse facial shape [30]. Here, we use KPCA to classify lower facial expressions by looking at the change in shape of the mouth as an expression forms.

In this paper we also discuss the effectiveness of LLE in conjunction with SVMs to classify expressions. LLE was introduced in 2000 as a non-linear dimensionality reduction technique that computes low dimensional neighborhood preserving embeddings of high dimensional data by unfolding the underlying manifold [24, 25].

The remainder of this paper is structured as follows: Section 2 documents our approach, Section 3 details experiments and results, and in Section 4 we conclude with some final remarks. This paper extends previous work detailed in [10, 21] and [12].

2. Proposed Methodology

Measuring facial expressions at varying expression intensity is a non trivial task as everyone's face is unique and interpersonal differences exist in how people perform facial expressions. Although numerous methodologies have been proposed to solve this problem, the main focus of our research is to measure facial expressions independent of identity and with varying degrees of intensity in a consistent and robust manner. In this section we discuss the techniques which we use to accurately classify lower facial expressions, namely; the facial action coding system, kernel principal component analysis, locally linear embedding, support vector machines and receiver operating characteristic curve analysis.

2.1. Facial Action Coding System - FACS

In this paper we use the *Facial Action Coding System* (FACS), which measures facial expressions according to the movement of muscles in the face. The system is based on an anatomical analysis of facial expressions [7]. The FACS allows us to subdivide our data into subsets where the variation in each expression is precisely characterized. The FACS provides an unambiguous, quantitative means of describing all movements of the face in terms of Action Units (AUs). An AU is composed of one or more muscles in the face that causes an atomic change in the faces appearance. All expressions can be described using the AUs defined by the FACS, providing a measurable set of criteria that define whether or not a particular facial expression is present.

However AUs do not always occur with the same expression intensity and for this reason the FACS also includes intensity ranges for the AUs. There are five intensities in total ranging from intensity A, where a trace change in appearance occurs, to intensity E, where an extreme or maximum appearance change occurs. Figure 1 provides a visual representation of the full FACS intensity range, which can be thought of as the spectrum of intensity for each AU. Note that this scale is divided non-uniformly - e.g. C takes up more of the spectrum than A [7].

Figure 1. *The scale of FACS intensity scores*

2.2. Kernel Principal Component Analysis - KPCA

$Principal\ Component\ Analysis$ (PCA, also known as the Karhunen-Loéve transform) is a technique used to lower the dimensionality of a feature space [29, 13]. *Kernel Principal Component Analysis* (KPCA) is similar to standard PCA except the data is projected into a higher dimensional feature space prior to performing eigenvector decomposition. We project the data into feature space through the use of the *kernel trick*. This *kernel trick* permits the computation of dot products in high dimensional *feature spaces*, using functions defined on pairs of input patterns.

More specifically, mapping from one space to a higher dimensional space involves a mapping from $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$, however, with an appropriate choice of kernel there exists a mapping ϕ such that $(\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) = K(\mathbf{x}_i, \mathbf{x}_j)$. This means that the inner products of the feature space can be calculated without computing $\phi(x)$ directly. This allows us to work in a potentially infinite high dimensional feature space. The choice of kernel is still a matter of debate, however, in this paper we use a *Gaussian* kernel. The Gaussian kernel is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)/2\sigma^2}. \quad (1)$$

Where σ determines the width of the kernel.

2.3. Local Linear Embedding - LLE

The LLE algorithm was introduced by Saul and Roweis in 2000 as an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embeddings of high dimensional data [24, 25]. Many extensions and adjustments to this core algorithm have been proposed, ranging from Robust-LLE [3, 14] to supervised [23, 15] and semi-supervised versions of LLE [22], however in this paper we are using the original algorithm as defined in [24, 25].

The LLE algorithm is based on simple geometric intuitions where the algorithm attempts to compute a low dimensional embedding with the property that nearby points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space. The LLE algorithm takes a dataset of N real valued vectors $\mathbf{X}_i$, each of dimensionality D, sampled from some smooth underlying manifold as its input. Provided there is sufficient data such that the manifold is well sampled, we can expect each data point and its neighbours to lie on or close to a locally linear patch of the manifold [25].

