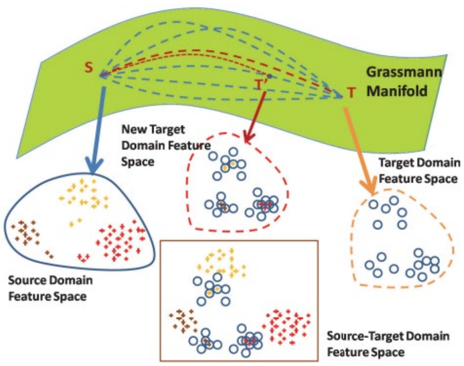
Title:	Recognizing Activities Under Domain Shift
Author(s):	Jamal, Arshad
Supervisor(s):	Venkatesh, K S Namboodiri, Vinay P Deodhare, Dipti
Keyword(s):	Domain Adaptation Action Recognition Deep Learning Subspace Learning
Subject(s):	Computer Vision

Abstract:

In the general settings of supervised learning, visual recognition has been extensively studied. In this setting, the models are learned under the assumption that the training and test data are sampled from the same underlying probability distribution. However, in most of the practical scenarios, this assumption is not true, resulting in a sub-optimal performance of the trained models. This problem, referred to as {\em Domain Shift}, has been extensively studied for the object classification task. However, this problem has not been as well studied for recognizing actions in videos. While, object recognition is well understood, the diverse variety of videos in action recognition make the task of addressing domain shift to be more challenging. An approach that has worked well for the image domain adaptation (DA) is based on the subspace modeling of the source and target domains, which works under the assumption that the two domains share a latent subspace where the domain shift can be reduced or eliminated. In our first contribution in this thesis, we propose a new eclectic domain mixing (EDM) approach, which reduces the domain discrepancy between the source and target domains and illustrate its effectiveness in adaptation by combining with two subspace based domain adaptation (DA) methods. Further, we use discrepancy measures such as {\em Symmetrized KL Divergence} and {\em Target Density Around Source} for empirical study of the proposed EDM approach. In our second contribution, we propose another subspace based method, named Action Modeling on Latent Subspace (AMLS). In the AMLS approach, the action videos in the target domain are modeled as a sequence of points on a latent subspace and adaptive kernels are successively learned between the source domain point and the sequence of target domain points on the latent subspace. It has been shown that the AMLS approach is better than the two subspace based methods. In our third contribution, we propose a deep network based adaptation techniques that we term as deep action domain adaptation (DADA). The main concept that we propose is that of explicitly modeling density based adaptation and using them while adapting domains for recognizing actions. We show that these techniques work well both for domain adaptation through adversarial learning to obtain invariant features or explicitly reducing the domain shift between distributions. It has been shown that the deep domain adaptation method outperforms all the subspace based methods. In our last contribution, we explore application areas for the domain adaptation methods developed in this thesis and choose video retrieval task to demonstrate the effectiveness of the proposed domain adaptation methods. We develop a pipeline for recognition-free action retrieval and compare its performance with recognition-based retrieval. Further, we propose a query expansion mechanism to

improve the retrieval performance for the recognition-free approach. In the end, we show out-of-vocabulary retrieval, which is a special case of retrieval for action classes not present in the training set. All the action domain adaptation experiments require multi-domain datasets with semantically similar classes in each of them. As there are no cross-domain benchmark dataset for action adaptation, we created our own multi-domain dataset collection using appropriate classes from UCF50, UCF101, HMDB51, Olympic Sports, KTH, MSR Action II and our own six-class SonyCam datasets. Specifically, we have created four multi-domain datasets, including (i) \textbf{UO} comprising six common classes of UCF50 and Olympic Sports datasets, (ii) \textbf{HU} comprising five common classes of HMDB51 and UCF50, (iii) \textbf{OH_UCF101} comprising eighteen common classes of UCF101 and Olympic Sports/HMDB51 and (iv) \textbf{KMS} comprising all the classes of KTH, MSR Action II and Sony Cam. As a pioneering effort in the area of action adaptation, we are presenting several benchmark results and techniques that could serve as baselines to guide future research in this area.



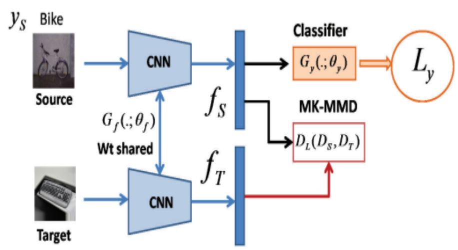


Figure 1.2: Network architecture for distribution matching based object domain adaptation.

