Peer reviewed version

Link to published version (if available):
10.1109/MMSP.2016.7813387

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

Laplacian One Class Extreme Learning Machines for Human Action Recognition

Vasileios Mygdalis[†], Alexandros Iosifidis[†*] Anastasios Tefas[†] and Ioannis Pitas[†*]
[†] Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
[*]Department of Signal Processing, Tampere University of Technology, Tampere, Finland
[*]Department of Electrical and Electronic Engineering, University of Bristol, UK

Abstract—A novel OCC method for human action recognition namely the Laplacian One Class Extreme Learning Machines is presented. The proposed method exploits local geometric data information within the OC-ELM optimization process. It is shown that emphasizing on preserving the local geometry of the data leads to a regularized solution, which models the target class more efficiently than the standard OC-ELM algorithm. The proposed method is extended to operate in feature spaces determined by the network hidden layer outputs, as well as in ELM spaces of arbitrary dimensions. Its superior performance against other OCC options is consistent among five publicly available human action recognition datasets.

I. INTRODUCTION

Human action recognition is a widely studied classification problem, due to its importance in media industry applications, such as semantic video annotation, human-computer interaction, movie (post-)production. As a classification problem, human action recognition presents challenges related to large within-class variance, due to the fact that the one action can be executed in a different manner by all subjects. Furthermore, the same action can be depicted by consecutive dissimilar frames, or captured from diverse viewing angles. Up-to-date, state of the art performance can be obtained by employing information extracted by multiple local video descriptors, classified with strong multi-class classification machines [1]–[6], or learned via deep neural network architectures [7], [8]. However, in specific industrial application scenarios (e.g., recognizing walking crowd scenes, hand-waving, lead-actor running), the discrimination of multiple actions with a complicated machine may not be as important as recognizing a single action. In order to determine whether the action of interest is present in the scene or not, binary classification machines can be considered as well. This scenario can be addressed as a One Class Classification (OCC) problem.

Perhaps the most widely adopted OCC method is the One-Class Support Vector Machines (OC-SVM) [9], which generates a hyperplane that separates the target class from the origin with the maximum possible margin. Another approach is the Support Vector Data Description (SVDD) [10], which encloses the target class with a the smallest possible hypersphere. Both OC-SVM and SVDD perform optimally by employing data mappings inherently obtained by using a kernel function, e.g., the Radial Basis Function (RBF) kernel. When the RBF kernel is employed, it has been found that both OC-SVM and SVDD provide equivalent solutions [10]. Additionally, OCC methods based on the Kernel Principal Component Analysis (KPCA) have also been proposed [11], [12], by calculating a proximity measure relative to the reconstruction error of a test sample. Recently, a single-hidden layer neural network-based method trained by using a variant of Extreme Learning Machines has been recently proposed [13], namely the One Class Extreme Learning Machines (OC-ELM), having comparable performance to other state of the art OCC methods.

As have been shown in the multi-class classification case, increased performance can be obtained when manifold regularization practices are employed in a classifier optimization process [14]–[16]. In the OCC case, the corresponding ideas have been employed in order to extend the OC-SVM and SVDD in the context of semi-supervised learning, by employing local geometric data relationships encoded in Nearest Neighbourhood (kNN) type graphs in the OC-SVM and SVDD optimization processes, have been proposed in [17] and [18], respectively. Since we emphasize in human action recognition applications, we consider employing the principles of manifold regularization in the context of OC-ELM, in view of the fact that Extreme Learning Machines have been found to provide superior performance against other approaches [4].

In this paper, a novel OCC method for human action recognition namely the Laplacian One Class Extreme Learning Machines is presented. The proposed method exploits local geometric data information within the OC-ELM optimization process. It is shown that emphasizing on preserving the local geometry of the data leads to a regularized solution, which models the target class more efficiently than the standard OC-ELM algorithm. The proposed method is extended to operate in feature spaces determined by the network hidden layer outputs, as well as in ELM spaces of arbitrary dimensions. The performance of the proposed method is evaluated against other OCC options in five publicly available human action recognition datasets. Experimental results confirm the superiority of the proposed method.

