Distance-based Human Action Recognition using optimized class representations

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, Greece

{*aiosif,tefas,pitas*}*@aiia.csd.auth.gr*

Abstract

We study distance-based classification of human actions and introduce a new metric learning approach based on logistic discrimination for the determination of a low-dimensional feature space of increased discrimination power. We argue that for effective distance-based classification, both the optimal projection space and the optimal class representation should be determined. We qualitatively and quantitatively illustrate the superiority of the proposed approach to metric learning approaches employing the class mean for class representation. We also introduce extensions of the proposed metric learning approach to allow for richer class representations and to operate in arbitrary-dimensional Hilbert spaces for non-linear feature extraction and classification. Experimental results denote that the performance of the proposed distance-based classification schemes is comparable (or even better) to that of Support Vector Machine classifier (in both the linear and kernel cases) which is currently the standard choice for human action recognition.

Keywords: Distance-based classification, Optimized class representations

1. Introduction

In this paper we focus on the problem of distance-based, multi-class classification of human actions and specifically on the Nearest Class Centroid (NCC) classification scheme that has been employed in many Computer Vision tasks, including image and action classification [22, 21, 35, 26, 5]. The success of NCC critically depends on the adopted distance function, which is usually learned by applying a learning process exploiting training samples. We follow this line of work and cast our classifier learning problem as one of learning a low-rank Mahalanobis distance which is shared across all classes. Such a Mahalanobis distance can be used in order to map the samples to a low-dimensional feature space of increased discrimination power, where classification is performed by employing the minimal Euclidean distance from the class representation.

Typically, NCC classification schemes employ the class mean vector for class representation, assuming that the classes forming the classification problem follow unimodal probability distributions having the same covariance structure. However, this is a strong assumption, which is difficult to be met in real classification problems. Consider the example illustrated in Figure 1. Figure 1a illustrates two classes formed by 2D data following different probability distributions and having different covariance structures. Figure 1b illustrates the projection space obtained by applying Linear Discriminant Analysis (LDA) [3] on the 2D data forming the two classes. As can be seen, LDA fails to determine a useful for classification subspace, since the two classes are mapped to the same region resulting to a classification rate equal to $46,45\%$. On the other hand, logistic discrimination is able to merely overcome these issues and increases class discrimination in the projection space, as illustrated in Figure 1c, leading to a classification rate equal to 94.17%. Finally, logistic discrimination employing the class vectors denoted by triangles in Figures 1a,d for class representation is able to perfectly discriminate the two classes in the projection space, leading to a classification rate equal to 100%. As can be seen in this, rather simple, example, critical role on the performance of NCC classifier plays, not only the adopted distance function, but also the adopted class representation.

In this paper, we propose a new metric learning algorithm based on multi-class logistic discrimination, where a sample is enforced to be closer to its class representation than to any other class representation in the projection space. The proposed algorithm determines both the optimal projection matrix and the optimal class representation that can be, subsequently, used for classification. In order to distinguish our approach from the NCC classifier, it is referred to as the Nearest Class Vector (NCV) classifier hereafter. In order to overcome the unimodality assumption that is inherently set by all the NCC, including the proposed NCV, classifiers, we introduce an extension, namely Nearest Subclass Vector (NSV) classifier, which exploits multiple representations per class. Finally, since kernel methods have been found to be very effective in many Computer Vision tasks, including human action recognition [31, 30, 29, 18], we extend both the NCV and the NSV classifiers in order to determine an optimal data projection matrix and optimal class representations in arbitrary-dimensional Hilbert spaces [24].

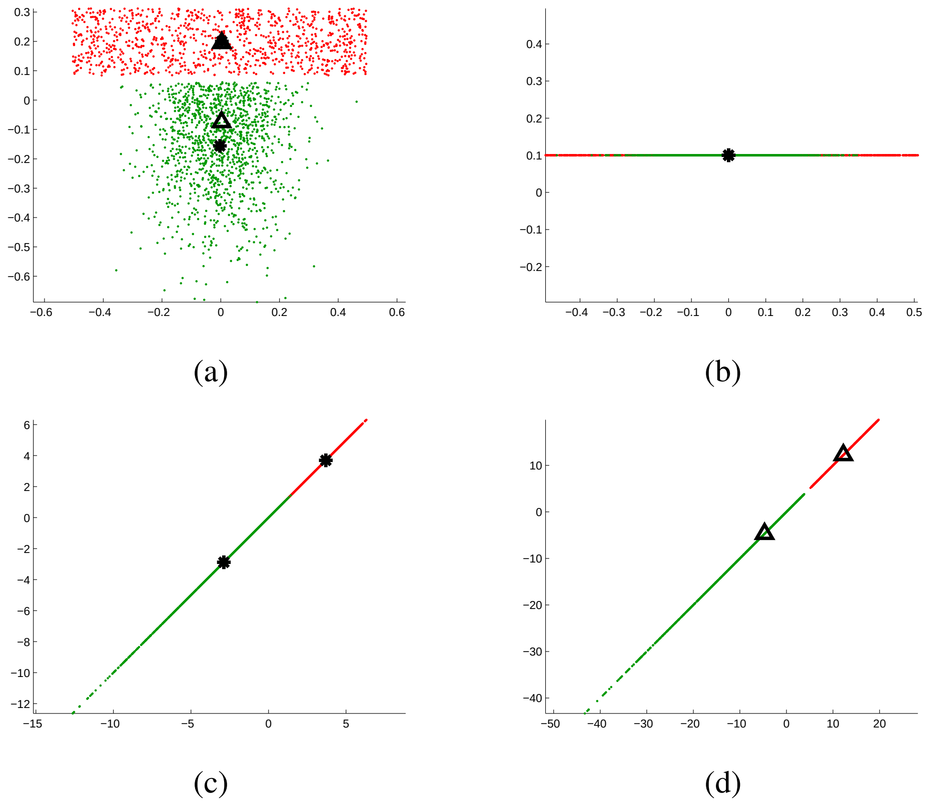

Figure 1: *a) two 2D classes having different probability class distributions and different covariance structures, b) projection space of LDA, c) projection space of logistic discrimination employing the class mean vectors (asterisks) for class representation and d) projection space of the proposed logistic discrimination scheme employing the optimal class vectors (triangles).*

We apply the proposed classification schemes on the Hollywood2 [19], Olympic sports [23] and the, recently introduced, ASLAN [6] datasets. As baseline approaches, we use the state-of-the-art methods proposed in [30, 6]: on the ASLAN dataset we employ a set of 12 similarity values calculated for histogram similarity measure between pairs of videos, represented by using the Bag of Words (BoW) model for HOG, HOF and HNF descriptors evaluated on STIP video locations [17]. This video pair similarity representation is employed for classification using a linear Support Vector Machine (SVM) classifier. We employ this baseline to evaluate the performance of the linear version of the proposed NCV and NSV classification schemes. For the remaining datasets we employ the BoW-based video representation by using HOG, HOF, MBH and Trajectory descriptors evaluated on the trajectories of densely sampled interest points [30]. Classification is performed by employing a kernel SVM classifier and the χ^2 kernel. We employ this baseline to evaluate the performance of the kernel version of the proposed NCV and NSV classification schemes.

The rest of the paper is structured as follows. We first discuss a selection of works related to this paper in Section 2. We describe the proposed metric learning algorithm for NCV classification in Sections 3. Extensions towards two directions, in order to exploit multiple representations per class and in order to operate in arbitrary-dimensional Hilbert spaces are presented in Subsections 3.1 and 3.2, respectively. Experimental results on human action recognition are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

Closely related to the proposed NCV classifier is the LDA algorithm and its variances [32, 14]. LDA determines an optimal discriminant subspace by adopting the between-class to within-class scatter ratio. LDA assumes unimodal class probability distributions having the same covariance structure and employs the mean class vectors for class representation. As has been discussed above, these are two strong assumptions that are difficult to be met in several real-world classification problems. A variant of LDA that tries to determine the optimal class representation for LDA-based data projection (in the linear case) is proposed in [8]. This idea has also been extended for nonlinear data projection in [11]. Our approach differs significantly in that: (i) we employ multi-class logistic discrimination for the determination of the data projection matrix and the optimal class representation. As it will be shown in Subsection 4.4, the adoption of logistic discrimination leads to increased performance compared to the criterion used in [8]. (ii) We extend the proposed NCV classification in order to exploit multiple representations per class.

Other works related to this paper include LESS [27], Taxonomy Embedding [34], the Sift-bag kernel [38], NCC classifier [22] and sample-to-class metric learning [33]. The LESS model [27], is used to learn a diagonal scaling matrix for the modification of the Euclidean distance by scaling the data dimensions and includes an l_1 penalty term in order to perform feature selection. Taxonomy Embedding [34] exploits a hierarchical cost function in order to map the samples to a lower dimensional feature space where each class is represented by the class mean vector. The Sift-bag kernel [38] determines a lower dimensional feature space that is orthogonal to the subspace with the maximal within-class variance, which is evaluated by employing the class mean vectors. The NCC classifier of [22] determines a Mahalanobis metric using the class mean vectors for class representation. Finally, the sample-to-class metric learning [33], learns a Mahalanobis metric by employing a Naive-Bayes Nearest Neighbor approach and, thus, requires the storage of all training samples, contrary to all the aforementioned methods (including the proposed one), where only some class vectors (the representative ones in the proposed method) are required for classification. Our approach differs significantly from these methods in that: (i) the proposed approach aims at determining both the optimal data projection matrix and the optimal class representation(s) for classification and (ii) the proposed approach is extended in order to operate in arbitrary-dimensional Hilbert spaces. As has been previously discussed, kernel methods have been proven effective in many Computer Vision tasks, including human action recognition.

3. The proposed NCV classifier

The proposed nearest class centroid (NCV) classifier assigns a sample $\mathbf{x}_i \in \mathbb{R}^D$ to the class $c^* \in \{1, \dots, C\}$ of the closest class vector:

$$c^* = \arg\min_{c} d(\mathbf{x}_i, \boldsymbol{\mu}_c), \tag{1}$$

where $\boldsymbol{\mu}_c \in \mathbb{R}^D$ is the representation of class c and may be any vector that enhances class discrimination, i.e., $\boldsymbol{\mu}_c$ is not necessarily the class mean vector. The adopted distance function, i.e.:

$$d(\mathbf{x}_i, \boldsymbol{\mu}_c) = (\mathbf{x}_i - \boldsymbol{\mu}_c)^T \mathbf{M} (\mathbf{x}_i - \boldsymbol{\mu}_c), \tag{2}$$

is the (squared) Mahalanobis distance between sample $\mathbf{x}_i$ and the class vector $\boldsymbol{\mu}_c$. We enforce $\mathbf{M}$ to be a symmetric metric, i.e. $\mathbf{M} = \mathbf{W}^T\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{d \times D}$. In the case where the dimensionality of the resulted feature space is lower than the sample dimension, $d < D$. By using $\mathbf{W}$, $d(\mathbf{x}, \boldsymbol{\mu}_c)$ can be written in the form:

$$\begin{aligned} d_{\mathbf{W}}(\mathbf{x}_i, \boldsymbol{\mu}_c) &= (\mathbf{x}_i - \boldsymbol{\mu}_c)^T \mathbf{W}^T \mathbf{W} (\mathbf{x}_i - \boldsymbol{\mu}_c) \\ &= \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\boldsymbol{\mu}_c\|_2^2. \end{aligned} \tag{3}$$

That is, $\mathbf{W}$ can be considered as a projection matrix used to map the data in a d-dimensional feature space, where classification is performed based on the minimal Euclidean distance from the class vectors.

We formulate the proposed NCV classifier by using a probabilistic model based on multi-class logistic regression. We define the conditional probability of class c given a sample vector $\mathbf{x}_i$ by:

$$p(c|\mathbf{x}_i) = \frac{e^{-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i,\boldsymbol{\mu}_c)}}{\sum_{l=1}^{C} e^{-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i,\boldsymbol{\mu}_l)}}. \tag{4}$$

Clearly, we would like to learn an appropriate set of $(\mathbf{W}^*, \boldsymbol{\mu}_c^*),\ c = 1, \dots, C$ that maximizes the probability of $p(y_i|\mathbf{x}_i)$, i.e., the probability to correctly classify all the training samples. y_i is used to denote the class label of training sample $\mathbf{x}_i$. In practice, it is convenient to maximize the mean log-likelihood of all the N training samples:

$$\mathcal{J}(\mathbf{W}, \boldsymbol{\mu}_c) = \frac{1}{N}\sum_{i=1}^{N} ln\, p(y_i|\mathbf{x}_i). \tag{5}$$

In the case where the distribution of training samples is not representative of the real class distributions, their contribution to $\mathcal{J}$ calculation can be appropriately weighted.

In order to learn both the optimal data projection matrix $\mathbf{W}^*$ and the optimal class vectors $\boldsymbol{\mu}_c^*$ we follow an Expectation Maximization-like iterative optimization approach. In the following, we introduce an additional index t denoting the t-th iteration of the adopted optimization scheme. For a given set of class vectors $\boldsymbol{\mu}_{c,t}$, we update the data projection matrix by following the gradient of $\mathcal{J}$ with respect to $\mathbf{W}$, i.e. $\mathbf{W}_{t+1} = \mathbf{W}_t + \eta_{\mathbf{W}}\nabla_{\mathbf{W}}\mathcal{J}$. By using $\mathbf{W}_{t+1}$ we, subsequently, update the class vectors by following the gradient of $\mathcal{J}$ with respect to $\boldsymbol{\mu}_c$, i.e., $\boldsymbol{\mu}_{c,t+1} = \boldsymbol{\mu}_{c,t} + \eta_{\boldsymbol{\mu}}\nabla_{\boldsymbol{\mu}_c}\mathcal{J}$. $\nabla_{\mathbf{W}}\mathcal{J}$, $\nabla_{\boldsymbol{\mu}_c}\mathcal{J}$ are given by:

$$\nabla_{\mathbf{W}}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big(p(c|\mathbf{x}_i) - \alpha_i^c\Big)\mathbf{W}\mathbf{q}_i^c\mathbf{q}_i^{c\,T}, \tag{6}$$

$$\nabla_{\boldsymbol{\mu}_c}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^c\Big(1 - p(c|\mathbf{x}_i)\Big)\mathbf{W}^T\mathbf{W}\mathbf{q}_i^c. \tag{7}$$

In (6,7), $\mathbf{q}_i^c = \boldsymbol{\mu}_c - \mathbf{x}_i$ and α_i^c is an index denoting if $\mathbf{x}_i$ belongs to class c, i.e., $\alpha_i^c = 1$ if $y_i = c$ and $\alpha_i^c = 0$ otherwise. $\eta_{\mathbf{W}}$ and $\eta_{\boldsymbol{\mu}}$ are the update rate parameters used to adapt $\mathbf{W}$ and $\boldsymbol{\mu}_{c,t}$, respectively. $\eta_{\mathbf{W}}$, $\eta_{\boldsymbol{\mu}}$ can either be a priori determined, e.g., $\eta_{\mathbf{W}} = 0.01$, $\eta_{\boldsymbol{\mu}} = 0.01$, or be dynamically determined.

In our experiments we have used an adaptive optimization process where $\eta_{\mathbf{W}}$, $\eta_{\boldsymbol{\mu}}$ are dynamically determined by following a linear search strategy. That is, in each iteration of the optimization process the criterion $\mathcal{J}$ is evaluated by using $\eta_{\mathbf{W},0} = 0.1$ (or $\eta_{\boldsymbol{\mu},0} = 0.1$). In the case where $\mathcal{J}_{t+1} > \mathcal{J}_t$, the criterion $\mathcal{J}$ is evaluated by using an update rate parameter equal to $\eta_{\mathbf{W},n+1} = 2\eta_{\mathbf{W},n}$ (or $\eta_{\boldsymbol{\mu},n+1} = 2\eta_{\boldsymbol{\mu},n}$). This process is followed until $\mathcal{J}_{t+1} < \mathcal{J}_t$ and the value providing the maximal increase in $\mathcal{J}$ is employed. In the case where, by using an update rate parameter equal to $\eta_{\mathbf{W},0} = 0.1$ (or $\eta_{\boldsymbol{\mu},0} = 0.1$), $\mathcal{J}_{t+1} < \mathcal{J}_t$, the criterion $\mathcal{J}$ is evaluated by using an update rate parameter equal to $\eta_{\mathbf{W},n+1} = \eta_{\mathbf{W},n}/2$ (or $\eta_{\boldsymbol{\mu},n+1} = \eta_{\boldsymbol{\mu},n}/2$). This process is followed until $\mathcal{J}_{t+1} > \mathcal{J}_t$ and the value increasing the criterion $\mathcal{J}$ is employed. We evaluate $\mathcal{J}$ after introducing all the training samples for the adaptation of $\mathbf{W}$, $\boldsymbol{\mu}_{c,t}$. However, (6,7) can be also employed by stochastic gradient ascent algorithms for the adoption of the proposed NCV in large-scale classification problems.

The above described iterative optimization scheme is performed until $(\mathcal{J}_{t+1} - \mathcal{J}_t)/\mathcal{J}_t < \epsilon$, where ϵ is a small positive value (equal to 10^{-8} in our experiments). We initialize the class representations to the class mean vectors, i.e., $\boldsymbol{\mu}_{c,1} = \mathbf{m}_c$, $c = 1, \ldots, C$, where $\mathbf{m}_c = \frac{1}{N_c}\sum_{i:y_i=c}\mathbf{x}_i$. For the initialization of the data projection matrix $\mathbf{W}$ one can employ either random projections [36], or PCA. In our preliminary experiments we have observed that the adoption of random projections generally outperforms the latter choice and, thus, we employ random projections in all our experiments.

3.1. Extension to multiple subclasses per class

For the case of multimodal classes, i.e., in the case where the classes forming the classification problem consist of multiple subclasses, we employ multiple class vectors for each class. In this case, the conditional probability of class c given a sample vector $\mathbf{x}_i$ is given by:

$$p(c|\mathbf{x}_i) = \sum_{j=1}^{C_c} p(c_j|\mathbf{x}_i) \quad (8)$$

$$p(c_j|\mathbf{x}_i) = \frac{e^{-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i,\boldsymbol{\mu}_{cj})}}{\sum_{l=1}^{C}\sum_{k=1}^{C_l} e^{-\frac{1}{2}d_{\mathbf{W}}(\mathbf{x}_i,\boldsymbol{\mu}_{lk})}}. \quad (9)$$

Here it is assumed that class c consists of C_c subclasses, each represented by the corresponding subclass vector $\boldsymbol{\mu}_{cj}$, $j = 1, \ldots, C_c$. Again, $\boldsymbol{\mu}_{cj}$ are not necessarily the subclass mean vectors, but are adapted to enhance class discrimination. Classification is performed by assigning the sample $\mathbf{x}_i$ to the class c^* providing the maximal conditional probability:

$$c^* = \arg\max_c p(c|\mathbf{x}_i), \tag{10}$$

In order to learn both the optimal data projection matrix $\mathbf{W}^*$ and the optimal subclass vectors $\boldsymbol{\mu}_{cj}^*$, we also apply the iterative optimization process described in Section 3. In this case the gradients of $\mathcal{J}$ with respect to $\mathbf{W}$ and $\boldsymbol{\mu}_{cj}$ are given by:

$$\nabla_{\mathbf{W}}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\sum_{j=1}^{C_c}\Big(p(c_j|\mathbf{x}_i) - \alpha_i^c\beta_i^{cj}\Big)\mathbf{W}\mathbf{q}_i^c\mathbf{q}_i^{cj\,T}, \tag{11}$$

$$\nabla_{\boldsymbol{\mu}_{cj}}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^c\beta_i^{cj}\Big(1 - p(c_j|\mathbf{x}_i)\Big)\mathbf{W}^T\mathbf{W}\mathbf{q}_i^{cj}, \tag{12}$$

where $\mathbf{q}_i^{cj} = \boldsymbol{\mu}_{cj} - \mathbf{x}_i$ and $\beta_i^{cj} = \frac{p(c_j|\mathbf{x}_i)}{\sum_{l=1}^{C_c} p(c_l|\mathbf{x}_i)}$. That is, each training sample $\mathbf{x}_i$ contributes to the adaptation of $\boldsymbol{\mu}_{cj}$ according to its membership value β_i^{cj}. Since the subclasses are not a priori known, in order to initialize the subclass vectors $\boldsymbol{\mu}_{cj}$ we cluster the training samples $\mathbf{x}_i$ belonging to class c. We have adopted K-Means algorithm to this end.

The above described NCV and NSV classifiers operate on the data feature space $\mathbb{R}^D$ in order to determine an optimal linear projection and optimal class representations for classification. An extension of NCV and NSV algorithms in the kernel case is described in the following Subsection.

3.2. Extension to the kernel case

In order to extend the proposed NCV classifier to the kernel case, the original input space is mapped to an arbitrary-dimensional space $\mathcal{F}$. The space $\mathcal{F}$ usually has the structure of a Hilbert space [24]. To do so, let $\phi : \mathbf{x}_i \in \mathbb{R}^D \rightarrow \phi(\mathbf{x}_i) \in \mathcal{F}$ be a non-linear mapping from the input space $\mathbb{R}^D$ to the space $\mathcal{F}$. In this space, we want to find an optimal linear projection to a low-dimensional feature space and the corresponding optimal class representations that will be used for classification. In this case, the adopted distance function used for the calculation of the conditional class probabilities is given by:

$$d_{\mathbf{W}_\phi}\Big(\phi(\mathbf{x}_i), \phi(\boldsymbol{\mu}_c)\Big) = \|\mathbf{W}_\phi\phi(\mathbf{x}_i) - \mathbf{W}_\phi\phi(\boldsymbol{\mu}_c)\|_2^2, \tag{13}$$

where $\mathbf{W}_\phi$ is a data projection matrix that will be used to map the samples from $\mathcal{F}$ to a low-dimensional feature space with enhanced discrimination power $\mathbb{R}^d$.

However, since $\mathbf{W}_\phi$ is a matrix of arbitrary (even infinite) dimensions, the distance in (13) cannot be directly calculated.

By expressing $\mathbf{W}_\phi$ and $\phi(\boldsymbol{\mu}_c)$ as linear combinations of the training vectors (represented in $\mathcal{F}$) [24], i.e., $\mathbf{W}_\phi = \mathbf{\Phi}\mathbf{A}$ and $\phi(\boldsymbol{\mu}_c) = \mathbf{\Phi}\mathbf{b}_c$, where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$ and $\mathbf{A} \in \mathbb{R}^{N \times d}$, $\mathbf{b}_c \in \mathbb{R}^N$ are a matrix and a vector containing the reconstruction weights for $\mathbf{W}_\phi$ and $\phi(\boldsymbol{\mu}_c)$ respectively, Equation (13) can be written in the form:

$$\begin{aligned} d_{\mathbf{A}}\Big(\phi(\mathbf{x}_i), \phi(\boldsymbol{\mu}_c)\Big) &= \|\mathbf{A}^T\mathbf{\Phi}^T\phi(\mathbf{x}_i) - \mathbf{A}^T\mathbf{\Phi}^T\phi(\boldsymbol{\mu}_c)\|_2^2 \\ &= \|\mathbf{A}^T\mathbf{k}_i - \mathbf{A}^T\mathbf{K}\mathbf{b}_c\|_2^2. \end{aligned} \tag{14}$$

Here $\mathbf{K}$ is the kernel matrix, having elements equal to $k_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j),\ i,j = 1,\ldots,N$, and $\mathbf{k}_i$ is the i-th column of $\mathbf{K}$, having elements equal to $k_{ji} = \phi(\mathbf{x}_j)^T\phi(\mathbf{x}_i)$, $j = 1,\ldots,N$.

By observing Equations (3,4,5,14), it can be seen that the problem to be solved has been transformed to the determination of the reconstruction weights $\mathbf{A}^*$ and $\mathbf{b}_c^*$ for optimal non-linear data projection and optimal class representation, respectively. In this case, the gradient of $\mathcal{J}$ with respect to $\mathbf{A}$ is given by:

$$\nabla_{\mathbf{A}}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big(p(c|\phi(\mathbf{x}_i)) - \alpha_i^c\Big)\mathbf{K}(\mathbf{z}_i^c\mathbf{z}_i^{c\,T})\mathbf{K}\mathbf{A}, \tag{15}$$

while the gradient of $\mathcal{J}$ with respect to $\mathbf{b}_c$ is given by:

$$\nabla_{\mathbf{b}_c}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^c\Big(1 - p(c|\phi(\mathbf{x}_i))\Big)\mathbf{K}\mathbf{A}\mathbf{A}^T\mathbf{K}\mathbf{z}_i^c. \tag{16}$$

In (15,16), $\mathbf{z}_i^c = \mathbf{b}_c - \mathbf{1}_i$, where $\mathbf{1}_i$ is a vector having all its elements equal to zero, except of the i-th element, which is equal to one. The class representations are initialized to the class mean vectors in $\mathcal{F}$. That is, $\mathbf{b}_c$ is initialized by setting all its elements equal to zero, expect of the elements corresponding to the training samples belonging to class c which are set equal to $1/N_c$, where N_c is the number of training samples belonging to class c. For the initialization of $\mathbf{A}$ we, also, employ random projections. An alternative could be the use of kernel PCA.

By using the same analysis for the NSV classifier, $\mathbf{A}$ and $\mathbf{b}_{c_j}$ are updated by using the following gradients:

$$\nabla_{\mathbf{A}}\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\sum_{j=1}^{C_c}\left(p(c_j|\phi(\mathbf{x}_i)) - \alpha_i^c\beta_i^{cj}\right)\mathbf{K}(\mathbf{z}_i^{cj}\mathbf{z}_i^{cj\,T})\mathbf{K}\mathbf{A}, \tag{17}$$

$$\nabla_{\mathbf{b}_{cj}} \mathcal{J} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i^c \beta_i^{cj} \Big(1 - p(c_j|\phi(\mathbf{x}_i))\Big) \mathbf{K}\mathbf{A}\mathbf{A}^T \mathbf{K} \mathbf{z}_i^{cj}. \tag{18}$$

$\mathbf{z}_i^{cj} = \mathbf{b}_{cj} - \mathbf{1}_i$ and $\beta_i^{cj} = \frac{p(c_j|\phi(\mathbf{x}_i))}{\sum_{l=1}^{C_j} p(c_l|\phi(\mathbf{x}_i))}$. That is, each training sample $\mathbf{x}_i$ contributes to the adaptation of $\mathbf{b}_{cj}$ according to its membership value β_i^{cj}, evaluated on $\mathcal{F}$.

Similarly to the linear case, since the subclasses are not a priori known, $\mathbf{b}_{cj}$ are initialized by applying clustering on the training data belonging to class c. However, in this case clustering should be performed on the training data representations in $\mathcal{F}$. We employ kernel K-Means [25] to this end, by using the kernel matrix of the training samples belonging to each class separately.

Here we should note that the choice of the kernel function $\phi(\cdot)$ is important for the performance of kernel-based classification schemes. The usual approach employs a kernel function, like the Radial Basis Function (RBF), and determines the values of the corresponding parameters, e.g., the value of the standard deviation σ in the RBF case, by employing performance criteria on a validation set, or by performing cross-validation on the training set. Another approach tries to learn an optimized kernel function $\phi^*(\cdot)$, leading to an optimized kernel matrix $\mathbf{K}^*$, like in [13, 16]. This is usually approached by expressing $\mathbf{K}^*$ as a linear combination of a set of pre-defined kernels, i.e., by setting $\mathbf{K}^* = \sum_k \theta_k \mathbf{K}_k$, and trying to determine an optimized vector $\boldsymbol{\theta}^*$ which is used to optimally weight the contribution of $\mathbf{K}_k$ in $\mathbf{K}^*$. In our experiments, we have adopted the multi-channel RBF-χ^2 kernel function, as will be discussed in subsection 4.4, which has been shown to be the state-of-the-art choice for BoW-based action recognition [29, 18, 37].

4. Experiments

In this section we present experiments conducted in order to evaluate the performance of the proposed classification schemes. We have employed three publicly available datasets, namely the ASLAN, the Olympic Sports and the Hollywood2 datasets. In the following, we describe the datasets and evaluation protocols used in our experiments. Experimental results are provided in subsection 4.4.

4.1. The ASLAN dataset

The Action Similarity Labeling (ASLAN) dataset [6] consists of thousands of videos collected from the web, in over 400 complex action classes. A “same/not-same” benchmark is provided, which addresses the action recognition problem as

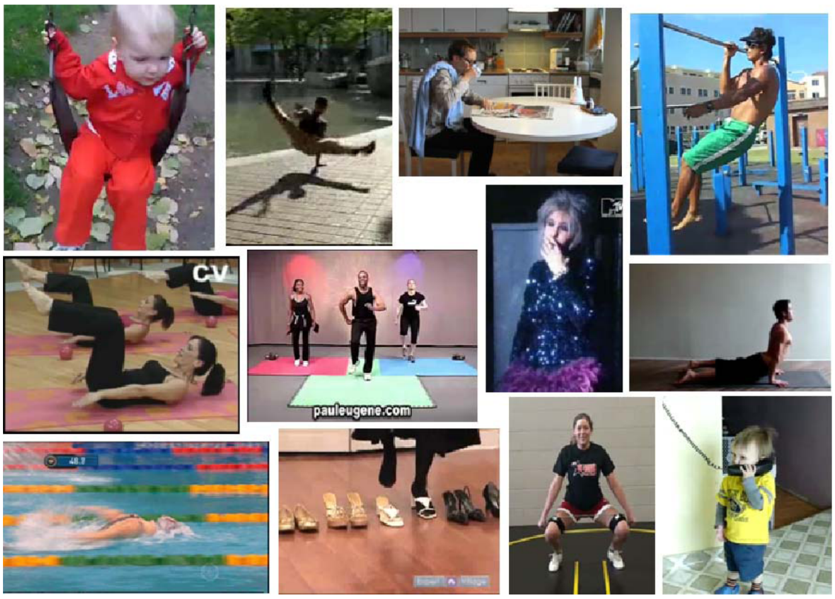

Figure 2: Video frames of the ASLAN dataset.

a video pair similarity problem. Specifically, the goal is to answer the following binary question: “Does a pair of videos depict the same action?”. Example video frames of the dataset are illustrated in Figure 2. We use the standard partitioning provided by the database. The dataset consists of ten splits of video pairs, each containing 300 pairs of same actions and 300 pairs of not-same actions. The splits contain mutually exclusive action classes. That is, action classes appearing in one split do not appear in any other split. Performance is evaluated by applying the ten-fold cross-validation procedure. In each fold, nine of the splits are used to train the algorithms and performance is measured on the remaining one. An experiment consists of ten folds, one for each test split, and performance is calculated by using the mean accuracy and the standard error of the mean (SE) over all folds.

4.2. The Olympic Sports dataset

The Olympic Sports dataset [23] consists of 783 videos depicting athletes practicing 16 sports, which have been collected from YouTube and annotated using Amazon Mechanical Turk. The actions appearing in the dataset are: ‘high jump’, ‘long jump’, ‘triple jump’, ‘pole vault’, ‘basketball lay-up’, ‘bowling’, ‘tennis serve’, ‘platform’, ‘discus’, ‘hammer’, ‘javelin’, ‘shot put’, ‘springboard’, ‘snatch’, ‘clean-jerk’ and ‘vault’. Example video frames of the dataset are illustrated in Figure 3. The dataset has rich scene context information, which is helpful for recognizing sport actions. We use the standard training-test split provided by the database (649 videos are used for training and performance is measured in the remaining 134 videos). The performance is evaluated by computing the mean Average Precision (mAP) over all classes, as suggested in [23]. In addition, since each videos depicts only one action, the mean classification rate can also be used for evaluation.

Figure 3: Video frames of the Olympic Sports dataset.

4.3. The Hollywood2 dataset

The Hollywood2 dataset [19] consists of 1707 videos depicting 12 actions. It has been collected from 69 different Hollywood movies. The actions appearing in the dataset are: ‘answering the phone’, ‘driving car’, ‘eating’, ‘ghting’, ‘getting out of car’, ‘hand shaking’, ‘hugging’, ‘kissing’, ‘running’, ‘sitting down’, ‘sitting up’, and ‘standing up’. Example video frames of the dataset are illustrated in Figure 4. We use the standard training-test split provided by the database (823 videos are used for training and performance is measured in the remaining 884

videos). Training and test videos come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP), as suggested in [19]. This is due to the fact that some videos of the dataset depict multiple actions.

Figure 4: Video frames of the Hollywood2 dataset depicting instances of all the twelve actions.

4.4. Experimental Results

Table 1 illustrates the mean accuracy and the standard error values obtained by applying the proposed NCV classifier on the ASLAN dataset. In our experiments we have employed the similarity vectors provided by the database. In Table 1, we also provide the mean accuracy and standard error values obtained by applying classification using linear SVM (we report the best performance obtained for $C = 10^{-6,\ldots,6}$), K-Nearest Neighbors (we report the best performance obtained for $K = 1, \ldots, 15$), Logistic Regression using the mean class vectors for class representation [22] (referred to as NCC), LDA and the method proposed in [8] for the determination of the optimal class representation in the LDA case (referred to as RCVLDA). It can be seen that NCC outperforms LDA and K-NN in all the cases, while SVM outperforms both K-NN, LDA and NCC in all cases. The determination of the optimal class representation for the LDA case leads to an increase of the performance of LDA. Specifically, it can be seen that, RCVLDA outperforms LDA and NCC in all the cases, while it outperforms SVM in three out of four cases. Finally, it can be seen that the proposed NCV algorithm clearly outperforms SVM, LDA and NCC in all cases, while it outperforms RCVLDA in three out of four cases. Overall, the proposed NCV algorithm provides the best

Table 1: Performance (Accuracy ± SE) on the ASLAN dataset.

	HOG	HOF	HNF	ALL
SVM	57.78 ± 0.82 %	56.68 ± 0.56 %	59.47 ± 0.66 %	60.88 ± 0.77 %
K-NN	52.58 ± 0.67 %	52.32 ± 0.98 %	52.63 ± 0.81 %	53.35 ± 1.05 %
LDA	50.33 ± 0.38 %	50.28 ± 0.27 %	49.82 ± 0.31 %	51.20 ± 0.43 %
NCC	56.83 ± 0.98 %	55.83 ± 0.73 %	57.83 ± 0.93 %	60.08 ± 0.92 %
RCVLDA	59.70 ± 0.91 %	**56.93 ± 0.63** %	59.17 ± 0.72 %	60.95 ± 0.81 %
NCV	**59.95 ± 0.6** %	56.58 ± 0.81 %	**60.08 ± 0.68** %	**61.4 ± 0.82** %

performance, equal to 61.4% (for $d = 5$), by concatenating the similarity values of all descriptor types provided by dataset. We have also applied the proposed NSV algorithm for different values of $C_c = \{2, 5, 10, 15, 20\}$. By using 10 subclasses per class, the proposed NSV algorithm further increases the performance to 61.66%. The performance obtained for values of $C_c = \{15, 20\}$ was slightly lower.

We evaluate the performance of the kernel version of the proposed NCV algorithm on the Olympic Sports and Hollywood2 datasets. In our experiments we have employed the improved version of dense trajectory-based video representation proposed in [30]. Since the BoW model is usually combined with kernel classification schemes, we have employed the BoW-based video representation for the evaluation of the kernel version of NCV classifier. We employ the pipeline used in [30] and constructed one codebook for each descriptor type ($K = 4000$). We adopted the RBF-χ^2 kernel, where different video representations are combined in a multi-channel approach [37]:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = exp\Big(- \sum_k \frac{1}{4A^k} D\left(\mathbf{x}_i^k, \mathbf{x}_j^k\right) \Big), \tag{19}$$

$D\left(\mathbf{x}_i^k, \mathbf{x}_j^k\right)$ is the χ^2 distance between the BoW-based video representation of $\mathbf{x}_i$ and $\mathbf{x}_j$ with respect to the k-th channel. A^k is the mean value of the χ^2 distances between the training samples for the k-th channel.

Table 2 illustrates the classification rates obtained by using different values of d. As can be seen, the use of a higher d value increases performance. It can also be seen in Table 2 that even the use of smaller values leads to satisfactory performance. The highest performance, equal to 67.16%, has been obtained by

Table 2: Mean classification rates on the Olympic Sports dataset for different target dimensions d.

50	100	200	300	400
63.41 %	64.18 %	65.67 %	**67.16** %	**67.16** %

Table 3: Mean classification rates on the Olympic Sports dataset.

SVM	KDA	KRDA	**NCV**	**NSV**
65.67 %	61.94 %	64.92 %	**67.16** %	**67.91** %

using the values $d = 300$ and $d = 400$. Thus, both these values are good choices for the Olympic Sports dataset. Expecting that for more complex datasets (like Hollwyood2) a high number of target space dimensions will probably provide satisfactory performance, we use the value $d = 400$ in all the remaining experiments.

In Table 3 we provide the mean classification rates obtained by using kernel SVM (we report the best performance obtained for $C = 10^{-6,\dots,6}$), Kernel Discriminant Analysis (KDA) [1] and Kernel Reference Discriminant Analysis (KRDA) [11] on the Olympic Sports dataset. We have also tested the performance of the kernel K-NN classifier [39], but its performance was far inferior to the remaining ones and, thus, we omit reporting it. As can be seen, the proposed NCV classifier outperforms the other three classification schemes. We have also applied the proposed NSV algorithm for different values of $C_c = \{2, 5, 10, 15, 20\}$. By using ten subclasses per action class, the proposed NSV classifier was able to outperform NCV, providing a classification rate equal to 67.91%.

Table 4 illustrates the mean average precision values obtained by applying the kernel version of the proposed NCV algorithm on the Olympic Sports and the Hollywood2 datasets. In this Table, we also provide the mean average precision values obtained by using kernel SVM (we report the best performance obtained for $C = 10^{-6,\dots,6}$), Minimum Class Variance Extreme Learning Machine (MCVELM) [7] (we report the best performance obtained for $C = 10^{-6,\dots,6}$), KDA and KRDA on the two datasets. We also provide the performance reported in [30], when employing the same video representation and the kernel SVM classifier. Finally, we have implemented a version of the proposed NCV classifier which learns only the data projection matrix $\mathbf{A}^*$ based on the class means in $\mathcal{F}$ (noted as NCC). As can be seen, NCC is quite effective, since it achieves performance similar to that of SVM, ELM and KDA. The proposed approach by learning

Table 4: Performance (mAP) on the Olympic Sports and Hollywood2 datasets.

	Olympic Sports	Hollywood2
SVM [30]	83.3 %	62.2 %
SVM (reproduced)	82.7 %	61.51 %
MCVELM	**86.07** %	58.66 %
KDA	81.27 %	59.06 %
KRDA	83.35 %	61.2 %
NCC	80,6 %	55.93 %
NCV	**84.14** %	**59.5** %
NSV	**85.49** %	**62.5** %

both the data projection matrix $\mathbf{A}^*$ and the class representations $\mathbf{b}_c$ enhances the performance of NCC. The proposed NCV algorithm outperforms both SVM and KDA in Olympic Sports dataset providing a mAP value equal to 84.14%. On the Hollywood2 dataset, the proposed NCV algorithm achieves a performance similar to that of the KDA classification scheme, while its performance is inferior to that of the SVM classifier. When compared to the MCVELM classifier, the performance of the proposed approach is inferior on the Olympic Sports dataset, while it ourperforms MCVELM on the Hollywood2 dataset. By using ten subclasses per action class, the proposed NSV classifier was able to outperform NCV, providing mean average precision values equal to 85.49% and 62.5% for the Olympic Sports and the Hollywood2 datasets, respectively.

In Table 5, we compare the performance of the adopted action recognition method with that of some other state-of-the-art methods evaluating their performance on Olympic Sports and Hollywood2 datasets. As can be seen, the proposed NCV and NSV algorithms, when combined with the improved trajectory-based video representation achieves satisfactory performance in both datasets.

Overall, it can be seen that distance-based classification exploiting optimized class representation(s), when approached by a probabilistic point of view, is a powerful approach which is able to provide comparable (or even better) performance with that of other state-of-the-art choices, like the SVM, MCVELM and LDA (KDA) classification schemes. When compared to the SVM and MCVELM choices, the adoption of the proposed method has the advantage that it leads to a lower-dimensional data representation preserving discriminant class information and, thus, compact data representations can be obtained. In addition, relevant

Table 5: Comparison of our results with some state-of-the-art methods on the Olympic Sports and Hollywood3 datasets.

	Olympic Sports	Hollywood2
Iosifidis et al. [9]	-	45.8 %
Brendel et al. [2]	77.33 %	-
Vig et al. [28]	-	61.9 %
Gaidon et al. [4]	82.7 %	-
Mathe et al. [20]	-	61 %
Jiang et al. [15]	80.6 %	59.5 %
Jain et al. [12]	83.2 %	**62.5** %
Iosifidis et al. [10]	82.12 %	58.2 %
Iosifidis et al. [11]	83.35 %	61.2 %
NCV+Improved Trajectories	**84.14** %	59.5 %
NSV+Improved Trajectories	**85.49** %	**62.5** %

work in image classification [22] denotes that the adoption of the NCC classification approach is able to outperform SVM-based classification in large-scale classification problems. The proposed approach by optimizing NCC with respect to both the data projection matrix and the class representation is expected to enhance its performance. When compared to the LDA (and KDA) approaches, the proposed method by adopting logistic discrimination overcomes the assumptions set by such methods. In addition, experimental results show that the proposed approach is able to achieve better performance.

5. Conclusion

In this paper we proposed a new metric learning for distance-based action classification. The proposed approach maximizes the log-likelihood of correct class prediction, which is calculated in a low-dimensional feature space of increased discrimination power by using optimized class representations. We have illustrated the superiority of the proposed approach to metric learning approaches employing the class mean for class representation. The proposed NCV classifier has been extended in order to exploit multiple representations per action class, as well as to operate in arbitrary-dimensional Hilbert spaces for non-linear data projection and classification. Experimental results on three action recognition datasets denote that the proposed classification scheme is able to enhance action recognition performance, when compared to the standard NCC approach, and provide comparable (or better) performance with that of the SVM (in both the linear and kernel cases) which is the current state-of-the-art choice for human action recognition.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

[1] Baudat, G., Anouar, F., 2000. Generalized discriminant analysis using a kernel approach. Neural Computation 12, 2385–2404.

[2] Brendel, W., Todorovic, S., 2011. Learning spatiotemporal graphs of human activities. International Conference on Computer Vision .

[3] Duda, R., Hart, P., Stork, D., 2000. Pattern classification. Wiley .

[4] Gaidon, A., Harchaoui, Z., Schmid, C., 2012. Recognizing activities with cluster-tries of tracklets. British Machine Vision Conference .

[5] Gross, O., Hassner, T., Wolf, L., 2011. The one shot similarity metric learning for action recognition. Similarity-Based Pattern Analysis and Recognition , 31–45.

[6] Gross, O., Hassner, T., Wolf, L., 2013. The action similarity labeling challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 615–621.

[7] Iosifidis, A., Tefas, A., Pitas, I., 2013a. Minimum class variance extreme learning machine for human action recognition. IEEE Transactions on Circuits and Systems for Video Technology 23, 1968–1979.

[8] Iosifidis, A., Tefas, A., Pitas, I., 2013b. On the optimal class representation in linear discriminant analysis. IEEE Transactions on Neural Networks and Learning Systems 24, 1491–1497.

[9] Iosifidis, A., Tefas, A., Pitas, I., 2014a. Discriminant bag of words based representation for human action recognition. Pattern Recognition Letters 49, 185–192.

[10] Iosifidis, A., Tefas, A., Pitas, I., 2014b. Human action recognition based on bag of features and multi-view neural networks. IEEE International Conference on Image Processing .

[11] Iosifidis, A., Tefas, A., Pitas, I., 2014c. Kernel reference discriminant analysis. Pattern Recognition Letters 49, 85–91.

[12] Jain, M., Jegou, H., Bouthemy, P., 2013. Better exploiting motion for better action recognition. Computer Vision and Pattern Recognition .

[13] Jain, P., Kulis, B., Dhillon, I., 2010. Inductive regularized learning of kernel functions. Neural Information Processing Systems .

[14] Jia, Y., Nie, F., Zhang, C., 2009. Trace ratio problem revisited. IEEE Transactions on Neural Networks 20, 729–735.

[15] Jiang, Y., Dai, Q., Xue, X., Liu, W., Ngo, C., 2012. Trajectory-based modeling of human actions with motion reference points. European Conference on Computer Vision .

[16] Kim, S., Magani, A., Boyd, S., 2006. Optimal kernel selection in kernel fisher discriminant analysis. International Conference on Machine Learning .

[17] Laptev, I., 2005. On space-time interest points. International Journal of Computer Vision 64, 107–123.

[18] Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies. Computer Vision and Pattern Recognition .

[19] Marszalek, M., Laptev, I., Schmid, C., 2009. Actions in context. Computer Vision and Pattern Recognition .

[20] Mathe, S., Sminchisescu, C., 2012. Dynamic eye movement datasets and learnt saliency models for visual action recognition. European Conference on Computer Vision .

[21] Mensink, T., Verbeek, J., Perronnin, F., Csurka, G., 2012. Metric learning for large scale image classication: Generalizing to new classes at near-zero cost. European Conference on Computer Vision .

[22] Mensink, T., Verbeek, J., Perronnin, F., Csurka, G., 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 2624–2637.

[23] Niebles, J., Chend, C., Fei-Fei, L., 2010. Modeling temporal structure of decomposable mition segemnts for activity classification. European Conference on Computer Vision .

[24] Scholkpf, B., Smola, A., 2001. Learning with kernels. MIT Press.

[25] Taylor, J., Cristianini, N., 2004. Kernel methods for pattern analysis. Cambridge University Press.

[26] Tran, D., Sorokin, A., 2008. Human activity recognition with metric learning. European Conference on Computer Vision .

[27] Veenman, C., Tax, D., 2005. Less: a model-based classifier for sparse subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1496–1500.

[28] Vig, E., Dorr, M., Cox, D., 2012. Space-variant descriptor sampling for action recognition based on saliency and eye movements. European Conference on Computer Vision .

[29] Wang, H., Klaser, A., Schmid, C., Liu, C., 2011. Action recognition by dense trajectories. Computer Vision and Pattern Recognition .

[30] Wang, H., Schmid, C., 2011. Action recognition with improved trajectories. International Conference on Computer Vision .

[31] Wang, H., Ullah, M., Klaser, A., Laptev, I., Schmid, C., 2009. Evaluation of local spatio-temporal features for action recognition. British Machine Vision Conference .

[32] Wang, H., Yan, S., Xu, D., Tang, X., Huang, T., 2007. Trace ratio vs. ratio trace for dimensionality reduction. Computer Vision and Pattern Recognition .

[33] Wang, Z., Hu, Y., Chia, L., 2010. Image-to-class distance metric learning for image classification. European Conference on Computer Vision .

[34] Weinberger, K., Chapelle, O., 2009. Large margin taxonomy embedding for document categorization. Neural Information Processing Systems .

[35] Wolf, L., Hassner, T., Taigman, Y., 2009. The one-shot similarity kernel. International Conference on Computer Vision , 897–902.

[36] Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y., 2009. Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 210–227.

[37] Zhang, J., Marszalek, M., Lazebnik, M., Schmid, C., 2007. Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision 73, 213–238.

[38] Zhou, X., Zhang, X., Yan, Z., Chang, S., Jonhson, M., Huang, T., 2008. Sift-bag kernel for video event analysis. ACM Multimedia .

[39] Zuo, W., Zhang, D., Wang, K., 2008. On kernel difference-weighted k-nearest neighbor classification. Pattern Analysis and Applications 11, 247–257.