ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

DA-AIM: Action Instance Mixing for Domain-adaptive Action Detection

Master's Thesis

Yifan Lu
Department of Information Technology and Electrical Engineering
ETH Zurich

Advisors: Dr. Gurkirt Singh
Dr. Suman Saha
Supervisor: Prof. Dr. Luc van Gool

April 19, 2022

Abstract

We propose a novel domain adaptive action detection approach that leverages the recent advancements in unsupervised domain adaptation (UDA) techniques. Self-training combined with cross-domain mixed sampling has shown remarkable performance gain in semantic segmentation. Motivated by this fact, we propose an approach for human action detection in videos that transfers knowledge from the source domain (annotated dataset) to the target domain (unannotated dataset) using mixed sampling and pseudo-label-based self-training. The existing UDA techniques follow ClassMix algorithm for semantic segmentation. However, simply adopting ClassMix for action detection does not work, mainly because these are two entirely different problems, i.e. pixel-label classification vs. instance-label detection. To tackle this, we propose a novel action instance mixed sampling technique that combines information across domains based on action instances instead of action classes. For source-to-target knowledge transfer, we adapt the Mean Teacher based self-training. We name our proposed framework as domain-adaptive action instance mixing (DA-AIM). We demonstrate that DA-AIM consistently outperforms prior works on two challenging benchmarks: Kinetics $\rightarrow$ AVA, Kinetics $\rightarrow$ Armasuisse.

Acknowledgements

I would like to thank my supervisors Gurkirt and Suman for their dedicated supports and guidance. They continuously provided encouragements and were always willing and enthusiastic to assist in any way they could despite their busy schedules. I would also like to thank Prof. Dr. Luc Van Gool who gave me this opportunity to undertake this project on the topic related to action detection, so that I could be exposed to this rapidly developing research area. Finally, many thanks to labmates Jizhao, Luca and Rish whose previous work has lent me a helping hand at the beginning of this project.

Contents

List of Figures

List of Tables

Acronyms

ccw counter-clockwise. 12

CNNs convolutional neural networks. 1, 5

DA domain adaptation. 1–3, 6, 11, 22, 23, 26, 28, 31

DA-AIM domain-adaptive action instance mixing. III, 2, 3, 5, 7, 9, 15, 18–20, 22–29, 31

DACS Domain Adaptation via Cross-domain Mixed Sampling. 2, 7

GAN generative adversarial nets. 6

GRL gradient reversal layer. III, 13, 14, 22, 26, 27

I3D inflated 3D CNN. 5

mAP Mean Average Precision. 23, 26, 29, 31

MDD Margin Disparity Discrepancy. 6

MLP multilayer perceptrons. 13, 14

MMD Maximum Mean Discrepancy. 6

R-CNN Region Convolution Neural Network. 5

RNNs Recurrent Neural Networks. 5

RoI Region-of-Interest. 6, 10, 11

SGD Stochastic Gradient Descent. 18, 22

SSL Semi-Supervised Learning. 7, 16

T-CNN Tube Convolutional Neural Network. 5

UDA unsupervised domain adaptation. III, 1, 3, 7, 11, 13, 14, 22, 28

Chapter 1

Introduction

Spatio-temporal human action detection, an important yet challenging problem in video understanding, aims to classify an input video of human action into one of pre-defined target classes as well as identify the temporal boundaries and spatial bounding box associated with an action instance. Its broad applications vary from surveillance, sport analysis, robotics to virtual reality and animation. With the emergence of Convolutional Neural Networks (CNNs) [32, 63], video action detection has made remarkable progress over the past few years. The most state-of-the-art methods that are built upon deep spatio-temporal convolutional architectures applied on short clips of RGB frames [4, 16, 46, 76] or develop two-stream convolution with scene optical flow [17, 62], have achieved impressive classification performance, with a top-1 accuracy over 77% on the Kinetics dataset [40] and a top-5 accuracy of more than 93%.

Despite their tremendous success on major open benchmark datasets, the most state-of-the-art methods still face challenges in practical tasks — they are hardly generalizable to real-world scenarios. Models trained on the publicly available datasets (source domain) often fail to produce accurate predictions on samples from unseen datasets (target domain), due to the distribution dissimilarity caused by several factors such as illuminations, color balances, perspective, background, etc. In particular, such differences between source and target domain is often referred to as "domain shift" and can not be easily remediated by further increasing the representation power [9, 34, 35]. A common practice when dealing with domain shift is to annotate some data from the target domain and re-train (fine-tune) the network on new data. However, collecting densely annotated data for training is usually a labor-intensive task. Especially for large scale video datasets, frame-level dense annotations are extremely expensive, time-consuming and error-prone [30, 40].

Unsupervised Domain Adaptation (UDA) seeks to overcome domain shift without target domain labels by aligning domain distributions [8] and learning domain-invariant features [23]. Most recent works mainly deploy three kinds of strategies, i.e. discrepancy losses [5, 83, 86], adversarial losses [48, 56, 82] and self-supervised training [64, 68, 77]. Researches of image-based domain adaptation (DA) have ranged from simple classification [21, 49, 61, 70, 71, 80] to more complex tasks like semantic segmentation [7, 11, 35, 69, 73, 81] and object detection [2, 10, 33, 37, 41, 85]. However, researches on video tasks are relatively limited. There are a few works contributing to video classification and action recognition [6, 12, 13, 38], nonetheless, DA for human action detection is still an untouched field. Compared to action recognition, human action detection ad-

ditionally requires the localization of spatial bounding boxes and temporal boundaries for action instances, which makes it inherently more complicated.

In this work, we address this challenging task of video unsupervised domain adaptation for human action detection. We propose Domain-adaptive Action Instance Mixing (DA-AIM) — a cross-domain video augmentation framework to tackle this problem. More specifically, we randomly select a few action instances from a source domain video clip and mix the corresponding resized 3D action tubes [27] onto an unlabelled target domain video. To ensure that over the course of training all augmented new videos contain instances from both source and target domain, oversized action tubes will be downscaled before mixing. Pseudo-labels and bounding boxes of the new augmented video clip are constructed by mixing the source domain ground-truth labels and bounding boxes together with pseudo-labels and bounding boxes created for the target domain video clip.

(a) source domain (b) target domain (c) adapted ClassMix (d) DA-AIM

Figure 1.1: Apply adapted ClassMix and DA-AIM respectively for video clips from (a) source domain and (b) target domain. The results are shown in (c) and (d). Only key frames are displayed.

Our method is inspired by a recently proposed method called Domain Adaptation Via Cross-domain Mixed Sampling (DACS), which uses mixed samples to solve DA problem for semantic segmentation [68]. However, DACS is identified and studied for static 2D image task and can not be naively employed for 3D video action detection problem. Meanwhile, the mechanism how it selects and mixes images from both domain by classes is not feasible for action detection, because semantic segmentation is pixel-level dense classification problem while the target objects in action detection are various instances highlighted by bounding boxes. Furthermore, action detection is most times formulated as multi-label classification, which makes selection by class impossible. Even in single-label classification settings, where the relevant actions are exclusive from each others, selection by class will lead to troubles. People appearing in one video are tend to do same or similar actions, e.g. football, dance. When selected by class, they will all be selected resulting in loss of information from target domain (Fig.1.1). These problems are effectively solved in DA-AIM by extending the cross-domain mixing framework further to temporal axis and replacing selection by class with more meaningful selection by action instance. We implement and analyze the efficacy of various DA stratigies including self-supervised learning [24, 64, 77], adversarial learning [21] and our proposed DA-AIM. We demonstrate our method achieves best performance on two different action detection domain adaptation benchmarks: Kinetics $\rightarrow$ AVA and Kinetics $\rightarrow$ Armasuisse.

In summary, our main contributions are: (1)To the best of our knowledge, we are the first to dig into the field of UDA for action detection.(2)We investigate several UDA techniques and verify their efficacy for action detection. (3) We propose DA-AIM, a novel cross-domain video mixing framework for action detection, and show it outperforms other DA strategies for action detection.