The remainder of the paper is structured as follows. In Section II, we briefly overview the standard OC-ELM. In Section III, we describe in detail the proposed Laplacian One Class Extreme Learning Machines classifier, where its kernel extension is described in Section IV. The conducted experiments are described in Section V. Finally, conclusions are drawn in Section VI.

II. One Class Extreme Learning Machines

Let a set of D-dimensional vectors $\boldsymbol{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$ be the training set, formed by N training samples belonging to the target class. We employ them in order in order to train a Single-hidden Layer Feed-forward Neural network, consisting of D input, L hidden and 1 output neuron, using the OC-ELM algorithm [13]. That is, the network input weights and bias values are randomly assigned, and the network output weights are analytically calculated. The training data are mapped from the input space to the ELM-space by an activation function $\Phi : \mathbb{R}^D \mapsto \mathbb{R}^L$. The network output weight vector $\boldsymbol{w} \in \mathbb{R}^L$ can be obtained by solving the following soft-margin optimization problem:

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{c}{2}\sum_{i=1}^N \xi_i^2, \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{w}^T\boldsymbol{\phi}_i = 1 - \xi_i, \quad i = 1, ..., N, \tag{2}$$

where 1 is the network target value for the training class (e.g., $t_i = 1$), $\boldsymbol{\phi}_i \in \mathbb{R}^L$ is the i-th training sample representation in the ELM space, corresponding to each training sample $\boldsymbol{x}_i$, ξ_i are the slack variables and $c > 0$ is a parameter allowing some training error in order to avoid overfitting. This optimization problem can be solved by obtaining the saddle points of the equivalent Lagrangian function:

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{c}{2}\sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \alpha_i(\boldsymbol{w}^T\boldsymbol{\phi}_i - 1 + \xi_i), \tag{3}$$

where α_i are the Lagrange multipliers corresponding to the constraints in (2). By setting the partial derivatives of the Lagrangian with respect to $\boldsymbol{w}, \boldsymbol{\xi_i}$ and λ_i equal to zero, two solutions for determining the network output weights can be obtained:

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\boldsymbol{I}_L\right)^{-1}\boldsymbol{\Phi}\mathbf{1}, \tag{4}$$

and

$$\boldsymbol{w} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{1}{c}\boldsymbol{I}_N\right)^{-1}\mathbf{1} = \boldsymbol{\Phi}\left(\boldsymbol{K} + \frac{1}{c}\boldsymbol{I}_N\right)^{-1}\mathbf{1}, \tag{5}$$

where $\boldsymbol{I}_L$ and $\boldsymbol{I}_N$ are identity matrices of sizes $L \times L$ and $N \times N$, respectively, $\mathbf{1}$ is a vector of ones corresponding to the training data labels, $\boldsymbol{\Phi} \in \mathbb{R}^{L \times N}$ is the matrix that contains the training data representations in the ELM space, and $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ is the so-called ELM kernel matrix, which expresses data similarity in the ELM space. In the case where the training data representations are calculated explicitly, both solutions can be adopted. The solution in (4) is preferred when $L < N$, or otherwise, (5) can be adopted. It has been shown that almost any non-linear piecewise continuous activation function $\Phi(\cdot)$ can be used for the calculation of the network hidden layer outputs, e.g., the sigmoid, polynomial, Radial Basis Function (RBF), RBF-χ^2, Fourier series, etc [19]–[21]. After the calculation of the network output weight $\boldsymbol{w}$, the network response for a given test sample $\boldsymbol{x}_t \in \mathbb{R}^D$ is given by:

$$o_t = \boldsymbol{w}^T\boldsymbol{\phi}_t. \tag{6}$$

and $\boldsymbol{x}_t$ is classified to the target class if it satisfies the following proximity measure:

$$(o_t - 1)^2 \leq \epsilon, \tag{7}$$

where $\epsilon \geq 0$ is a threshold that can be determined by using the network responses for the training data (i.e., a value of $\epsilon = 0.05\bar{o}$ was used in all our experiments, where $\bar{o}$ is the mean network response for the training data). When L is of infinite dimensions (e.g., RBF was employed as network activation function), we can employ the implicit representation of $\boldsymbol{w}$, i.e., $\boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\alpha}$, which can be found by setting the derivative of the Lagrangian function with respect to $\boldsymbol{w}$ equal to zero, $\frac{\vartheta\mathcal{L}}{\vartheta\boldsymbol{w}} = 0$. Thus (6) takes the following form:

$$o_t = \boldsymbol{w}^T\boldsymbol{\phi}_t = \boldsymbol{\alpha}^T\boldsymbol{k}_t, \tag{8}$$

where $\boldsymbol{k}_t = \boldsymbol{\Phi}^T\boldsymbol{\phi}_t$, is a $\mathbb{R}^N$ vector containing the similarities of the test sample $\boldsymbol{x}_t$ with the training data.

III. Laplacian One Class Extreme Learning Machines

In this Section, we describe in detail the proposed Laplacian One-Class Extreme Learning Machines (L-OC-ELM) algorithm. The data relationships can be encoded with an undirected weighted graph, such that $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \boldsymbol{A}\}$, where the vertex set $\mathcal{V} = \{\boldsymbol{x}_i\}_1^N$ can be formed either from the training data in the input space, or $\mathcal{V} = \{\boldsymbol{\phi}_i\}_1^N$ can be formed from the training data representations in the ELM space, with the latter having the advantage of describing non-linear relationships between the training data, $\mathcal{E}$ contains the connections between the graph vertices and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the graph weigh matrix. In order to employ the graph weights for manifold regularization [14], the following function needs to be minimized:

$$\frac{1}{2}\sum_{i,j}\|\boldsymbol{\phi_i} - \boldsymbol{\phi_j}\|^2 A_{ij} = \boldsymbol{\Phi}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{\Phi}^T = \boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T, \tag{9}$$

where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is the degree matrix, having its diagonal elements $D_{ii} = \sum_j^N A_{ij}, i = 1, \ldots, N$ or zeros otherwise, and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the graph Laplacian matrix.

We would like to initiate the graph weights so that they express local geometric data relationships, by employing the following function:

$$A_{ij} = \left\{ \begin{array}{ll} exp\left(-\frac{\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|_2^2}{2\sigma^2}\right), & \text{if } \boldsymbol{\phi}_j \in \mathcal{N}_i \\ 0, & \text{otherwise.} \end{array} \right\}, \tag{10}$$

where $\mathcal{N}_i$ denotes whether $\boldsymbol{\phi}_j$ belongs to the neighborhood of $\boldsymbol{\phi}_i$. In our experiments, we constructed $k-$Nearest Neighborhood graphs using $k = 5, 10, 15$ neighbors.

In order to minimize training error and respect local geometric data relationships at the same time, we propose the following optimization problem:

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T\boldsymbol{w} + \frac{c}{2}\sum_{i=1}^N \xi_i^2, \tag{11}$$

$$\text{s.t.} \quad \boldsymbol{w}^T\boldsymbol{\phi}_i = 1 - \xi_i, \quad i = 1, ..., N. \tag{12}$$

The proposed optimization problem can be solved by finding the saddle points of the Lagrangian:

$$\mathcal{L} = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T\boldsymbol{w} + \frac{c}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(\boldsymbol{w}^T\boldsymbol{\phi}_i - 1 + \xi_i), \quad (13)$$

By setting the partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{w}$, ξ and α_i, equal to zero, we obtain:

$$\frac{\vartheta\mathcal{L}}{\vartheta\boldsymbol{w}} = 0 \Rightarrow \boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T\boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\alpha}, \quad (14)$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\xi_i} = 0 \Rightarrow \boldsymbol{\xi} = \frac{1}{c}\boldsymbol{\alpha}, \quad (15)$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\alpha_i} = 0 \Rightarrow \boldsymbol{\Phi}^T\boldsymbol{w} = \boldsymbol{1} - \boldsymbol{\xi}, \quad (16)$$

where $\boldsymbol{1}$ is a vector of ones, corresponding to the training data labels. By substituting (15) in (16), and multiplying both sides with $\boldsymbol{\Phi}$, we obtain:

$$\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{w} + \frac{1}{c}\boldsymbol{\Phi}\boldsymbol{a} = \boldsymbol{\Phi}\boldsymbol{1}. \quad (17)$$