There are three main steps in the LLE algorithm. Firstly the manifold is sampled and the *K* nearest neighbors per data point are identified. Secondly each point $\mathbf{X}_i$ is approximated as a linear combination of its neighbors $\mathbf{X}_j$, these linear combinations are then used to construct the sparse weight matrix $\mathbf{W}_{ij}$. Reconstruction errors are then measured by the cost function Equation 2, which adds up the squared distances between all the data points and their reconstructions.

$$\sum \mathbf{W} = \sum_i |\overrightarrow{\mathbf{X}_i} - \sum_j \mathbf{W}_{ij} \overrightarrow{\mathbf{X}_j}|^2 \tag{2}$$

In the final step of the LLE algorithm, each point $\mathbf{X}_i$ in the high dimensional space is mapped to a point $\mathbf{Y}_i$ in the low dimensional space which best preserves the structure and geometry of $\mathbf{X}_i$'s neighborhood. This geometry and structure is represented by the weight matrix $\mathbf{W}_{ij}$. The mapping from $\mathbf{X}_i$ to $\mathbf{Y}_i$ is achieved by fixing the weights $\mathbf{W}_{ij}$ and selecting the bottom d non zero coordinates of each output $\mathbf{Y}_i$ to minimize Equation 3.

$$\phi \mathbf{Y} = \sum_i |\overrightarrow{\mathbf{Y}_i} - \sum_j \mathbf{W}_{ij} \overrightarrow{\mathbf{Y}_j}|^2. \tag{3}$$

2.4. Support Vector Machines - SVMs

SVMs are used to classify an expression using the outputs from both the KPCA model and the LLE model as inputs. The SVM algorithm can be separated into two distinct procedures, the *kernel trick*, which we have already discussed, and the *base algorithm*. Suppose we have a dataset $(x_1, y_1), ..., (x_m, y_m) \in \mathbf{X} \times \{\pm 1\}$ where $\mathbf{X}$ is some space from which the x_i have been sampled. We can construct a dual Lagrangian of the form

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4}$$

which are subject to the constraints $\alpha_i \geq 0 \; \forall i$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$. The solution to Equation 4 is a set of α values which defines a hyperplane that is positioned in an optimal location between the two classes. Further details of the construction of this equation can be found in [10].

2.5. Receiver Operating Characteristic (ROC) Curves

Receiver Operating Characteristic (ROC) analysis is a technique for visualizing, organizing and selecting classifiers based on their performance, and was first used during World War 2 as a means of assessing the capabilities of radar systems in distinguishing random interferences from actual targets [9]. Today, ROC analysis is becoming increasingly important in the area of cost sensitive classification, classification in the presence of unbalanced classes, robust comparison of classifier performance under imprecise class distribution and misclassification costs.

Given a classifier and an instance, there are four possible outcomes: *True Positive* occurs when a test correctly returns a positive result; *True Negative* occurs when a test correctly returns a negative results; *False Positive* occurs when a test incorrectly returns a positive result; *False Negative* occurs when a test incorrectly returns a negative result. These four values can then be used to create a *Confusion Matrix*, which can be seen in Table 1. The numbers along the major diagonal represent the correct outcomes, and the numbers off this diagonal represents the errors - confusion between the various classes [9]. The values in the confusion matrix are used to compile various ratios such as the *True Positive Rate* shown in Equation 5 and the *False Positive Rate* shown in Equation 6.

	Y	**N**
p	**T**rue **P**ositives	**F**alse **N**egatives
n	**F**alse **P**ositives	**T**rue **N**egatives

Table 1. *Confusion Matrix*

$$TruePositiveRate = \frac{TP}{TotalPositives} \tag{5}$$

$$FalsePositiveRate = 1 - \frac{FP}{TotalNegatives} \tag{6}$$

ROC graphs are two dimensional graphs with the true positive rate plotted on the y-axis, and the false positive rate plotted on the x-axis. A ROC graph depicts relative trade offs between the benefits and costs of a particular classifier. The most frequently used performance measure in ROC analysis is the *area under the ROC curve*, more commonly referred to as the AUC. The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. The AUC ranges from 0 - 1, however as the random classifier has an AUC of 0.5, no realistic classifier should have an AUC less than 0.5.

3. Experiments and Results

In our research to date we have used the *Cohn-Kanade AU-Coded Facial Expression Database* [4]. This database contains approximately 2000 image sequences from over 200 subjects. The subjects came from a cross-cultural background and were aged approximately 18 to 30. This database contains full AU coding and partial intensity coding of facial images and while it is not ideal in that it isn't completely intensity coded, it is the most comprehensive database currently available.

In this paper we classify four lower facial expressions; AU20+25, AU25+27, AU10+20+25, and AU12. The effect that these have on the mouth is illustrated in Table 2. For a detailed description of each expression refer to [7].