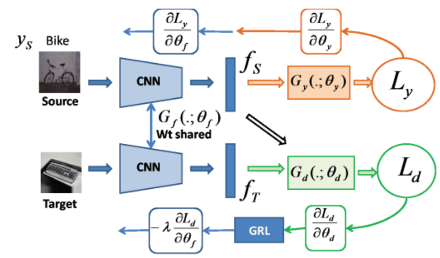


Figure 1.3: Network Architecture for Adversarial Learning.

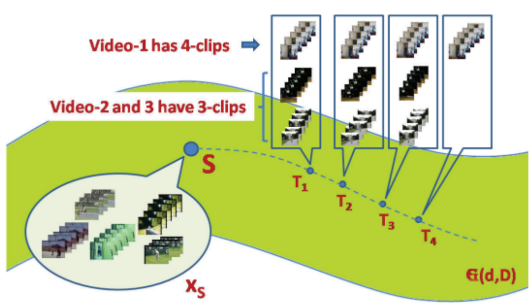


Figure 4.1: Concept of the proposed Action Modeling on Latent Subspace (AMLS) approach. For all the clips in the source domain, a subspace representation, shown as point $\mathbf{S} \in \mathbb{R}^{D \times d}$ on a Grassmann manifold $\mathcal{G}(D, d)$, is obtained using the Principal Component Analysis (PCA) method. Similarly, the target domain action videos are modeled as a sequence of points ($\mathbf{T}_i \in \mathbb{R}^{D \times d}$) on the manifold, where each point corresponds to a collection of features from each video as shown in the figure. In the end, adaptive kernels are learned between the successive points, which are then used for action domain adaptation.

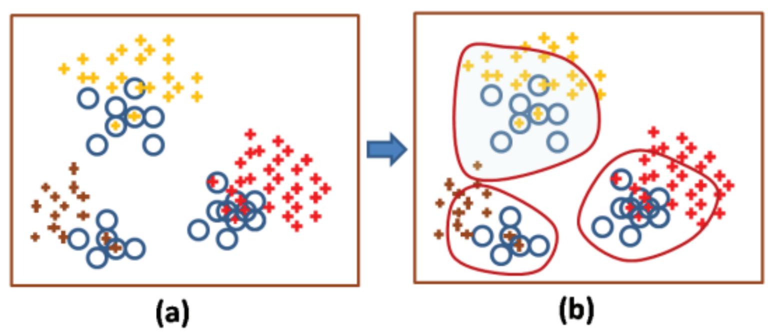


Figure 5.1: Conceptual diagram of source sample selection to maximize positive transfer and minimize negative transfer. The samples of three classes from source domain are shown in three colors with '+' symbols. The unlabelled target domain samples are shown with 'o' symbol. Fig (b) shows the selected source samples. *Best viewed in color.*

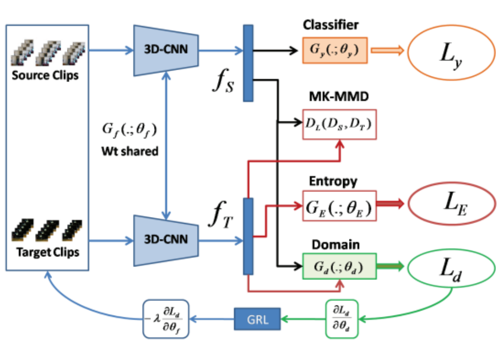


Figure 5.2: Architecture of the proposed Deep Action Domain Adaptation (DADA) network. $G_f(.;\theta_f)$ is the feature mapping function, $G_y(.;\theta_y)$ is the class discriminator function, $G_E(.;\theta_E)$ is the entropy function and $G_d(.;\theta_d)$ is the domain discriminator function. L_y, L_E and L_d are the corresponding loss functions. $D_L(D_S, D_T)$ is the distribution matching function. GRL is gradient reversal layer. Other back-propagation layers have been omitted for simplicity. *Best viewed in color.*

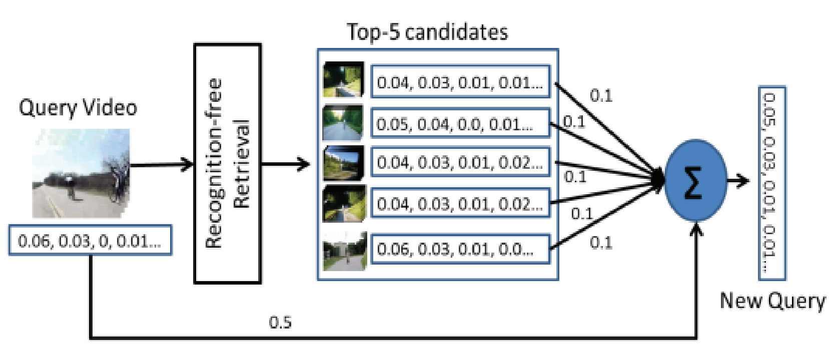


Figure 6.1: Query expansion pipeline