Afterwards, by replacing (14) in (17), we obtain the following solution for the network output weights:

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T\right)^{-1}\boldsymbol{\Phi}\boldsymbol{1}. \quad (18)$$

In the case where $L > N$, we might address singularity issues. Thus, we adopt a regularized solution, adding a small value to the diagonal elements of the expression to be inverted as follows:

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T + \frac{r}{c}\boldsymbol{I}_L\right)^{-1}\boldsymbol{\Phi}\boldsymbol{1}, \quad (19)$$

where $\boldsymbol{I}_L$ is an identity matrix of appropriate dimensions and the parameter $r > 0$ can be set to a small value (e.g., $r = 10^{-3}$), increasing the rank of the expression. Finally, we can decide whether a test sample $\boldsymbol{x}_i$ belongs to the target class or not by employing the same proximity measure as defined in equations (7).

IV. Kernel Laplacian One Class Extreme Learning Machines

In the previous section, we described the proposed L-OC-ELM classifier, when the explicit data representations in the ELM space $\mathbb{R}^L$ are available. However, in multiclass classification problems, ELM exploiting kernel formulations have been found to outperform ELM networks exploiting random hidden layer parameters [20], [21]. For example, in the RBF kernel case, the network output weights would be of infinite dimensions, thus needs to be implicitly expressed by exploiting the Representer Theorem, as a linear combination of the training data representations in the ELM space and a reconstruction vector i.e.:

$$\boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\beta}, \quad (20)$$

where $\boldsymbol{\beta}$ is a $\mathbb{R}^N$ vector containing the reconstruction weights of $\boldsymbol{w}$ with respect to $\boldsymbol{\Phi}$. Since $L \gg N$ in the kernel space,

we adopt a regularized version of the proposed optimization problem defined in (11), i.e., using $\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T + r\boldsymbol{I}_N$ instead of $\boldsymbol{\Phi}\boldsymbol{L}\boldsymbol{\Phi}^T$, where $r > 0$ is a parameter set to a small value and $\boldsymbol{I}_N$ is a $N \times N$ identity matrix. By replacing (20) in proposed optimization problem (11), the Lagrangian function defined in (13) takes the following form:

$$\mathcal{L} = \frac{1}{2}\boldsymbol{\beta}^T\left(\boldsymbol{K}\boldsymbol{L}\boldsymbol{K} + r\boldsymbol{K}\right)\boldsymbol{\beta} + \frac{c}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(\boldsymbol{\beta}^T\boldsymbol{k}_i - 1 + \xi_i), \quad (21)$$

where $\boldsymbol{k}_i = \boldsymbol{\Phi}^T\boldsymbol{\phi}_i$ contains the similarities of the $i-$th training sample with the rest of the training data and $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ is the ELM kernel matrix. By setting the partial derivatives of $\mathcal{L}$ equal to zero, with respect to $\boldsymbol{\beta}$, ξ_i and a_i, we obtain:

$$\frac{\vartheta\mathcal{L}}{\vartheta\boldsymbol{\beta}} = 0 \Rightarrow \left(\boldsymbol{L}\boldsymbol{K} + r\boldsymbol{I}_N\right)\boldsymbol{\beta} = \boldsymbol{\alpha}, \quad (22)$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\xi_i} = 0 \Rightarrow \boldsymbol{\xi} = \frac{1}{c}\boldsymbol{\alpha}, \quad (23)$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\alpha_i} = 0 \Rightarrow \boldsymbol{K}\boldsymbol{\beta} = \boldsymbol{1} - \boldsymbol{\xi}. \quad (24)$$

By replacing (23) in (24), and (24) in (22), we obtain the following solution for the reconstruction vector $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = \left(\boldsymbol{K} + \boldsymbol{L}\boldsymbol{K} + \frac{r}{c}\boldsymbol{I}_N\right)^{-1}\boldsymbol{1}. \quad (25)$$

In the test phase, the network output for a test vector $\boldsymbol{x}_t$, is given by:

$$o_t = \boldsymbol{\beta}^T\boldsymbol{k}_t, \quad (26)$$

where $\boldsymbol{k}_t$ contains the similarities of the test sample $\boldsymbol{x}_t$ with the training data in the ELM space. Finally, $\boldsymbol{x}_t$ is classified to the target class, using (7).

