Human Activity Recognition by Class Label LLE

Juliana Valencia-Aguirre[1], Andrés M. Álvarez-Meza[1], Genaro Daza-Santacoloma[1], Carlos Acosta-Medina[1,2], and Germán Castellanos-Domínguez[1]

[1] Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales, Colombia
[2] Scientific Computing and Mathematical Modeling Group, Universidad Nacional de Colombia, Manizales, Colombia
{jvalenciaag,amalvarezme,gdazas,cdacostam,cgcastellanosd}@unal.edu.co

Abstract. Human motion analysis has emerged as an important area of research for different fields and applications. However, analyzing image and video sequences to perform tasks such as action recognition, becomes a challenge due to the high dimensionality of this type of data, not mentioning the restrictions in the recording conditions (lighting, angle, distances, etc). In that sense, we propose a framework for human action recognition, which involves a preprocessing stage that decreases the influence of the record conditions in the analysis. Further, our proposal is based on a new supervised feature extraction technique that includes class label information in the mapping process, to enhance both the underlying data structure unfolding and the margin of separability among classes. Proposed methodology is tested on a benchmark dataset. Attained results show how our approach obtains a suitable performance using straightforward classifiers.

Keywords: Motion recognition, video processing, dimensionality reduction.

1 Introduction

Human action recognition is a growing area of study, which arises a lot of associated applications of interest, e.g. detection of suspicious behavior for security systems, development of interactive game environments, performance evaluation of sport players, motion assessment for patients in rehabilitation programs, among others. As known, the main purpose of human action recognition is to assign a specific label to a motion [1]. Previous approaches found in the state-of-the-art report adequate performance in the recognition of different human actions, traditionally, running and walking. However, some of these works analyze data recorded under constrained conditions such as: distance from the camera to the subject must be fixed, lightning and angle can not have several changes, the background must present homogeneity along the video, among others.

Generally, there are two tightly related steps in building a motion recognition system: extracting motion features and training a classifier using these features.

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 260–267, 2012.

The most relevant works about motion recognition focuses on motion feature selection, including extracting features from 2-D tracking data [2], 3-D tracking information [3], or extracting motion information directly from images [1,4,5]. Some works use Support Vector Machines (SVM) to perform human motion recognition [1,3]. Moreover, in [6], the motion representation strategy is based on local spatial and temporal features, and the learning procedure is achieved by employing the k-means clustering algorithm. In [3], a 3-D characterization of the spatial and temporal correspondences for each frame are used to describe the motion behavior, next, a SVM classifies the activities. The major disadvantage of these works is to find suitable values for the SVM parameters.

In this work, we propose a framework for human action recognition, which involves a preprocessing stage that detects frames with no motion in the video and decreases the influence of the record conditions and noise in the analysis. Also, an Infinite Impulse Responde (IIR) filter is used in order to extract motion information from the video frames. Further, our proposal is based on a new supervised nonlinear dimensionality reduction technique that includes class label information in the mapping process, to enhance both the underlying data structure and the classification performance. In this regard, our scheme searches a low-dimensional space, in which a straightforward classifier can be used, for obtaining a suitable human action recognition performance. This work is organized as follows. Section 2 introduces the proposed methodology for human action recognition. In Section 3, the experimental results are described and discussed. Finally, in Section 4, we conclude about the attained results.

2 Motion Analysis

In order to analyze videos taken under real conditions, we proposed a methodology (see Figure 1) that is less sensitive to restrictions such as lighting control, specific distance between the subject and the camera, cloth control, angle, etc. The main purpose is to consider as few constrains as possible in the input data.