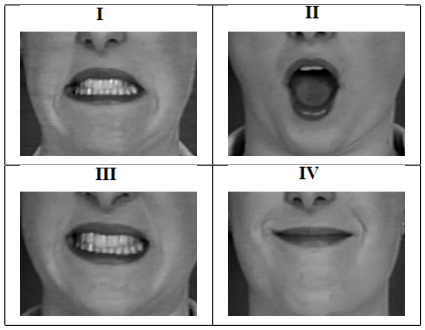

Table 2. *This table illustrates the effect of portraying four different expressions. The AUs portrayed are: I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12*

Our training dataset consisted of 73 images of one subject performing 4 sequences from neutral to extreme expression. Our test dataset consisted of 522 images of multiple subjects from multi-cultural backgrounds performing the four lower facial expressions as shown in Table 2. In our test dataset we sampled the sequences at each intensity rating including neutral, 6 in total, while in our training set we labeled each frame in the sequence with an intensity score. Prior to experimentation we manually placed 24 landmark points placed at 24 specific locations, shown in Figure 2. The optimal position of these landmark points is defined by Cootes in [5].

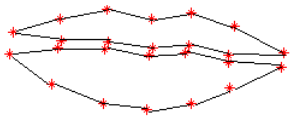

Figure 2. *An example of a shape of a neutral mouth.*

To analyze the variance of the points that describe the shape of the mouth it is necessary that the mouth shapes in the training and test sets are as closely aligned as possible. Therefore, before applying the LLE and KPCA algorithms we align our data by firstly, centering the data and then performing *Generalized Procrustes Alignment* (GPA) [13]. GPA aligns two shapes with respect to position, rotation and scale by minimizing the weighted sum of the squared distances between the corresponding landmark points. More information on this technique can be found in [10].

Once GPA has been performed, we can either apply KPCA and LLE directly or we can further reduce the variance in the dataset by performing *shape differencing*, whereby we subtract the neutral shape from each subsequent shape in the sequence. The reason being is that we are interested in the difference between the two shapes and not the actual shapes themselves. Once this preprocessing has been completed we calculate our shape models by performing LLE and KPCA on the aligned training data, the outputs of which are used to train the SVM classifiers. New unseen data is projected into both shape spaces and the outputs are used as inputs to the previously trained SVM classifiers.

3.1. Locally Linear Embedding Results

Figure 3 shows the LLE expression shape space created as a result of applying LLE to our training set without using shape differencing and Figure 4 displays the LLE shape space after performing the additional step of *Shape Differencing*. It is clear from these diagrams that LLE has successfully separated out the input dataset into the four lower facial expressions which we wish to classify, however, as an improved separation of the expressions has occurred due to the using the additional shape differencing step, we use this expression space as the basis for performing SVM classification.

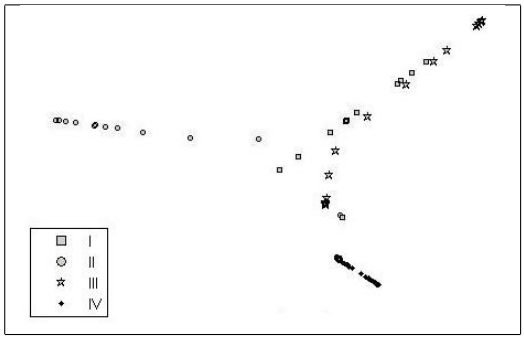

Figure 3. *LLE low dimensional expression space when shape differencing is not used. From the legend the expression are: I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12*

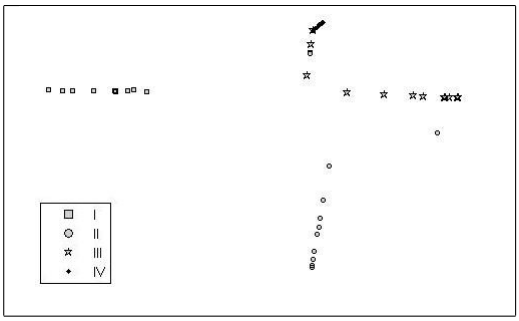

Figure 4. *LLE low dimensional expression space when shape differencing is used. From the legend the expression are: I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12*

The aim of this experiment was to establish how our LLE-SVM based technique performs at classifying expressions at varying degrees of intensity, therefore for each of our expressions we separated our testing and training data into intensity five groupings: ABCDE; BCDE; CDE; DE; E, all of which contain the neutral expression (for example the first grouping contained the neutral along with intensities A, B, C, D and E). As we had four expressions to classify we need four one-against-all nonlinear SVM classifiers, the optimal SVM for each facial expression was selected based on the results of the classifier-test cases defined by the intensity groupings, an example of these groupings for expression 1 are shown in Table 3.

		Expression 1 - test cases				
Expression I training	ABCDE	ABCDE	BCDE	CDE	DE	E
	BCDE	ABCDE	BCDE	CDE	DE	E
	CDE	ABCDE	BCDE	CDE	DE	E
	DE	ABCDE	BCDE	CDE	DE	E
	E	ABCDE	BCDE	CDE	DE	E

Table 3. *Expression I Intensity groupings, which are used to evaluate the SVM. For each expression we have 20 SVM training - testing pairs.*

Once our classifiers were trained we tested them using the various test groupings to determine at what level the classifier broke down. For example we trained our classifier using the neutral and extreme shapes and tested it on data containing the entire intensity range to ascertain if the intervening intensities were correctly classified as belonging to the expression.