V. Experiments

In this section, we present the experiments conducted in order to evaluate the performance of the proposed L-OC-ELM classifier in Human Action Recognition problems. Along with the proposed method, we have also trained the OC-ELM [13] algorithm, as well as the OC-SVM algorithm [9], the Kernel PCA for novelty detection [11] (KPCS) and Kernel Null Space Methods for Novelty Detection [12] (KNFST). For the proposed method and OC-ELM, we examined different values of $c = 10^n, n = -6, \ldots, 6$. For OC-SVM, we have employed a $\nu-$SVM implementation, where have set ν equal to $\nu = \{0.01, 0.1, \ldots, 0.9\}$. For KPCS and KNFST, we have set the corresponding reconstruction error parameters equal to ν, keeping an energy of $p = \{0.90, 0.95, 0.98\}$. The optimal parameter settings for each method were determined with a $5-$fold cross validation procedure.

For our experiments, we have employed the the IMPART Multi-modal/Multi-view Dataset [22], the i3DPost multi-view action database [23], the Olympic Sports dataset [24], the Hollywood2 [25] and the Hollywood3D [26] publicly available datasets. Example frames of the employed datasets can be seen in Figures 1, 2, 3, 4 and 5. In the IMPART and i3DPost datasets, we have employed a 3-fold cross validation

TABLE I
AVERAGE G-MEANS RATES IN HUMAN ACTION RECOGNITION DATASETS

	IMPART	i3DPost	Olympic Sports	Hollywood2	Hollywood3D
OC-SVM [9]	61.45	74.53	60.72	**58.54**	55.90
KPCS [11]	43.97	78.30	51.98	55.99	28.98
KNFST [12]	69.61	77.47	57.71	53.87	55.28
OC-ELM [13]	67.23	83.77	62.04	55.74	56.74
L-OC-ELM	**70.52**	**86.31**	**65.01**	57.42	**58.02**

procedure, where we have split the datasets in 3 sets, mutually exclusive. In the Olympic Sports, Hollywood2 and Hollywood 3D datasets, we employed the standard train and test videos, as given by the dataset providers.

Fig. 1. Example frames from the IMPART dataset

Fig. 2. Example frames from the i3DPost dataset

Fig. 3. Example frames for the Olympic Sports dataset

In order to obtain vectorial video representations for each video segment depicting one activity, we have employed the dense trajectory-based video description [1]. This video description calculates five descriptor types, namely the Histogram of Oriented Gradients (HOG), Histogram of Optical

Fig. 4. Example frames for the Hollywood2 dataset

Fig. 5. Example frames for the Hollywood3D dataset

Flow (HOF), Motion Boundary Histogram along direction x (MBHx), Motion Boundary Histogram along direction y (MBHy) and the normalized trajectory coordinates (Traj), on the trajectories of densely-sampled video frame interest points that are tracked for a number of consecutive video frames (7 frames are used in our experiments). The five descriptors are calculated on the trajectory of each video frame interest point. We haved employed these video segment descriptions in order to obtain five video segment representations by using the Bag-of-Words model [2]. Thus, by following this process, each video segment was represented by 5 vectors, i.e. $\mathbf{x}_i^d, d = 1, \ldots, 5$. In order to fuse the information described in different video representations, we have combined the video segment representations with kernel methods, as in [1]. That is, we have employed the RBF kernel function, combining different descriptor types using a multi-channel approach [27]:

$$k(\mathcal{X}_i, \mathcal{X}_j) = exp\left(-\frac{1}{d}\sum_d \frac{\|\mathbf{x}_i^d - \mathbf{x}_j^d\|_2^2}{2\sigma_d^2}\right), \qquad (27)$$

where σ_d is a parameter scaling the Euclidean distance between $\mathbf{x}_i^d$ and $\mathbf{x}_j^d$. In our experiments, we set the value of σ_d proportional to the mean Euclidean distance between the $\mathbf{x}_i^d$, $i = 1, \ldots, N$ (i.e., $\gamma = a\sigma_d$, where $a = 0.01, 0.1, 1, 10, 100$), which is the natural scaling factor for the

Euclidean distances for each descriptor type on each dataset. After calculating the kernel matrices for the training and test samples, we employed them in each classification problem.

