Semi-Supervised Subclass Support Vector Data Description for image and video classification

Vasileios Mygdalis*, Alexandros Iosifidis*†, Anastasios Tefas*, Ioannis Pitas*
*Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
† Department of Engineering, Electrical and Computer Engineering, Aarhus University, Denmark
Email: {tefas,pitas}@aiia.csd.auth.gr

***Abstract*—In this paper, the Semi-Supervised Subclass Support Vector Data Description is presented, a method that operates in both the supervised and the semi-supervised One-class classification case. The proposed method consists a novel extension of the standard SVDD method, by introducing two additional terms its optimization problem. These two terms correspond to expressing global and local geometric data information respectively, during the classifier optimization process. Global geometric data information is employed by minimizing the global target class variance, assuming that subclasses may have been formed within as well. In addition, by exploiting the semi-supervised learning smoothness assumption, local neighborhood information between all available (labeled and unlabeled) data is preserved, even in the supervised learning case. We show that the adoption of both terms results in a regularized feature space, where low variance directions have been emphasized, while local geometric data information have been preserved. The proposed method has been evaluated in classification problems related to face recognition, human action recognition and generic One-class classification problems, comparing favorably against related One-class classification methods in both the semi-supervised and the supervised learning cases.**

***Index Terms*—One-class classification, Support Vector Data Description, Semi-supervised SVDD, Subclass SVDD, S^{3}SVDD**

I. Introduction

One-class classification (OCC) involves only a single class, the so-called target class, which must be distinguished from the rest of the world. It is commonly employed when sampling for different classes (negative examples) is difficult, expensive or even impossible, providing invaluable applications in failure detection tasks, medical diagnosis, mobile fraud detection [1]. It have also been applied to hyperspectral image classification [2], image segmentation [3], face authentication [4] video summarization [5], human action recognition and face recognition [6], [7], [8]. One of the most successful OCC methods is the Support Vector Data Description (SVDD) classifier [9].

The SVDD training phase determines the minimum bounding hypersphere which encloses the labeled examples of the target class. Test patterns that fall inside this hypersphere are classified to the target class, or are considered as outliers, otherwise. Over the years, many SVDD extensions have been proposed to increase training/test speed [10], [11], improve classification accuracy [6], [8], [12] and perform semi-supervised learning [13], [14]. For example, SVDD have been extended in order to automatically determine optimized Gaussian kernel parameters (i.e., the sigma) [10], or employ fuzzy rough feature sets [15]. A training approach that obtains a solution in a regularized space, resembling a hyperellipsoid in the input space, can be obtained by employing the whitening transform in the training data [5], [12], [16], [17]. Moreover, as in every Support Vector based classification method, removing the non-support vectors from the training set would not affect the final SVDD classification model, as highlighted in [18]. A method that allows fast SVDD testing was presented in [11], where the authors propose the calculation of feature vector preimages, in order to apply the relationships between this feature vector and the SVDD hypersphere center to re-expresses the center with a single vector, rather than as a linear combination of the support vectors. Recently, SVDD has been extended in the context of semi-supervised learning [19], by employing relationships between labeled and unlabeled training patterns, expressed with Nearest Neighbourhood (kNN) graph structures, in order to learn the optimal hypersphere in a regularized space, where locality information is preserved [2], [20].

Despite the important advancements of OCC methods over the past years, particular characteristics commonly being exploited by state-of-the-art image and video multiclass classification methods [21], [22], [23], [24], are yet to be examined in the OCC case. Such particular characteristics include modeling and recognizing innate diverse classes, such as recognizing crowd scenes, action recognition in scenes shot using different settings (e.g., illumination changes, indoor and outdoor scenes), by addressing each diversion within a class as a different subclass. Thereby, within-class class dispersion is minimized by exploiting subclass information that has been extracted by employing e.g., an additional unsupervised step. Moreover, there are cases where the labeled data only represent a subset of all the data available to train the classifer. To each respective end, methods exploiting subclass information [8] and methods exploiting unlabeled data [19] have been devised. However, none of the available OCC methods is able to combine semi-supervised classification and subclass information at the same time.

In this paper, the Semi-Supervised Subclass Support Vector Data Description (S^{3}SVDD) is presented, a method that operates in both the supervised and the semi-supervised One-class classification case. The proposed method consists a novel extension of the standard SVDD method, by introducing two additional terms its optimization problem. These two terms correspond to expressing global and local geometric data information respectively, during the classifier optimization process. Global geometric data information is employed by minimizing

the global target class variance, assuming that subclasses may have been formed within as well. In addition, by exploiting the semi-supervised learning smoothness assumption, local neighborhood information between all available (labeled and unlabeled) data is preserved, even in the supervised learning case. We show that the adoption of both terms results in a regularized feature space, where low variance directions have been emphasized, while local geometric data information have been preserved. The proposed method has been evaluated in classification problems related to face recognition, human action recognition and generic One-class classification problems, comparing favorably against related One-class classification methods in both the semi-supervised and the supervised learning cases.

The rest of the paper is structured as follows. in Section II, we review the related work in SVDD based classification, i.e., the standard SVDD [9] and the Subclass SVDD [8] methods. The proposed method is detailed in Section III. Experiments for evaluating the performance of the proposed method for both supervised and semi-supervised classification are provided in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

In this section, we present the preliminaries required to introduce the proposed method. We consider the supervised classification case, where the labeled $D-$dimensional feature vectors $\boldsymbol{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$ originate from a target single class and are, therefore, employed to train the classifiers. The Standard SVDD [9] classifier is briefly described in Subsection II-A. In Subsection II-B, we describe the Subclass SVDD [8], which features one of the core regularization terms of the proposed method.

A. Support Vector Data Description

The standard SVDD method [9] aims at generating a hypersphere, having center $\boldsymbol{a} \in \mathbb{R}^D$ and radius R, which encloses the target class training vectors $\boldsymbol{x}_i, i = 1, \ldots, N$. The primal SVDD optimization problem is defined as follows [9]:

$$\begin{aligned} \underset{R,\xi_i,\boldsymbol{a}}{\text{Minimize:}} \;\; & R^2 + c\sum_{i=1}^{N}\xi_i \\ \text{subject to :} \;\; & \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, N, \end{aligned} \tag{1}$$

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:

$$L = \sum_{i=1}^{N}\gamma_i\ \boldsymbol{x}_i^T\boldsymbol{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_i\gamma_j\boldsymbol{x}_i^T\boldsymbol{x}_j, \tag{2}$$

subject to :

$$0 \leq \gamma_i \leq c,\ \ \sum_{i=1}^{N}\gamma_i = 1, \tag{3}$$

where γ_i is the Lagrange multiplier corresponding to each constraint of the primal SVDD optimization problem (1). For each training sample $\boldsymbol{x}_i$ that satisfies the constraint $\xi_i = 0$, the corresponding Lagrange multiplier γ_i is equal to zero. The optimal hypersphere center is a linear combination of the Lagrange multipliers and the support vectors:

$$\boldsymbol{a} = \sum_{i=1}^{N}\gamma_i\boldsymbol{x}_i. \tag{4}$$

The hypersphere radius is the distance R of the hyperpshere center to the boundary. The radius can be calculated by using any of the support vectors $\boldsymbol{x}_k$ whose coefficient satisfies $\gamma_k > 0$, excluding items that fall outside of the description [9] (i.e., the support vectors whose coefficient are $\gamma_i = c$), as follows:

$$R^2 = \|\boldsymbol{x}_k - \boldsymbol{a}\|^2. \tag{5}$$

By expressing the center $\boldsymbol{a}$ in terms of support vectors, the radius can be obtained as follows:

$$R^2 = \boldsymbol{x}_k^T\boldsymbol{x}_k - \sum_{i=1}^{N}\gamma_i\boldsymbol{x}_i^T\boldsymbol{x}_k - \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_i\gamma_j\boldsymbol{x}_i^T\boldsymbol{x}_j. \tag{6}$$

By observing (6), the hypersphere radius is expressed in a dot product form. In order to determine solutions in feature spaces of increased dimensionality, dot products can be replaced with kernel products. The kernel products represent data similarity in a feature space $\mathcal{F}$, with a kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$, where $\phi(\cdot) : \mathbb{R}^D \mapsto \mathcal{F}$ is a (usually nonlinear) function which maps the training data from the input space to the feature space, e.g., the polynomial or the Radial Basis Function (RBF) function.

