Nonlinear dimensionality reduction in texture classification: is manifold learning better than PCA?*

Cédrick Bamba Nsimba[1][†]and Alexandre L. M. Levada[1][‡]

Department of Computer Science, Federal University of São Carlos, São Carlos, SP, Brazil
cedrick.bamba@ifsp.edu.br, alexandre@dc.ufscar.br

Abstract

This paper presents a comparative analysis of algorithms belonging to manifold learning and linear dimensionality reduction. Firstly, classical texture image descriptors, namely Gray-Level Co-occurrence Matrix features, Haralick features, Histogram of Oriented Gradients features and Local Binary Patterns are combined to characterize and discriminate textures. For patches extracted from several texture images, a concatenation of the image descriptors is performed. Using four algorithms to wit Principal Component Analysis (PCA), Locally Linear Embedding (LLE), Isometric Feature Mapping (ISOMAP) and Laplacian Eigenmaps (Lap. Eig.), dimensionality reduction is achieved. The resulting learned features are then used to train four different classifiers: k-nearest neighbors, naive Bayes, decision tree and multilayer perceptron. Finally, the non-parametric statistical hypothesis test, Wilcoxon signed-rank test, is used to figure out whether or not manifold learning algorithms perform better than PCA. Computational experiments were conducted using the Outex and Salzburg datasets and the obtained results show that among twelve comparisons that were carried out, PCA presented better results than ISOMAP, LLE and Lap. Eig. in three comparisons. The remainder nine comparisons did not presented significant differences, indicating that in the presence of huge collections of texture images (bigger databases) the combination of image feature descriptors or patches extracted directly from raw image data and manifold learning techniques is potentially able to improve texture classification.

1 Introduction

Texture analysis plays an important role in the computer vision area. It is responsible for extracting meaningful information from texture images. Texture is considered as an essential attribute, among all characteristics present in an image, which can be used as a rich source of information in many application areas, such as object recognition, remote sensing, content-based image retrieval and so on.

In the computational context, texture analysis can be defined as a set of techniques capable of processing texture information in every perspective or in certain regions of interest (ROIs) of an image. According to Materka, Strzelecki et al. [12], this set is basically composed of four major categories, namely: feature extraction, texture classification, texture segmentation and shape reconstruction from texture. Feature extraction is the first stage of texture analysis. It is responsible for translating the correlation between visual patterns contained in an image into quantitative values. The methods responsible of performing this function are known as texture descriptors. The purpose of texture classification is to associate textured samples with

---

two or more classes according to a given criteria of similarity [8, 21, 12]. Texture segmentation, unlike classification, focuses on the delimitation of regions based on texture information. In this case there is no need for prior knowledge of the characteristics of each surface meaning that unsupervised grouping techniques can be used [22, 21, 12]. Shape reconstruction through textures is directly linked to the recomposition of three-dimensional surfaces by means of texture information [13, 22, 12, 21]. This work is confined mainly to texture classification which is composed of two main processes, regarding feature extraction and classification of patterns.

Many methods of texture analysis have been developed over the years, each one exploring a novel approach to extract the images texture information. For instance, we have classical methods based on second-order statistics (such as Co-occurrence Matrices (GLCM)), Haralick, Histogram of Oriented Gradients (HoG) and Local Binary Patterns (LBP). Over many years, these four descriptors have gained a large amount of interest in many computer vision researching groups. The features extracted using the aforementioned descriptors have proven to be discriminative in classifying texture patterns.

By using those four descriptors to choose a subset of features that really represent a texture image separately, some texture information can be lost which may result in decreased recognition performance. According to this paper, in order to compensate for lost texture information and to fill the gaps left by each descriptor separately, the features extracted by those four descriptors are combined. Feature vector representing the patches are then taken through dimensionality reduction methods which give an improvement in recognition performance. On the other hand, we also propose to consider as feature vector a concatenation of 32x32 patches extracted directly from an image raw data.

Finally, we use this effective combination of the features of those four descriptors and also those ones extracted directly from an image raw data (after concatenating them and applying dimensionality reduction algorithms) with four classifiers (KNN [10], Naive Bayes [23], Decision Trees [14] and Multilayer Perceptron [5]) on Outex [19] and Salzburg [11] datasets. Finally, a comparative analysis is achieved using the Wilcoxon test [24] to figure out if nonlinear dimensionality reduction methods are better than the linear ones using the two proposed approaches.

The rest of this paper is organized as follows. We give a brief overview of PCA and manifold learning methods in Sections 2 and 3 respectively. Section 4 is dedicated to present our proposal. Computational experiments and results for this work are presented in Sections 5 and 6 respectively. Finally, we conclude the paper's work in Section 7.

2   Principal Component Analysis

High-dimensional datasets (i.e datasets with many features) may present difficulty in visualization process, may require high computational performance or may contain noise. To overcome these issues, there exist methods related with linear dimensionality reduction with the purpose of shrinking datasets by transformation and/or selection of features, while minimizing information loss. In this section, we will focus particularly on PCA [9].