This work consists of five chapters. In Chapter 2, we present related works and compare our work with them. In Chapter 3, we briefly explain DA techniques we used as baseline and introduce our DA-AIM framework in details. Experiment results will be unveiled and analysed in Chapter 4. In the end, we draw conclusions and make outlook on future research in Chapter 5.

Chapter 2

Related Work

2.1 Action Recognition and Detection

Action Recognition. Action recognition using deep neural networks has shown quick progress recently. There exist different approaches to it using Recurrent Neural Networks (RNNs) [15], two-stream CNNs [17, 62], and 3D CNNs [4, 66, 67, 76]. Two-stream approaches [17, 62] adopt two-stream networks with RGB and optical flow streams, which has spatial 2D CNNs to learn still features from RGB frames and temporal 2D CNNs to capture information about motion changes from optical flow. Carreira *et al.* [4] introduced the inflated 3D CNN (I3D) that expands ImageNet [60] pre-trained kernels of 2D CNNs to 3D. They verified 3D CNNs with largescale in-context action datasets such as Kinetics [40], which becomes strong baselines in action recognition. Du *et al.* [67] and Xie *et al.* [76] factorized 3D convolutions into 2D spatial and 1D temporal convolutions for more nonlinearities and efficient prediction.

Action Detection. Action detection is a more challenging problem due to the additional requirement for localisation of actions in a large spatial-temporal search space. Inspired by Region Convolution Neural Network (R-CNN) [25] and Fast R-CNN [26], Peng *et al.* [57] proposed a multi-region two-stream R-CNN model to improve frame-level detection by stacking optical flow over frames. Later, Hou *et al.* [36] suggested an end-to-end 3D CNN model called Tube Convolutional Neural Network (T-CNN) that is capable to recognize and localize action based on 3D convolution features. Recently, Feichtenhofer *et al.* [16] proposed SlowFast network, which capture spatial semantics and motion separately by applying different spatio-temporal resolutions for two pathways.

In this work, SlowFast is used as backbone in our experiments. However, we address domain-adaptive action detection problem that aims to improve the model performance in the presence of a domain shift between the labeled source and unlabeled target domain. In contrast, the above mentioned works test their models on datasets sharing similar distributions as the training datasets. When tested on unseen dataset (target domain), they usually suffer from performance degradation. Therefore, domain adaptation framework, such as our proposed DA-AIM, is the key to deploy

them in real-world scenario.

2.2 Domain Adaptation

Image-Based Domain Adaptation. Domain adaptation strives to address the performance drop caused by the different distributions of training data and testing data. It has been extensively explored for image-based objectives. Discrepancy-based DA [5, 29, 72, 83, 86] is one of the major classes of methods seeking domain-invariant features in task-specific layer through various statistical moment matching techniques, such as Maximum Mean Discrepancy (MMD) [29, 72], Margin Disparity Discrepancy (MDD) [83] and JDDA [5] that attempts to match the covariance of source and target with discriminative information preserved. An alternative branch of DA applies self-supervised learning. The emphasis lies on designing an auxiliary task (or pretext task) that is related to the main task and the labels can be self-annotated. Such auxiliary tasks include solving jigsaw puzzle [3, 53], image inpainting [55], image colorization [43], image rotation [24, 64] and depth prediction [74]. Adversarial-based DA [21, 22, 33–35, 48, 56, 71] is also popular with similar concepts as generative adversarial nets (GAN) [28] by using domain discriminators. With carefully designed adversarial objectives, the domain discriminator and the feature extractor are optimized through min-max training. Moreover, self-training and data augmentation are frequently used for image-based domain adaptation as well. Due to the close relation with our work, we discuss them in separate paragraphs below. These works tackle image-level domain adaptation while domain-adaptive action detection is video-based problem.

Video-Based Domain Adaptation. Unlike image-based domain adaptation, video-based domain adaptation receives less attention for its difficulty even though it supports many important applications. There are only a few works on video domain adaptation [6, 19, 39, 77]. Jing *et al.* [39] proposed 3DRotNet to learn spatial-temporal features by rotating videos and predicting such rotations as a pretext task. After learning, 3DRotNet was able to understand the semantic concepts and motions in video and was used to improve video understanding task on small datesets. Fernando *et al.* [19] formulated a novel auxiliary tasks, odd-one-out learning. The main idea was to let models identify unrelated or odd element from a set of related elements. More specifically, they sampled a set of video clips and rearranged the order of frames in one of the clips. Models are trained to predict the odd video subsequently. Similar to Fernando's approach, Xu *et al.* [77] performed an auxiliary task to predict the order of shuffled clips from the video, which encourages models to learn spatial-temporal features of the video. Chen *et al.* [6] employed attention mechanism and attended to temporal relation features to overcome domain shift. These works differ from ours by focusing on different video task. They dedicated themselves to video classification or action recognition, whereas our work focuses on action detection. We implement their methods and analyze their efficacy for action detection. The results serve as baseline in our work. Some necessary adaptations are made on their works to fit into action detection problem. For example, in our work frame-level features are replaced by region based features because our later stage action classification is derived from Region-of-Interest (RoI) features and more attention should be paid

on them instead of frame-level features including background information. Furthermore, some implementation details are modified to align with our backbone.

Data Augmentation (Mixing). Augmentation strategies that combine more than one image include Mixup [79] and CutMix [78]. Mixup averages the RGB values of two images and the ground truth labels to create new samples. CutMix has improved upon regional dropout by filling in image patches from other images in the dropped-out region. Kim *et al.* [42] and French *et al.* [20] demonstrated the potency of CutMix in Semi-Supervised Learning (SSL) for semantic segmentation. Later, Olsson *et al.* proposed ClassMix [54] developed from CutMix, where the mask used for mixing is instead created dynamically based on the predictions of the network. Specifically, the algorithm selects half of the classes predicted for a given image, and the corresponding pixels are pasted onto a second image. In comparison, DA-AIM mixes 3D video clips not only in spatial space but also along temporal axis. More importantly, DA-AIM is tailored for action detection, for which previous mixing strategies are not feasible. Mixup [79] and CutMix [78] may lead to ambiguity of actions. ClassMix [54] suits for segmentation, i.e. pixel-able classification but not instance-label detection.

Self-Training (Pseudo-Labeling). Pseudo-label refinement under a self-training framework has achieved competitive results in UDA for image classification and semantic segmentation. By iteratively using gradually-improved target pseudo-labels to train the network, the performance on the target domain are boosted. One of the problems for pseudo-labeling identified in previous studies is the bias in the target domain towards initially easy-to-transfer classes [75, 87, 88]. Although we are tackling more complex video-based UDA for action detection, we observe similar phenomenon in our experiments. To combat the problem of faulty pseudo-labels for UDA, existing works have suggested careful selection and adjustment procedures, accounting for the domain gap. Variants include specialised sampling [50, 87] and handling of uncertainty [84, 87]. Compared with their works, we use cross-domain mixing of samples, a simple but effective method, to solve this problem.

DACS. The idea of using cross-domain mixing to correct erroneous pseudo-label generation arising due to the domain shift is inspired by DACS [68]. DACS combines self-training and ClassMix [54], which mixes images from source and target domain to create new augmented samples during training. However, DACS is not applicable to action detection, since it is designed for semantic segmentation, i.e. pixel-label classification problem, whereas action detection is instance-label detection problem. The mixing procedure and label combination need to be treated differently. We propose DA-AIM that mixes action tubes from source and target domain and introduce mechanisms like enlarging bounding boxes and resizing to avoid information loss on target domain. Ground-truth and pseudo-labels are mixed reasonably based on the overlapping situation.

Chapter 3

Methodology

Fig.3.1 shows an overview of our proposed Domain-adaptive Action Instance Mixing (DA-AIM). Labeled and unlabeled video clips from source and target domain are mixed together to create new augmented training samples. Ground-truth labels and pseudo labels are then correspondingly mixed. These mixed samples are then trained on, in addition to the labeled data itself.

