OPEN ACCESS

University of BRISTOL

Peer reviewed version

Link to published version (if available):
10.1109/SPLIM.2016.7528409

Link to publication record in Explore Bristol Research
PDF-document

## University of Bristol - Explore Bristol Research
### General rights

# Kernel Subclass Support Vector Description for Face and 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 Electical and Electronic Engineering, University of Bristol, UK

*Abstract*—In this paper, we present the Kernel Subclass Support Vector Data Description classifier. We focus on face recognition and human action recognition applications, where we argue that sub-classes are formed within the training class. We modify the standard SVDD optimization problem, so that it exploits subclass information in its optimization process. We extend the proposed method to work in feature spaces of arbitrary dimensionality. We evaluate the proposed method in publicly available face recognition and human action recognition datasets. Experimental results have shown that increased performance can be obtained by employing the proposed method.

## I. Introduction

Face recognition and human action recognition are widely studied classification problems in image analysis. A typical face recognition framework consists of four processing steps, i.e., face detection, feature extraction, dimensionality reduction and classification. Similarly, a human action recognition framework consists of three processing steps, i.e., video segmentation, feature extraction and classification. In the classification step, face recognition and human action recognition are commonly addressed as multi-class classification problems. However, there are cases when the use of one-class classification is also considered since application scenarios of multi-class classification methods have limitations related to model expansion and class-specific classification. Moreover, the multi-class classification model does not take class importance into consideration. For example, in movie post-production applications, recognizing the lead actor correctly all the time might be more important than recognizing actors having a peripheral role. In order to overcome the above described limitations, we consider employing the Support Vector Data Description (SVDD) [1] method.

Support Vector Data Description (SVDD) is a support vector based one-class classifier, initially proposed in [1]. The training phase of this classifier involves determining the minimum bounding hypersphere which encloses the target class. Test patterns that fall inside this hypersphere belong to the target class, or considered as outliers, otherwise. The SVDD optimization problem attempts to minimize the radius of the hypersphere with respect to its center. Finally, the hyperpshere center is expressed as a linear combination of the determination of support vectors. SVDD has been extended in order to exploit the fact that removing the non support vectors from the training set does not affect the classification model, thus a method that determines redundand training vectors has been proposed in [2]. Additionally, a method that determines an optimized gaussian kernel parameter (i.e., the sigma) in a fast manner, have been proposed in [3]. Recently, SVDD has been extended in the context of semi-supervised learning [4]. To this end, relationships between labeled and unlabeled training patterns, expressed with Nearest Neighbourhood ($k$NN) graph structures, are employed in order to learn a regularized manifold [5], [6].

SVDD is a state-of-the-art classification method, that can efficiently model the target class with a hypersphere. In order to improve the SVDD classification performance, a method that performs the whitening transform in the training data has been proposed in [7]. That is, the within-class variance is employed at the SVDD optimization process. In essence, instead of hypersphere, the solution of [7] resembles a hyperellipse that tightly encloses the target class. However, in face recognition and human action recognition problems, we assume that there are cases where the target class forms sub-classes related to, e.g., different illumination conditions, different viewing angles, different image scaling. This assumption is based on previous work in related fields [8], [9], [10], [11], [12], [13]. Thus, we argue that in order to improve the classification performance of SVDD, instead of minimizing the global variance, we should be minimizing the within-class variance with respect to subclass information.

In this paper, we focus on the supervised one-class classification case, for face recognition/verification applications. We consider the case where the available training data information originates from only one class, or an important class is present in the training set. We argue that sub-classes are formed within the training class, which are not considered by related methods. We modify the standard SVDD optimization problem, so that it exploits subclass information in its optimization process. We extend the proposed method to work in feature spaces of arbitrary dimensionality. We evaluate the proposed method in publicly available face recognition and human action recognition datasets.

The rest of the paper is structured as follows. In Section II, we briefly overview the standard SVDD. In Section III, we describe in detail the linear case of the proposed subclass SVDD. The kernel extension of the proposed method is described in Section IV. The conducted experiments are described in Section V. Finally, conclusions are drawn in Section VI.