Finally, a given test sample $\boldsymbol{x} \in \mathbb{R}^D$ is classified to the target class if it satisfies the following inequality:

$$\kappa(\boldsymbol{x},\boldsymbol{x}) - 2\sum_{i=1}^{N}\gamma_i\kappa(\boldsymbol{x},\boldsymbol{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_i\gamma_j\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) \leqslant R^2. \tag{7}$$

B. SVDD exploiting subclass information

In the case where the target class covariance matrix is not the unity matrix, or when the class has subclasses, the hypersphere generated by the standard SVDD, might be suboptimal. A recently proposed extension of the SVDD, namely the Subclass SVDD [8], handles this case by minimizing 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, e.g., [25]. By considering the case where s subclasses are formed within the target class, the within class dispersion can be expressed with a matrix $\boldsymbol{S} \in \mathbb{R}^{D\times D}$, as follows:

$$\boldsymbol{S} = \sum_{i=1}^{N}\sum_{j=1}^{s}\frac{N_j}{N}e_i^j(\boldsymbol{x}_i - \bar{\boldsymbol{x}_j})(\boldsymbol{x}_i - \bar{\boldsymbol{x}_j})^T, \tag{8}$$

where e_i^j is an index denoting that the training sample $\boldsymbol{x}_i$ 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 s can either be set manually, based on previous knowledge about the problem at hand, or be automatically determined as

an additional classifier hyper-parameter during the classifier training process, by applying k-fold (e.g., $5-$fold) cross-validation on the training data. In order to incorporate subclass information in the SVDD optimization process, the following optimization problem have been proposed [8]:

$$\begin{aligned} \underset{R,\xi_i,\boldsymbol{a}}{\text{Minimize:}} \;\; & R^2 + c\sum_{i=1}^{N}\xi_i \\ \text{subject to :} \;\; & (\boldsymbol{x}_i-\boldsymbol{a})^T\boldsymbol{S}^{-1}(\boldsymbol{x}_i-\boldsymbol{a}) \leq R^2+\xi_i, \\ & \xi_i \geq 0, \qquad i=1,\ldots,N, \end{aligned} \tag{9}$$

where the description no longer represents a hypersphere, but a hyperellipsoid with hyperellipsoid center $\boldsymbol{a}$, R is the Mahalanobis distance from the hyperellipsoid center, ξ_i are the slack variables and c is a trade-off parameter between training error and generalization performance.

The above described optimization problem is equivalent to the following dual optimization problem:

$$\underset{\gamma_i}{\text{Minimize:}} \sum_{i=1}^{N}\gamma_i\boldsymbol{x}_i\boldsymbol{S}^{-1}\boldsymbol{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_i\gamma_j\boldsymbol{x}_i\boldsymbol{S}^{-1}\boldsymbol{x}_j \tag{10}$$

$$\text{subject to :} 0 \leq \gamma_i \leq c, \;\; \sum_{i=1}^{N}\gamma_i = 1, \tag{11}$$

where γ_i are the support vector coefficients (i.e., Lagrange multipliers) for each training sample $\boldsymbol{x}_i$. Values of $\gamma_i > 0$ denote that $\boldsymbol{x}_i$ is a support vector. Here, it should be noted that the parameter c can take any positive value. A value $c = 0$, eliminates the chance of convergence, since the constraints in (11) 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 $(0, 1]$.

The primal variable $\boldsymbol{a}$ (i.e., the hyperellipsoid center) can be recovered as follows:

$$\boldsymbol{a} = \boldsymbol{S}^{-1}\boldsymbol{X}\boldsymbol{\gamma}, \tag{12}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N]^T$ is the datamatrix and $\boldsymbol{\gamma} \in \mathbb{R}^N$ is a column vector whose ith column contains the support vector coefficient γ_i of $\boldsymbol{x}_i$.

The primal variable R can be recovered from a support vector $\boldsymbol{x}_k$ whose coefficient satisfies $0 < \gamma_i < c$, as follows:

$$R^2 = \|\boldsymbol{x}_k - \boldsymbol{a}\|^2 = \|\boldsymbol{x}_k - \boldsymbol{S}^{-1}\boldsymbol{X}\boldsymbol{\gamma}\|^2. \tag{13}$$

In order to decide whether a test sample $\boldsymbol{x} \in \mathbb{R}^D$ falls inside the hyperellipsoid, the following decision value is obtained:

$$f(\boldsymbol{x}) = R^2 - \|\boldsymbol{x} - \boldsymbol{a}\|^2, \tag{14}$$

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

By expressing the primal variables in terms of support vectors, using the equations (12) and (13), the following solution is obtained:

$$f(\boldsymbol{x}) = \|\boldsymbol{x}_k - \boldsymbol{S}^{-1}\boldsymbol{X}\boldsymbol{\gamma}\|^2 - \|\boldsymbol{x} - \boldsymbol{S}^{-1}\boldsymbol{X}\boldsymbol{\gamma}\|^2. \tag{15}$$

In the cases where a mapping function $\phi(\cdot)$ has been employed, the matrix $\boldsymbol{S}$ is defined the arbitrary dimensionality space $\mathcal{F}$. Moreover, in the case where the feature space dimensionality is higher than N, the matrix $\boldsymbol{S}$ is of rank N, thus not invertible. To this end, a linear classifier can be applied in a subspace determined by applying kernel PCA on the training data [5], [12], [16], [17].

In Subclass SVDD [8], a two step approach is followed. First, the matrix $\boldsymbol{S}$ is decomposed as follows:

$$\boldsymbol{S} = \boldsymbol{\Phi}\left(\frac{1}{N}\sum_{j=1}^{k}N_j\boldsymbol{e}_j\boldsymbol{e}_j^T\right)\boldsymbol{\Phi}^T = \boldsymbol{\Phi}\boldsymbol{M}\boldsymbol{\Phi}^T, \tag{16}$$

where $\boldsymbol{e}_j \in \mathbb{R}^N$ is a vector having elements $e_{ji} = 1$, if the training data $\boldsymbol{x}_i$ belongs to the $j-$th subclass (having N_j elements), or zero otherwise, $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is the matrix that encodes subclass information in a pairwise manner and the matrix $\boldsymbol{\Phi}$ contains the training data representations in $\mathcal{F}$, i.e., $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1),\ldots,\phi(\boldsymbol{x}_N)]^T$.

Next, a regularized version of $\boldsymbol{S}$ is employed, such that $\tilde{\boldsymbol{S}} = \boldsymbol{S} + r\boldsymbol{I}$, 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:

$$\tilde{\boldsymbol{S}}^{-1} = \frac{1}{r}\boldsymbol{I} - \frac{1}{r^2}\boldsymbol{\Phi}\left(\boldsymbol{M}^{-1} + \frac{1}{r}\boldsymbol{K}\right)^{-1}\boldsymbol{\Phi}^T, \tag{17}$$

where $\boldsymbol{K} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ is the so-called kernel matrix. By employing (17) in (10), following function should be minimized:

$$\begin{aligned} L = & \sum_{i=1}^{N}\gamma_i\left(\frac{1}{r}k_{ii} - \frac{1}{r^2}\boldsymbol{k}_i^T(\boldsymbol{M}^{-1}+\frac{1}{r}\boldsymbol{K})^{-1}\boldsymbol{k}_i\right) - \\ & - \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_i\gamma_j\left(\frac{1}{r}k_{ij} - \frac{1}{r^2}\boldsymbol{k}_i^T(\boldsymbol{M}^{-1}+\frac{1}{r}\boldsymbol{K})^{-1}\boldsymbol{k}_j\right), \end{aligned} \tag{18}$$

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 $\boldsymbol{K}$.

Function (18) is of the same form as the standard SVDD optimization function (2), while using the modified kernel:

$$\tilde{\kappa}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{r}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{r^2}\boldsymbol{k}_i^T\left(\boldsymbol{M}^{-1} + \frac{1}{r}\boldsymbol{K}\right)^{-1}\boldsymbol{k}_j. \tag{19}$$

Finally, in order to decide whether a test sample $\boldsymbol{x} \in \mathbb{R}^D$ belongs to the training class, the standard SVDD solution (7) can be employed, using the modified kernel found in (19).

III. S^3 SVDD

In this section, we describe in detail the proposed method and its properties. Consider a set $\mathcal{X} = \{\boldsymbol{x}_i,\ldots,\boldsymbol{x}_N\}$, containing $N = \ell + u$ available data, from which ℓ are labeled and u are unlabeled. In the OCC case, we expect that all labeled data belong to the target class, while no information for the unlabeled data is available.