In order to rate the performance of each method, we have employed the $g-$mean metric, which is the geometric mean of the precision and recall, which is suitable for binary classification settings. For each class, we have determined the best g-mean metric for each trained class. Finally, for each dataset, we report the average g-mean metrics obtained for all classes in each dataset. The performance of each method is depicted in Table I.

As can be seen, the proposed method outperformed the OC-ELM in all cases. That is, the Laplacian graph settings provided additional information in the classifier, allowing to determine more precise output weights. Moreover, the proposed method outperformed other OCC options in most of the cases.

VI. CONCLUSION

A novel OCC method for human action recognition namely the Laplacian One Class Extreme Learning Machines was presented. Improved performance over other OCC options was obtained by exploiting information regarding the local geometric data relationships, encoded in graph structures. Exploiting graph settings in the OC-ELM optimization problem leads to a regularized solution, which models the target class more efficiently than the standard OC-ELM. Since Laplacian type graphs have been exploited for semi-supervised classification, the proposed method could be extended to work in semi-supervised classification settings.

REFERENCES

[1] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[2] A. Iosifidis, A. Tefas, and I. Pitas, "Discriminant bag of words based representation for human action recognition," *Pattern Recognition Letters*, vol. 49, pp. 185–192, 2014.

[3] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, pp. 1–20, 2015.

[4] A. Iosifidis, A. Tefas, and I. Pitas, "Human action recognition based on multi-view regularized Extreme Learning Machine," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 05, p. 1540020, 2015.

[5] G. Varol and A. A. Salah, "Efficient large-scale action recognition in videos using Extreme Learning Machines," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8274–8282, 2015.

[6] A. Iosifidis, A. Tefas, and I. Pitas, "Regularized Extreme Learning Machine for large-scale media content analysis," *Procedia Computer Science*, vol. 53, pp. 420–427, 2015.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with Convolutional Neural Networks," *Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[8] E. P. Ijjina and C. K. Mohan, "Hybrid deep neural network model for human action recognition," *Applied Soft Computing*, 2015.

[9] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[10] D. Tax and R. Duin, "Support Vector Data Description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[11] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[12] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-Class Classification with Extreme Learning Machine," *Mathematical Problems in Engineering*, pp. 1–12, 2014.

[14] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[15] G. Arvanitidis and A. Tefas, "Exploiting graph embedding in Support Vector Machines," *In: IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2012.

[16] A. Iosifidis, A. Tefas, and I. Pitas, "Graph Embedded Extreme Learning Machine." *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.

[17] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 8, pp. 3188–3197, 2010.

[18] P. Duong, V. Nguyen, M. Dinh, T. Le, D. Tran, and W. Ma, "Graph-based semi-supervised Support Vector Data Description for novelty detection," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2015.

[19] G.-B. Huang and L. Chen, "Convex incremental Extreme Learning Machine," *Neurocomputing*, vol. 70, no. 16, pp. 3056–3062, 2007.

[20] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[21] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel Extreme Learning Machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[22] H. Kim and A. Hilton, "Influence of colour and feature geometry on multi-modal 3d point clouds data registration," *International Conference on 3D Vision (3DV)*, vol. 1, pp. 202–209, 2014.

[23] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," *Conference for Visual Media Production (CVMP)*, pp. 159–168, 2009.

[24] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 392–405.

[25] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2929–2936, 2009.

[26] S. Hadfield and R. Bowden, "Hollywood 3d: Recognizing actions in 3d natural scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3398–3405, 2013.

[27] J. Zhang, M. Marszalek, M. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.