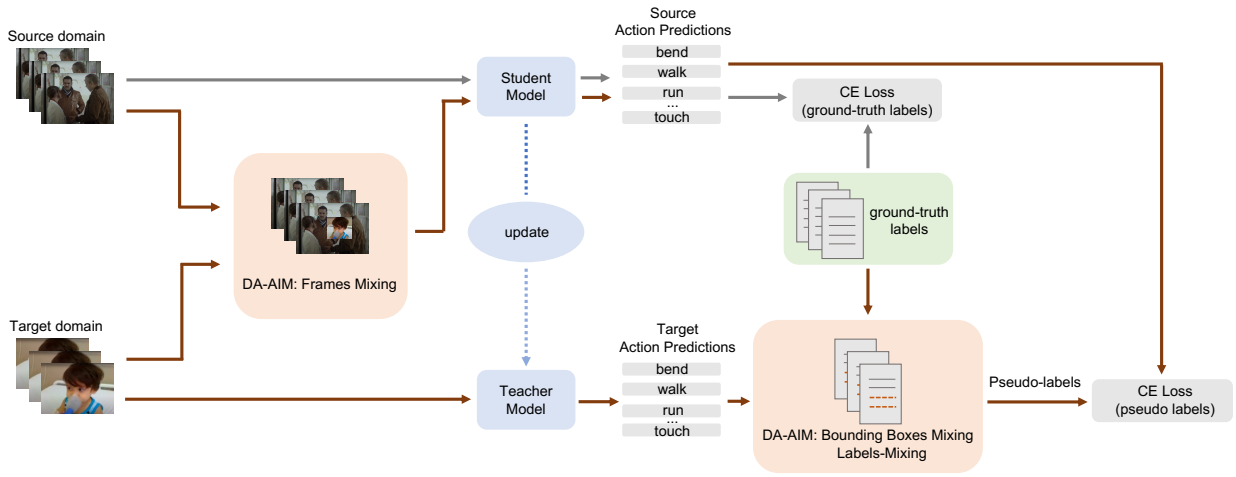


Figure 3.1: Overview of our proposed framework.

3.1 Backbone

We use SlowFast [16] as our backbone architecture throughout this work for both baseline and DA-AIM experiments. SlowFast is a state-of-the-art model for action recognition and detection, which uses two pathways to imitate the biological structure of primate visual system [14, 18, 47].

3.1.1 SlowFast

As name suggests, the encoder of SlowFast is composed of two pathways, i.e. the slow pathway and the fast pathway. The slow pathway is mainly used to capture spatial information, so it is required to have a low frame rate. One way to achieve this is using a large temporal stride τ on input frames, i.e. it only processes one frame out of τ frames. In parallel to the slow pathway, the fast pathway works at a high frame rate, i.e. with a small temporal stride of τ/α, where $\alpha > 1$ is the frame rate ratio between the fast and slow pathways. If the raw clip length of one input video equals T, then the lengths of the inputs for slow and fast pathway are T/τ and $\alpha T/\tau$ respectively, from which it is clear to see that the fast pathway incepts a factor of α more frames and thus more detailed and faster temporal information. Additionally, these two pathways also differ in terms of capacity where the fast pathway occupies obviously lower channel capacity and this behaviour of the fast pathway can be understood as a weaker ability of representing spatial semantics. The channel capacity is controlled by a ration parameter $\beta(\beta < 1)$. Throughout this work, we use $T = 64$, $\beta = 1/8$, $\tau = 8$ and $\alpha = 4$, so the slow pathway operates 8 frames while the fast pathway works on 32 frames. PySlowFast delivers an implementation of several different backbone architectures and specifically in this work we use ResNet-50.

While extracting features, the information from slow and fast pathway needs to be fused. Due to the difference of the number of channels, it is required to match the sizes of features at first. Three different transfomations in the lateral connections are proposed [16], i.e. Time-to-channel, Time-strided sampling and Time-strided convolution. In this project, time-strided convolution is used, it performs a 3D convolution with 5×1^2 kernel, $2\beta C$ output channels and stride equals α.

3.1.2 Action Detection

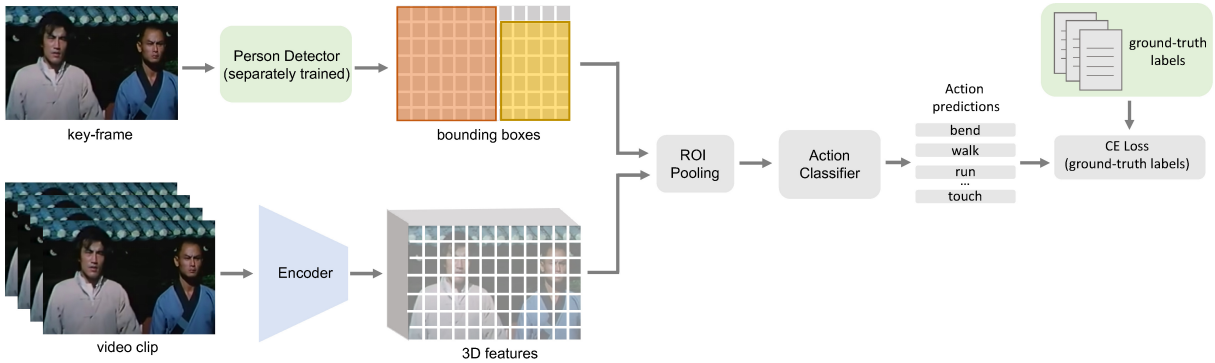


Figure 3.2: Action detection pipeline for SLowFast

The extracted features are then fed into an action detection pipeline (Fig.3.2) to convert the given feature representation of the input clips into corresponding predictions for action class. The action detector architecture is similar to Faster R-CNN [59] with minimal modifications adapted for video. The RoI features [26] are extracted at the last feature map of the encoder. Each 2D RoI at a frame is firstly extended into a 3D RoI by replicating it along the temporal axis according to

[30]. Then the RoI features are calculated by RoIAlign [31] spatially, and global average pooling temporally, which are later max-pooled and fed to the classifier for action predictions.

3.2 Domain Adaptation Techniques (Baselines)

In this section, we briefly introduce several frequently used DA techniques, which we implement and conduct experiments to serve as baselines. We separate them into two categories, namely self-supervised learning and adversarial learning.

3.2.1 Self-supervised Learning

Self-supervised learning is one kind of DA technique where the supervisory signal can be obtained directly from the data itself. We consider two different self-supervised auxiliary tasks in this work, i.e.rotation and clip order prediction. They share the same pipeline (Fig.3.3). Our main task, namely action detection, is jointly trained with the auxiliary task in order to find domain-invariant features. The training data for the auxiliary task comes from both source and target domain. The features are extracted by SlowFast and then fed into both action classifier and auxiliary task classifier. The main difference and keystone, however, come from the auxiliary task data pre-processor.