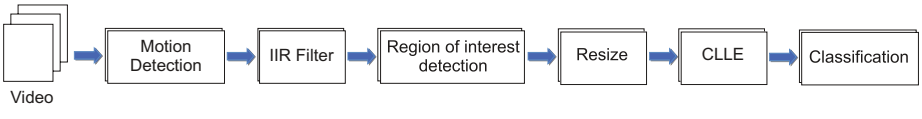


Fig. 1. Proposed methodology

2.1 Motion Detection

It is possible to find videos that do not present activity or motion 100% of the time. Thence, perform motion detection becomes an essential step. Let $\mathbf{X}_{n_v \times p}$ the input data, n_v is the number of analyzed videos and p is the input dimension. Then $\mathbf{X} = \begin{bmatrix} \mathbf{V}_1 \ \mathbf{V}_2 \cdots \mathbf{V}_{n_v} \end{bmatrix}^\top$, where $\mathbf{V}_r \in \mathbb{R}^{n_{fr} \times p}$, n_{fr} is the number of frames of r-th video, $r \in \{1, 2, \ldots, n_v\}$, $p = h * w$, being h the number of rows pixels and w the number of column pixels for each frame in gray scale.

The first step in this stage is to apply a median filter to each frame in $\mathbf{X}$. Given a video $\widehat{\mathbf{V}}_r = \left[\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_{n_{fr}}\right]^\top$, where $\mathbf{v}_i \in \mathbb{R}^{1\times p}$ are the vectorized version of the filter frames, we propose to detect motion in each $\widehat{\mathbf{V}}_r$ by $\mathbf{d}_i = |\mathbf{v}_i - \mathbf{v}_{i-1}|$, being $\mathbf{d}_i$ the derivative of $\mathbf{v}_i$, and $i = 1, 2, \ldots, n_{fr}$. This operation allows to detect significant changes between video frames, in order to eliminate parts of the video with minimum variability. As a result we obtain the derivative matrix $\mathbf{De} = \left[\mathbf{d}_1\ \mathbf{d}_2\ \cdots\ \mathbf{d}_{n_{fr}-1}\right]^\top$.

Now, a series of morphological operations are applied to $\mathbf{De}$, to facilitate the detection of constant frames, and to avoid consider noise as motion in the video: Calculate the extended-maxima transform, perform a morphological closing on the images, and finally fill holes in the binary images. Afterward this procedure, the constant frames are identifying by performing a sum of the derivative value pixel by pixel. When the number of pixels with intensity 0 in a frame are more than an specific value (e.g. 90%) then is consider with no motion. Once we know if there is motion and where is located, then we proceed to extract the frames with no movement. As a result we obtained the matrix $\mathbf{Z}_r = \left[\mathbf{T}_1\ \mathbf{T}_2\ \cdots\ \mathbf{T}_{n_s}\right]^\top$, where $\mathbf{T}_j \in \mathbb{R}^{n_{fT}\times p}$, being n_{fT} the amount of frames that have motion in constant sequence, $n_{fT} \in [n_{fT_{\min}}, n_{fT_{\max}}]$, $j \in \{1, 2, \ldots, n_s\}$, and n_s is the number of sub-sequences with motion detected in $\mathbf{V}_r$.

2.2 Recursive Filtering - IIR Filter

The IIR filter response acts like a measure that allows to identify recent motion. It works by adding a fraction of a previous frame to the current frame, then, the added fraction represents the degree of filtering. As the degree of filtering is increased, moving objects are also blurred in time. This can make moving objects appear to have a *ghost* or *comet-tail*, which we used as motion information. At each frame, motion information is represented by a feature image [4].

Considering a subsequence $\mathbf{T}_j = \left[\mathbf{t}_1\ \mathbf{t}_2\ \cdots\ \mathbf{t}_{n_{fT}}\right]^\top$ found in a video $\mathbf{V}_r$, a weighted average at time a, is computed as $\boldsymbol{\rho}_a = \tau\mathbf{t}_{a-1} + (1-\tau)\,\boldsymbol{\rho}_{a-1}$, where $\boldsymbol{\rho}_a \in \mathbb{R}^{1\times p}$, $\mathbf{t}_a \in \mathbb{R}^{1\times p}$ is the image at time a, $a = 1, 2, \ldots, n_{fT}$, and τ is a scalar between 0 and 1. The feature image $\mathbf{f}_a$ is calculated by $\mathbf{f}_a = |\rho_a - \mathbf{t}_a|$.