We consider the case where a non-linear continuous function $\phi(\cdot) : \mathbb{R}^D \mapsto \mathcal{F}$ has been employed to all available data, mapping them from the input space to the feature space $\mathcal{F}$, and the ith data representation is denoted as $\boldsymbol{\phi}_i = \phi(\boldsymbol{x}_i)$. Also let $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{F}|\times\ell}$ be the matrix containing the labeled data representations in $\mathcal{F}$ and $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{|\mathcal{F}|\times N}$ the corresponding

matrix containing labeled and unlabeled data mappings. Then, $\boldsymbol{K} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}, \boldsymbol{K} \in \mathbb{R}^{\ell\times\ell}$ denotes the kernel matrix between the labeled data, while $\hat{\boldsymbol{K}} = \tilde{\boldsymbol{\Phi}}^T\tilde{\boldsymbol{\Phi}}, \hat{\boldsymbol{K}} \in \mathbb{R}^{N\times N}$ denotes the kernel matrix containing data similarity between all data (labeled and unlabeled) and $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{\Phi}}^T\boldsymbol{\Phi}$, denotes the $N \times \ell$ matrix that contains data similarity of labeled data, with all available data. Both $\boldsymbol{K}$ and $\tilde{\boldsymbol{K}}$, are submatrices of $\hat{\boldsymbol{K}}$.

The mathematical formulation of the proposed S^{3}SVDD classifier is derived as follows. First, by following the basic semi-supervised learning smoothness assumption, we expect that items that fall close to each other, are more likely to share the same label. According to data similarity, the outputs of the decision function should be smooth on adjacent data. To this end, we assume that a kNN graph $\mathcal{G} = \{\mathcal{X}, \boldsymbol{W}\}$ is formed between all available data $\mathcal{X}$ and $\boldsymbol{W}$ is the graph weight matrix, whose elements W_{ij} are initiated with a heat-kernel function:

$$W_{ij} = \begin{cases} exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2}{2\sigma^2}\right), & \text{if } \boldsymbol{x}_i \text{ is labeled and } \boldsymbol{x}_j \in \mathcal{N}_i \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

where σ^2 is the variance between the training data, which the normal scaling factor of the distances between the training samples, $i, j = 1, \ldots, N$ are the indices of labeled and unlabeled data respectively and $\mathcal{N}_i$ denotes if the element $\boldsymbol{x}_j$ is among the kNN neighbors of any labeled sample $\boldsymbol{x}_i$. We discard the kNN data relationships between unlabeled data as in [19], since we only require the solution to be supported by labeled data. In all our experiments, we have fixed $k = 5$ since such a value usually provides good results [26], limiting the outliers regularizing the hyperellipsoid center.

Additionally, we require that the Subclass SVDD regularization term to be expressed for the ℓ labeled examples (which belong to the target class). In all our experiments, we have fixed the number of subclasses to be $s = 3$, in order to restrict the available parameters to be tuned and moreover, it has shown to provide stable performance in our previous work [5], [8] in supervised learning. Finally, by introducing the two terms, the proposed optimization problem is formed as follows:

$$\begin{aligned} \underset{R,\xi_i,\boldsymbol{a}}{\text{Minimize:}} \quad & R^2 + c\sum_{i=1}^{\ell}\xi_i + c'\sum_{i,j}^{N} w_{ij}\left(f_i - f_j\right)^2 \\ \text{subject to :} \quad & \left(\boldsymbol{\phi}_i - \boldsymbol{a}\right)^T \boldsymbol{S}^{-1}\left(\boldsymbol{\phi}_i - \boldsymbol{a}\right) \le R^2 + \xi_i, \\ & \xi_i \ge 0, \qquad i = 1, \ldots, \ell, \end{aligned} \tag{21}$$

where $\xi_i \ge 0$ are the slack variables, c is the standard SVDD parameter, W_{ij} contains data similarity between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ (if they belong to the same neighborhood, or zero otherwise), f_i is the output of the standard SVDD decision function defined in (22) for $\boldsymbol{\phi}_i$ and $c' \ge 0$ is an additional parameter allowing unlabeled data information to be incorporated in the Subclass SVDD optimization problem. In the case where $c' = 0$, the optimization problem degenerates to the Subclass SVDD optimization problem (9). For the case where $c' > 0$, we would like to simplify the third term of the optimization problem. First, we consider the decision function for a training sample $\boldsymbol{x} \in \mathbb{R}^D$, combined with a kernel function, as follows:

$$f(\boldsymbol{x}) = R^2 - \|\phi(\boldsymbol{x}) - \boldsymbol{a}\|^2, \tag{22}$$

and the difference between the outputs of f_j and f_i for all kernels that satisfy $\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i = \boldsymbol{\phi}_j^T\boldsymbol{\phi}_j = z, z$ constant for every i, j which is true for e.g., the RBF kernel, can be simplified as follows [11]:

$$f_i - f_j = 2(\boldsymbol{\phi}_j - \boldsymbol{\phi}_i)^T\boldsymbol{a}. \tag{23}$$

Finally, we substitute the third term of the proposed optimization problem with the following equal expression:

$$\sum_{i,j}^{N} w_{ij}\left(f_i - f_j\right)^2 = 4\boldsymbol{a}^T\tilde{\boldsymbol{\Phi}}^T\boldsymbol{L}\tilde{\boldsymbol{\Phi}}\boldsymbol{a} = \boldsymbol{a}^T\boldsymbol{D}\boldsymbol{a}, \tag{24}$$

where $\boldsymbol{L}$ is the graph Laplacian matrix corresponding to the graph weight matrix $\boldsymbol{W}$ and $\boldsymbol{D} = 4\tilde{\boldsymbol{\Phi}}^T\boldsymbol{L}\tilde{\boldsymbol{\Phi}}$.

The optimization problem defined in (21) can be solved by finding the saddle points of the Lagrangian:

$$\begin{aligned} L = R^2 + c\sum_{i=1}^{N}\xi_i + c'\boldsymbol{a}^T\boldsymbol{D}\boldsymbol{a} - \sum_{i=1}^{N}\beta_i\xi_i - \\ - \sum_{i=1}^{N}\gamma_i\left(R^2 + \xi_i - \left(\boldsymbol{\phi}_i - \boldsymbol{a}\right)^T\boldsymbol{S}^{-1}\left(\boldsymbol{\phi}_i - \boldsymbol{a}\right)\right), \end{aligned} \tag{25}$$

where β_i and γ_i are Lagrange multipliers. By zeroing gradient the gradients of the Lagrangian with respect to R, ξ_i and $\boldsymbol{a}$, we obtain the following formula for the primal variable $\boldsymbol{a}$:

$$\boldsymbol{a} = \left(\boldsymbol{S}^{-1} + c'\boldsymbol{D}\right)^{-1}\boldsymbol{S}^{-1}\boldsymbol{\Phi}\boldsymbol{\gamma}. \tag{26}$$

Its detailed derivation is explained in Appendix A. The Lagrange multipliers β_i can be discarded, by demanding that $0 \le \gamma_i \le c$. After derivations described in Appendix B, we obtain the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\gamma_i}}{\text{Minimize:}} \quad & \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T(\frac{1}{c'}\boldsymbol{D}^{-1} + \boldsymbol{S})^{-1}\boldsymbol{\phi}_j\gamma_j \\ & - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j + \sum_{i=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_i \\ \text{subject to :} \quad & 0 \le \gamma_i \le c. \end{aligned} \tag{27}$$

The S^{3}SVDD optimization problem can be solved using standard SVDD implementations [27], by employing a kernel matrix[1] $\boldsymbol{Q} \in \mathbb{R}^{\ell\times\ell}$, having the following values:

$$q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_j - \boldsymbol{\phi}_i^T(\frac{1}{c'}\boldsymbol{D}^{-1} + \boldsymbol{S})^{-1}\boldsymbol{\phi}_j. \tag{28}$$

Since we work in spaces of arbitrary dimensionality, we require that both matrices $\boldsymbol{S}$ and $\boldsymbol{D}$ are invertible. Thus, we employ their regularized versions, defined as

$$\tilde{\boldsymbol{S}} = \boldsymbol{S} + r_1\boldsymbol{I} \quad \text{and} \quad \tilde{\boldsymbol{D}} = \boldsymbol{D} + r_2\boldsymbol{I}, \tag{29}$$

[1]Employing this kernel matrix essentially solves (27) by ignoring its 3rd term as in [9], which is constant for all kernels having the property $\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_i = d$, where d is a constant.

where r_1 and r_2 are regularization parameters increasing the ranks of the matrices, and $\boldsymbol{I}$ are identity matrices of appropriate dimensions. Moreover, r_1 and r_2 can tuned to control the amount of regularization. In all our experiments, we have employed values of r_1 and r_2 equal to 10^l, where $l = -3, \ldots, 3$.