It is common that some datasets follow certain distributions that are majorly embedded in a few orthogonal components, where a component is the result of a linear combination of the original features. PCA is a statistical technique that aims at transforming a n-dimensional dataset X to a m-dimensional dataset Y, where, obviously, $m \leq n$. Moreover, the dimensions of Y are captured in the direction in which the variance of samples in X is maximum and should necessarily be orthogonal components, commonly known as principal components [9]. In Figure 1, the orange and purple arrows are the principal components of the synthetic dataset K, with 1000 samples and 2 features (2 principal components). In addition, the Algorithm 1

2
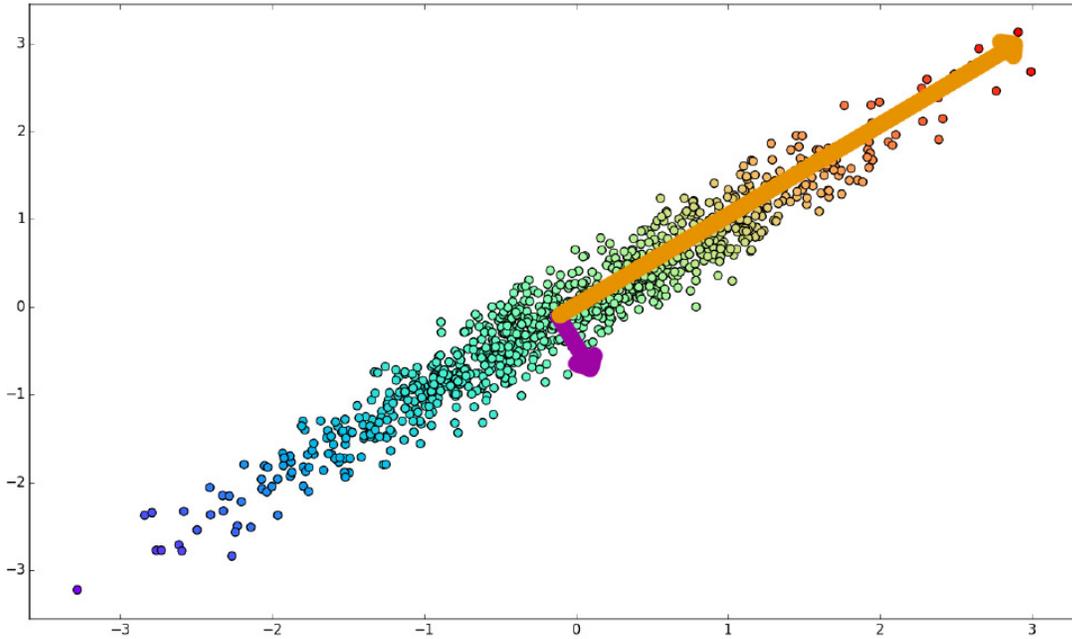
Figure 1: The principal components of dataset K (orange and purple arrows).

summarizes in a few simple steps the idea behind PCA method.

Data: a dataset D with n samples and f features and m as being the number of dimensions desired for the reduced dataset [17, 15].

Result: the matrix consisting of the principal components

1. Find $X = HD$, where H is the centering matrix defined as $H = I_n - \frac{1}{n}11^T$, where I_n and 1 are, respectively, the identity matrix of order n and the column vector of 1's.

2. Calculate the covariance matrix $\sum_X$.

3. Use singular value decomposition to find the eigenvalues $\sum = \{\sigma_i\}$ and eigenvector $V = \{v_i\}$ of $\sum_X$.

4. Sort the eigenvalues by their absolute value in descending order and select the first m ones and their respective eigenvectors.

Algorithm 1: Simple idea behind PCA Algorithm

3   Manifold Learning

Manifold learning is a popular approach to nonlinear dimensionality reduction. Algorithms for this task are based on the idea that the dimensionality of many datasets is only artificially high; though each data point consists of perhaps thousands of features, it may be described as a function of only a few underlying parameters. That is, the data points are actually samples from a low-dimensional manifold that is embedded in a high-dimensional space. Manifold learning algorithms attempt to uncover these parameters in order to find a low-dimensional representation of the data. Manifold learning algorithms include methods such as ISOMAP, LLE, Lap. Eig. that we are going to present briefly in this section.

3

3.1   ISOMAP

Firstly, ISOMAP has been suggested by Tenenbaum, de Silva and Langfor [20] and was one of the first algorithms introduced for manifold learning. It can be thought of as an extension of Multidimensional Scaling (or simply MDS). ISOMAP algorithm can be summarized in three big steps:

1. Construct the weighted graph G from the distances pairwise for all points in the input and find the graph G' by applying the nearest-neighbor algorithm on the graph G.

2. Compute the shortest path graph G'' between all pairs of nodes from graph G'. This might be done by the all-pairs Dijkstra's [6] or by the Floyd Warshall algorithm [1].

3. Use G'' to construct the p-dimensional embedding using the MDS algorithm. In other words, the MDS method can now be used to construct a representation in sub-spaces of the R^n.

3.1.1   Multidimensional Scaling (MDS)

Isomap algorithm finds points whose Euclidean distances equal the geodesic distances that were calculated in steps 1 and 2. In fact, such points exist and are unique, since the manifold is isometrically embedded. Multidimensional Scaling [7] is a classical technique that may be used to find such points. The main goal of the MDS algorithm can be formalized as follows: Let $\{x_i, y_i\}$ be the input dataset for $i = 1, 2, ..., n$ where x_i denotes the feature vector that represents the ith sample and y_i denotes the class or category to which the vector belongs, being generally an integer greater than zero. Given a Euclidean pairwise distances matrix, retrieve which are the coordinates of $x_r \in R^k, r = 1, 2, ..., n$, where k is defined by the user (plan, space 3D, etc...). A distance matrix is given by $D = \left\{d_{rs}^2\right\}, r, s = 1, 2, ..., n$ where d_{rs}^2 is a distance between x_r and x_s vectors. Its expression is given by

$$d_{rs}^2 = \|x_r - x_s\|^2.$$

Now, let B be the matrix of inner products defined as $B = \{b_{rs}\}$, where $b_{rs} = x_r^T x_s$. Based on these configurations, the MDS method seeks to address two main problems:

i) From the distance matrix D find a matrix B.

ii) From B retrieve the coordinates $x_r \in R^p$ where p is the rank of B.