Then, for $\mathbf{T}_j$, the filter output is $\mathbf{F}_j = \left[\mathbf{f}_1\ \mathbf{f}_2\ \cdots\ \mathbf{f}_{n_{fT}}\right]^\top$, with $\mathbf{F}_j \in \mathbb{R}^{n_{fT}\times p}$. If $\tau = 1$ then $\mathbf{t}_a$ will be equal to the previous frame, if $\tau = 0$ then $\mathbf{t}_a$ will remain constant, and thence $\mathbf{F}_j$ will be equal to the foreground. The idea is that the feature image captures the temporal changes in the video sequence, and therefore, moving objects result in blurring.

Finally, to remove possible noise present after the IIR filter, a median filter is applied to $\mathbf{F}_j$, and a morphological filter is employed to facilitate the next stage. This morphological operation sets a pixel to 1 if five or more pixels in its 3-by-3 neighborhood are 1s; otherwise, it sets the pixel to 0 (considering a binary image). The outcome of the morphological filter is noted as $\mathbf{Fm}_j = \left[\mathbf{fm}_1\ \mathbf{fm}_2\ \cdots\ \mathbf{fm}_{n_{fT}}\right]^\top$. It is important to emphasize that the main advantage of the recursive filtering, is that is suitable for real-time applications.

2.3 Region of Interest Detection and Resize

To identify what kind of activity is generating the motion, a region of interest is found (i.e. for hand waving, the region of interest should be located around the hands and arms, which are the ones that produce changes between frames). Considering $\mathbf{Fm}_j$, the idea is to find the height and width of the subjects by calculating their position in each image, thence, we look for changes in pixel intensity to determine the first pixel in both rows and columns, in which begin the silhouette of the subject.

Regarding to the resize stage, the goal is to set a standard resolution for the video frames. Thus, the images were resized to an specific number of rows and columns, $h_r \times w_r$. This process is performed using a bicubic interpolation and antialiasing. Then, output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood, and an antialiasing is performed when shrinking the images. The resize operation is executed considering the region of interest found based on $\mathbf{Fm}_j$. Thence, the obtained output is $\tilde{\mathbf{F}}_j \in \mathbb{R}^{n_{fT} \times p_r}$, being $p_r = h_r w_r$, h_r is the resized height and w_r is the resized width.

2.4 Class Label Locally Linear Embedding – CLLE

In [7] an extension of the LLE method is presented, which we called Class Label Locally Linear Embedding - CLLE. This technique employs class labels as extra information to guide the dimensionality reduction procedure, to preserve the local geometry of the data while providing a discriminative strategy during the mapping. Given the input data matrix $\mathbf{A} \in \mathbb{R}^{N \times p_r}$, containing the N total number of preprocessed frames with constant motion from a set of videos $\mathbf{X}$ (see section 2.1), the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is computed by minimizing $\varepsilon(\mathbf{W}) = \sum_{i=1}^{N} \|\mathbf{a}_i - \sum_{j=1}^{N} w_{ij}\mathbf{a}_j\|^2$, subject to $w_{ij} = 0$ if $\mathbf{a}_j$ is not k-neighbor of $\mathbf{a}_i$, and $\sum_{j=1}^{N} w_{ij} = 1$. Then, C-LLE computes a low-dimensional space $\mathbf{Y} \in \mathbb{R}^{N \times m}$ $(m \leq p_r)$, by minimizing

$$\min_{\mathbf{Y}} \Psi(\mathbf{Y}, \beta) = \min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} w_{ij}\mathbf{y}_j \right\|^2 - \beta \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} \gamma_{ij}\mathbf{y}_j \right\|^2 \right\}, \quad (1)$$

with $\beta \in \Re^+$, and subject to $\sum_{i=1}^{N} \mathbf{y}_i = \mathbf{0}$ and $\sum_{i=1}^{N} \mathbf{y}_i \mathbf{y}_i^\top / N = \mathbf{I}_{m \times m}$. Furthermore, $\gamma_{ij} = 0$ if $i = j$, $\gamma_{ij} = \frac{1}{N-1}$, if $i = j$ if $\mathcal{P}(\mathbf{y}_i) \neq \mathcal{P}(\mathbf{y}_j)$, and $\gamma_{ij} = -\frac{1}{N-1}$ if $\mathcal{P}(\mathbf{y}_i) = \mathcal{P}(\mathbf{y}_j)$, being $\mathcal{P}(\cdot)$ a function that determines the class label of the objects, and β is a tradeoff between the preservation of the data local geometry and the representation induced by the class labels. For solving the minimization problem, it is possible to rewrite (1) as