As proven in Appendix C, the kernel matrix $\boldsymbol{Q}$ is of the following form:

$$\boldsymbol{Q} = \left[\alpha \boldsymbol{I} - \beta \left(\boldsymbol{M} + \beta^{-1}\boldsymbol{K}^{-1}\right)^{-1} \boldsymbol{M}\right] \boldsymbol{K} + \gamma \tilde{\boldsymbol{K}}^T \left(\tilde{\boldsymbol{L}}^{-1} + \delta \hat{\boldsymbol{K}}\right)^{-1} \tilde{\boldsymbol{K}}, \quad (30)$$

where $\tilde{\boldsymbol{L}} = \left[\left(\boldsymbol{L}^{-1} + \frac{1}{r_2}\hat{\boldsymbol{K}}\right)^{-1} + \tilde{\boldsymbol{M}}\right]$ is a matrix that combines global geometric information with local data relationships (for every $r_2 \neq 0$), $\tilde{\boldsymbol{M}} \in \mathbb{R}^{N \times N}$ is a matrix of the same rank as $\boldsymbol{M}$ (described in Subsection II-B), being equal to $\boldsymbol{M}$ in the positions of the labeled data, and having zeros otherwise (i.e., in the positions of the unlabeled data) and for notation simplicity, we employ the parameters α, β, γ and δ, where $\alpha = \left(\frac{1}{r_1} - \frac{r_2}{r_1 r_2 + 1}\right)$, $\beta = \frac{1}{r_1}$, $\gamma = \frac{(r_2+1)(r_2-1)}{(r_1 r_2+1)^2}$ and $\delta = \frac{(r_2+1)(r_2-1)}{r_1 r_2^2 + r_2}$. Finally, the decision value for a test vector $\boldsymbol{x} \in \mathbb{R}^D$ can be obtained through (7), replacing the kernel matrix with $\boldsymbol{Q}$.

The proposed method provides an extension of the Subclass SVDD [8], in the context of semi-supervised learning. Global within-class variance information about the labeled data, as well as local neighborhood information pairwise relationships between labeled and unlabeled data are incorporated in the SVDD optimization process. The resulting hypershere center is regularized by both terms. In the case where no unlabeled data are available (i.e., the supervised learning case), the proposed method has the effect of combining local and global geometric data information described by the matrices $\boldsymbol{M}$ and $\boldsymbol{L}$. For the supervised learning case, and by ignoring local geometric data information, the proposed S^{3}SVDD method degenerates to Subclass SVDD. Moreover, by assuming that no subclasses are formed within the target class (i.e., $s = 1$), then the matrix $\boldsymbol{S}$ becomes the standard within-class scatter matrix. Thus, for a value of $s = 1$, when no unlabeled data are available and when local geometric information is ignored, the proposed method degenerates to Ellipsoid SVDD methods [16], [17]. Additionally, in the semi-supervised case, we have implemented the smoothness assumption from a graph-theoretic perspective. If the first term of the proposed method is ignored, i.e, $\boldsymbol{S} = \boldsymbol{I}$, the proposed method degenerates to Graph-based SVDD [19]. Thus, the methods described in [16], [17], [19] can be considered as special cases of the proposed method.

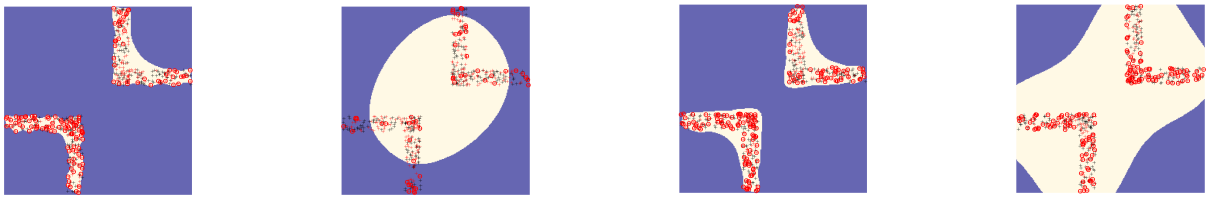
We demonstrate the regularization effects introduced by the proposed method by using a 2D example, as depicted in Figure 1. In this example, although the set of labeled data belong to a single class only, its items form two distinct subclasses. Moreover, let us assume that there also exists a set of unlabeled data, that fall very close to the labeled data. We have employed the standard SVDD and the proposed method in this dataset, in order to obtain the generated classification boundaries. We evaluate the boundaries empirically, by examining two different parameter settings cases. First, we determine the parameters such that the boundaries generated by both SVDD and the proposed method tightly enclose all training data, as can be seen in Figures 1.a and 1.c, for standard SVDD and the proposed method, respectively. Next, we depict the classification boundaries using modified parameter settings, such that the positive test space is expanded, as can be seen for standard SVDD in Figure 1.b and the proposed method in Figure 1.d,. We examine this case since this is a commonly followed procedure, in order to increase a classifier generalization performance. As be seen, when appropriate parameter settings are employed (Figures 1.a and 1.c, a toy 2D dataset can be modeled well enough by both methods. However, when trying to expand the positive classification space, the boundaries generated by the proposed method, seem to follow the data distribution more appropriately than the ones generated by standard SVDD. This can be explained by the additional regularization introduced by the two terms, and the improved distribution modeling using the unlabeled data. These properties are very important in realistic applications.

Finally, we discuss the parameters r_1 and r_2 defined in (29), which control the regularization effect. In practice, different combinations of r_1 and r_2 should be employed depending on the application at hand and the target class distribution. That is, the parameters r_1 and r_2 along with the standard SVDD parameter c should be optimally chosen, which increases the training algorithm complexity. In our experiments, we have employed cross-validation for determining the optimal parameters r_1 and r_2 from a set of predefined values, discussed in the Section IV. However, no additional computational complexity is introduced in the optimization process, other than calculating the kernel matrix in (30), since standard SVDD implementations are thereby employed.

IV. Experiments

In this section, we describe the experiments conducted in order to evaluate the performance of the proposed S^{3}SVDD classifier, in supervised and semi-supervised OCC problems. For comparison reasons, we have also applied the standard SVDD classifier [9], the Semi-Supervised One-class Support Vector Machines classifier (S^{2}OC-SVM) [2] and the Graph-based SVDD (S^{2}SVDD) [19]. We refer to the competing methods with their respective acronyms, hereafter.

For all methods, we have determined the optimal set of parameters using a cross-validation procedure, by applying grid search on a set of predefined values. The set of predefined values was the same for all methods. Unless stated otherwise in the following subsections, we have employed the RBF kernel function and Euclidean distances, where the corresponding scaling factor was equal to $a = \{0.01, 0.1, 1, 5, 10, 100\}$. The regularization parameters r_1 and r_2, along with the corresponding regularization parameter r of S^{2}SVDD and S^{2}OC-SVM, were set equal to 10^l, where $l = -3, \ldots, 3$. The kNN graphs employed by the proposed method, S^{2}OC-SVM and S^{2}SVDD were formed using (20), for $k = 5$. The SVDD parameter c and the corresponding SVM parameter ν, were determined from a set of values equal to



(a) SVDD, C1 (b) SVDD, C2 (c) S^{3}SVDD, C1 (d) S^{3}SVDD, C2

Fig. 1: Red crosses represent positive labeled data, black crosses represent unlabeled data, circles represent support vectors. The light color represent the classification space where the classifier desicion is positive (items in that space belong to the target class), and dark color represent the negative decision (items in that space are classified as outliers). C1 denotes the use of tight training data enclosure parameter settings. C2 presents the case of more loose boundaries that could be used in order to alleviate overfitting, in order to improve generalization performance.

$c = \nu = \{0.001, 0.01, 0.05, 0.1, \ldots, 0.9\}$, using a cross-validation procedure.

Since we focus on video and image data classification applications, we have formed One-class classification problems, related to Face Recognition and Human Action Recognition. Moreover, in order to demonstrate the generic effectiveness of the proposed method, we have also conducted experiments in generic OCC problems. We have employed publicly available datasets to this end. For datasets not providing training and test splits, we have employed the $5-$fold cross validation procedure, i.e., we have split the datasets into 5 sets and for each fold, we have employed 4 sets for training the classifiers and 1 set for testing. The reported performance is the average obtained performance for all folds. In all other cases, we have employed the standard training and test partitions provided by the dataset providers.

In order to apply the methods in the semi-supervised learning scenario, we have followed an additional procedure. We have used the labels of a randomly selected subset of the training data, using a variable $p \in (0, 1]$, such that $\ell = pN$, where ℓ is the number of the labeled data and N is the total number of all training data. Value of $p = 1$ denotes the supervised OCC scenario ($\ell = N$), while a value of $p = 0.2$ suggests that only 20% of the employed training data were labeled. In order to derive correct conclusions for the performance of each method, we have repeated each experiment (for each value of p) 5 times, and report the mean performance and the corresponding standard deviation. Here we should note that the exact same set of randomly selected hidden labels of the training data were employed for all competing methods. We should also note that, for the standard SVDD case, unlabeled data information is not employed at all in the training process.

In Subsections IV-A and IV-B, we describe in detail the experiments conducted in face recognition and human action recognition, respectively. Finally, experimental results in generic OCC problems are described in Subsection IV-C.