To address the first problem, MDS method assume that points are mean-centered i.e. $\sum_{r=1}^{n} x_r = 0$. After some algebraic operations on D and using the initial form of matrix B, the final expression of matrix B is as $B = HAH$ such that $A = -\frac{1}{2}D$ and $H = I - \frac{1}{n}11^T$.

Finally, for retrieving the coordinates of the points, some properties of matrix B can be used. In fact B has three important properties, namely B is symmetric, the rank of B is p (this corresponds to the maximal number of linearly independent rows/columns of B that generates a base in R^p) and B is positive semidefinite. This implies that the matrix B has p non-negative eigenvalues and $n - p$ eigenvalues are null. Thus, by the spectral decomposition of B one can write $B = \vee' \wedge' \vee'^T$ where $\wedge' = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is a diagonal matrix of eigenvalues of B and $\vee'$ with $n \times p$ dimension is defined as

4

$$\begin{bmatrix} | & | & \cdots & | \\ | & | & \cdots & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & \cdots & | \\ | & | & \cdots & | \end{bmatrix}$$

However, as B can be written in these following format $B = X_{n \times p} X_{p \times n}^T$, thus, the observed $X_{n \times p}$ can be expressed as $X_{n \times p} = \vee'_{n \times p} \wedge'^{\frac{1}{2}}_{p \times p}$ where $\wedge'^{\frac{1}{2}} = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, ..., \sqrt{\lambda_p})$.
Each row of $X_{n \times p}$ will have the coordinate of a vector $x_i \in R^p$, where p is a parameter that controls the number of dimensions of the output space: if we want a $2D$ plot, $p = 2$, in case of a $3D$ plot, $p = 3$. The Algorithm 2 summarizes the process of MDS.

Data: $D = \left\{ d_{rs}^2 \right\}$
Result: points in low-dimensional (k) Euclidean space whose interpoint distances match
        the geodesic distances found in step 1 of ISOMAP
1. Do $A = -\frac{1}{2} D$
2. Do $H = I - \frac{1}{n} 11^T$
3. Compute $B = HAH$
4. Find the eigenvalues and the eigenvectors of B
5. Take the k eigenvectors associated with the largest k eigenvalues of B and set $\vee'_{nxk}$
   and $\wedge'^{\frac{1}{2}} = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, ..., \sqrt{\lambda_k})$
6. Compute $X_{nxk} = \vee'_{nxk} \wedge'^{\frac{1}{2}}_{kxk}$
Algorithm 2: MDS Algorithm

It should be noted that the ISOMAP algorithm is totally unsupervised, in the sense that it does not use any information about class distribution. It attempts to find a more compact representation of the original data only by preserving the distances.

3.2 LLE

LLE has been suggested by Saul and Roweis [16] and was introduced at about the same time as ISOMAP, but it is based on different idea than that of ISOMAP. The idea comes from visualizing a manifold as a collection of overlapping coordinate patches. If the neighborhood sizes are small and the manifold is sufficiently smooth, then these patches will be approximately linear. Moreover, the chart from the manifold to R^d (d is the dimensionality of the manifold that the input is assumed to lie on and, accordingly, the dimensionality of the output i.e. $y_i \in R^d$) will be roughly linear on these small patches. The idea is to identify these linear patches, characterize the geometry of them, and find a mapping to R^d that preserves this local geometry and is nearly linear. It is assumed that these local patches will overlap with one another so that the local reconstructions will combine into a global one [3]. LLE algorithm can be summarized in two big steps:

1. Model the manifold as a collection of linear patches and attempts to characterize the geometry of these linear patches.

2. Find a configuration in d-dimensions (the dimensionality of the parameter space) whose local geometry is characterized well by W (W_i is a characterization of the local geometry around x_i of the manifold). Unlike ISOMAP, d must be known a priori or estimated.

3.3   Laplacian Eigenmaps

The idea behind the Lap. Eig. [2] is to use spectral graph theory in order to capture local information about the manifold. Given a graph and the corresponding matrix of edge weights, W, the Laplacian graph is defined as $L = D - W$ in which D represents the diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$.

Here, a matrix W is used as a local similarity in order to measure the degree to which points are near to one another. Thus $W_{ij} = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$ if x_j is one of the k-nearest neighbors of x_i and equals 0 otherwise. The main goal is to use W in order to find output points $y_1, ..., y_n \in R^d$ that are the low-dimensional analogs of input points $x_1, ..., x_n \in R^D$ where $d < D$. Lap. Eig. algorithm can be summarized in two big steps:

1. Given $x_1, ..., x_n \in R^D$, d, k, σ as input, define a local similarity matrix W_{ij}.

2. Let U be defined as the matrix whose columns are the eigenvectors of $L\mathbf{y} = \lambda D\mathbf{y}$ associated to the non-zero eigenvalues. Here L represents Lagrange multipliers. Y is then given by the $n \times d$ matrix whose columns are these d eigenvectors and whose rows are the embedded points $(y_1, ..., y_n \in R^d)$.