Using the outputs of our various SVMs we performed ROC curve analysis, the optimal ROC curve for each of our four expressions is shown in Table 4, and the corresponding mean AUC for the expressions are displayed in Table 6, extracts from the confusion matrics are shown in Table 5, with the complete classifier results in Table 10. We can clearly see from these results that the overall average AUC achieved by our LLE-SVM based technique is 0.81. As expected we achieved lower AUCs for the two similar facial expressions I and III, while our other two expressions II and IV performed significantly better with mean AUCs of 0.92 and 0.98 respectively. This drop in AUC for expressions I and III can be explained by looking at the detailed classifier results in Table 10, where we can see a higher number of reported false positives.

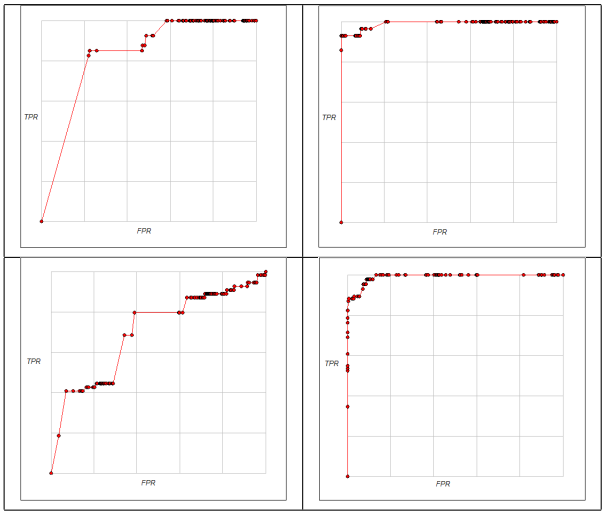

Table 4. *LLE-SVM Optimal ROC curves. The four facial expressions are: top left AU20+25; top right AU25+27; bottom left AU10+20+25; bottom right AU12*

	Y	*N*	*Y*	*N*	*Y*	*N*	*Y*	*N*
p	52	93	69	1	25	31	62	0
n	48	329	71	381	65	401	43	417

Table 5. *LLE: Extract from confusion matrices of the four expressions. From left to right AU20+25, AU25+27, AU10+20+25 and AU12. Correct outcomes lie on the major diagonals and the numbers off this diagonal represent the errors.*

3.2. Kernel Principal Component Analysis Results

Figure 5 shows the output space established as a result of applying KPCA to the training dataset when shape differencing is not used, and Figure 6 shows the results achieved when the additional shape differencing step is performed.

AU	$\bar{AUC}$	$\pm$
I-v-all	0.7250	0.0666
II-v-all	0.9164	0.0712
III-v-all	0.6236	0.0516
IV-v-all	0.9798	0.0202
$AverageAUC$	0.8110	

Table 6. *Mean AUC for LLE-SVM based technique. In the table above I=AU20+25, II=AU25+27, III=AU10+20+25 and IV=AU12.*

While there is not a large variation in how KPCA separated out these datasets, in order to compare the two techniques, we used the latter expression space in our experimentation. As with the previous experiment, the aim was to determine how our KPCA-SVM technique performs at classifying the four lower facial expressions at varying degrees of intensity therefore we used the same intensity groupings as for the LLE experiments as shown in Table 3.

Once our data was separated out into intensity groupings we performed KPCA on the data, and used the resulting shape spaces to train one-against-all SVMs. However, unlike the previous LLE based experiment, as the kernel trick has already been performed inside the KPCA algorithm, we use linear SVMs to classify the facial expressions.

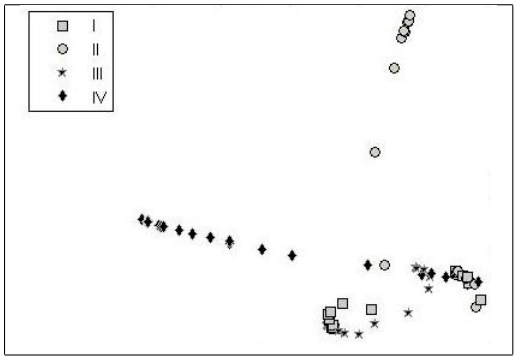

Figure 5. *KPCA low dimensional expression space when shape differencing is not used. Expression I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12*

Due to the fact that our KPCA based technique failed to separate out expressions I and III in the lower dimensional space, it is not surprising that the ROC curves for these expressions are concave. In order to to correctly interpret the results of such a seemingly weak classifier we look at extracts from the confusion matrices in Table 8 and also the mean AUC in Table 9. Here we can see that the AUC of expression I and III is very low, and when we look at the detiled results in Table 11 we also see that for these two