## II. Support Vector Data Description

The SVDD method aims at generating a hypersphere, with center $\mathbf{a} \in \mathbb{R}^D$ and radius $R$, which encloses the training vectors onto a bounded, spherically shaped, area. Let the vectors $\boldsymbol{x}_i \in \mathbb{R}^D$, $i = 1, \ldots, N$ form the target class, from which we wish to generate the one-class classification model, by employing the SVDD method. The optimal hypersphere can be found by solving the following optimization problem:

$$\underset{R,\xi_i,\boldsymbol{a}}{\text{minimize:}} \quad R^2 + c \sum_{i=1}^{N} \xi_i \tag{1}$$

$$\text{subject to :} \quad \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 \leq R^2 + \xi_i, \tag{2}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, N, \tag{3}$$

where $\xi_i$, $i = 1, \ldots, N$ are the slack variables and $c > 0$ is a free parameter that allows some training error (i.e., soft margin formulation), in order to increase the generalization performance. The equivalent dual-Wolf optimization problem is given by minimizing:

$$\sum_{i=1}^{N} \gamma_i \ \kappa(\boldsymbol{x}_i \boldsymbol{x}_i) - \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \kappa(\boldsymbol{x}_i \boldsymbol{x}_j) \tag{4}$$

subject to :

$$0 \leq \gamma_i \leq c, \ \sum_{i=1}^{N} \gamma_i = 1, \tag{5}$$

where $\gamma_i$ is the Lagrange multiplier corresponding to each constraint (2). For each object $\boldsymbol{x}_i$ that satisfies (2), the corresponding Lagrange multiplier $\gamma_i$ is equal to zero. Thus, the optimal hypershere center is a linear combination of the Lagrange multipliers and the support vectors:

$$\boldsymbol{a} = \sum_{i=1}^{N} \gamma_i \boldsymbol{x}_i, \tag{6}$$

The hypersphere radius $R$ can be calculated by using any support vector $\boldsymbol{x}_k$ whose coefficient satisfies $\gamma_k > 0$ [1], as follows:

$$R^2 = \|\boldsymbol{x}_k - \boldsymbol{a}\|^2. \tag{7}$$

By expressing the center $\boldsymbol{a}$ in terms of support vectors we obtain:

$$R^2 = (\boldsymbol{x}_k \cdot \boldsymbol{x}_k) - \sum_{i=1}^{N} \gamma_i (\boldsymbol{x}_i \cdot \boldsymbol{x}_k) - \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j). \tag{8}$$

The radius (8) is expressed in dot product form. As have been suggested by [1], all dot products can be replaced with a kernel function $\kappa(\cdot, \cdot)$, which expresses data similarity in spaces different than $\mathbb{R}^D$. This allows the SVDD to find the optimal hypersphere in spaces of increased dimensionality, i.e., by employing a mapping function such as the RBF kernel function, such that $\phi(\cdot) \mapsto \mathcal{F}$.

Finally, for a given test sample $\boldsymbol{x} \in \mathbb{R}^D$, we decide that it belongs to the target class, if it satisfies the following inequality:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}) - 2 \sum_{i=1}^{N} \gamma_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \leqslant R^2. \tag{9}$$

## III. SVDD Exploiting Subclass Information

In the case where the training data form sub-classes, the standard SVDD optimization would create a loose hypersphere. Thus, we would wish to minimine the dispersion of the training data with respect to subclass information. Subclasses can be determined in the input space by applying the $k-$means algorithm. By considering the case where $k$ subclasses are formed within the target class, the within class dispersion can be expressed as follows:

$$\boldsymbol{S} = \sum_{i=1}^{N} \sum_{j=1}^{k} \frac{N_j}{N} e_i^j (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T, \tag{10}$$

where $e_i^j$ is an index denoting that the training data $\boldsymbol{x}_i$ belongs to the $j-$th subclass (i.e., $e_i^j = 1$), and $\bar{\boldsymbol{x}}_j$ is the average vector of the $j-$th subclass. The number of subclasses $k$ can either be set manually based on the properties of the problem at hand, or be automatically determined by applying $k$-fold (e.g., 5-fold) cross-validation. In order to incorporate subclass information in the SVDD optimization process, we solve the following optimization problem:

$$\underset{R,\xi_i,\boldsymbol{a}}{\text{minimize:}} \quad R^2 + c \sum_{i=1}^{N} \xi_i \tag{11}$$

$$\text{subject to :} \quad (\boldsymbol{x}_i - \boldsymbol{a})^T \boldsymbol{S}^{-1} (\boldsymbol{x}_i - \boldsymbol{a}) \leq R^2 + \xi_i, \tag{12}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, N, \tag{13}$$

where $\boldsymbol{a}$ is the hypersphere center, $R$ is the hypersphere radius, $\xi_i$ are the slack variables and $c$ is a trade-off parameter between training error and generalization performance. By employing a vector $\boldsymbol{u} = \boldsymbol{S}^{\frac{1}{2}} \boldsymbol{a}$, the optimization problem can be solved by determining the saddle points of the Lagrangian:

$$\mathcal{L} = R^2 + c \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \beta_i \xi_i - $$
$$- \sum_{i=1}^{N} \gamma_i \left( R^2 + \xi_i - \|\boldsymbol{S}^{-\frac{1}{2}} \boldsymbol{x}_i - \boldsymbol{u}\|^2 \right), \tag{14}$$

which lead to the following optimality conditions:

$$\frac{\vartheta \mathcal{L}}{\vartheta R} = 0 \Rightarrow \sum_{i=1}^{N} \gamma_i = 1, \tag{15}$$

$$\frac{\vartheta \mathcal{L}}{\vartheta \xi_i} = 0 \Rightarrow \beta_i = c - \gamma_i, \tag{16}$$

$$\frac{\vartheta \mathcal{L}}{\vartheta \boldsymbol{u}} = 0 \Rightarrow \boldsymbol{u} = \sum_{i=1}^{N} \gamma_i \boldsymbol{S}^{-\frac{1}{2}} \boldsymbol{x}_i. \tag{17}$$

Condition (16) can always be met if we demand $0 \leq \gamma_i \leq c$, thus the Lagrange multipliers $\beta_i$ can be removed. From (17), the hypersphere center $\boldsymbol{a}$ can be found as follows:

$$\boldsymbol{a} = \boldsymbol{S}^{-1} \boldsymbol{X} \boldsymbol{\gamma}, \tag{18}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^N$ is a vector containing the Lagrange multipliers. Every training pattern $\boldsymbol{x}_i$ which satisfies (12) (i.e., $\xi_i = 0$), falls inside the hypersphere and, thus, its corresponding Lagrange multiplier is equal to zero.

In any other case, $\gamma_i > 0$ and $\boldsymbol{x}_i$ is a support vector. The optimal radius can be recovered from any support vector $\boldsymbol{x}_k$ as follows:

$$R^2 = \|\boldsymbol{x}_k - \boldsymbol{a}\|^2 = \|\boldsymbol{x}_k - \boldsymbol{S}^{-1} \boldsymbol{X} \boldsymbol{\gamma}\|^2. \tag{19}$$

Having calculated the optimal center and radius, in order to make a decision whether a test pattern $\boldsymbol{x} \in \mathbb{R}^D$ falls inside the hypersphere, we calculate the following decision value:

$$f(\boldsymbol{x}) = R^2 - \|\boldsymbol{x} - \boldsymbol{a}\|^2, \tag{20}$$

where the test pattern is classified to the target class when $f(\boldsymbol{x}) \geq 0$, or considered an outlier otherwise.