A. Experiments in Face Recognition

In our first set of experiments in face recognition, we have employed the PubFig+LFW [28] dataset. This dataset consists of $13,002$ 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 form the distractor set from LFW. For each facial image, the extracted features include the Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gabor wavelet features. The extracted features were reduced to 2048 dimensions with Principal Component Analysis (PCA), from which we only employed the first 1536 dimensions as in [28]). In order to create balanced classification problems, 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 the first 500 images (fixed for each OCC problem) from the distractor set. That is, each (of the 83) test set consists of a number of test images expected to belong in the target class, while the 500 from the distractor set should not. This scenario represents the face authentication scenario [4].

For evaluation metric in face recognition, we have employed the g-mean metric [29], which is the geometric mean of the recall and precision:

$$g = \sqrt{prec \times rec}, \tag{31}$$

which have been found to be suitable for binary classification problems, especially when the data are imbalanced.

Experimental results in PubFig+LFW dataset are depicted in Table I. Bold values denote the maximum obtained performance. We report the average g-mean metric obtained in each of the 83 OCC problems. For the semi-supervised classification scenario, since each experiment is repeated 5 times, we report the average g-mean metric, as well as the standard corresponding deviation. By observing the experimental results, it can be seen that the proposed S^{3}SVDD method outperforms the semi-supervised classification methods by a large extent. A surprising fact in this dataset, is that the standard SVDD outperformed the semi-supervised methods, although kNN regularization is not employed at all in the SVDD case. This can be explained by the fact that the distribution of the dataset features, might lie in spaces where 5NN-based regularization might not be optimal. On the other hand, the proposed method, although it employs the same 5NN regularization process, alleviates overfitting by additionally employing the within-

class covariance matrix in its optimization process. This is even more visible in the supervised classification case.

In our second set of experiments, we have employed classic face recognition datasets, i.e. the AR [30], Yale [31] and ORL [32] datasets. The datasets contain 2600, and 2432 and 400 frontal facial images belonging to 100, 38 and 40 subjects, respectively. In order to image feature vectors, we have resized the images to 40×30 pixels and employed the pixel luminance to produce a $D = 1200$ dimensional vector for each facial image. Since no standard experimental protocol is defined on these datasets, we have performed a cross-validation procedure. That is, we have performed data splits resulting into 5 sets. We have employed 4 sets for training the classifiers and left the 1 set for testing purposes. We have repeated the procedure 5 times (for each test set) and report the average performance obtained.

Experimental Results are depicted in Table II. The reported performance was obtained in a similar fashion to the PubFig+LFW case. In the ORL case, values of $p < 0.5$ are not reported, since the remaining labeled examples for each class are too few, thus no statistically important results can be obtained. The proposed method outperforms the competition in most supervised classification cases. The semi-supervised methods outperformed the standard SVDD, unlike the PubFig+LFW case. Since the employed feature vectors contain essentially the pixel luminance information, which may not be as discriminating as the features employed in the PubFig+LFW scenario, the classification power of semi-supervised methods is enhanced by adopting the additional criteria in their optimization process.

In our third set of experiments, we have performed experiments in AR, Yale and ORL datasets for the supervised learning case, only using augmented feature vectors, in order to include subclass information in all competing methods. To this end, we have created an alternative data representation $\mathbf{y} \in \mathbb{R}^D$, by employing the following approach. First, we have determined 3 subclasses in the input space using $k-$means, in exactly the same way as they are calculated for the proposed S^{3}SVDD. Next, contrary to the proposed method, we have calculated the matrix $\mathbf{S}$ explicitly (in the input space), using equation (8). Then, we calculated a transformation, such that $\mathbf{Y} = \mathbf{W}\mathbf{X}$, where $\mathbf{W}^T\mathbf{W} = \mathbf{S}^{-1}$ and $\mathbf{X}$ includes all the (labeled) training data. We have calculated $\mathbf{W}$ in a similar manner as in calculating the whitening transform, i.e., by performing eigenanalysis of $\mathbf{S}$, since $\mathbf{S}$ is in essence a modified version of the standard within-class covariance matrix. Therefore, the augmented datamatrix $\mathbf{Y}$ includes subclass information. Finally, we have employed $\mathbf{Y}$ instead of $\mathbf{X}$ in the all competing methods.

Experimental results are shown in Table III. As can be seen, subclass information employed in the feature vectors, affected the performance of the competing methods in a similar fashion. There was an overall increase of performance among the ORL and Yale datasets, and an overall decrease in the AR dataset. Moreover, there was no significant change in the performance rankings. Experimental results denote that subclass information extracted in the input space, is not carried optimally to the feature space, when it is introduced as a feature. This can be explained by the fact that by using this approach, subclass information is separated from the classifier optimization process. Moreover, the combination of local and global data relationships, has different regularization effects in the feature space than employing only local data relationships, regardless of having subclass information in the input space or not. Therefore, our approach of combining such information in the feature space is superior.

Finally, in our fourth set of experiments in face recognition, we have employed two additional datasets, namely the Caltech Faces [33] and Georgia Tech face database [34]. We have determined the performance of the competing methods in the supervised learning case, using features extracted by a deep neural network architecture. To this end, we have employed the VGG faces descriptor [35], i.e., we have performed a forward pass to the network, and extracted the 4096-dimensional features from the fc7 convolution layer, the final layer before the classifier. With the specific features, we should note that instead of Euclidean distances, we have employed cosine distances in order to form the RBF kernel for this specific experiments, since it is known that they work well with such features. Experimental results are shown in Table III. The proposed method outperformed the state-of-the-art in almost our experiments in face recognition.

B. Experiments in Human Action Recognition

In our human action recognition experiments, we have employed the i3DPost multi-view action database [36], the IMPART Multi-modal/Multi-view Dataset [37], as well as the Hollywood2 [38], Hollywood3D [39] and Olympic Sports [40] publicly available datasets. The i3DPost dataset contains 512 segmented high-resolution (1080×1920 pixel) videos depicting eight human actors performing eight activities. The IMPART dataset was collected using a multi-camera outdoor setup, which consists of 14 fixed cameras placed around each person 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 training and 307 test stereoscopic video segments depicting 14 actions. In our experiments, we have employed only the right video channel. Finally the Olympic Sports dataset consists of 783 videos depicting athletes performing 16 sport activities, which have been collected from YouTube and were annotated using Amazon Mechanical Turk. In terms of performance metrics for the human action recognition case, we report the mean average precision (mAP), which is the standard metric employed in Human Action recognition problems [22], [41], [42].

In order to obtain vectorial video representations for each video segment depicting each action, we have employed the dense trajectory-based video description [41]. 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

TABLE I: Average g-means and standard deviations within semi-supervised classification folds in PubFig+LFW dataset.

Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	65.44 ± 0.48	65.35 ± 0.14	59.45 ± 0.18	**70.78** ± 1.92
p=0.3	72.44 ± 0.51	64.90 ± 0.13	58.35 ± 0.49	**74.64** ± 0.35
p=0.5	**75.17** ± 0.22	64.24 ± 0.15	57.44 ± 0.22	72.62 ± 0.51
p=0.7	**75.27** ± 0.11	63.93 ± 0.07	57.44 ± 0.20	73.31 ± 0.84
Supervised	74.86	63.60	57.67	**79.31**

TABLE II: Average g-means rates and standard deviations within semi-supervised classification folds in various face recognition datasets, using standard features.

Dataset	AR			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	55.67 ± 1.26	**70.49** ± 0.39	69.09 ± 0.44	59.08 ± 0.97
p=0.3	64.04 ± 1.00	**72.66** ± 0.51	71.28 ± 0.61	59.65 ± 1.43
p=0.5	70.39 ± 0.59	**74.96** ± 0.41	74.06 ± 0.27	68.42 ± 1.05
p=0.7	72.44 ± 0.46	**75.99** ± 0.30	75.23 ± 0.41	73.13 ± 1.06
Supervised	74.20	**77.16**	76.42	76.95
Dataset	YALE			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	62.18 ± 0.75	**69.76** ± 0.51	66.05 ± 0.53	59.64 ± 1.50
p=0.3	64.20 ± 0.55	**70.89** ± 0.42	67.51 ± 0.42	62.10 ± 1.29
p=0.5	66.70 ± 0.57	**71.98** ± 0.47	69.48 ± 0.41	65.23 ± 1.35
p=0.7	67.74 ± 0.41	**72.47** ± 0.37	70.23 ± 0.36	70.81 ± 1.15
Supervised	69.00	73.22	71.09	**79.91**
Dataset	ORL			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.5	-	78.89 ± 3.10	73.52 ± 2.29	**90.52** ± 1.94
p=0.7	-	83.46 ± 4.79	79.71 ± 2.14	**93.94** ± 2.06
Supervised	77.08	85.45	82.34	**96.86**

TABLE III: Average g-means rates in Face Recognition datasets, using advanced features, in the supervised learning case.