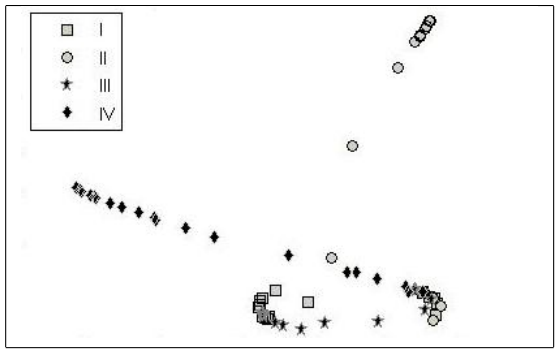

Figure 6. *KPCA low dimensional expression space when shape differencing is used. Expressions I=AU20+25; II=AU25+27; III=AU10+20+25; IV=AU12*

expressions there are no true positive classifications.

4. Conclusion

The accurate classification of facial expressions is a growing problem within several domains. The solution described in this paper takes a multidisciplinary approach drawing together psychological tools, statistical models and machine learning techniques.

The two shape models were calculated using KPCA and LLE to lower the dimensionality of the problem. We trained one-against-all SVMs to classify the four expressions (AU20+25, AU25+27, AU10+20+25 and AU12) at varying degrees of intensity. We found that overall the LLE-SVM based technique consistently performed better than the KPCA-SVM based technique. We can see from the LLE ROC curves in Table 4 that even though there is a drop in the AUCs for the two similar facial expressions when compared with the AUCs of the other expressions, this is still a satisfactory result as the differentiation of two such similar expressions is a non trivial task.

5. Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

References

[1] M. S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. *IEEE International conference on systems, man and cybernetics*, pages 592–597, October 2004.

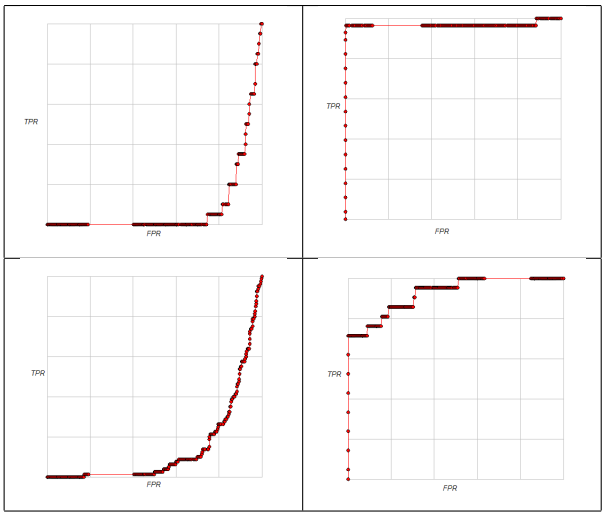

Table 7. *KPCA Optimal ROC curves showing results of the application of KPCA and linear SVMs. The four facial expressions are: top left AU20+25; top right AU25+27; bottom left AU10+20+25; bottom right AU12*

	Y	N	Y	N	Y	N	Y	N
p	0	0	46	0	0	0	17	0
n	100	422	94	302	90	432	85	417

Table 8. *KPCA extract from confusion matrices of the four expressions. From left to right AU20+25, AU25+27, AU10+20+25 and AU12. Correct outcomes lie on the major diagonals and the numbers off this diagonal represent the errors.*

AU	$\bar{AUC}$	$\pm$
I-v-all	0.1122	0.0567
II-v-all	0.8982	0.0639
III-v-all	0.1336	0.0176
IV-v-all	0.8855	0.0404
$AverageAUC$	0.5070	

Table 9. *KPCA mean AUC In the table above I=AU20+25, II=AU25+27, III=AU10+20+25 and IV=AU12.*

[2] C. Campbell. kernel methods: A survey of current techniques. *Neurocomputing*, 48:63–84, 2002.

[3] Y. Chang and D. Y. Yeung. Robust local linear embedding. Technical report, Department of Computer Science, Hong Kong University of Science and Technology, HKUST-CS05-12, 2005.

[4] J. Cohn and Kanade. Cohn-kanade au-coded facial expression database. Technical report, Pittsburgh University, 1999.

[5] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. Technical report, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, October 2001.

[6] C. Darwin and P. Ekman. *The expression of the emotions in man and animal*. Chicago: The University of Chicago Press, 1998. 1st edition in 1872, 2nd edition in 1889, 3rd edition with additional commentry by Paul Ekman in 1998.

[7] P. Ekman, W. Friesen, and J. Hager. Facial action coding system. *Consulting Psychologists Press*, 1978.