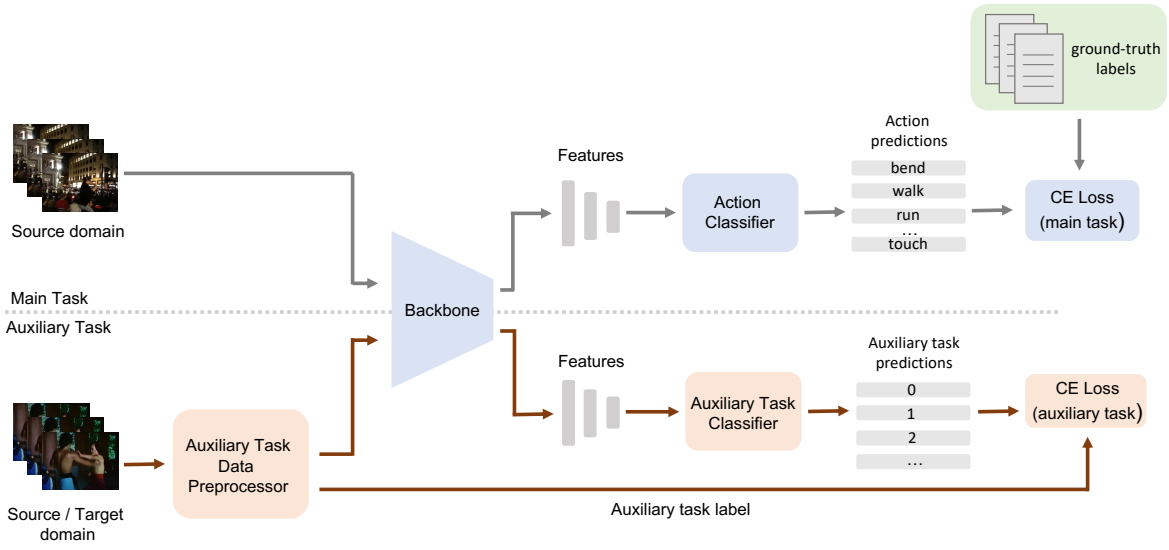
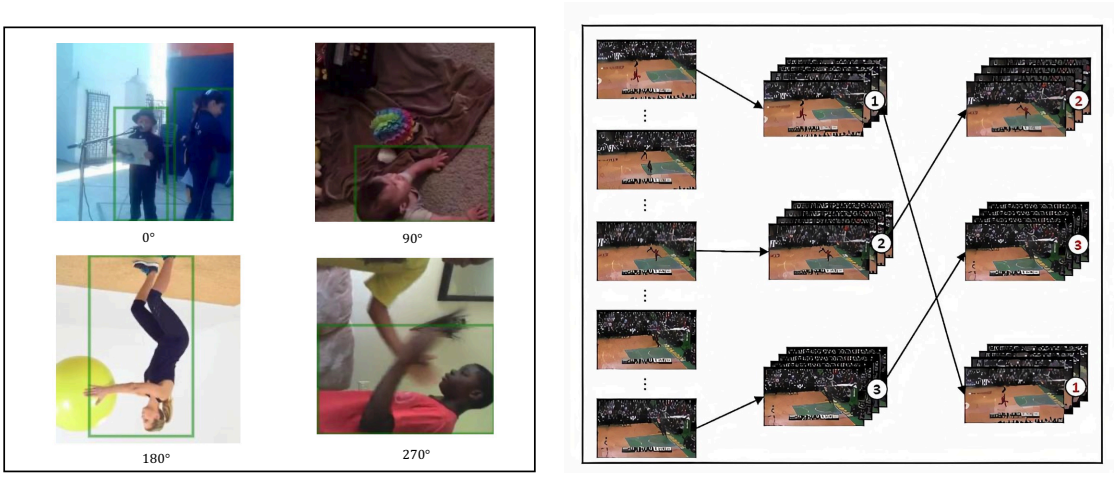


Figure 3.3: Pipeline of self-supervised learning UDA for action detection. We implement two different auxiliary tasks rotation and clip-order prediction. The differences lie inside auxiliary task data pre-processor.

Rotation Rotation prediction is a common used techniques in self-supervised learning for both image and video [24, 64]. In our work, rotation prediction is conducted on the RoI features, since our aim is to improve action classification on the basis of given bounding boxes that focus only on the image area containing action instances. Introducing region based rotation prediction differs our

work from the previous studies. To load the images for rotation prediction, action tubes (images and bounding boxes) need to be rotated into 1 of 4 configurations (0°, 90°, 180°, 270°) counter-clockwise (ccw) and the corresponding label (4 classes) needs to be created. Given bounding boxes as a tuple (x_1, y_1, x_2, y_2), where (x_1, y_1) corresponds to the top left corner and (x_2, y_2) corresponds to the bottom right corner, as well as H being the image length and W the image width, Equ.3.1 - 3.4 describe the transformations to compute bounding boxes coordinates after rotation. Some examples after rotation are shown in Fig.3.4 (a).

$$0°ccw : (x_1, y_1, x_2, y_2) \tag{3.1}$$
$$90°ccw : (y_1, W - x_2, y_2, W - x_1) \tag{3.2}$$
$$180°ccw : (W - x_2, H - y_2, W - x_1, H - y_1) \tag{3.3}$$
$$270°ccw : (H - y_2, x_1, H - y_1, x_2) \tag{3.4}$$



(a) rotation (b) clip-order

Figure 3.4: Examples of (a) rotation and (b) clip-order after transformation.

Clip-Order To explore the temporal information of video, recent works attempt to leverage the temporal relations among frames, such as order verification [51] and order prediction of frames [44] for self-supervise learning. However, regarding action detection, the order can at times not be uniquely determined when referring to frames merely because of the bi-directional characteristic of some actions. Therefore, in this work we use clip-order prediction instead of frame-order compared to previous studies. The task is reformulate as a classification problem. For one input video clip, we firstly divide it into 3 sub-clips and then permute the sub-clips randomly to form the input data of auxiliary task (Fig.3.4 (b)) while the actual order is served as the label. There are $3! = 6$ possible orders to rearrange the sub-clips, which equals the 6 classes for the auxiliary task.

3.2.2 Adversarial Learning

In this work, we also implement one of the standard UDA technique called gradient reversal layer (GRL) as baseline. GRL acts as an identity function during forward propagation, yet during backward propagation it multiplies input with a negative factor. Hence, GRL actually does gradient ascent rather than gradient descent during backward propagation. Fig.3.5 shows how to utilize GRL to do UDA [21]. GRL is inserted between the feature extractor and the domain classifier, where feature extractor is trying to fool domain classifier, so that the domain classifier cannot differ from which domain features are extracted, i.e. it promotes feature extractor to concentrate on domain-invariant features.

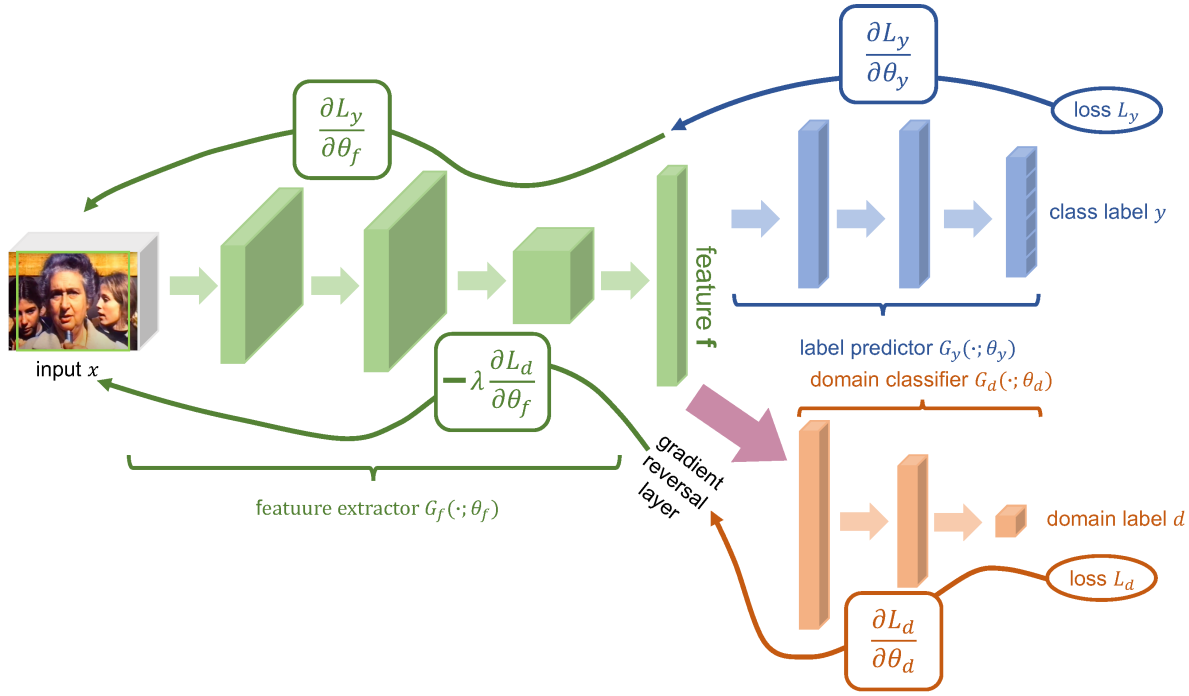


Figure 3.5: The working principle of GRL in UDA

Following similar settings in [6], our pipeline is shown in Fig.3.6. The training samples (video clips) from the source and target domains are fed as inputs to our backbone for feature extraction. The supervised action detection loss is computed on the source domain. At the same time, the source and the target features are used to do domain alignment with the help of three components, spatial module, temporal module and the temporal feature fusion.