Feature type	Subclass info			VGG fc7	
Algorithm/Dataset	AR	Yale	ORL	Caltech	Georgiatech
SVDD	68.94	70.48	92.24	64.95	65.10
S^2OC-SVM	75.01	**78.55**	95.40	94.46	93.24
S^2SVDD	73.11	76.36	94.54	91.92	90.68
S^3SVDD	**77.20**	76.25	**97.37**	**99.93**	**99.99**

video frames (7 frames are used in our experiments). We have employed these video segment descriptors in order to obtain five video segment representations by using the Bag-of-Words model [22], and combined them with kernel methods using a late fusion approach [43], i.e.,:

$$k(\mathcal{X}_i, \mathcal{X}_j) = exp\left(-\frac{1}{d}\sum_d \frac{\|\mathbf{x}_i^d - \mathbf{x}_j^d\|_2^2}{2\sigma_d^2}\right), \qquad (32)$$

$\boldsymbol{x}_i^d \in \mathbb{R}^D$ is a video feature vector for $d = 5$ (number of descriptor types) and σ_d is a parameter scaling the Euclidean distance between $\boldsymbol{x}_i^d$ and $\boldsymbol{x}_j^d$.

In the i3DPost and IMPART datasets, we have employed a 3-fold cross validation procedure, where we have split the datasets in 3 mutually exclusive sets. 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. This procedure was repeated for all activities, and repeated 3 times for each fold. In the Hollywood2, Hollywood 3D and Olympic Sports datasets, we employed the standard training and test videos, provided by the dataset providers [38], [39], [40].

In Table IV, we report the obtained mAP rates for each dataset. That is, we have formed OCC classification problems, where each classifier was trained using labeled examples from a single class, as well as unlabeled examples from their 5NN neighbors (except the standard SVDD case). As can be seen, the proposed method provides enhanced performance in every case for the semi-supervised classification cases. Moreover, the proposed method outperformed the competitors in every supervised classification case, and in some cases, by a large margin. Moreover, it can be also seen that both S^2SVDD and S^2OC-SVM outperformed the standard SVDD. This was also reported in our face recognition experiments, especially when the VGG features were employed. In both cases, the feature vectors employed were of high dimensionality and the datamatrices in all cases where sparse. Thus, it can be assumed that data were lying in manifold whose actual dimensionality was lower, and this effect was described by the kNN term. The proposed method was able to outperform the competition because in addition to the kNN term, global within-subclass variance was minimized at the same time. Thus, we conclude that both local and global regularization terms have contributed

to the increased classification performance.

C. Experiments in Generic OCC problems

In our final set of experiments, we have evaluated the performance in generic OCC problems, that are publicly available in the UCI repository [44]. The corresponding OCC versions were obtained from the Netherlands Pattern Recognition Laboratory [27], which have been modified to include binary labels only. Since no specific train and test data are predefined by the dataset providers, we have performed the 5-fold cross validation procedure. For each fold, we kept 80% for training purposes and 20% for testing. In order to implement the semi-supervised scenario, we have followed the procedure described in the beginning of Section IV. We note that for semi-supervised methods, we have employed both labeled and unlabeled training data, except for the standard SVDD case, where we only used the positive labeled ones.

In Table V, we report the obtained mAP rates for all competing methods. The proposed method outperformed the competition in amost every case. Additionally, it can be seen that the standard SVDD outperformed the semi-supervised learning methods in almost every case. By combining both facts, it can be explained that local geometric data relationships between $k = 5$ neighboring data did not contribute in a positive manner in regularizing the obtained classification space, and based on our insights, a different value of neighbors may did have a positive effect in the obtained results, e.g., $k = 10, 15$. However, the same value of $k = 5$ was used for the proposed method as well, where it was shown that it still outperformed the standard SVDD and alleviated the negative effects by exploiting subclass information.

V. Conclusion

In this paper, we have described a novel semi-supervised OCC method based on SVDD, that is regularized by two additional terms, combing global and local geometric data relationships. Our experiments indicated that there are cases, where either or both of of the additional terms contribute to derive the classification space, where enhanced classification performance is obtained. Therefore, the proposed method is superior to existing OCC methods in both the semi-supervised and the supervised learning case as well.

Future work could include applications or extensions in different classification problems and methods. Moreover, since we have proven that exploiting two regularization parameters at the same time for the SVDD case is beneficial, adding even more regularization terms describing different properties of the data, could be promising. However, we should note that every additional regularization term usually requires optimizing an additional hyperparameter. To this end, methods automatically determining the optimal parameters settings would be another research direction. Finally, the proposed method exploits spaces that have been explicitely estimated using transformation on a standard RBF kernel matrix. Estimating this space directly with a continuous piece-wise mapping function, i.e., novel kernel functions perhaps based on deep learning, could be promising.

Acknowledgment

This project has received funding from the European Union's European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART) and Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

Appendix A
Deriving the S^3 SVDD hypersphere center

For easier derivative calculation, we employ a vector be defined as $\boldsymbol{u} = \boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{a}$, where $\boldsymbol{a}$ is the actual hypersphere center. In addition, by employing $\boldsymbol{u}$ in equation (24), the proposed S^{3}SVDD optimization problem takes the following form:

$$\begin{aligned} \underset{R,\xi_i,\boldsymbol{u}}{\text{Minimize:}} \;\; & R^2 + c\sum_{i=1}^{\ell}\xi_i + c'\boldsymbol{u}^T\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{u} \\ \text{Subject to :} \;\; & \|\boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{\phi}_i - \boldsymbol{u}\|_2^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, \ell, \end{aligned} \tag{33}$$

which can be solved by finding the saddle points of the Lagrangian:

$$\begin{aligned} L = R^2 + c\sum_{i=1}^{N}\xi_i + c'\boldsymbol{u}^T\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{u} - \sum_{i=1}^{N}\beta_i\xi_i - \\ - \sum_{i=1}^{N}\gamma_i\left(R^2 + \xi_i - \|\boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{\phi}_i - \boldsymbol{u}\|^2\right), \end{aligned} \tag{34}$$

leading to the following optimality conditions:

$$\frac{\vartheta L}{\vartheta R} = 0 \Rightarrow \sum_{i=1}^{N}\gamma_i = 1, \tag{35}$$

$$\frac{\vartheta L}{\vartheta \xi_i} = 0 \Rightarrow \beta_i = c - \gamma_i, \tag{36}$$

Condition (36) can always be met if we demand $0 \leq \gamma_i \leq c$, thus the Lagrange multipliers β_i can be discarded. Finally, by zeroing the gradient of L with respect to $\boldsymbol{u}$, we obtain:

$$\begin{aligned} & \frac{\vartheta L}{\vartheta \boldsymbol{u}} = 0 \Rightarrow \\ & \Rightarrow 2c'\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{D}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{u} + 2\boldsymbol{u} = 2\boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{\Phi}\boldsymbol{\gamma} \Rightarrow \\ & \Rightarrow c'\boldsymbol{S}\boldsymbol{D}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{u} + \boldsymbol{S}^{\frac{1}{2}}\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\gamma} \Rightarrow \\ & \Rightarrow c'\boldsymbol{S}\boldsymbol{D}\boldsymbol{a} + \boldsymbol{a} = \boldsymbol{\Phi}\boldsymbol{\gamma} \Rightarrow \\ & \Rightarrow c'\boldsymbol{D}\boldsymbol{a} + \boldsymbol{S}^{-1}\boldsymbol{a} = \boldsymbol{S}^{-1}\boldsymbol{\Phi}\boldsymbol{\gamma} \Rightarrow \\ & \Rightarrow \boldsymbol{a} = \left(\boldsymbol{S}^{-1} + c'\boldsymbol{D}\right)^{-1}\boldsymbol{S}^{-1}\boldsymbol{\Phi}\boldsymbol{\gamma}, \end{aligned} \tag{37}$$

which is the S^{3}SVDD hypersphere center.

TABLE IV: Mean Average Precision and standard deviation within semi-supervised classification folds in Human Action Recognition Datasets.