[8] M. J. Er, S. Wu, J. Lu, and H. L. Toh. Face recognition with radial basis function neural networks. *IEEE transactions on neural networks*, 13(3):697–710, 2002.

[9] T. Fawcett. Roc graphs: Notes and practical considerations for data mining researchers. 2003.

[10] J. Ghent. *A Computational Model of Facial Expression*. PhD thesis, National University of Ireland Maynooth, Co. kildare, Ireland, July 2005.

[11] J. Ghent and J. McDonald. Holistic facial expression classification. *Opto-Ireland*, April 2005.

[12] J. Ghent, J. Reilly, and J. McDonald. Classification of facial expressions using kernel principal component analysis. *Proceedings of the Irish machine vision and image processing conference*, 2006.

[13] J. C. Gower. Generalised procrustes analysis. *Psychometrika*, 40:33–50, 1975.

[14] A. Hadid and M. PietikAanien. Efficient locally linear embeddings of imperfect manifolds. *Proceedings of the Third International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany*, pages 188–201, 2003.

[15] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. PietikAainen. Beyond locally linear embedding algorithm. Technical report, Department of Electrical and Information Engerring, University of Oulu, Oulu, Finland, MVG-01-2002, 2002.

[16] G. Littlewort, M. S. Bartlett, I. F. andJ. Chenu, T. Kanda, H. Ishiguro, and J. Movellan. Towards social robots: automatic evaluation of human-robot interaction by face detection and expression classification. *Advances in Neural Information Processing Systems*, 16:1563–1570, 2004.

[17] G. Littlewort-Ford, M. S. Bartlett, and J. R. Movellan. Are your eyes smiling? detecting genuine smiles with support vector machines and gabor wavelets. *Proceedings of the 8th annual joint symposium on neural computation*, 2001.

[18] M. J. Lyons, J. Budyek, and S. Akamatsu. Automatic classification of single facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 21(12):1357–1362, 1999.

[19] S. Mika, B. Schlkopf, A. Smola, K. Miller, M. Scholz, and G. Rtsch. Kernel pca ans de-noising of feature spaces. *Advances in Neural Infrmation Processing Systems*, 11.

[20] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: the state of the art. *IEEE transactions on pattern analysis and machine learning*, 22(12), 2000.

[21] J. Reilly, J. Ghent, and J. McDonald. Investigating the dyanmics of facial expression. *Proceedings of the 2nd International Symposium on Visual Computing*, November 2006.

Exp I	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	422	100	0	0	422	100	**0.223**	422	80	0	0	422	80	**0.135**	422	60	0	0	422	60	**0.083**	422	40	0	0	422	40	**0.073**
	NBCDE	422	100	0	0	422	100	**0.100**	422	80	0	0	422	80	**0.143**	422	60	0	0	422	60	**0.100**	422	40	0	0	422	40	**0.078**
	NCDE	422	100	0	0	422	100	**0.023**	422	80	0	0	422	80	**0.114**	422	60	0	0	422	60	**0.077**	422	40	0	0	422	40	**0.060**
	NDE	422	100	0	0	422	100	**0.236**	422	80	0	0	422	80	**0.146**	422	60	0	0	422	60	**0.103**	422	40	0	0	422	40	**0.081**
	NE	422	100	0	0	422	100	**0.230**	422	80	0	0	422	80	**0.141**	422	60	0	0	422	60	**0.098**	422	40	0	0	422	40	**0.076**

Exp II	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	382	140	46	0	382	94	**0.804**	382	112	46	0	382	66	**0.850**	382	84	45	0	382	39	**0.921**	382	56	38	0	382	18	**0.950**
	NBCDE	382	140	43	0	382	97	**0.804**	382	112	43	0	382	69	**0.850**	382	84	42	0	382	42	**0.921**	382	56	35	0	382	21	**0.950**
	NCDE	382	140	38	0	382	102	**0.801**	382	112	38	0	382	74	**0.850**	382	84	37	0	382	47	**0.921**	382	56	32	0	382	24	**0.950**
	NDE	382	140	43	0	382	97	**0.801**	382	112	43	0	382	69	**0.850**	382	84	42	0	382	42	**0.921**	382	56	35	0	382	21	**0.950**
	NE	382	140	15	0	382	125	**0.801**	382	112	15	0	382	97	**0.850**	382	84	15	0	382	69	**0.921**	382	56	13	0	382	43	**0.950**

Exp III	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	432	90	0	0	432	90	**0.169**	432	72	0	0	432	72	**0.146**	432	54	0	0	432	54	**0.133**	432	36	0	0	432	18	**0.141**
	NBCDE	432	90	0	0	432	90	**0.163**	432	72	0	0	432	72	**0.135**	432	54	0	0	432	54	**0.120**	432	36	0	0	432	18	**0.128**
	NCDE	432	90	0	0	432	90	**0.161**	432	72	0	0	432	72	**0.131**	432	54	0	0	432	54	**0.117**	432	36	0	0	432	18	**0.126**
	NDE	432	90	0	0	432	90	**0.164**	432	72	0	0	432	72	**0.137**	432	54	0	0	432	54	**0.124**	432	36	0	0	432	18	**0.128**
	NE	432	90	0	0	432	90	**0.159**	432	72	0	0	432	72	**0.128**	432	54	0	0	432	54	**0.113**	432	36	0	0	432	18	**0.114**

Exp IV	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	417	105	17	0	417	88	**0.776**	417	84	17	0	417	67	**0.868**	417	63	16	0	417	47	**0.903**	417	42	13	0	417	29	**0.921**
	NBCDE	417	105	13	0	417	92	**0.816**	417	84	13	0	417	71	**0.868**	417	63	12	0	417	51	**0.903**	417	42	11	0	417	31	**0.920**
	NCDE	417	105	8	0	417	97	**0.816**	417	84	8	0	417	76	**0.868**	417	63	8	0	417	55	**0.903**	417	42	7	0	417	35	**0.920**
	NDE	417	105	8	0	417	97	**0.816**	417	84	8	0	417	76	**0.868**	417	63	8	0	417	55	**0.903**	417	42	7	0	417	35	**0.920**
	NE	417	105	5	0	417	100	**0.816**	417	84	5	0	417	79	**0.868**	417	63	5	0	417	58	**0.903**	417	42	4	0	417	38	**0.920**

Exp I	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	422	100	52	93	329	48	**0.647**	422	80	51	93	329	29	**0.729**	422	60	45	93	329	15	**0.787**	422	40	33	93	329	7	**0.826**
	NBCDE	422	100	53	128	294	47	**0.614**	422	80	52	128	294	28	**0.697**	422	60	46	128	294	14	**0.751**	422	40	34	128	294	6	**0.790**
	NCDE	422	100	53	123	299	47	**0.616**	422	80	52	123	299	28	**0.701**	422	60	46	123	299	14	**0.755**	422	40	34	123	299	6	**0.795**
	NDE	422	100	42	109	313	58	**0.613**	422	80	41	109	313	39	**0.692**	422	60	37	109	313	23	**0.746**	422	40	28	109	313	12	**0.786**
	NE	422	100	32	99	323	68	**0.610**	422	80	31	99	323	49	**0.682**	422	60	27	99	323	33	**0.732**	422	40	21	99	323	19	**0.772**

Exp II	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	382	140	69	1	381	71	**0.776**	382	112	68	1	381	44	**0.874**	382	84	60	1	381	24	**0.922**	382	56	41	1	381	15	**0.937**
	NBCDE	382	140	58	0	382	82	**0.802**	382	112	58	0	382	54	**0.904**	382	84	55	0	382	29	**0.954**	382	56	44	0	382	12	**0.978**
	NCDE	382	140	54	0	382	86	**0.801**	382	112	54	0	382	58	**0.903**	382	84	52	0	382	32	**0.953**	382	56	41	0	382	15	**0.978**
	NDE	382	140	67	4	378	73	**0.799**	382	112	67	4	378	45	**0.901**	382	84	61	4	378	23	**0.950**	382	56	48	4	378	8	**0.974**
	NE	382	140	74	4	378	66	**0.801**	382	112	55	0	382	57	**0.904**	382	84	52	0	382	32	**0.954**	382	56	42	0	382	14	**0.978**

Exp III	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	432	90	25	31	401	65	**0.595**	432	72	24	31	401	48	**0.663**	432	54	22	31	401	32	**0.669**	432	36	17	31	401	19	**0.720**
	NBCDE	432	90	27	95	337	63	**0.546**	432	72	28	95	237	46	**0.601**	432	54	23	95	337	31	**0.641**	432	36	17	95	337	19	**0.671**
	NCDE	432	90	59	256	167	31	**0.560**	432	72	57	265	167	15	**0.617**	432	54	25	104	328	29	**0.660**	432	36	19	104	328	17	**0.688**
	NDE	432	90	10	17	415	80	**0.546**	432	72	10	17	415	62	**0.560**	432	54	9	17	415	45	**0.588**	432	36	9	17	415	27	**0.627**
	NE	432	90	28	78	354	62	**0.530**	432	72	27	78	354	45	**0.588**	432	54	25	78	354	29	**0.637**	432	36	19	78	354	17	**0.670**