4   Proposed method

This section presents a dimensionality reduction based approach for texture classification. Our method is firstly inspired by the observation of how dimensionality reduction techniques can effectively learn relevant information from the raw data, and secondly, by a wide variety of characteristics that present the feature vectors generated by each one of the classical descriptors adopted in this work. Based on this, we propose a straightforward solution which concatenates all these feature vectors (referred as first approach) and also those feature vectors constructed from patches extracted directly from the raw data (referred as second approach) and makes a feature selection through PCA and manifold learning techniques to choose the most effective features for class separability.

The idea consists in creating an approach for texture classification strongly based on the application of dimensionality reduction techniques. According to the block diagram of the proposed method (refer to Figure 2), the two stages in Figure 2(a) are part of the first approach and consist in extracting HoG, GLCM, Haralick and LBP features followed by their concatenation. We concatenated these four feature vectors aiming at constructing one big feature vector to serve as an input for dimensionality reduction using PCA and manifold learning algorithms. The stage in Figure 2(b) is part of the second approach and consists in extracting 32x32 patches from an image which will then be concatenated to form a feature vector of the underlying image. In the next stage of the diagram flow, feature selection is achieved through PCA, ISOMAP, LLE and Lap. Eig. algorithms during which thresholds are used to maintain only the most effective features among all belonging to the concatenated feature vector. From this process, we have as an output a discriminative and reduced feature vector that contains only the best elements (the most relevant informations) of the input descriptors. Then, the selected feature vector is taken through the classification process using KNN, Naive Bayes, Decision Trees and ML Perceptron. Finally, a comparative analysis is achieved using the Wilcoxon test to figure out if nonlinear dimensionality reduction methods are better than the linear ones using the two proposed approaches.

6

5 Experimental setup

5.1 Image datasets

Two texture image datasets were used in the experiments:

- **Salzburg** [11] contains a collection of 476 color texture images that have been captured around Salzburg (Austria) and, herein, depicted in Figure 4. For each texture class (from total of 10 classes used in this paper), there were 128×128 source images, of which 80% was used for training the classifier, while the other 20% was used for testing.

- **Outex** [19] has a total of 960 ($24 \times 20 \times 2$) image instances of illuminant "inca". The training set consists of 480 (24×20) images and the other 480 (24×20) images are part of the test set. The test suite for Outex used in this work is Outex_TC_00011-r. Some of the texture instances for each class are depicted in Figure 5.

5.2 Description of experiments

In Figure 2 we present the flow diagram of the proposed method, including details of the pipeline, and the methods used in the experiments. Note that our contribution is to perform a statistical comparative analysis to show which approach is better than other between PCA and manifold techniques and how these approaches can be used to reduce the dimensionality of the feature space in texture classification task.

In this paper, we perform the following two types of experiments:

- **Experiments using the full feature vector created by concatenating all descriptors,** followed by classification with feature selection;

- **Experiments using the feature vector created by extracting 32 x 32 patches,** followed by classification with feature selection.

The main topic under investigation is the effect of different methods in reducing dimensionality for concatenation of descriptors and 32 x 32 image patches-based feature vector in texture classification. In our experiments we analyze if PCA reduction produces better results than manifold algorithms for both feature extraction approaches adopted in this paper. It is for this reason that we used the Wilcoxon test to find which method has significant difference. A significance level of $\alpha = 0.05$ was used in all tests. In this case the comparison is pair-wise. In addition, all experiments were performed with a stratified $k - fold$ cross validation setting. The accuracy proved to be a suitable measure to evaluate the classification performance, since all datasets are balanced and also the sampling for the cross validation is stratified [18].

6 Results and Discussion

6.1 Full feature space experiments

The accuracies for the first set of experiments are shown in Table 1 and Table 3. They present a general perspective of the results. For each dataset and dimensionality reduction method among PCA, ISOMAP, LLE and Lap. Eig., we displayed ten accuracies, corresponding to the dimensionality reduction using 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 features (attributes).

It is evident that, in general, the methods used to reduce the dimensionality of the full feature space have significant impact on classification accuracy. In this way, the choice of

7
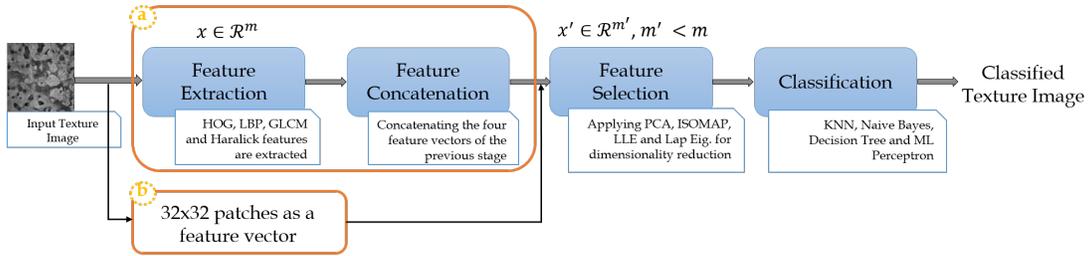
Figure 2: Block diagram of the proposed method.
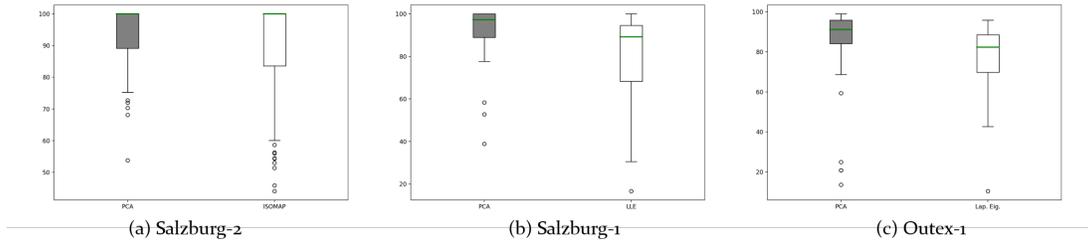
Figure 3: Accuracies of PCA, ISOMAP, LLE and Lap. Eig. for the Salzburg dataset using the 32 x 32 image patch-based feature vector (a), Salzburg dataset using the concatenated feature vector (b) and Outex dataset using the concatenated feature vector (c). Boxplots in gray correspond to significances when compared to the accuracy of other method with p-value $<$ 0.05. Here, the name Salzburg-2 (a) refers to the second approach applied to Salzburg dataset. Analogously, the names Salzburg-1 (b) and Outex-1 (c) refer to the first approach applied to Salzburg and Outex datasets, respectively.

Figure 4: A sample from Salzburgs texture database.

8
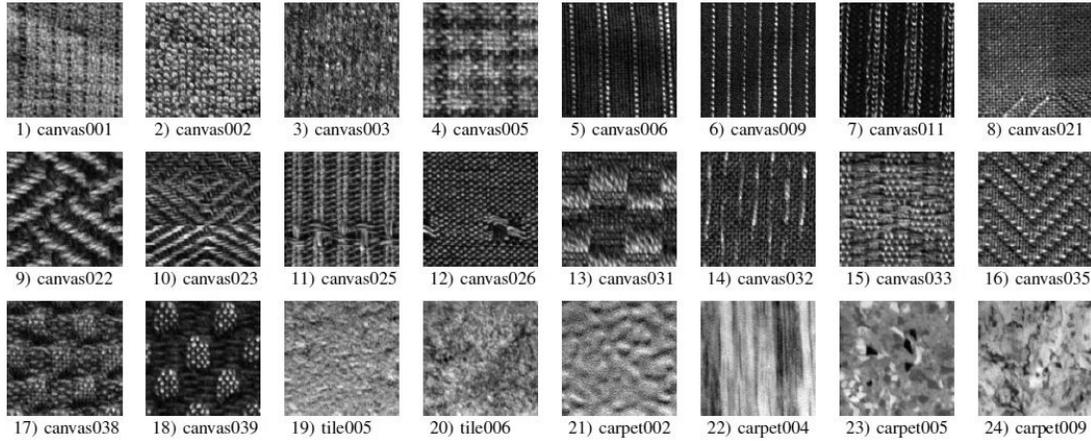
Figure 5: 128×128 blocks from the 24 texture classes of the Outex database.

d (target feature vector size) can improve or hamper the discriminative capacity of texture descriptors. While the brute force search for the best value of d seems a major drawback of many works in feature selection using dimensionality reduction techniques, in this paper, we adopted an intrinsic dimensionality estimation approach to find the intrinsic value of d. This approach computes the accumulated percentual of variance of the ith component from a list of n components (dimensions). Then the intrinsic dimensionality of the reduced space will be the first component with accumulated variance ratio greater than 95% for ISOMAP algorithm, for instance. One of the important observations from this result is that the reduction from the computed d to a higher number of dimensions often maintains the accuracies and sometimes can even slightly increase them. Figure 6 shows the observed value of d, varying from 0 to 100, based on the percentage of variance retained in the attributes. It also shows that the intrinsic dimensionalities of Salzburg and Outex datasets for ISOMAP algorithm are 10 and 24, respectively.

Since the results showed in Table 1 and Table 3 are only an overall view, we performed a deeper analysis using the following pairs of methods: PCA with ISOMAP; PCA with LLE and PCA with Lap. Eig. Also, because of the comparison must be carried out in pairs and under no assumption about the distribution of the data , the Wilcoxon's test was performed to compare the accuracies obtained by the reduction methods showed in Tables 1 and 3. A significance level of $\alpha = 0.05$ was used in all tests. To observe the results in more detail, boxplots for Salzburg and Outex datasets using PCA with LLE and PCA with Lap. Eig are shown in Figures 3 (b) and 3 (c), respectively. The boxplots shaded in gray correspond to data with significant difference when compared to the accuracy of the other method, obtaining p-value < 0.05.

We obtained significantly better results for PCA when compared with the LLE method for Salzburg (see Figure 3 (b)) and with the Lap. Eig method for Outex (see Figure 3 (c)), according to the Wilcoxon's statistical test. On a more general note, it can be observed that the results were not significantly worse, except for the value of d equals to 1 for all the methods in the two datasets.

9
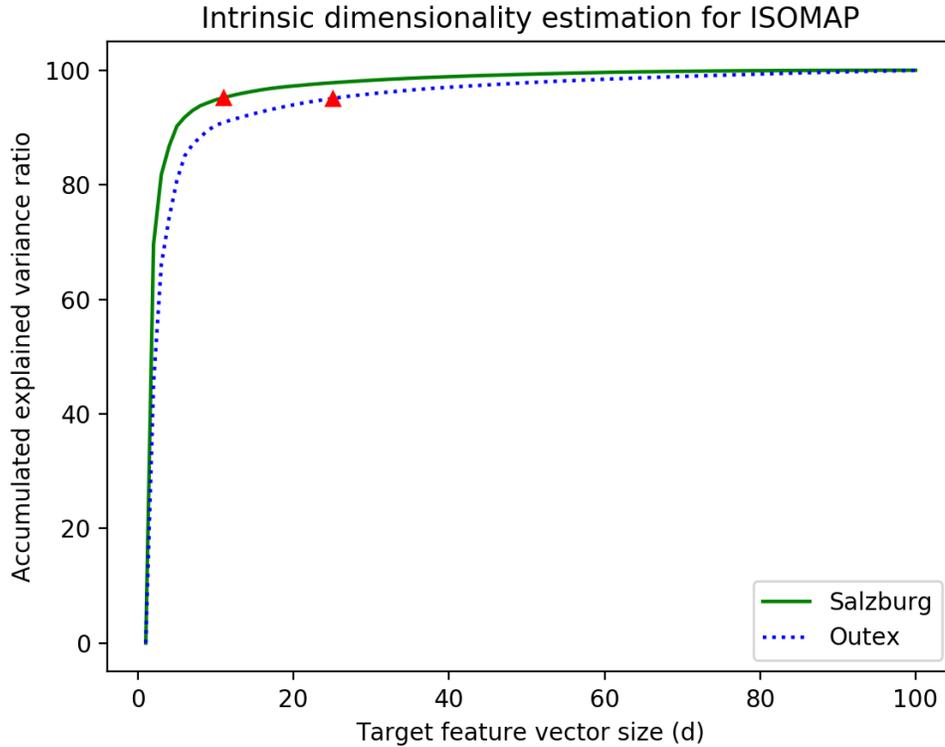
Figure 6: Intrinsic dimensionality estimation for ISOMAP for Salzburg and Outex datasets using the full feature space.

6.2   32 x 32 patch-based feature space experiments

An overview of the results for the second set of experiments is shown in Table 2 and Table 4. For each data set and dimensionality reduction method, we showed eleven accuracies, corresponding to the dimensionality reduction with the value of d equals to 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 and 100. Here, the idea is to test if the extraction of patches directly from a texture image raw data can improve or maintains the results, and also to compare the use of PCA and manifold learning methods in this way.

The only result that significantly presented difference among the pair-wise comparisons was between PCA and ISOMAP for Salzburg data set as shown in Figure 3 (a). PCA significantly presented better accuracies than ISOMAP for salzburg data set, according to the Wilcoxon's statistical test for p-value < 0.05. More importantly, on a more general perspective (i.e., for all accuracies), the results were not significantly worse.

10

		Number of Features									
		1	2	3	4	5	6	7	8	9	10
PCA	KNN	58.33	88.89	88.89	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	N.B.	80.56	86.11	97.22	100.00	97.22	100.00	97.22	100.00	100.00	100.00
	MLP	38.89	77.78	80.56	94.44	97.22	91.67	97.22	100.00	100.00	100.00
	DT	52.78	86.11	88.89	94.44	94.44	88.89	94.44	94.44	97.22	88.89
ISOMAP	KNN	86.11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	97.22	100.00
	N.B.	75.00	100.00	100.00	97.22	100.00	97.22	100.00	94.44	88.89	100.00
	MLP	44.44	77.78	91.67	91.67	94.44	94.44	100.00	97.22	97.22	100.00
	DT	75.00	97.22	88.89	94.44	97.22	94.44	83.33	83.33	91.67	83.33
LLE	KNN	66.67	69.44	94.44	100.00	97.22	97.22	100.00	100.00	100.00	97.22
	N.B.	41.67	68.75	78.13	83.33	82.29	82.29	94.79	92.71	89.58	90.63
	MLP	30.56	16.67	55.56	55.56	47.22	50.00	69.44	47.22	94.44	83.33
	DT	63.89	75.00	91.67	94.44	91.67	97.22	97.22	88.89	94.44	94.44
Lap. Eig.	KNN	72.22	83.33	100.00	100.00	97.22	100.00	100.00	94.44	100.00	100.00
	N.B.	72.22	83.33	100.00	97.22	100.00	97.22	97.22	94.44	100.00	97.22
	MLP	11.11	55.56	91.67	100.00	91.67	94.44	97.22	91.67	100.00	97.22
	DT	63.89	80.56	100.00	100.00	94.44	94.44	94.44	94.44	91.67	91.67

Table 1: Quantitative results when comparing PCA method with four techniques of manifold learning in Salzburg dataset. Here we considered the full feature vector created by concatenating all descriptors.

		Number of Features									
		5	10	15	20	25	30	35	40	45	50	100
PCA	KNN	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	N.B.	53.82	68.14	70.33	72.68	71.99	75.23	78.95	79.15	79.15	80.78	83.24
	MLP	91.11	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	DT	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
ISOMAP	KNN	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	N.B.	44.08	45.85	52.97	51.37	54.35	56.01	54.48	56.24	58.69	60.03	68.89
	MLP	88.46	98.76	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	DT	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
LLE	KNN	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	N.B.	22.91	36.01	41.08	43.50	48.04	49.51	49.58	51.80	52.75	53.89	63.33
	MLP	43.46	52.55	54.80	57.78	63.89	68.82	69.12	71.86	72.42	77.25	90.20
	DT	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Lap. Eig.	KNN	99.97	99.97	100.00	100.00	100.00	99.97	100.00	100.00	100.00	100.00	100.00
	N.B.	47.78	46.54	54.64	52.97	55.33	56.34	55.39	56.67	54.41	56.70	61.01
	MLP	66.86	76.18	80.36	86.44	91.57	92.88	95.26	98.56	98.14	99.08	99.74
	DT	99.90	99.93	99.93	99.93	99.87	99.84	99.97	99.97	99.84	99.87	99.80

Table 2: Quantitative results when comparing PCA method with four techniques of manifold learning in Salzburg dataset. Here a 32 x 32 patch of a texture image is considered as the feature vector.

7   Conclusions

This research presented an important contribution regarding the use of dimensionality reduction techniques for texture classification task. We stated that an original feature vector of D dimensions could be reduced to only d features (where $d < D$) that can better represent the data, achieving similar or higher accuracies. To produce an higher feature vector size to be used as input of the dimensionality reduction methods, we proposed two approaches: one using the full feature vector created by concatenating HoG, LBP, GLCM and Haralick descriptors; and other using the feature vector created by extracting 32 x 32 patches of each texture im-

		Number of Features									
		1	2	3	4	5	6	7	8	9	10
PCA	KNN	20.83	77.08	94.79	90.63	95.83	95.83	98.96	97.92	96.88	97.92
	N.B.	25.00	73.96	91.67	87.50	91.67	92.71	91.67	95.83	96.88	98.96
	MLP	13.54	59.38	81.25	84.38	88.54	88.54	94.79	95.83	96.88	98.96
	DT	20.83	68.75	88.54	83.33	84.38	85.42	86.46	88.54	92.71	92.71
ISOMAP	KNN	31.25	83.33	89.58	92.71	92.71	96.88	97.92	97.92	98.96	95.83
	N.B.	31.25	70.83	81.25	88.54	84.38	94.79	92.71	96.88	94.79	94.79
	MLP	17.71	60.42	82.29	87.50	88.54	91.67	92.71	92.71	94.79	95.83
	DT	22.92	69.79	85.42	83.33	78.13	90.63	88.54	84.38	87.50	86.46
LLE	KNN	40.63	67.71	79.17	88.54	89.58	92.71	98.96	97.92	100.00	96.88
	N.B.	41.67	68.75	78.13	83.33	82.29	82.29	94.79	92.71	89.58	90.63
	MLP	30.56	16.67	55.56	55.56	47.22	50.00	69.44	47.22	94.44	83.33
	DT	63.89	75.00	91.67	94.44	91.67	97.22	97.22	88.89	94.44	94.44
Lap. Eig.	KNN	46.88	84.38	88.54	91.67	88.54	95.83	92.71	88.54	88.54	91.67
	N.B.	50.00	73.96	75.00	78.13	77.08	82.29	82.29	82.29	83.33	86.46
	MLP	10.42	42.71	58.33	58.33	66.67	67.71	73.96	69.79	77.08	86.46
	DT	43.75	69.79	77.08	83.33	83.33	85.42	88.54	88.54	88.54	88.54

Table 3: Quantitative results when comparing PCA method with four techniques of manifold learning in Outex dataset. Here we considered the full feature vector created by concatenating all descriptors.

		Number of Features									
		5	10	15	20	25	30	35	40	45	50	100
PCA	KNN	56.51	62.24	61.98	60.42	58.33	54.69	53.65	52.60	48.96	45.83	37.50
	N.B.	61.98	70.05	73.95	76.30	79.43	78.91	79.69	80.47	79.95	80.47	80.47
	MLP	63.80	73.70	77.08	75.26	76.82	71.88	69.79	67.97	63.80	63.02	49.23
	DT	53.39	56.25	61.20	58.33	57.81	58.85	58.33	55.73	57.81	57.29	53.13
ISOMAP	KNN	70.31	76.82	76.56	76.30	75.78	76.56	76.56	76.56	75.78	75.50	72.14
	N.B.	55.99	68.23	72.66	73.18	73.18	73.70	73.96	72.92	73.96	74.48	73.70
	MLP	51.82	60.42	65.63	63.28	66.41	62.76	60.94	57.81	63.02	56.25	54.95
	DT	58.07	65.36	66.93	66.41	64.58	62.50	63.02	61.20	61.46	63.02	64.32
LLE	KNN	59.11	68.23	66.15	70.57	72.14	70.57	71.88	71.88	73.44	72.40	65.10
	N.B.	58.59	64.32	68.23	71.88	75.00	75.52	77.08	79.17	78.13	79.95	82.55
	MLP	37.24	37.50	42.19	42.71	44.53	45.83	48.18	46.88	46.09	47.66	53.91
	DT	53.13	53.65	50.26	58.33	59.11	55.73	59.11	56.77	53.65	51.82	52.34
Lap. Eig.	KNN	61.46	69.27	69.27	71.61	73.70	73.96	76.56	75.78	74.48	74.48	67.45
	N.B.	47.66	58.59	63.02	61.72	64.58	66.93	67.71	68.23	68.49	68.49	67.45
	MLP	59.38	67.71	74.22	71.09	71.88	75.26	74.22	74.22	74.22	73.96	67.97
	DT	58.85	63.28	62.76	62.50	61.98	65.10	66.41	64.32	63.80	62.50	60.42

Table 4: Quantitative results when comparing PCA method with four techniques of manifold learning in Outex dataset. Here a 32 x 32 patch of a texture image is considered as the feature vector.

age. In this paper we especially carried out investigations through Wilcoxon's statistical test with regard to figure out whether or not manifold learning algorithms perform better than PCA. In this case for each data set, we performed six pair-wise comparisons, namely PCA with ISOMAP; PCA with LLE and PCA with Lap. Eig such as each comparison was done for both approaches. Thus, we had twelve pair-wise comparisons in total for all experiments.

In the first place, it is important to note that the use of the two approaches (full feature vector and 32 x 32 patches-based feature vector) followed by feature selection using both PCA and manifold learning produced, on average, superior results than the individuals descriptors for texture classification task. In the second one, among all the pair-wise comparisons involved in the experiments, only three results (from a set of twelve ones) were significantly differents, according to the boxplots in Figure 3. These results clearly indicated that PCA overperformed the ISOMAP in Salzburg using the second approach, also overperformed the LLE in Salzburg using the first approach, and finally, overperformed the Lap. Eig in Outex using the first approach.

Since the results for the rest of nine comparisons did not significantly show the difference between the accuracies of PCA and each one of the manifold learning method, we believe that increasing the dimension of the feature vectors of both approaches by respectively adding more descriptors and by extracting 8 x 8 patches from texture image, may promote better performance in favour of manifold learning methods. The reason for this is that manifold learning methods are more efficient in dimensionality reduction than PCA in the presence of huge collections of texture images (bigger databases) and especially for texture classification task [4][25].

Future studies can take into account new approaches for estimating the intrinsic dimensionality value of d for each manifold learning method, can also explore different manifold learning methods and variations of the supervised classifier's algorithms in order to produce better accuracies, as well as investigate the sub-spaces generated by such methods in order to understand better their discriminative power.

Acknowledgment

References

[1] Jack Yang Alex Chumbley, Karleigh Moore and 2 others contributed. Floyd-warshall algorithm. Available at https://brilliant.org/wiki/floyd-warshall-algorithm/, last viewed november 2018.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2002.

[3] Lawrence Cayton. Algorithms for manifold learning. 2005.

[4] Jiazhong Chen, Bingpeng Ma, Hua Cao, Jie Chen, Yebin Fan, Rong Li, and Weimin Wu. Updating initial labels from spectral graph by manifold regularization for saliency detection. *Neurocomputing*, 266:79 – 90, 2017.

[5] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. *Deep Big Multilayer Perceptrons for Digit Recognition*, pages 581–598. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] Rivest Stein Cormen, Leiserson. *Introduction to Algorithms*. The MIT Press, 2009.

[7] Cox T. Cox M. *Multidimensional Scaling.* In: Handbook of Data Visualization. Springer Handbooks Comp.Statistics. Springer, Berlin, Heidelberg, 2008.

[8] Sonali Dash, K Chiranjeevi, U.R. Jena, and Akula Trinadh. Comparative study of image texture classification techniques. pages 1–6, 01 2015.

[9] Dunteman George H. *Principal components analysis.* Sage Publications, No. 69, Newbury Road, CA, USA, 1989.

[10] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In Robert Meersman, Zahir Tari, and Douglas C. Schmidt, editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 986–996, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[11] R. Kwitt and P. Meerwald. Salzburg texture image database. Available at `http://www.wavelab.at/sources/STex/`, last viewed February 2018.

[12] Andrzej Materka and Michal Strzelecki. Texture analysis methods – a review. Technical report, INSTITUTE OF ELECTRONICS, TECHNICAL UNIVERSITY OF LODZ, 1998.

[13] Majid Mirmehdi, Xianghua Xie, and Jasjit Suri. *Handbook of Texture Analysis.* Imperial College Press, London, UK, UK, 2009.

[14] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986.

[15] Sebastian Raschka. Principal component analysis (pca) step by step. 04 2014.

[16] Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.*, 4:119–155, December 2003.

[17] Lindsay I Smith. A tutorial on principal components analysis. Technical report, Cornell University, USA, February 26 2002.

[18] Yanmin Sun. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40:3358–3378, 2007.

[19] M. Pietikainen J. Viertola J. Kyllonen T. Ojala, T. Maenpaa and S. Huovinen. Outex new framework for empirical evaluation of texture analysis algorithms. volume ICPR/1, pages 701–706, 2002.

[20] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[21] MIHRAN TUCERYAN and ANIL K. JAIN. *TEXTURE ANALYSIS*, pages 207–248.

[22] Payal Dilip Wankhade. A review on aspects of texture analysis of images. *International Journal of Application or Innovation in Engineering Management (IJAIEM)*, Volume 3:229–232, 2014.

[23] Geoffrey I. Webb. *Naïve Bayes*, pages 713–714. Springer US, Boston, MA, 2010.

[24] Frank Wilcoxon. *Individual Comparisons by Ranking Methods*, pages 196–202. Springer New York, New York, NY, 1992.

[25] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3166–3173, Washington, DC, USA, 2013. IEEE Computer Society.