Dataset	i3DPost			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	75.14 ± 1.48	81.25 ± 0.68	80.30 ± 0.58	**84.21** ± 0.59
p=0.3	77.60 ± 1.11	82.57 ± 0.60	81.67 ± 0.64	**86.00** ± 0.66
p=0.5	80.58 ± 0.95	83.36 ± 0.36	82.84 ± 0.70	**87.24** ± 0.78
p=0.7	82.57 ± 0.47	83.82 ± 0.18	83.32 ± 0.30	**87.72** ± 0.39
Supervised	83.78	84.11	83.70	**88.93**
Dataset	IMPART			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	51.46 ± 1.13	62.74 ± 0.52	61.46 ± 0.76	**68.19** ± 1.09
p=0.3	51.56 ± 0.58	62.64 ± 0.67	61.07 ± 0.70	**69.80** ±1.90
p=0.5	52.11 ± 0.65	62.81 ± 0.61	61.27 ± 0.34	**71.50** ± 1.69
p=0.7	53.15 ± 0.64	62.76 ± 0.31	61.27 ± 0.24	**71.28** ±1.05
Supervised	53.44	62.94	61.24	**72.44**
Dataset	Hollywood2			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	26.29 ± 0.72	26.37 ± 0.30	26.40 ± 0.33	**33.97** ± 0.51
p=0.3	26.04 ± 0.63	26.63 ± 0.29	26.64 ± 0.27	**35.16** ± 1.20
p=0.5	26.08 ± 0.78	26.45 ± 0.31	26.50 ± 0.28	**35.72** ± 0.66
p=0.7	26.03 ± 0.42	26.41 ± 0.15	26.45 ± 0.09	**35.51** ± 1.29
Supervised	25.22	26.75	26.52	**33.12**
Dataset	Hollywood3D			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	24.65 ± 0.23	27.53 ± 0.17	27.41 ± 0.17	**37.46** ± 1.79
p=0.3	24.19 ± 0.45	27.36 ± 0.21	27.58 ± 0.25	**37.67** ± 0.54
p=0.5	24.26 ± 0.27	27.49 ± 0.19	27.54 ± 0.19	**37.72** ± 1.29
p=0.7	23.87 ± 0.26	27.37 ± 0.14	27.50 ± 0.09	**38.47** ± 2.01
Supervised	23.38	27.44	27.64	**38.47**
Dataset	Olympic Sports			
Algorithm/P	SVDD	S^2OC-SVM	S^2SVDD	**S^3SVDD**
p=0.2	36.71 ± 2.14	45.54 ± 0.23	44.97 ± 0.45	**62.48** ± 1.50
p=0.3	35.82 ± 1.29	45.85 ± 0.41	45.04 ± 0.16	**62.81** ± 1.55
p=0.5	35.75 ± 0.89	45.30 ± 0.22	44.94 ± 0.16	**62.20** ± 1.67
p=0.7	36.58 ± 0.92	45.29 ± 0.17	45.05 ± 0.14	**65.34** ± 1.63
Supervised	36.89	44.98	44.99	**66.84**

APPENDIX B
DERIVATION OF THE LAGRANGIAN OF S^3 SVDD

After replacing (35), (36) in the Lagrangian function (34) (using the hypershere center $\boldsymbol{a}$ instead of $\boldsymbol{u}$), we obtain the following:

$$L = c'\boldsymbol{a}^T\boldsymbol{D}\boldsymbol{a} + \sum_{i=1}^{\ell}\gamma_i(\boldsymbol{\phi}_i - \boldsymbol{a})^T\boldsymbol{S}^{-1}(\boldsymbol{\phi}_i - \boldsymbol{a}). \quad (38)$$

Let $\boldsymbol{B} = \boldsymbol{S}^{-1} + c'\boldsymbol{D}$. By substituting (37) in (38), the Lagrangian takes the following form:

$$\begin{aligned} L = &\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{B}^{-1}c'\boldsymbol{D}\boldsymbol{B}^{-1}\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j \\ &+\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{B}^{-1}\boldsymbol{S}^{-1}\boldsymbol{B}^{-1}\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j \\ &-2\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{B}^{-1}\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j \\ &+\sum_{i=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_i = \\ &=\sum_{i=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_i - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{B}^{-1}\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j \end{aligned} \quad (39)$$

Moreover, by using the Woodbury identity:

$$(\boldsymbol{S}^{-1} + c'\boldsymbol{D})^{-1} = \boldsymbol{S} - \boldsymbol{S}(\frac{1}{c'}\boldsymbol{D}^{-1} + \boldsymbol{S})^{-1}\boldsymbol{S}, \quad (40)$$

TABLE V: Mean Average Precision and standard deviation within semi-supervised classification folds in generic OCC problems.

Dataset	Breast Malignant			
Algorithm/P	SVDD	S^{2}OC-SVM	S^{2}SVDD	**S^{3}SVDD**
p=0.2	98.49 ± 0.35	97.92 ± 0.09	97.94 ± 0.06	**99.00** ± 0.14
p=0.3	98.48 ± 0.27	97.97 ± 0.11	97.93 ± 0.09	**99.11** ± 0.22
p=0.5	98.62 ± 0.22	97.95 ± 0.70	97.91 ± 0.10	**99.19** ± 0.17
p=0.7	98.72 ± 0.17	97.97 ± 0.11	97.98 ± 0.15	**99.16** ± 0.20
Supervised	98.71	97.95	97.95	**99.16**
Dataset	Breast Benign			
Algorithm/P	SVDD	S^{2}OC-SVM	S^{2}SVDD	**S^{3}SVDD**
p=0.2	98.82 ± 0.28	98.80 ± 0.09	98.86 ± 0.04	**99.23**± 0.19
p=0.3	98.83 ± 0.29	98.82 ± 0.07	98.85 ± 0.08	**99.25** ± 0.21
p=0.5	98.97 ± 0.26	98.79 ± 0.08	98.79 ± 0.08	**99.31** ± 0.28
p=0.7	99.04 ± 0.16	98.85 ± 0.06	98.86 ± 0.11	**99.28** ± 0.19
Supervised	99.02	98.80	98.80	**99.07**
Dataset	Diabetes			
Algorithm/P	SVDD	S^{2}OC-SVM	S^{2}SVDD	**S^{3}SVDD**
p=0.2	**67.82** ± 2.52	52.78 ± 1.19	51.25 ± 1.34	66.27 ± 2.35
p=0.3	**68.88** ± 3.01	52.89 ± 1.17	52.22 ± 1.52	66.84 ± 1.81
p=0.5	**69.18** ± 1.20	52.74 ± 1.20	52.32 ± 1.13	66.68 ± 1.33
p=0.7	**69.36** ± 1.69	52.55 ± 0.81	52.07 ± 1.24	65.36 ± 1.75
Supervised	**68.91**	52.04	52.66	65.94
Dataset	Heart			
Algorithm/P	SVDD	S^{2}OC-SVM	S^{2}SVDD	**S^{3}SVDD**
p=0.2	78.22 ± 5.12	65.01 ± 1.61	65.79 ± 1.56	**78.32** ± 1.59
p=0.3	79.78 ± 4.56	65.01 ± 1.01	65.88 ± 1.44	**80.80** ± 1.77
p=0.5	80.78 ± 2.24	64.43 ± 0.62	66.17 ± 1.89	**82.36** ± 2.37
p=0.7	**83.10** ± 2.82	64.12 ± 0.45	65.63 ± 0.56	82.70 ± 1.19
Supervised	**82.87**	63.91	65.33	82.01
Dataset	Liver			
Algorithm/P	SVDD	S^{2}OC-SVM	S^{2}SVDD	**S^{3}SVDD**
p=0.2	75.76 ± 3.99	75.50 ± 0.83	75.17 ± 0.61	**82.84** ± 1.58
p=0.3	75.16 ± 3.10	75.59 ± 0.76	75.45 ± 0.71	**83.46** ± 1.77
p=0.5	77.72 ± 2.76	75.38 ± 0.49	75.20 ± 0.38	**83.25** ± 1.57
p=0.7	80.17 ± 2.92	75.28 ± 0.66	75.11 ± 0.59	**83.30** ± 1.55
Supervised	81.10	75.78	75.29	**83.34**

the Lagrangian takes its final form as follows:

$$L = \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T(\frac{1}{c'}\boldsymbol{D}^{-1}+\boldsymbol{S})^{-1}\boldsymbol{\phi}_j\gamma_j - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_j\gamma_j + \sum_{i=1}^{\ell}\gamma_i\boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_i. \quad (41)$$

APPENDIX C
DERIVATION OF THE KERNEL MATRIX

The S^{3}SVDD optimization problem can be solved using standard SVDD implementations, by employing the following kernel matrix $\boldsymbol{Q} \in \mathbb{R}^{\ell\times\ell}$:

$$q(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{\phi}_i^T\boldsymbol{S}^{-1}\boldsymbol{\phi}_j - \boldsymbol{\phi}_i^T(\frac{1}{c'}\boldsymbol{D}^{-1}+\boldsymbol{S})^{-1}\boldsymbol{\phi}_j. \quad (42)$$

Since we work in spaces of arbitrary dimensionality, we require that both matrices $\boldsymbol{S}$ and $\boldsymbol{D}$ are invertible. Thus, we employ their regularized versions, defined as

$$\tilde{\boldsymbol{S}} = \boldsymbol{S} + r_1\boldsymbol{I} \quad \text{and} \quad \tilde{\boldsymbol{D}} = \boldsymbol{D} + r_2\boldsymbol{I}, \quad (43)$$

where r_1 and r_2 are parameters set to a small value increasing the ranks of the matrices, and $\boldsymbol{I}$ are identity matrices of appropriate dimensions.