Exp IV	TEST ▶	N-ALL							NBCDE							NCDE							NDE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	417	105	62	0	417	43	**0.989**	417	84	58	0	417	26	**0.990**	417	63	49	0	417	14	**0.989**	417	42	35	0	417	7	**0.990**
	NBCDE	417	105	22	0	417	83	**0.987**	417	84	22	0	417	62	**0.990**	417	63	21	0	417	42	**0.990**	417	42	17	0	417	25	**0.992**
	NCDE	417	105	81	1	416	24	**0.986**	417	84	69	1	416	15	**0.989**	417	63	53	1	416	10	**0.989**	417	42	36	1	416	6	**0.991**
	NDE	417	105	54	0	417	51	**0.987**	417	84	51	0	417	33	**0.990**	417	63	45	0	417	18	**0.990**	417	42	32	0	417	10	**0.992**
	NE	417	105	13	0	417	92	**0.928**	417	84	13	0	417	71	**0.935**	417	63	13	0	417	50	**0.943**	417	42	10	0	417	32	**0.948**

Exp I	TEST ▶	NDE							NE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	422	40	0	0	422	40	**0.076**	422	20	0	0	422	20	**0.077**
	NBCDE	422	40	0	0	422	40	**0.078**	422	20	0	0	422	20	**0.080**
	NCDE	422	40	0	0	422	40	**0.060**	422	20	0	0	422	20	**0.063**
	NDE	422	40	0	0	422	40	**0.081**	422	20	0	0	422	20	**0.082**
	NE	422	40	0	0	422	40	**0.077**	422	20	0	0	422	20	**0.074**

Exp II	TEST ▶	NDE							NE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	382	56	21	0	382	7	**0.968**	382	20	0	0	382	20	**0.968**
	NBCDE	382	56	19	0	382	9	**0.968**	382	20	0	0	382	20	**0.968**
	NCDE	382	56	19	0	382	9	**0.968**	382	20	0	0	382	20	**0.968**
	NDE	382	56	19	0	382	9	**0.968**	382	20	0	0	382	20	**0.968**
	NE	382	56	8	0	382	20	**0.968**	382	20	0	0	382	20	**0.968**

Exp III	TEST ▶	NDE							NE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	432	36	18	0	432	18	**0.950**	432	20	18	0	432	2	**0.990**
	NBCDE	432	36	18	0	432	18	**0.950**	432	20	18	0	432	2	**0.990**
	NCDE	432	36	18	0	432	18	**0.950**	432	20	18	0	432	2	**0.990**
	NDE	432	36	18	0	432	18	**0.950**	432	20	18	0	432	2	**0.990**
	NE	432	36	18	0	432	18	**0.950**	432	20	18	0	432	2	**0.990**

Exp IV	TEST ▶	NDE							NE						
	TRAIN ▼	N	P	TP	FP	TN	FN	AUC	N	P	TP	FP	TN	FN	AUC
	N-ALL	417	42	4	0	417	38	**0.920**	417	20	0	0	417	18	**0.951**
	NBCDE	417	42	4	0	417	38	**0.920**	417	20	0	0	417	18	**0.951**
	NCDE	417	42	4	0	417	38	**0.920**	417	20	0	0	417	18	**0.951**
	NDE	417	42	4	0	417	38	**0.920**	417	20	0	0	417	18	**0.951**
	NE	417	42	4	0	417	38	**0.920**	417	20	0	0	417	18	**0.951**

Table 10. *LLE-SVM detailed classifier results and ROC analysis. Expression I=AU20+25; II=AU25+27; III=AU10+20+25 and IV=AU12.*

Table 11. *KPCA-SVM detailed classifier results and ROC analysis. Expression I=AU20+25; II=AU25+27; III=AU10+20+25 and IV=AU12.*

[22] D. D. Ridder and R. P. W. Duin. Locally linear embedding for classification. Technical report, Pattern Recognition Group, Department of Imaging Science and Technology, Delft University of Technology, Delft, The Netherlands, PH-2002-01, 2002.

[23] D. D. Ridder, O. Kouropteva, O. Okun, M. PietikAanien, and R. P. W. Duin. Supervised locally linear embedding. *ICANN 2003*, pages 333–341, 2003.

[24] L. K. Saul and S. T. Roweis. An introduction to locally linear embedding. http://www.cs.toronto.edu/ roweis/lle/publications.html, 2001.

[25] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4(119), 2003.

[26] B. Schlkopf, S. Mika, A. Smola, G. Rtsch, and K. Miller. Kernel pca patter reconstruction via approximate pre-images. *Proceedings of the 8th International Confrence on Artifical Neural Networks*, 1998.

[27] B. Schlkopf, A. Smola, and K. Miller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[28] B. ScholKopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[29] S. Theodoridis and K. Koutroumbas. *Pattern Recongition*. Academic press, first edition, 1999.

[30] C. J. Twining and C. J. Taylor. Kernel principal component analyis and the construction of non-linear active shape models. *Proceedings of BMVC*, 2001.

[31] M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Facial recognition using kernel methods. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.