$$\min_{\mathbf{Y}} \Psi(\mathbf{Y}, \beta) = \min_{\mathbf{Y}} \left\{ \mathrm{tr}\left(\mathbf{Y}^\top \left(\mathbf{M} - \beta \tilde{\mathbf{M}} \right) \mathbf{Y} \right) \right\} \quad \text{s.t.} \quad \begin{cases} \mathbf{1}_{1 \times N} \mathbf{Y} = \mathbf{0}_{1 \times N} \\ \frac{1}{N} \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_{m \times m} \end{cases}, \quad (2)$$

where $\mathbf{M} = \left(\mathbf{I}_{N\times N} - \mathbf{W}^\top\right)\left(\mathbf{I}_{N\times N} - \mathbf{W}\right)$, and $\tilde{\mathbf{M}} = \left(\mathbf{I}_{N\times N} - \Gamma^\top\right)\left(\mathbf{I}_{N\times N} - \Gamma\right)$, being $\Gamma \in \Re^{N\times N}$ a matrix whose elements γ_{ij} are computed as mentioned above. It is possible to calculate the m eigenvectors of $\mathbf{M} - \beta\tilde{\mathbf{M}}$ associated to its m smallest eigenvalues, discarding the eigenvector related to some eigenvalue equal to zero. Obtained eigenvectors constitute the embedding space $\mathbf{Y}$. Note that the β parameter in (1) is a tradeoff between the reconstruction error and the margin between objects belonging to different classes. If $\beta = 0$, we have the original mapping of LLE, and as β increases the separation between classes is larger. For a given β it is possible to find the output $\mathbf{Y}_\beta$ that minimizes the cost function (1). Next, the reconstruction error e_R and the margin μ can be computed as $e_R(\beta) = \mathrm{tr}\left(\mathbf{Y}_\beta^\top \mathbf{M}\mathbf{Y}_\beta\right)$, and $\mu(\beta) = \mathrm{tr}\left(\mathbf{Y}_\beta^\top \tilde{\mathbf{M}}\mathbf{Y}_\beta\right)$. Looking forward the minimization of $e_R(\beta)$ and the maximization of $\mu(\beta)$, the parametric plot $e_R(\beta)$ vs $\mu(\beta)$ can be used to study the behavior of these quantities [7].

3 Experimental Results

In order to validate the proposed methodology, we provide comparison with commonly known techniques for different processing conditions. Firstly, the preprocessing stage is carried out, but without performing dimensionality reduction. Then, based on the scheme described in Figure 1, besides proposed CLLE, we use three dimensionality reduction techniques: the traditional linear Principal Components Analysis (PCA) and as nonlinear algorithms, the Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LEM) are employed [8,9]. Testing of considered methods is carried out on a real world database, namely, the Action dataset, which is a benchmark in the state-of-the-art [6]. The database is conformed by six kinds of human actions (walking, jogging, running, boxing, handwaving and handclapping) performed by 25 subjects. All sequences were taken with a static camera with 25 *fps*. The sequences having four-seconds-length in average were down-sampled to 120×160 pixels. For testing, the videos with scale variation (zoom) and/or noticeable shadows were discarded. Besides, we randomly choose 30 videos ($n_v = 30$); 5 videos for each one of the six considered activities. Figure 2 shows some Action dataset samples.

The generalization abilities for the provided experimental conditions are tested by using a 10-folds-cross-validation scheme. In regard to the early stages of the proposed methodology, the parameters are set as follows: the median filter applied in the motion detection stage is performed using a 3-by-3 neighborhood. A morphological filter *disc* type is used. Then for the motion detection, a minimum window of 10 and maximum 20 frames is established to extract those parts of the video with no activity. The frames are removed if the 90% (or more) of the image pixels are zero content, otherwise, a frame is considered as motion. The parameter τ in the IIR filter is set to 0.5. The final frame size is set to 25×30. As a result, 141 videos (sub-video sequences found in the motion detection stage) are employed for training and testing. We obtained an space $\tilde{\mathbf{F}}$ with $n_{fT} = 141$, $p = 750$ and $C = 6$, considering the resizing procedure. The specific amount of

videos for each activity is: Walking - 25, Jogging - 21, Running - 20, Boxing - 25, Hanwaving - 25, Handclapping - 25.