Since the matrix $\boldsymbol{S}$ is the same matrix employed in Subclass SVDD described in Section II, its inverse is obtained from equation (17), replacing the regularization parameter r with r_1, as follows:

$$\tilde{\boldsymbol{S}}^{-1} = \frac{1}{r_1}\boldsymbol{I} - \frac{1}{r_1^2}\boldsymbol{\Phi}\left(\boldsymbol{M}^{-1}+\frac{1}{r_1}\boldsymbol{K}\right)^{-1}\boldsymbol{\Phi}^T, \quad (44)$$

where $\boldsymbol{M}$ is a $\ell\times\ell$ matrix encoding subclass information between the labeled data, having values as described below equation (16). In a similar fashion, the inverse of $\boldsymbol{D}$ is given by:

$$\tilde{\boldsymbol{D}}^{-1} = \frac{1}{r_2}\boldsymbol{I} - \frac{1}{r_2^2}\tilde{\boldsymbol{\Phi}}\left(\boldsymbol{L}^{-1}+\frac{1}{r_2}\hat{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{\Phi}}^T, \quad (45)$$

where $\tilde{\boldsymbol{\Phi}}$ contains labeled and unlabeled data representations in the feature space $\mathcal{F}$. We employ the same notation, as in Section III. Hereafter, we consider the case where $c' = 1$, since similar regularization effects to the kernel matrix can be obtained with the parameter r_2.

Then, we calculate the quantity in the parenthesis of (42):

$$(\tilde{\boldsymbol{S}}+\tilde{\boldsymbol{D}}^{-1}) = \frac{r_1 r_2+1}{r_2}\boldsymbol{I}- \\ -\frac{1}{r_2^2}\tilde{\boldsymbol{\Phi}}\left(\boldsymbol{L}^{-1}+\frac{1}{r_2}\hat{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{\Phi}}^T+\boldsymbol{\Phi}\boldsymbol{M}\boldsymbol{\Phi}^T. \tag{46}$$

The matrix $\boldsymbol{\Phi}$ is a submatrix of $\tilde{\boldsymbol{\Phi}}$, containing the labeled data representations. Let us replace $\boldsymbol{\Phi}$ with $\tilde{\boldsymbol{\Phi}}$. The added values included in $\tilde{\boldsymbol{\Phi}}$ should not be employed in the calculations. To this end, let us replace $\boldsymbol{M}$, by defining a matrix $\tilde{\boldsymbol{M}} \in \mathbb{R}^{N\times N}$, which is of the same rank as $\boldsymbol{M}$, being equal to $\boldsymbol{M}$ in the positions of the labeled data, and having zeros otherwise (i.e., in the positions of the unlabeled data). Without loss of generality, the quantities can therefore be added.

Let $\tilde{\boldsymbol{L}} = \left[\left(\boldsymbol{L}^{-1}+\frac{1}{r_2}\hat{\boldsymbol{K}}\right)^{-1}+\tilde{\boldsymbol{M}}\right]$ be the matrix that describes this summation for every $r_2 \neq 0$. Then, the inverse of (46) is:

$$(\tilde{\boldsymbol{S}}+\tilde{\boldsymbol{D}}^{-1})^{-1} = \left(\frac{r_1 r_2+1}{r_2}\boldsymbol{I}+\frac{r_2^2-1}{r_2^2}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{L}}\tilde{\boldsymbol{\Phi}}^T\right)^{-1} = \\ = \frac{r_2}{r_1 r_2+1}\boldsymbol{I}- \\ -\frac{(r_2+1)(r_2-1)}{(r_1 r_2+1)^2}\tilde{\boldsymbol{\Phi}}\left(\tilde{\boldsymbol{L}}^{-1}+\frac{(r_2+1)(r_2-1)}{r_1 r_2^2+r2}\hat{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{\Phi}}^T. \tag{47}$$

Finally, after applying the derived equations in (42), we obtain the following kernel:

$$\boldsymbol{Q} = \left[\alpha\boldsymbol{I}-\beta\left(\boldsymbol{M}+\beta^{-1}\boldsymbol{K}^{-1}\right)^{-1}\boldsymbol{M}\right]\boldsymbol{K} \\ +\gamma\tilde{\boldsymbol{K}}^T\left(\tilde{\boldsymbol{L}}^{-1}+\delta\hat{\boldsymbol{K}}\right)^{-1}\tilde{\boldsymbol{K}}, \tag{48}$$

where: $\alpha = \left(\frac{1}{r_1}-\frac{r_2}{r_1 r_2+1}\right)$, $\beta = \frac{1}{r_1}$, $\gamma = \frac{(r_2+1)(r_2-1)}{(r_1 r_2+1)^2}$ and $\delta = \frac{(r_2+1)(r_2-1)}{r_1 r_2^2+r_2}$.

REFERENCES

[1] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[2] 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.

[3] B. Cyganek, "One-class support vector ensembles for image segmentation and classification," *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 103–117, 2012.

[4] M. Bicego, E. Grosso, and M. Tistarelli, "Face authentication using one-class support vector machines," *Advances in Biometric Person Authentication*, pp. 15–22, 2005.

[5] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, "Video summarization based on subclass Support Vector Data Description," *In: IEEE Symposium Series on Computational Intelligence (SSCI), Computational Intelligence for Engineering Solutions (CIES)*, pp. 183–187, 2014.

[6] ——, "Graph embedded one-class classifiers for media data classification," *Pattern Recognition*, vol. 60, pp. 585–595, 2016.

[7] A. Iosifidis, V. Mygdalis, A. Tefas, and I. Pitas, "One-class classification based on extreme learning and geometric class information," *Neural Processing Letters*, pp. 1–16, 2016.

[8] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, "Kernel subclass support vector description for face and human action recognition," *In: International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, pp. 1–5, 2016.

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

[10] 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.

[11] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1296–1313, 2010.

[12] 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.

[13] N. Görnitz, M. Kloft, and U. Brefeld, "Active and semi-supervised data domain description," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 407–422, 2009.

[14] T. Le, D. Tran, T. Tran, K. Nguyen, and W. Ma, "Fuzzy entropy semi-supervised support vector data description," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5, 2013.

[15] R. Sadeghi and J. Hamidzadeh, "Automatic support vector data description," *Soft Computing*, pp. 1–12, 2016.

[16] K. Wang, H. Xiao, and Y. Fu, "Ellipsoidal support vector data description in kernel pca subspace," *In: International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, pp. 13–18, 2016.

[17] K. Teeyapan, N. Theera-Umpon, and S. Auephanwiriyakul, "Ellipsoidal support vector data description," *Neural Computing and Applications*, pp. 1–11, 2016.

[18] 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.

[19] P. Duong, V. Nguyen, M. Dinh, T. Le, D. Tran, and W. Ma, "Graph-based semi-supervised support vector data description for novelty detection," *In: International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2015.

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

[21] 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.

[22] 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.

[23] ——, "Graph Embedded Extreme Learning Machine," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.

[24] A. Iosifidis and M. Gabbouj, "Multi-class support vector machine classifiers using intrinsic and penalty graphs," *Pattern Recognition*, vol. 55, pp. 231–246, 2016.

[25] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

[26] A. Iosifidis, A. Tefas, and I. Pitas, "Regularized extreme learning machine for multi-view semisupervised action recognition," *Neurocomputing*, no. 145, pp. 250–262, 2014.

[27] D. Tax, "Ddtools, the data description toolbox for matlab," June 2015, version 2.1.2.

[28] 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.

[29] 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.

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

[31] 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.

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

[33] "Caltech image datasets." [Online]. Available: http://www.vision.caltech.edu/archive.html

[34] "Georgia tech face database." [Online]. Available: http://www.anefian.com
[35] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," *British Machine Vision Conference*, 2015.
[36] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3DPost multi-view and 3D human action/interaction database," *In: European Conference on Visual Media Production (CVMP)*, pp. 159–168, 2009.
[37] H. Kim and A. Hilton, "Influence of colour and feature geometry on multimodal 3D point clouds data registration," *In: International Conference on 3D Vision (3DV)*, pp. 202–209, 2015.
[38] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2929–2936, 2009.
[39] 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.
[40] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," *European conference on computer vision*, pp. 392–405, 2010.
[41] 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.
[42] 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.
[43] 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.
[44] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml