Dimensionality Reduction of Face Images for Gender Classification

Samarasena Buchala, Neil Davey, Ray J.Frank, Tim M.Gale

Abstract—**Data in most of the real world applications are high dimensional and learning algorithms like neural networks have problems in handling high dimensional data. However, the Intrinsic Dimension is often much less than the original dimension of the data. Here, we use a fractal based method to estimate the Intrinsic Dimension and show that a nonlinear projection method called Curvilinear Component Analysis can effectively reduce the original dimension to the Intrinsic Dimension. We apply this approach for dimensionality reduction of the face images data and use neural network classifiers for Gender Classification.**

Index Terms—**Curvilinear Component Analysis, Dimensionality Reduction, Gender Classification, Intrinsic Dimension, Principal Component Analysis.**

I. Introduction

High dimensional data usually contain redundancies and may have many irrelevant variables. Classifiers like neural networks may need huge networks, with many free parameters, to cover the high dimensional data. Networks, on such datasets, even if successfuly trained, often perform badly on their test sets. This bad generalization may be due to the large number of free parameters representing irrelevant information. To learn relevant information from such datasets, a large number of datapoints would be needed, which is often impractical, and the training time needed for learning also increases to a great extent. This problem with high dimensional data is often referred in the literature as “curse of dimensionality” [1].

The intrinsic dimension which is the true dimension, of the data may be much smaller than the original data dimension. The problem with high dimensional data can be circumvented by reducing the data to its Intrinsic Dimension.

Principal Component Analysis (PCA) [2], [3] and Independent Component Analysis (ICA) [4] are linear projection methods and are the most popular statistical methods for dimensionality reduction. Being linear methods, they work perfectly well on the linear data. However real world data are often nonlinear, in which case linear techniques are not appropriate. Here, we will use a powerful recent nonlinear projection method, Curvilinear Component Analysis (CCA), for dimensionality reduction and show that it is possible to reduce the data to its Intrinsic Dimension. We apply this technique on face images data and use two classifiers, Multi Layer Perceptron (MLP), and Support Vector Machine (SVM) with a linear kernel, for Gender Classification.

II. Intrinsic Dimension

Intrinsic Dimension (ID) can be defined as the minimum number of free variables required to define the data without any significant information loss.

Due to correlations among the data, linear and nonlinear, a *D* dimensional data may actually lie on a *d* dimensional manifold ($D > d$) and the ID of such data is said to be *d*. For example a plane embedded in a three-dimensional space, as shown in Fig. 1(a) has an ID of 2, as two axes are linearly dependent. Fig. 1(b) shows the well known three dimensional horseshoe data distribution. However any point in the data can be defined by a linear axis and a curvilinear axis, indicating that it’s ID is 2.

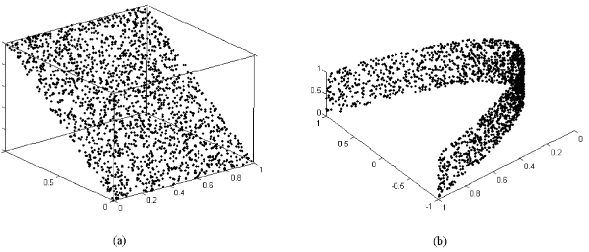

Fig. 1. (a) A two dimensional plane embedded in a three dimensional space has an ID value of 2. (b) Three dimensional horseshoe data distribution with an ID value of 2.

A. ID Estimation

Dimension reduction algorithms, reduce the data to a user defined dimension but do not inform about the number of dimensions the data should be reduced to. ID estimation can be a prior step to dimensionality reduction. There are few methods in the literature, for estimating the ID, which are mainly *fractal* based. As the name suggests a fractal dimension can be a non integer value. Here, we use a fractal based dimension called *Correlation Dimension* [5]. This method assumes that the data is spatially correlated and a measure of this property is called the *Correlation Integral* and can be calculated by (1).

$$C(l) = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} K \quad where \quad \begin{array}{l} K = 1, \, if \;\, d_{i,j}^{X} \le l \\ K = 0, \, if \;\, d_{i,j}^{X} > l \end{array} \qquad (1)$$

S. Buchala is working for a PhD degree in the Department of Computer Science at the University of Hertfordshire, College Lane, Hatfield , AL10 9AB, UK (e-mail: S.Buchala@herts.ac.uk).

N. Davey is a lecturer in the Department of Computer Science at the University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK (e-mail: N.Davey@herts.ac.uk).

R.J. Frank is a lecturer in the Department of Computer Science at the University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK (e-mail: R.J.Frank@herts.ac.uk).

T. Gale is a visiting Research Fellow at the University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK and also works in the department of Psychiatry at the QEII Hospital, Welwyn Garden City, AL7 4HQ, UK (e-mail: T.Gale@herts.ac.uk).

Where N is the number of data points, l is the length variable and $d_{i,j}^{X}$ is the Euclidean distance between points X_i and X_j in dataset X. The idea is that in a d dimensional dataset, the total number of pair wise points closer to each other than length l is proportional to l raised to d. From this assumption the Correlation Dimension d can be calculated from (2).

$$d = \lim_{l \to 0} \frac{\log(C(l))}{\log(l)} \tag{2}$$

The above equation can be approximated by calculating the slope of the graph plotted of the logarithmic values of the Correlation Integral and length l.

Fig. 2(a) shows the Correlation Dimension plot of a 2000 data point horseshoe distribution. The correlation Dimension shown in Fig. 2(b) is the slope of the linear part of the curve shown in Fig. 2(a), and is calculated at 1.8768.

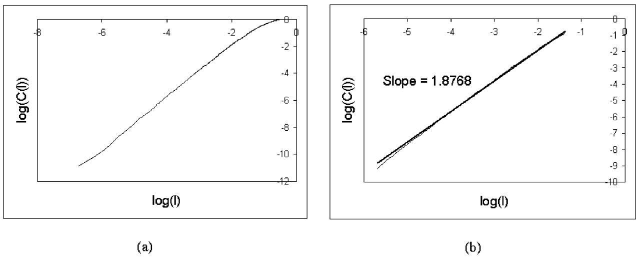

Fig. 2. (a) Correlation Dimension plot of the horseshoe data. (b) The Correlation Dimension is calculated as the slope of the linear part of the curve.

Accurate ID estimation for most real world data, including our face images data, is difficult because of the availability of only relatively few data points and noise in the data. However a rough estimation of ID can be done by the above method. Subsequently a few dimensions near to the estimated ID can be tried and the dimension which gives the best result can be considered as the true dimension of the data.

III. Dimensionality Reduction

Many techniques for dimensionality reduction have been proposed in the literature. However, Principal Component Analysis (PCA) and recently Independent Component Analysis (ICA) are the ones mostly used. PCA, especially, is a well known technique in the field of Face Recognition [6], [7]. These are linear techniques and look for linear dependencies in the data. They work perfectly on the data shown in Fig. 1(a), but fail on the data shown in Fig. 1(b). Nonlinear methods such as Nonlinear Multidimensional Scaling [8] and Sammon's Nonlinear Mapping [9] have the ability to reduce the dimensionality of nonlinear data. However, these methods suffer from huge computational costs and the inability to unfold strongly nonlinear data [10]. We use a recent algorithm called Curvilinear Component Analysis (CCA) proposed by Demartines and Herault [10], which overcomes some of the shortcomings of the other mentioned methods and has the ability to reduce the dimensionality of strongly nonlinear data.

A. Curvilinear Component Analysis

The structure of the CCA network consists of two layers, the first one of which performs vector quantization on the dataset and the second layer called the projection layer performs a topographic mapping of the structure obtained by the vector quantization layer. The projection layer is a free space, which takes the shape of the submanifold of the data.

While dimensionality reduction methods reduce the dimension of the data, vector quantization methods reduce the number of data points. The main purpose of vector quantization in CCA is to reduce the computational cost. As our face images dataset is relatively small (400 faces), we do not perform vector quantization and hence we discuss, here, only the projection part of the CCA.

The idea of CCA is to preserve distances in the input and output spaces; all the possible distances between points in the input space should match the respective distances in the output space. However, preservation of larger distances many not be possible in the case of nonlinear data, as a global unfolding of the manifold is required to reduce the dimension. In this case, it is important that at least local (smaller) distances should be preserved. For this, CCA uses a neighbourhood function which ensures the condition of distance matching is satisfied for smaller distances while it is relaxed for larger distances. Preservation of smaller distances (local mapping), may then lead to the stretching of larger distances (global unfolding).

The projection layer of CCA minimizes an error function which is given as

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(d_{i,j}^{X} - d_{i,j}^{Y} \right)^2 F_\lambda\left(d_{i,j}^{Y} \right) \quad \forall\, j \neq i \tag{3}$$

Where $d_{i,j}^{X}$ and $d_{i,j}^{Y}$ are the Euclidean distances between points i and j in the input space X and output space Y respectively. $F_\lambda\left(d_{i,j}^{Y}\right)$ is the neighbourhood function, selected such that it favours smaller distances over larger ones. Minimizing the error function with respect to the point Y_i in the output space by a normal stochastic gradient would give the following update rule.

$$\nabla Y_i = \alpha(t) \sum_{j=1}^{N} \left[2F_\lambda\left(d_{i,j}^{Y}\right) - \left(d_{i,j}^{X} - d_{i,j}^{Y}\right) F_\lambda^{'}\left(d_{i,j}^{Y}\right) \right] \cdot \left[\frac{d_{i,j}^{X} - d_{i,j}^{Y}}{d_{i,j}^{Y}} \right] \left(Y_i - Y_j \right) \quad \forall\, j \neq i \tag{4}$$

$\alpha(t)$, the learning rate, and the neighbourhood function $F_\lambda\left(d_{i,j}^{Y}\right)$ can be time varying.

The stochastic gradient update method of (4) can be conceived as selecting a point Y_i in the output space, while the remaining points are pinned. The selected point is moved (updated) according to the average influence of all the pinned points. This method of updating has the following drawbacks [10].

- The computational cost is of the order of $O(N^2)$ as all the possible $N(N-1)/2$ distances need to be calculated at each time step.
- The sum of all influences may lead to an averaging effect, which leads to a small update amount resulting in slow convergence.

For these reasons CCA uses a different update method, where the selected point is pinned while the remaining points are moved according to its influence. Then, by ignoring the derivative part of (4), the update rule of CCA can be written as:

$$\nabla Y_j = \alpha(t) F_\lambda\left(d_{i,j}^Y\right)\frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y}\left(Y_j - Y_i\right) \quad \forall\ j \neq i \tag{5}$$

The algorithm for projection of the training data can be summarized as follows.

```
Calculate the Euclidean distances
between all pairs of points in the
input space.
Initialize the points in the output
space randomly or using PCA.
Initialize epoch t=0
For each epoch t,
Begin
  Calculate $\alpha(t)$ and $\lambda$.
  For each point $Y_j$ in the output
  space,
  Begin
      $\nabla Y_j = \alpha(t) F_\lambda\left(d_{i,j}^Y\right)\frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y}\left(Y_j - Y_i\right) \quad \forall\ j \neq i$
  End
  Increment t
End
```

Mapping of a new point (test data) from the input space X to the output space Y, in CCA, involves reducing the error function of (3) and is iterative in the same sense as the actual learning process. However, the update rule is the stochastic gradient of (4) without the derivative part. The algorithm for projecting a new point can be summarized as follows.

```
Calculate the Euclidean distances
between the new test point and all
the training points.
Initialize the test point in the
output space randomly or using PCA.
Initialize epoch t=0.
For each epoch t,
Begin
  Calculate $\alpha(t)$ and $\lambda$.
  $\nabla Y_i = \alpha(t)\sum_{j=1}^{N} F_\lambda\left(d_{i,j}^Y\right)\frac{d_{i,j}^X - d_{i,j}^Y}{d_{i,j}^Y}\left(Y_j - Y_i\right)$
                                          $\forall\ j \neq i$
  Increment t
End
```

We use the first few variables obtained by the PCA projection, for initialization of the points in the output space. This initialization, rather than a random one, induces some prior information about the submanifold of the data. The learning rate and the neighbourhood width are calculated as an exponential decay.

B. Projection Evaluation

The quality of a projection can be evaluated by the "*dy-dx*" representation [10]. It is a plot of all the possible distances in the input space, *dx's*, versus their respective distances in the output space, *dy's*. For a linear projection the "*dy-dx*" plot should be linear. Fig. 3(a) shows the projection of the plane in a three dimensional space, of Fig. 1(a), in a two dimensional space. This "*dy-dx*" plot, shown in Fig. 3(b) indicates a linear projection as the *dy's* and *dx's* are proportional at all scales. However, for a nonlinear projection a complete distance match at all scales may not be possible. Fig. 4(a) shows the projection of the horseshoe data, of Fig. 1(b), in a two dimensional space. The projection is nonlinear with only small *dy's* matching *dx's*, shown in Fig. 4(b). Unfolding can be observed as (*dy* > *dx*) occurring for larger distances.

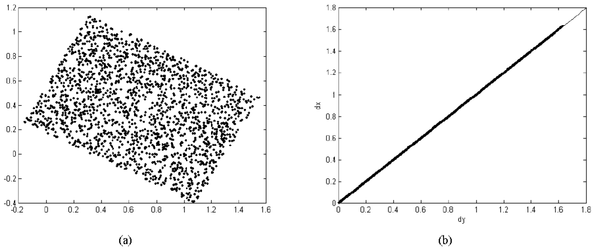

Fig. 3. (a) Projection of Fig. 1(a) from a three dimensional space to a two dimensional space by CCA (b) The "*dy-dx*" representation indicates a complete linear projection with no unfolding.

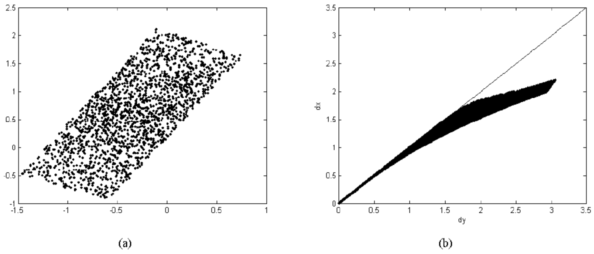

Fig. 4. (a) Projection of the horseshoe data of Fig. 1(b) from a three dimensional space to a two dimensional space by CCA (b) The "*dy-dx*" representation indicates a nonlinear projection with unfolding (*dy* > *dx*) occurring at higher scales.

IV. Gender Classification

A. Datasets

Two different datasets are used. The first one consists of 100 distinct adult, frontal face grey scale images (50 male and 50 female). The face images are from the following databases: AR [11], FERET [12], and JAFFE [13]. The dataset consists of faces of different races and age groups, taken under different lighting conditions. Some examples are shown in Fig. 5.

Fig. 5. Examples of the raw face images of *dataset1*

Taking the midpoint of the two eyes as a reference point a 60 × 90 part is extracted from each of the 128 × 128 face images. Histogram equalization is applied on the extracted images to normalize for different lighting conditions. Some of the extracted and histogram equalized faces are shown in Fig. 6. The dimensionality of this dataset is 5400. We refer to this resultant dataset as *dataset1*.

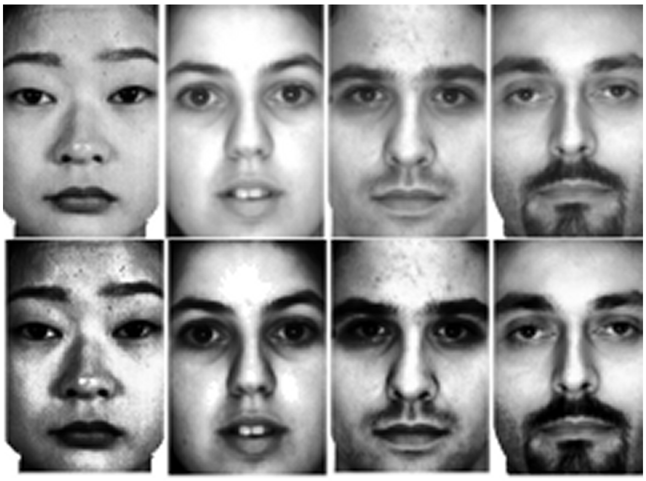

Fig. 6. The first row faces are the 60 × 90 extractions of the original 128 × 128 face images. The second row shows face images after histogram equalization.

The second dataset, a much larger dataset, was used by Sun et al. [14]. This dataset consists of 400 adult frontal face grey scale images (200 male and 200 female) each with 100 × 100 resolution. All face images were warped to the same scale, orientation and position, by geometric alignment of centres of the eyes and corners of the mouth. Histogram Equalization was then applied on the resultant images. The dimensionality of this dataset is 10000. We refer to this dataset as *dataset2*.

B. Dataset1

This dataset is randomly divided into 5 subsets, with each subset having 80 (40 male and 40 female) for training and 20 (10 male and 10 female) for testing. The test sets are not overlapped with their respective training sets and other test sets.

Intrinsic Dimensionality of this dataset is calculated using the fractal method discussed in Section II.A. As stated earlier, ID estimation of real world data is difficult. Fig. 7(a) shows the Correlation Dimension plot for *dataset1*. As the plot is not linear like the plot of the horseshoe data shown in Fig. 2, we select different intervals and measure the slope of the linear fit of that interval. The ID values from these plots of different intervals are different; the ID estimation of the plot in Fig. 7(b) is 7 while it is 11 in both Fig. 7(c) and 7(d). We select the worst case dimension 11 as the Intrinsic Dimension of the *dataset1*.

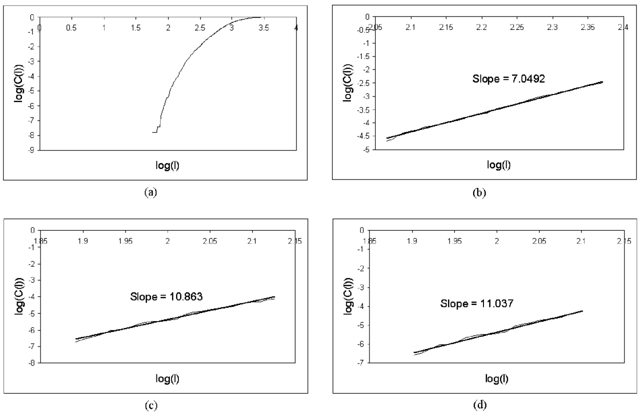

Fig. 7. (a) The Correlation Dimension plot for *dataset1*. The ID estimation varies at different intervals. The interval taken in (b) gives ID as 7, while (c) and (d) gives ID as 11.

As the ID value of 11 selected for this dataset can be considered as only a rough estimation, we tried different dimensions around this value.

We use two classifiers, a MLP and a SVM with a linear kernel (Other kernels like Radial Basis function and Polynomial function are tried, but linear kernel produced better results). The average error rates over the 5 sets on this dataset, for different CCA dimensions are shown in Table I. The error on CCA data with 6 dimensions was quite high with both MLP and SVM and the error went down as the dimension is increased. The minimum dimension with optimum result is 14. The "*dy-dx*" plots of the CCA projections, shown in Fig. 8 can explain the results of Table I. Fig. 8(a) shows CCA projection to 6 dimensions. The plot is distorted, with distance linearity occurring only at very small distances, indicating a bad projection. The projection quality improves as the dimension is increased. Fig. 8(d) and 8(e) with CCA projections to 14 and 16 dimensions respectively, has distance linearity occurring at larger distances.

TABLE I
AVERAGE ERROR RATES OVER 5 TESTSETS OF *DATASET1*, WITH DIFFERENT CCA DIMENSIONS

Method	MLP (%)	SVM (%)
CCA-6	40	43
CCA-8	31	28
CCA-10	31	24
CCA-12	28	24
CCA-14	23	17
CCA-15	26	17
CCA-16	25	20
CCA-18	26	19

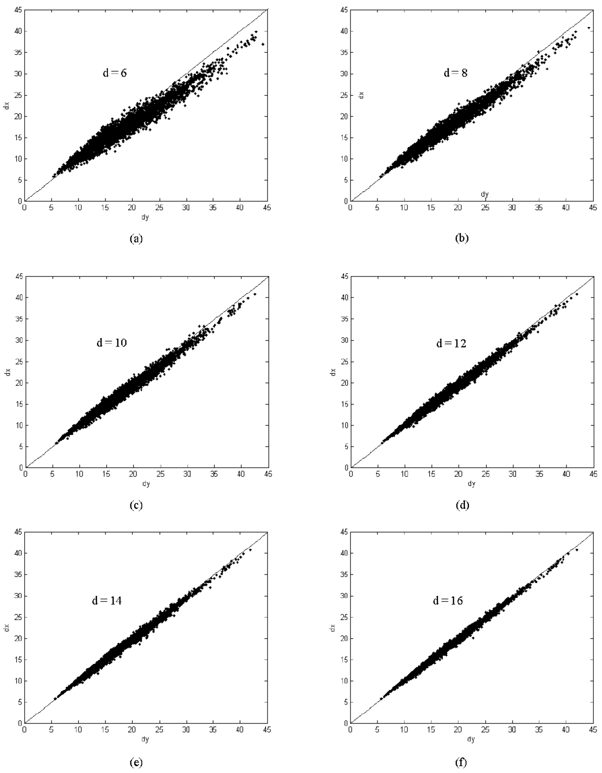

Fig. 8. The "*dy-dx*" plots of CCA projections to (a) 6 dimensions (b) 8 dimensions (c) 10 dimensions (d) 12 dimensions (e) 14 dimensions (f) 16 dimensions. Projection to 6 dimensions is more distorted with only few starting small distances being linear. The projection quality improves as the dimension is increased.

For comparision, we tried classification on data obtained by PCA reduction. For an *N* data point dataset, there will be *N-1* meaningful *Principal Components*. More details can be found in [6]. As this dataset has 80 faces there will be 79 meaningful principal components. However the first 67 components accounted for 95% of the total variance of the data. By projecting the data onto these 67 components we were able to reduce the 5400 dimensional data to a 67 dimensional data. We refer to this data as PCA-67 data. We also tried classification on the actual data, without any dimensionality reduction, and we refer to this data as RAW data. Table II shows that both dimensionality reduction approaches produced better results than the RAW data, with PCA-67 faring better than CCA-14. For comparison, another PCA reduction to 14 dimensions is obtained, by projecting the data onto the first 14 components. We refer to this as the PCA-14 data. The classification in Table II shows that PCA-14 performance is worse than that on the RAW data. The "*dy-dx*" plots of PCA-67 and PCA-14 data shown in Fig. 9 explains their performance. The PCA-14 plot, in Fig. 9(b), shows mismatch of distances at all scales. It can also be seen, from Table II, that the SVM gave a better classification than the MLP.

TABLE II
AVERAGE ERROR RATES OVER 5 TESTSETS OF *DATASET1*

Method	**MLP (%)**	**SVM (%)**
RAW	30	27
PCA-67	19	12
CCA-14	23	17
PCA-14	34	32

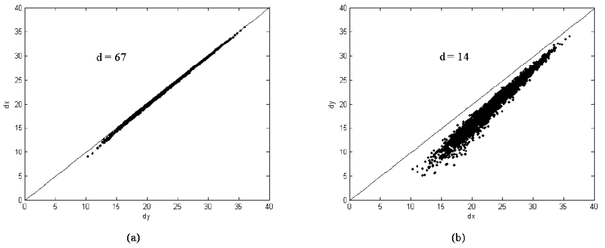

Fig. 9. The "*dy-dx*" plots of PCA projections to (a) 67 dimensions (b) 14 dimensions. The plot is distorted for 14 dimensions with nonlinearity of distances at all scales, while the plot for 67 dimensions is mostly linear.

C. *Dataset2*

This dataset is divided into 5 subsets with each subset having 320 faces (160 male and 160 female) for training, 50 faces (25 male and 25 female) for testing, and 30 faces (15 male and 15 female) as a validation set. The validation sets are used for stopping criteria for the training of the MLP.

A rough ID estimation was performed, similar to the process for *dataset1*. The ID is measured as approximately 14. Again different CCA dimensions are tried as shown in Table III. The PCA gave 273 components accounting for 95% of the total variance of the data. The projection of the data onto these 273 components resulted in a 273 dimensional data. We refer to this data as PCA-273. Table III shows the average error rates over 5 testsets. It can be seen that the average error rates for CCA above 10 dimensions is similar, however the minimum dimension with optimum result is CCA-12. The performance of the PCA-273 is similar to that of CCA-12. There is not much difference in the classification performances of the MLP and SVM.

TABLE III
AVERAGE ERROR RATES OVER 5 TESTSETS OF *DATASET2*

Method	**MLP (%)**	**SVM (%)**
PCA-273	6.55	6.25
CCA-6	13.75	11.25
CCA-8	9.5	9.25
CCA-10	8.25	8
CCA-12	6.75	7
CCA-14	6.75	7.5
CCA-16	6.5	7.5
CCA-18	8.25	7
CCA-20	8.75	7.5
CCA-22	7.75	7.5

V. DISCUSSION

PCA projection to account for 95% of the total variance of the data resulted in 67 dimensions for *dataset1*, while it resulted in 273 dimensions for *dataset2*. As the number of data points increases, the number of such PCA dimensions also increases. This, however, does not necessarily mean that the ID also increases, and our results show similar ID estimation for both datasets. CCA is able to successfully reduce the original dimension to the ID for both datasets.

If the "*dy-dx*" plot of the CCA-14 of *dataset1*, Fig. 8(e) is considered, the projection can be seen as reasonably linear, with no strong unfolding (*dy* > *dx*). The larger distances in the original dataspace are replicated with good fidelity in the output space. This indicates the projection of the data in a 14 dimensional space by CCA is not strongly nonlinear. In contrast the PCA projection in a 14 dimensional space is highly distorted as shown by Fig 9(b). This shows the inability of the PCA to deal with even slight nonlinearities. CCA favours smaller distances over larger ones. It can be seen in all the "*dy-dx*" plots of Fig.8, that smaller distances in both input and output spaces are matched. Even in a distorted plot of Fig. 8(a), there are few small distances that are matched. PCA projections, in contrast, seem to favour larger distances. The "*dy-dx*" plot of PCA projection in a 14 dimensional space, shown in Fig. 9(b), shows distortion at all scales. However, the smaller distances are more distorted than the larger distances. Even in a fairly uniform PCA projection in a 67 dimensional space, shown in Fig. 9(a), there is a slight mismatch in smaller distances. This may suggest a bad local mapping by PCA.

Based on our experiments, we can make the following conclusions:

- The ID of our face images data is much lower than their original dimension.
- Linear methods like PCA are unable to effectively reduce the nonlinear data to its ID, whereas nonlinear methods like CCA can effectively do this.
- Classification in the ID space works.

VI. Acknowledgment

We thank Zehang Sun for providing with the face images data which we referred as *dataset2* in our experiments.

References

[1] Bellman, R.E., *Adaptive control processes: A guided tour*: Princeton University Press, 1961.

[2] Karhunen, K., "*Uber lineare methoden in der wahrscheinlichkeitsrechnung.,"* Annales Academiae Scientiarum Fennicae, Series A1, Mathematica-Physica, . vol. 37, p. 3-79, 1947.

[3] Loeve, M. "*Functions aleatoire du second ordre,"*. in *Processus stochastiques et mouvement Brownien*: Gauthier-Villars, Paris, 1948.

[4] Comon, P., "*Independent component analysis -A new concept?,"* Signal Processing, . vol. 36, p. 287-314, 1994.

[5] Grassberger, P. and I. Proccacia, "*Measuring the strangeness of strange attractors.,"* Physica D, . vol. 9, p. 189-208, 1983.

[6] Sirovich, L. and M. Kirby, "*Low -dimensional procedure for the characterization of human faces.,"* Journal of the Optical Society of America A, . vol. 4, p. 519-524, 1987.

[7] Turk, M. and A. Pentland, "*EigenFaces for recognition.,"* J.Cognitive Neuroscience, . vol. 3, p. 71-86, 1991.

[8] Shepard, R.N. and J.D. Carroll. "*Parametric representation of nonlinear data structures,"*. in *International Symposium on Multivariate Analysis*: Academic Press, 1965.

[9] Sammon, J.W., "*A nonlinear mapping algorithm for data structure analysis.,"* IEEE transactions Computers, . vol. C-18, no. 5, p. 401-409, 1969.

[10] Demartines, P. and J. Herault, "*Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets.,"* IEEE Transactions on Neural Networks, . vol. 8, no. 1, p. 148-154, 1997.

[11] Martiniz, A.M. and R. Benavente, "*The AR face database.,"* Technical Report : 24, CVC. , June, 1998.

[12] Phillips, P.J., H. Wechsler, J. Huang, and P. Rauss, "*The FERET database and evaluation procedure for face recognition algorithms.,"* Image and Vision Computing, . vol. 16, no. 5, p. 295-306, 1998.

[13] Lyons, M.J., S. Akamatsu, M. Kamachi, and J. Gyoba. "*Coding facial expressions with gabor wavelets,"*. in *IEEE International Conference on Automatic Face and Gesture Recognition*. Nara, Japan, 1998.

[14] Sun, Z., X. Yuan, G. Bebis, and S. Louis, J. "*Neural-Network-based gender classification using genetic search for eigen-feature selection,"*. in *IEEE international joint conference on neural networks*, 2002.