The number of nearest neighbors for LLE, LEM and CLLE is chosen using a proposed approach in [10], which computes an specific number of neighbors for each input object. The dimension of the embedding space is fixed looking for a 95% of expected local variability, leading in an output dimension of $m = 4$. Three classifiers are tested: linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), and k-nearest neighbors classifier (KNNC). The number of neighbors for this classifier is optimized with respect to the leave-one-out error of the training set. Finally, the procedure to analyze new samples after training the system is shown in Figure 3.

Fig. 2. Action dataset samples

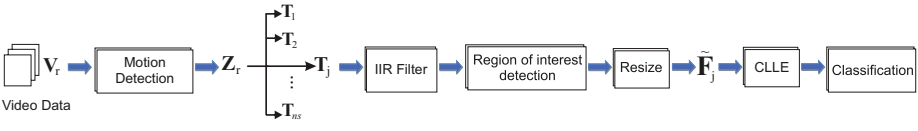


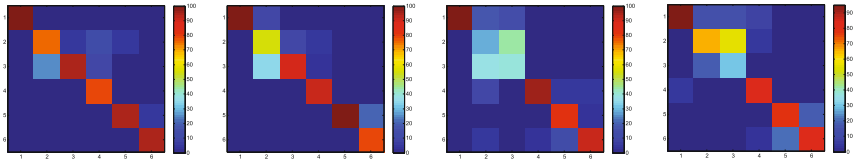
Fig. 3. Scheme used for new samples

Table 1. Classification Accuracy and Confidence Interval

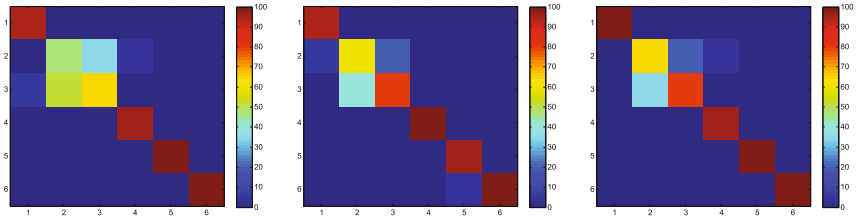
	Classification Accuracy ± Standard Deviation				
	Without DR	PCA	LLE	LEM	CLLE
LDC	48.7473 ± 11.3255	50.3883 ± 7.7237	52.5238 ± 13.7744	33.7573 ± 16.0388	85.8832 ± 5.3789
	$CI = [40.64, 56.84]$	$CI = [44.86, 55.91]$	$CI = [42.67, 62.37]$	$CI = [22.28, 45.22]$	$CI = [82.03, 89.730]$
QDC	18.4725 ± 3.7750	63.5788 ± 10.4543	60.1575 ± 14.3836	50.9240 ± 10.5782	90.1360 ± 5.4782
	$CI = [15.77, 21.17]$	$CI = [56.10, 71.05]$	$CI = [49.86, 70.44]$	$CI = [43.35, 58.49]$	$CI = [86.21, 94.05]$
KNN	90.8022 ± 9.5041	87.4615 ± 8.2779	75.7106 ± 7.4377	73.4840 ± 9.9066	91.8246 ± 8.0512
	$CI = [84.00, 97.60]$	$CI = [81.53, 93.38]$	$CI = [70.45, 80.96]$	$CI = [66.39, 80.57]$	$CI = [86.06, 97.58]$

Regarding to the PCA results (see Table 1, Figure 4(b)), this simplest reduction dimension approach exhibits a poor performance for simple classifiers (LDC, QDC), showing that linear transformations are not suitable to unfold the underlying data structure. Indeed, PCA decreases the classification accuracy for the KNN classifier in comparison to the case when dimensionality reduction is not carried out. According to Figure 4(b), the greatest difficulties become when identifying Jogging, Running, Handwaving and Handclapping.

In case of the nonlinear methods (LLE and LEM), the achieve classification accuracy decreases. In particular, these techniques have drawbacks to identify properly between Walking, Jogging and Running, and between Handwaving and



(a) Without DR-KNN (b) PCA - KNN (c) LLE -KNN (d) LEM - KNN



(e) CLLE - LDC (f) CLLE - QDC (g) CLLE - KNN

Fig. 4. Confusion matrices best classification results. Order of the actions: Walk, Jog, Run, Box, H. Waving, H. Clapping

Handclapping (see Figure 4(c) and 4(d)). LLE and LEM have some limitations when data is coming from different manifolds. In contrast, discussed CLLE approach preserves the local geometry of the data, and provides a discriminative strategy during the embedding procedure. Thus, CLLE improves the classification results in comparison to conventional LLE or other techniques like LEM, as can be corroborated in Table 1 and Figures 4(e), 4(f) and 4(g). According to the attained results, CLLE allows to use a classifier with very simple decision boundary, and therefore, leading to high classification performance. Improved performance may be explained, since the technique can unfold the non-linear input data structures. Hence, obtained mappings produce simpler low dimensional manifolds, which preserves the high-dimensional data topology while the class label information is considered, ensuring the class separability. In this sense, one may infer that CLLE technique does not require complex classifiers.

It must be quoted that due to the characteristics of the KNN classifier, it exhibits a good performance even without the dimensionality reduction procedure, and for each technique tested, this classifier always obtained the best results. So, the main contribution of the proposed methodology is that CLLE improves the performance of the classification stage even for simpler classifiers such as LDC, also decreasing the standard deviation of the average accuracy.

4 Conclusion

In this paper, a new human action recognition methodology is proposed. This methodology involves a preprocessing stage that is robust to noise and perturbations, and that allows to extract frames with no motion in the input data.

Moreover, the motion information is extracted directly from the frames by an IIR filter. On the other hand, we used a supervised form of LLE, called CLLE, proposed in [7], for the dimensionality reduction process. However, we tested with other unsupervised feature extraction techniques (PCA, LEM and LLE), and we also consider no dimensionality reduction. According to the results, CLLE allows to use a classifier with very simple decision boundary and obtain high classification performance. The proposed methodology is robust against noise conditions and/or unexpected changes of the given environment, and it improves the performance of the classification stage even for simpler classifiers such as LDC.

Acknowledgments. Research carried out under grants provided by a PhD. scholarship and the project 20201006570 funded by Universidad Nacional de Colombia, and project 20201006594 funded by Universidad Nacional de Colombia and Universidad de Caldas.

References

1. Cao, D., Masoud, O.T., Boley, D., Papanikolopoulos, N.: Human motion recognition using support vector machines. Comput. Vis. Image Underst. 113, 1064–1075 (2009)
2. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Proceedings of Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 726–733 (2003)
3. Mori, T., Shimosaka, M., Sato, T.: Svm-based human action recognition and its remarkable motion features discovery algorithm. In: ISER 2004. Springer Tracts in Advanced Robotics, vol. 21, pp. 15–25. Springer (2004)
4. Masoud, O., Papanikolopoulos, N.: A method for human action recognition. Image and Vision Computing 21, 729–743 (2003)
5. Meng, H., Pears, N., Freeman, M., Bailey, C.: Motion history histograms for human action recognition. Embedded Computer Vision 139, 139–162 (2009)
6. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR, pp. 32–36 (2004)
7. Daza-Santacoloma, G., Castellanos-Dominguez, G., Principe, J.C.: Locally linear embedding based on correntropy measure for visualization and classification. Neurocomputing 80(0), 19–30 (2012)
8. Saul, L.K., Roweis, S.T.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Machine Learning Research 4, 119–155 (2003)
9. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
10. Álvarez Meza, A., Valencia-Aguirre, J., Daza-Santacoloma, G., Castellanos-Domínguez, G.: Global and local choice of the number of nearest neighbors in locally linear embedding. Patter Recognition Letters 32(16), 2171–2177 (2011)