By expressing the radius and the center in terms of support vectors, using the equations (18) and (19), we obtain the following solution:

$$f(\boldsymbol{x}) = \|\boldsymbol{x}_k - \boldsymbol{S}^{-1} \boldsymbol{X} \boldsymbol{\gamma}\|^2 - \|\boldsymbol{x} - \boldsymbol{S}^{-1} \boldsymbol{X} \boldsymbol{\gamma}\|^2. \tag{21}$$

Next, in order to obtain $\boldsymbol{\gamma}$, we reformulate the Lagrangian defined in (14), exploiting (15), (16) and (17), as follows:

$$\mathcal{L} = \sum_{i=1}^{N} \gamma_i \boldsymbol{x}_i \boldsymbol{S}^{-1} \boldsymbol{x}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \boldsymbol{x}_i \boldsymbol{S}^{-1} \boldsymbol{x}_j \tag{22}$$

Finally, the solution is obtained by solving the following optimization problem:

$$\underset{\gamma_i}{\text{minimize:}} \sum_{i=1}^{N} \gamma_i \boldsymbol{x}_i \boldsymbol{S}^{-1} \boldsymbol{x}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \boldsymbol{x}_i \boldsymbol{S}^{-1} \boldsymbol{x}_j \tag{23}$$

$$\text{subject to :} 0 \leq \gamma_i \leq c, \ \sum_{i=1}^{N} \gamma_i = 1. \tag{24}$$

Here, it should be noted that parameter $c$ can take any positive value. However, setting a value $c = 0$, eliminates the chance of convergence, since the constraints in (24) will never be met. Moreover, setting any value $c \geq 1$, leads to the same solution for $c = 1$, since the support vector coefficients should satisfy $\sum_{i=1}^{N} \gamma_i = 1$. Thus, the parameter $c$ should be limited to values of $(0, 1]$.

## IV. KERNEL SUBCLASS SVDD

In the previous section we have described the linear case where the subclasses where determined in the input space $\mathbb{R}^D$.

In order to express subclass information in spaces of arbitrary dimensionality, we decompose the matrix $\boldsymbol{S}$ as follows:

$$\boldsymbol{S} = \sum_{i=1}^{N} \sum_{j=1}^{k} \frac{N_k}{N} e_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)^T$$
$$= \boldsymbol{X} \left( \frac{1}{N} \sum_{j=1}^{k} N_j \boldsymbol{e}_j \boldsymbol{e}_j^T \right) \boldsymbol{X}^T = \boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^T, \tag{25}$$

where $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ is the datamatrix and $\boldsymbol{e}_j \in \mathbb{R}^N$ is a vector having elements $e_{ji} = 1$ if the training data $\boldsymbol{x}_i$ belongs to the $j-$th subclass, or zero otherwise.

By employing the RBF kernel function, the matrix $\boldsymbol{S}$ would have infinite dimensions, i.e.:

$$\boldsymbol{S} = \boldsymbol{\Phi} \boldsymbol{M} \boldsymbol{\Phi}^T, \tag{26}$$

where $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]$ is a matrix that contains the training data representations in the feature space $\mathcal{F}$. However, if $\mathbb{R}^D$ is of very high dimensionality, e.g., $D \gg N$, then the matrix $\boldsymbol{S}$ might not be invertible in such space. Moreover, this would be the case when a mapping function such as the RBF kernel function is employed. In order to avoid singularity issues, we employ a regularized version of $\boldsymbol{S}$, such that:

$$\tilde{\boldsymbol{S}} = \boldsymbol{S} + r\boldsymbol{I}, \tag{27}$$

where $r$ is a regularization parameter allowing the matrix $\boldsymbol{S}$ to be invertible, and $\boldsymbol{I}$ is an identity matrix of appropriate dimensions. By exploiting the Woodbury identity, the inverse of $\tilde{\boldsymbol{S}}$ is given by:

$$\tilde{\boldsymbol{S}}^{-1} = \frac{1}{r} \boldsymbol{I} - \frac{1}{r^2} \boldsymbol{\Phi} \left( \boldsymbol{M}^{-1} + \frac{1}{r} \boldsymbol{K} \right)^{-1} \boldsymbol{\Phi}^T, \tag{28}$$

where $\boldsymbol{K} = \boldsymbol{\Phi}^T \boldsymbol{\Phi}$ is the so-called kernel matrix. Finally, when a kernel function is employed (i.e., $\phi(\boldsymbol{x}_i)$ instead of $\boldsymbol{x}_i$) and by replacing (28) in the the Lagrangian function in (22), we obtain:

$$\mathcal{L} = \sum_{i=1}^{N} \gamma_i \left( \frac{1}{r} k_{ii} - \frac{1}{r^2} \boldsymbol{k}_i^T (\boldsymbol{M}^{-1} + \frac{1}{r} \boldsymbol{K})^{-1} \boldsymbol{k}_i \right) -$$
$$- \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i \gamma_j \left( \frac{1}{r} k_{ij} - \frac{1}{r^2} \boldsymbol{k}_i^T (\boldsymbol{M}^{-1} + \frac{1}{r} \boldsymbol{K})^{-1} \boldsymbol{k}_j \right), \tag{29}$$

where $k_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ expresses data similarity in $\mathcal{F}$ between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and $\boldsymbol{k}_i$ is the $i-$th column of the kernel matrix $\boldsymbol{K}$. By observing the optimization problem in (29), it is of the same form as the standard SVDD optimization problem (4), that can be solved using the modified kernel:

$$\tilde{\kappa}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{r} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{r^2} \boldsymbol{k}_i^T \left( \boldsymbol{M}^{-1} + \frac{1}{r} \boldsymbol{K} \right)^{-1} \boldsymbol{k}_j. \tag{30}$$

Finally, in order to decide whether a test sample $\boldsymbol{x} \in \mathbb{R}^D$ belongs to the training class, we can employ the standard SVDD solution (9), using the modified kernel found in (30).

## V. Experiments

This section presents the experiments conducted in order to evaluate the performance of the proposed Subclass SVDD method in face recognition and human action recognition. To this end, we have employed publicly available datasets, which have been widely adopted in relevant work. In all our experiments, we have applied the proposed subclass SVDD, along with the One-Class SVM [14] (OC-SVM), the standard SVDD [1] and the minimum variance SVDD [7] (MV-SVDD), which is a special case of the proposed method when we assume that no subclasses are formed (i.e., the number of subclasses is equal to 1). For all cases, we report the average obtained g-mean rate between all classes [15], which is suitable for imbalanced binary classification problems ($g_{mean} = \sqrt{\text{precision} \times \text{recall}}$).

In our first set of experiments, we have employed classic face recognition datasets, including the AR [16], ORL [17] and Yale [18] datasets. The datasets contain 2600, 400, and 2432 frontal facial images from 100, 40 and 38 subjects, respectively. For all cases, we have resized the images to $40 \times 30$ and vectorized them to produce a $D = 1200$ vector for each facial image. We have split each dataset in 5 training and test sets. We have employed 4/5 of the dataset for training and left the 1/5 for testing separately. We have repeated the procedure 5 times (each for a different test set) and report the average performance obtained for each class. Additionally, we have employed the competing algorithms in the PubFig83 + LFW Dataset [19]. We have employed the feature vectors (HOG, LBP, and Gabor wavelet features reduced to 2048 dimensions with PCA), which were extracted from 13,002 facial images representing 83 individuals from PubFig83, divided into 2/3 training (8720 faces) and 1/3 testing set (4,282 faces), as well as 12,066 images representing over 5,000 faces which were used as a distractor set from LFW. For each of the 83 individuals, we have employed the training images for this class and tested on the respective test set of this class, as well as 200 randomly selected images for the distractor set. Experimental results in Face recognition datasets are shown in Table I.

#### TABLE I
AVERAGE G-MEANS RATES IN FACE RECOGNITION DATASETS

|  | ORL | AR | YALE | PubFig83+ LFW |
|---|---|---|---|---|
| OC-SVM | 76.34 | 71.41 | 63.71 | 76.02 |
| SVDD | 77.08 | 70.39 | 63.42 | 76.55 |
| MV-SVDD | 77.08 | 71.88 | 65.13 | 77.45 |
| **Proposed** | **78.03** | **73.40** | **66.59** | **77.94** |

For human action recognition, we have employed the i3DPost multi-view action database [20], the IMPART Multimodal/Multi-view Dataset [21], as well as the Hollywood2 [22] and Hollywood3D [23] publicly available datasets. The i3DPost dataset contains 512 segmented high-resolution ($1080 \times 1920$ pixel) videos depicting eight human actors performing eight activities. The IMPART dataset consists of a multi-camera outdoor setup, which consists of 14 fixed cameras placed around each subject, where each subject is performing 12 actions. The Hollywood2 dataset consists of 810 training and 884 test video segments, of 12 activities. Finally, the Hollywood3D dataset consists of 359 train and 307 test stereoscopic video segments depicting 14 actions. In our experiments, we have employed only the right video channel.

In order to obtain vectorial video representations for each video segment depicting one activity, we have employed the dense trajectory-based video description [24]. This video description calculates five descriptor types, namely the Histogram of Oriented Gradients, Histogram of Optical Flow, Motion Boundary Histogram along direction $x$, Motion Boundary Histogram along direction $y$ and the normalized trajectory coordinates, 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). We haved employed these video segment descriptions in order to obtain five video segment representations by using the Bag-of-Words model [10], and combined them with kernel methods using a late fusion approach [25].

In the i3DPost and IMPART datasets, we have employed a 3-fold cross validation procedure, where we have split the datasets in 3 sets, mutually exclusive. Each set included videos depicting all activities. We have employed the videos depicting each distinct activity from two sets in order to train the classifiers, and tested on the remaining one. For each activity, we have obtained g-mean metric. This procedure was repeated for all activities, and repeated 3 times for each fold. In the Hollywood2 and Hollywood 3D datasets, we employed the standard train and test videos, provided by the authors of [22], [23]. The average g-mean metrics obtained for all activities between the folds is depicted in Table II. As can

#### TABLE II
AVERAGE G-MEANS RATES IN HUMAN ACTION RECOGNITION DATASETS

|  | IMPART | i3DPost | Hollywood2 | Hollywood3D |
|---|---|---|---|---|
| OC-SVM | 61.45 | 74.53 | 58.54 | 55.90 |
| SVDD | 61.55 | 74.19 | 58.29 | 56.15 |
| MV-SVDD | 62.70 | 74.43 | 59.84 | 58.61 |
| **Proposed** | **63.10** | **74.86** | **61.19** | **58.72** |

be seen in both tables, exploiting subclass information in the SVDD optimization process leads to increased classification performance.

## VI. Conclusion

In this work, we have described an extension of the SVDD so that it exploits subclass information in its optimization process. We have evaluated the proposed method in face recognition and human action recognition problems, obtaining increased performance. Future work could include employing additional optimization criteria in the SVDD optimization process.

REFERENCES

[1] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[2] Y. Xiao, B. Liu, Z. Hao, and L. Cao, "A k-farthest-neighbor-based approach for support vector data description," *Applied intelligence*, vol. 41, no. 1, pp. 196–211, 2014.

[3] S. Khazai, A. Safari, B. Mojaradi, and S. Homayouni, "Improving the svdd approach to hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 4, pp. 594–598, 2012.

[4] 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.

[5] 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.

[6] 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.

[7] S. Zafeiriou and N. Laskaris, "On the improvement of support vector techniques for clustering by means of whitening transform," *IEEE Signal Processing Letters*, vol. 15, pp. 198–201, 2008.

[8] G. Arvanitidis and A. Tefas, "Exploiting graph embedding in support vector machines," *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2012.

[9] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1968–1979, 2013.

[10] ——, "Discriminant bag of words based representation for human action recognition," *Pattern Recognition Letters*, 2014.

[11] ——, "Graph embedded extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.

[12] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, "Video summarization based on subclass support vector data description," *IEEE Symposium Series on Computational Intelligence (SSCI), IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pp. 183–187, 2014.

[13] ——, "Exploiting subclass information in one-class support vector machine for video summarization," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2259–2263, 2015.

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

[15] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine learning*, vol. 30, no. 2-3, pp. 195–215, 1998.

[16] A. M. Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.

[17] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," *IEEE Workshop on Applications of Computer Vision*, 1994.

[18] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[19] B. Becker and E. Ortiz, "Evaluating open-universe face identification on the web," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 904–911, 2013.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.