Spatial Module Spatial module consists of multilayer perceptrons (MLP) and is used to convert the feature vectors into task-driven feature vectors. The converted features are then passed to the adversarial discriminators G_{sd}, which is composed of GRL and a domain classifier used for domain classification task in the frame-level. The aim of spatial module is to close the domain shifts between the source and target domain along the spatial direction.

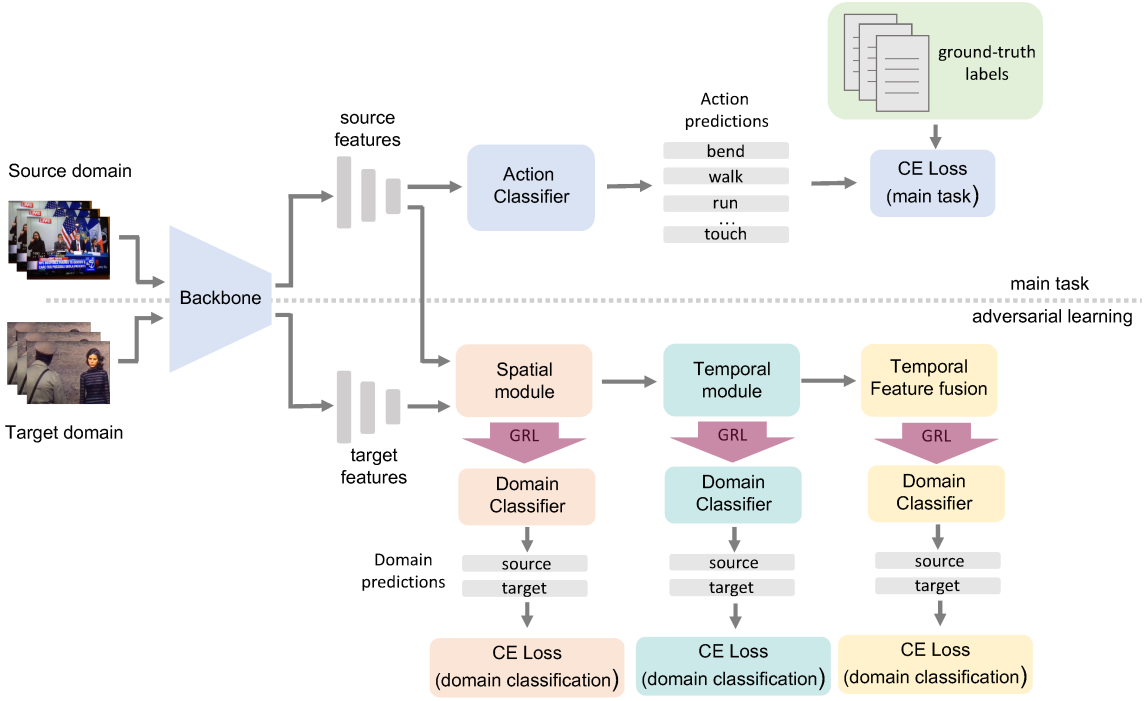


Figure 3.6: Pipeline of adversarial learning UDA for action detection.

Temporal Module The emphasis of temporal module is to build different frame relation sets to capture temporal relation of video. Assume the i-th video clip V_i contains n frames $v_i^1, v_i^2, \ldots, v_i^n$ and the number of frames K used to build a relation set can be chosen from 2 to n. Thus, there are 2-frames relation set V_i^2, 3-frames relation set V_i^3, ...n-frames relation set V_i^n. Each relation set V_i^K includes $\binom{k}{n}$ elements and has a corresponding function $g_{\phi(K)}$ which is MLP with parameters $\phi(K)$. The K-frame relation features of the i-th video V_i is defined as follows:

$$R_K(V_i) = \sum_m g_{\phi(K)}((V_i^K)_m), \tag{3.5}$$

where m is the m-th element of the relation set V_i^K. To reduces the domain discrepancy caused at the different time scales, individual adversarial discriminators G_{rd}^K consisting of GRL and domain classifiers are applied to $R_2(V_i)...R_n(V_i)$ respectively.

Temporal Feature Fusion After all frame relation features are obtained, the temporal feature fusion is generated by applying element-wise summation on $R_2(V_i)$ ... $R_n(V_i)$, which stands for the final video representation of the i-th video V_i. Nonetheless, relation features that have larger domain discrepancy and are easier to be discriminated by a domain classifier should be given more

attention. Hence, the domain attention mechanism is introduced by the means of entropy criterion. The domain attention value for each relation feature is defined as:

$$\omega_i^K = 1 - H(\widehat{d}_i^K), \tag{3.6}$$

where $\widehat{d}_i^K$ represents the output of the frame relation domain classifier G_{rd}^K for the i-th video. H is the entropy function measuring uncertainty and the definition is (for discrete case):

$$H(p) = -\sum_k p_k \log p_k \tag{3.7}$$

With the help of attentive values, the final representation of i-th video h_i can be expressed as:

$$h_i = \sum_{K=2}^{n} (\omega_i^K + 1) \cdot R_i^K \tag{3.8}$$

Similarly as in spatial module and temporal module, after obtaining this final video clip representation, another adversarial discriminators G_{td} is applied to further close the domain shifts in video-level.

3.3 Domain-adaptive Action Instance Mixing (DA-AIM)

In this section, we will introduce our proposed DA-AIM. Our framework can be decomposed into two main steps, namely pseudo-labeling and mixing. We will firstly explain them in details respectively and then demonstrate specifically how we combine them and present our DA-AIM algorithm.

3.3.1 Pseudo-Labeling

In order to construct pseudo-labels for unlabeled target domain video clips, they are fed into the teacher model before mixing, where the teacher model is an average of consecutive student models [65], as averaging model weights over training steps tends to produce a more accurate model than using the final weights directly [58].

More formally, we define the weights of the student model at training step t as θ_t and the weights of the teacher model as θ'_t. At each training step t, weights of the teacher model θ'_t are not optimized with loss function but updated according to the following equation:

$$\theta'_t = \alpha\theta'_{t-1} + (1 - \alpha)\theta_t \tag{3.9}$$

where α is a smoothing coefficient hyperparameter. In our experiments, we preprocess datasets and focus on exclusive actions, which means those actions can not be done at the same time. Consequently, the problem is a single-label classification problem. Hence, the pseudo-label of an action instance is the action class obtaining highest confidence score from the current teacher model.

3.3.2 Mixing

Within-domain and cross-domain mixing have been widely used for dense prediction tasks in SSL and unsupervised learning, such as image classification [1, 78] and semantic segmentation [20, 42, 68]. Despite the effectiveness on the 2D image problems, mixing has rarely been studied for videos. To tackle 3D video action detection problem, we extend the 2D cross-domain mixing strategy in [68] to temporal axis and meanwhile introduce bounding boxes expansion and frame resizing to fit the special property of action detection. Fig.3.7 indicates the procedure of our cross-domain video mixing.

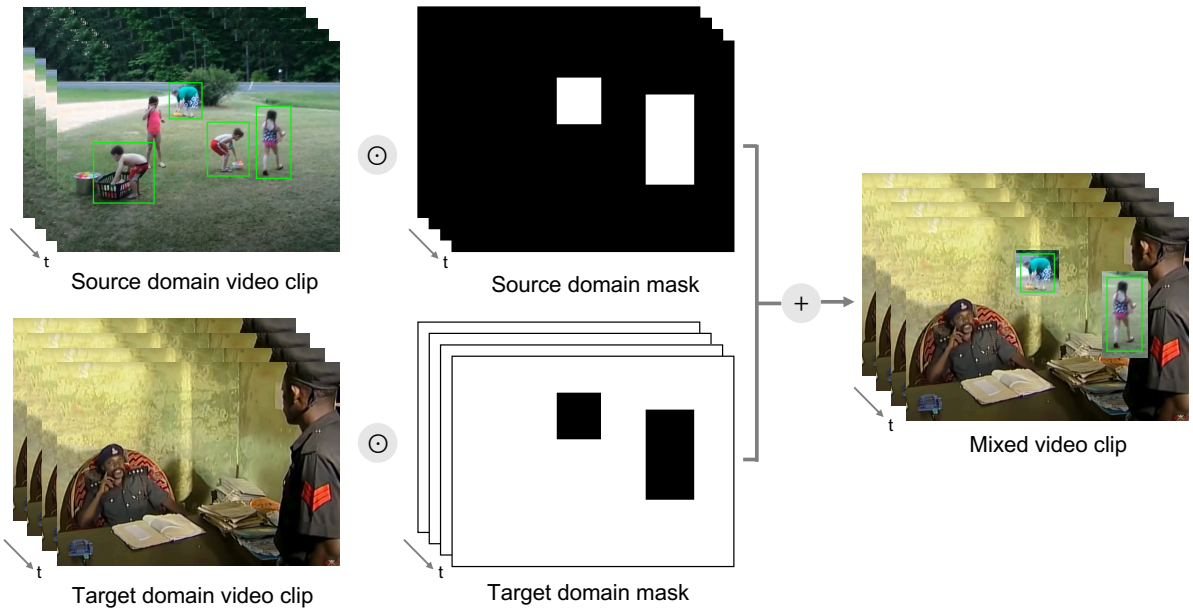


Figure 3.7: The procedure of cross-domain mixing.

Given video clips from source and target domain, as well as annotations of source domain, we firstly select half of the action instances from the source domain randomly. Since the bounding boxes are created for the key-frame located in the middle of a clip, considering fast movements such as running, we expand each bounding boxes by 20% when creating the source domain mask. The 3D source domain mask $M \in \{0,1\}^{T\times W\times H}$ is constructed by replicating 2D mask $M_{key\text{-}frame} \in \{0,1\}^{W\times H}$ in temporal axis, where $M_{key\text{-}frame}$ is nothing but a binary matrix containing 1 only at the places corresponding to selected and expanded bounding boxes. Then, our mixed video clips can be obtained through:

$$x_M = M \odot x_S + (1 - M) \odot x_T, \tag{3.10}$$

where $x_M, x_S, x_T \in \mathbb{R}^{T\times W\times H}$ represent the mixed video clip, input source and target domain video clips respectively.

Another point needs to be noticed in action detection is that video samples from datasets like Kinetics are very often short and focused on one specific action instance. If such a video clip is

used without resizing, it will lead to imbalance of information from source and target domain in mixed video clip, where most information comes from source domain while only border pixels are from target domain. To solve this problem, if the area of all selected bounding boxes takes up more than half of the entire area of one frame, we will downscale the frame sequence by factor 0.5, i.e. the width and length of frames are decreased by half. Bounding boxes and the mask are correspondingly modified to suit the resized video clip (Fig.3.8). Given bounding boxes as a tuple (x_1, y_1, x_2, y_2), where (x_1, y_1) corresponds to the top left corner and $(x2, y2)$ corresponds to the bottom right corner, as well as H being the image length and W the image width, coordinates of bounding boxes after resizing (x'_1, y'_1, x'_2, y'_2) can be expressed as:

$$x'_1 = [\tfrac{W}{4}] + [\tfrac{x_1}{2}] \tag{3.11}$$
$$y'_1 = [\tfrac{H}{4}] + [\tfrac{y_1}{2}] \tag{3.12}$$
$$x'_2 = [\tfrac{W}{4}] + [\tfrac{x_2}{2}] \tag{3.13}$$
$$y'_2 = [\tfrac{H}{4}] + [\tfrac{y_2}{2}] \tag{3.14}$$

where $[\cdot]$ indicates the rounding function to find the nearest integer. The empty borders after resizing will be filled with 0.

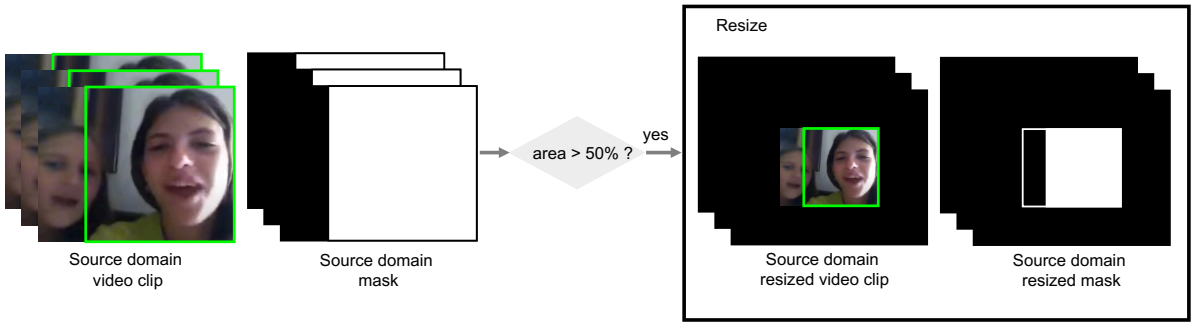


Figure 3.8: Frames need to be downscaled if selected bounding boxes take up more than half of the entire area. Bounding boxes and the mask are correspondingly reformed to fit resied frames.

3.3.3 Combination of Bounding-Boxes and Labels

Bounding boxes and labels can not be simply concatenated, because original target domain action instances might be covered after mixing source domain action instances onto the target domain video clips. Due to the possibility of lacking important information to identify the action, if a bounding box from target domain overlaps with any pasted bounding boxes from source domain, it should be removed and not be used to compute the loss function anymore.

Given two bounding boxes as tuples $b_1 = (x_1, y_1, x_2, y_2)$ and $b_2 = (x'_1, y'_1, x'_2, y'_2)$, we can verify whether the two bounding boxes overlap or not by the following boolean expression:

$$overlap = \neg\big((x'_2 < x_1) \vee (y'_1 > y_2) \vee (x'_1 > x_2) \vee (y'_2 < y_1)\big). \tag{3.15}$$

For each bounding box from target domain, we loop through all selected bounding boxes from source domain and check whether they overlap or not with Eq.3.15. If Eq.3.15 returns true, we delete the bounding box and the corresponding pseudo-label from target domain lists. This procedure is repeated for all bounding boxes from target domain. After that, selected ground-truth bounding boxes and labels from source domain can be concatenated with cleaned bounding boxes and pseudo-labels from target domain, which do not contain overlapping action instances anymore.

3.3.4 Loss Function

In DA-AIM, the student network parameters θ are trained by minimizing the following loss:

$$\arg\min_{\theta} \mathcal{L}(\theta) = \arg\min_{\theta} \mathbb{E}\Big[H\big(f_\theta(X_S, B_S), Y_S\big) + \lambda H\big(f_\theta(X_M, B_M), Y_M\big)\Big] \tag{3.16}$$

where the expectation is over batches of random variables X_S, B_S, Y_S, X_M, B_M and Y_M. Video clips in X_S are sampled uniformly from the source domain distribution, B_S and Y_S are the corresponding bounding boxes and labels. Furthermore, X_M is the new mixed video clips, created by performing cross-domain mixing between the video clips from the source and target domain, as explained above. B_M and Y_M are mixed bounding boxes and mixed labels constructed by combining ground-truth bounding boxes or labels with predicted bounding boxes or pseudo-labels. Lastly, as we focus on exclusive actions and formulate the problem as single-label classification, we use cross-entropy loss H computed between the predictions and the corresponding labels (ground-truth or pseudo) averaged over all action instances. λ is a hyper-parameter that decides how much the unsupervised part of the loss affects the overall training. Adapted from [68], we use an adaptive schedule for λ, where it is the proportion of instances in the whole unlabeled instances in the mixed video clip, of which the predictions have a confidence above a certain threshold. Training is performed by Stochastic Gradient Descent (SGD) on the loss using batches with the same number of source-domain video clips and augmented video clips.

3.3.5 Domain-adaptive Action Instance Mixing Algorithm

DA-AIM is an integration of cross-domain instance mixing and pseudo-labeling, whereas they can also be used separately. We visually compare the effects of applying them separately without integration and DA-AIM in Fig.3.9. When used alone, pseudo-labeling will not create new augmented video clips but operates on the target domain video clips directly. In another word, $H\big(f_\theta(X_M, B_M), Y_M\big)$ in Eq.3.16 becomes $H\big(f_\theta(X_T, \widehat{B}_T, \widehat{Y}_T\big)$, where X_T, $\widehat{B}_T$ and $\widehat{Y}_T$ are unlabeled video clips from target domain, pre-computed bounding boxes and pseudo-labels for target domain respectively. In contrast, cross-domain instance mixing only augments the video clips without any pseudo-labels from target domain, which means $H\big(f_\theta(X_M, B_M), Y_M\big)$ in Eq.3.16 becomes $H\big(f_\theta(X_M, B'_S), Y'_S\big)$, where B'_S and Y'_S are parts of ground-truth bounding boxes and labels corresponding to selected action instances from source domain.

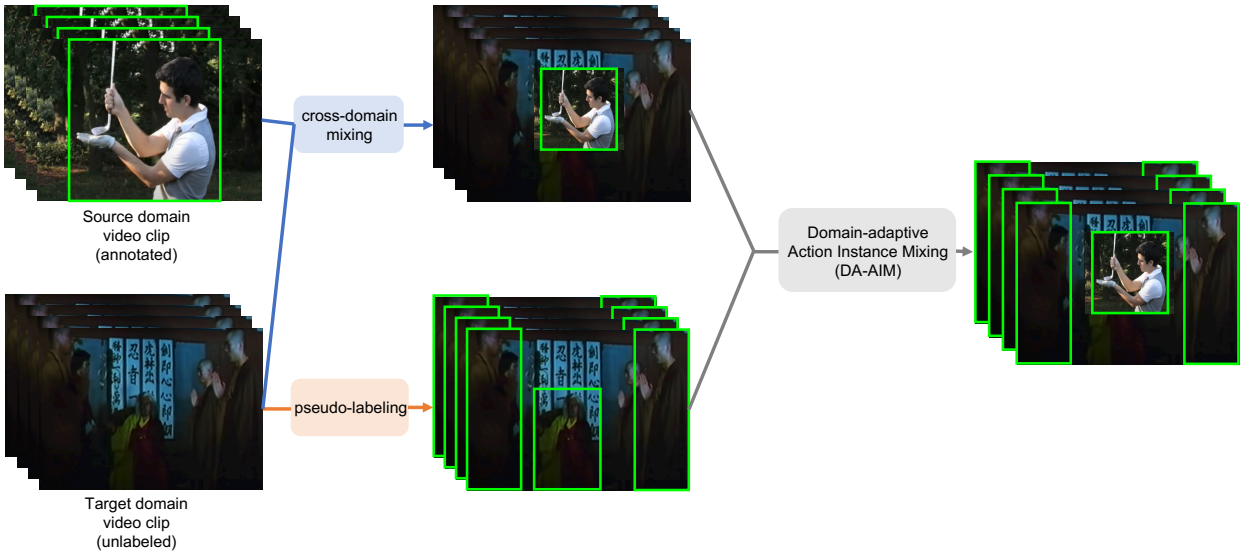


Figure 3.9: Compare the effects of pseudo-labeling, mixing and DA-AIM.

The overall DA-AIM algorithm is summarized in Alg.1. The source-domain and target-domain datasets are referred to as $\mathcal{D}_S$ and $\mathcal{D}_T$. A batch of video clips, bounding boxes and labels, X_S, B_S and Y_S, is sampled from $\mathcal{D}_S$, and another batch of video clips, X_T from $\mathcal{D}_T$. $\widehat{B}_T$ represents bounding boxes of target domain video clips estimated by a pre-trained person detector. The unlabeled video clips X_T and bounding boxes $\widehat{B}_T$ are firstly fed to the teacher network $f_{\theta'}$, from which pseudo-labels $\widehat{Y}_T$ are obtained. Then, the augmented video clips X_M are created by mixing X_S and X_T.The pseudo-labels Y_M and bounding boxes B_M are correspondingly constructed by mixing Y_S, $\widehat{Y}_T$ and B_S, $\widehat{B}_T$. From this point forward, the algorithm resembles a supervised learning approach: compute predictions, compare them with the labels (ground-truth or pseudo-labels), estimate loss function (cross-entropy loss in our case), perform backprogragation, and conduct a step of gradient descent. This process is then repeated for a predetermined amount of iterations N.

Algorithm 1 DA-AIM Algorithm

Input: Source-domain dataset $\mathcal{D}_S$, target-domain dataset $\mathcal{D}_T$, teacher network $f_{\theta'}$, student network f_θ, pretrained person detector d_p.

Output: Trained student network f_θ.

// Initialization

Initialize student and teacher network parameters θ and θ' with checkpoint.

// Training loop

for $t \leftarrow 1, 2, ..., N$ **do**

  Sample mini-batch from source domain $X_S, B_S, Y_S \sim \mathcal{D}_S$;

  Sample mini-batch from unlabeled target domain $X_T \sim \mathcal{D}_T$;

  // Pseudo-labeling

  Update teacher model parameters $\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$

  Compute bounding boxes $\widehat{B}_T \leftarrow d_p(X_T)$

  Compute pseudo-labels $\widehat{Y}_T \leftarrow \operatorname{argmax}\big(f_{\theta'}(X_T, \widehat{B}_T)\big)$

  // Mixing

  Augment video clips $X_M \leftarrow M \odot X_S + (1-M) \odot X_T$

  // Combine bounding boxes and labels

  Construct bounding boxes B_M and pseudo-labels Y_M from B_S, $\widehat{B}_T$, Y_S, $\widehat{Y}_T$

  Compute predictions $\widehat{Y}_S \leftarrow f_\theta(X_S, B_S)$, $\widehat{Y}_M \leftarrow f_\theta(X_M, B_M)$

  Compute loss $\ell = \mathcal{L}(\widehat{Y}_S, Y_S, \widehat{Y}_M, Y_M)$

  Compute gradient $\nabla_\theta \ell$ by backpropagation

  Optimize θ with stochastic gradient descent

end for

return f_θ

Chapter 4

Experiments

4.1 Datasets

We use three datasets in our experiments: Kinetics [40, 45], AVA [30] and Armasuisse. In this section, we will briefly introduce them separately and then describe how we preprocess and reduce them to fit our experiment settings.

4.1.1 AVA 2.2

AVA stands for atomic visual action and consists of 430 densely annotated 15-minute video clips with 80 visual actions. In total, roughly $1.62M$ action annotations are provided with the possibility that multiple annotations are made for one action instance, i.e. each action instance can perform multiple actions at the same time. Actions are localized in both space and time. We use version 2.2 of the annotation files throughout this work.

4.1.2 Kinetics700

Kinetics700 [40] contains more than $650k$ videos covering 700 action classes. Video clips last around 10s and possess a single label describing the dominant action occurring in the video. Due to lack of the spatial localisation, Kinetics700 is originally used for entire frame based action recognition. Recently, Li *et al.* [45] has adapted the Kinetics700 dataset to AVA-style bounding boxes and atomic action annotations, which makes it also suitable for action detection.

4.1.3 Armasuisse

Armasuisse is an in-house dataset from Federal Office for Defence Procurement focusing on 16 actions. The current version contains over 8.6k action annotations created for 25 videos lasting 1-4 minutes. The background of videos are relatively simple and not relevant to actions. The annotations are much less noisy compared to large scale datasets such as AVA and Kinetics.

4.1.4 Dataset Reduction

For the sake of time and resources consumption, large scale datasets need to be decreased in the number of classes as well as the number of training samples. In this work, we stick to single-label classification setting, where we select several exclusive actions that can not be done at the same time. For Kinetics $\rightarrow$ AVA experiments we focus on the following 6 actions: bend/bow (at the waist), lie/sleep, run/jog, sit, stand and walk, while for Kinetics $\rightarrow$ Armasuisse experiments we use 3 other actions: touch (an object), throw and take a photo.

To reduce the number of training samples, we set 5000 as the maximum number of training samples for each action class. For the case of insufficient training samples owing to the class imbalance inside large scale datasets, the highest possible number of samples from that class will be taken. Regarding validation datasets, there is no restriction on the amount of samples, i.e. we use all the samples from those specific action classes metioned above during the validation.

Overall statistics of reduced datasets are provided in Tab.4.1. More details can be found in Appendix A.

	AVA-6-5k		**Kinetics-6-5k**		**Armasuisse-3-5k**		**Kinetics-3-5k**	
	Train	Val	Train	Val	Train	Val	Train	Val
Annotations	28,281	89,481	29,009	27,173	441	339	6,686	1,920
Unique boxes	28,281	89,481	29,009	27,173	441	339	6,686	1,920
Key-frames	14,248	48,741	15,453	19,205	441	339	6,115	1,779
Videos	235	64	15,453	19,205	12	7	6,115	1,779

Table 4.1: Overall statistics of reduced datasets.

4.2 Experimental Setup

We implement and evaluate 6 different UDA strategies for action detection: self-supervised learning with rotation prediction (Rotation), self-supervised learning with clip-order prediction (Clip-order), adversarial learning with GRL (GRL), self-training (pseudo-labeling), data augmentation (mixing) and domain-adaptive action instance mixing (DA-AIM). The comparison and relation among pseudo-labeling, mixing and DA-AIM are explained in Section 3.3.5. DA-AIM can be seen as an integration of pseudo-labeling and mixing. Nonetheless, these two major steps can also be used separately for UDA. Additionally, we conduct experiments without any DA strategy, i.e. training model on source domain and evaluating directly on the target domain, which we refer as baseline experiment without DA techniques (Baseline w/o DA).

For all the experiments in this work, we adopt the widely used PySlowFast [16] with ResNet50 as backbone. To further reduce the training time, we initialize the model with weights pre-trained on MiT dataset [52]. We use SGD with Nesterov acceleration, and a base learning rate of 1×10^{-2} for baseline experiments while 1.25×10^{-2} for others, which is then decreased using cosine scheduler with final learning rate equal to $1/100$ of base learning rate. Warm-up lasts 1 epoch

and starts from $1/10$ of base learning rate. Weight decay is set to 1×10^{-7} and momentum to 0.9. For Kinetics $\rightarrow$ AVA experiments, we train on 4 GPUs with batch size 24 for 6 epochs, whereas for Kinetics $\rightarrow$ Armasuisse experiments we use batch size 8 and train on 2 GPUs for 4 epochs. The quantitative results shown in Section 4.3.2 are evaluated as an average of two runs and Mean Average Precision (mAP) is used as metric to indicate overall performance of various DA techniques.

4.3 Experiment Results

4.3.1 Mixed Samples

In this section, we provide some real training samples from source and target domain along with the augmented training samples created by DA-AIM. During the training, source domain samples and the augmented samples are used to compute supervised and unsupervised loss according to Eq.3.16. Samples from Kinetics $\rightarrow$ AVA and Kinetics $\rightarrow$ Armasuisse experiments are illustrated individually in Fig.4.1 and Fig.4.2.

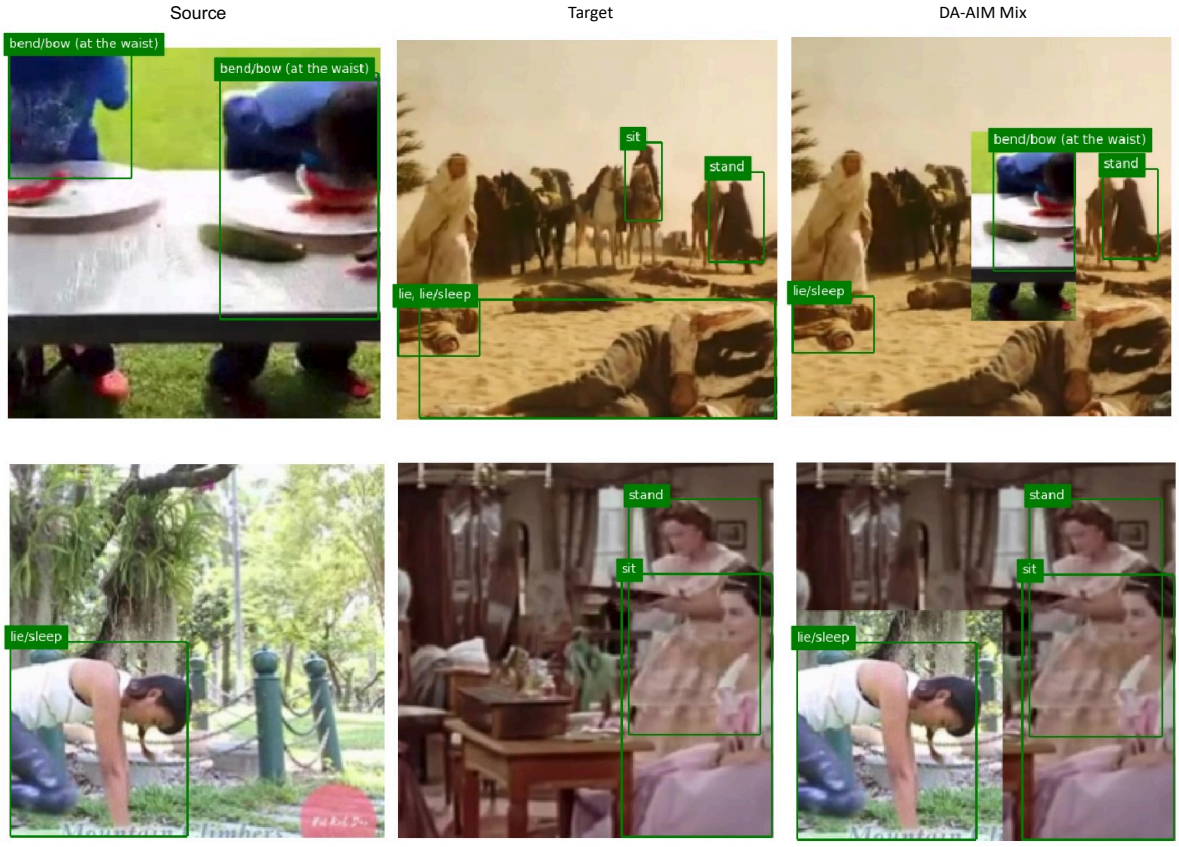


Figure 4.1: Kinetics $\rightarrow$ AVA: Training samples from source and target domain together with the corresponding augmented samples created by DA-AIM. Only key frames are displayed here.

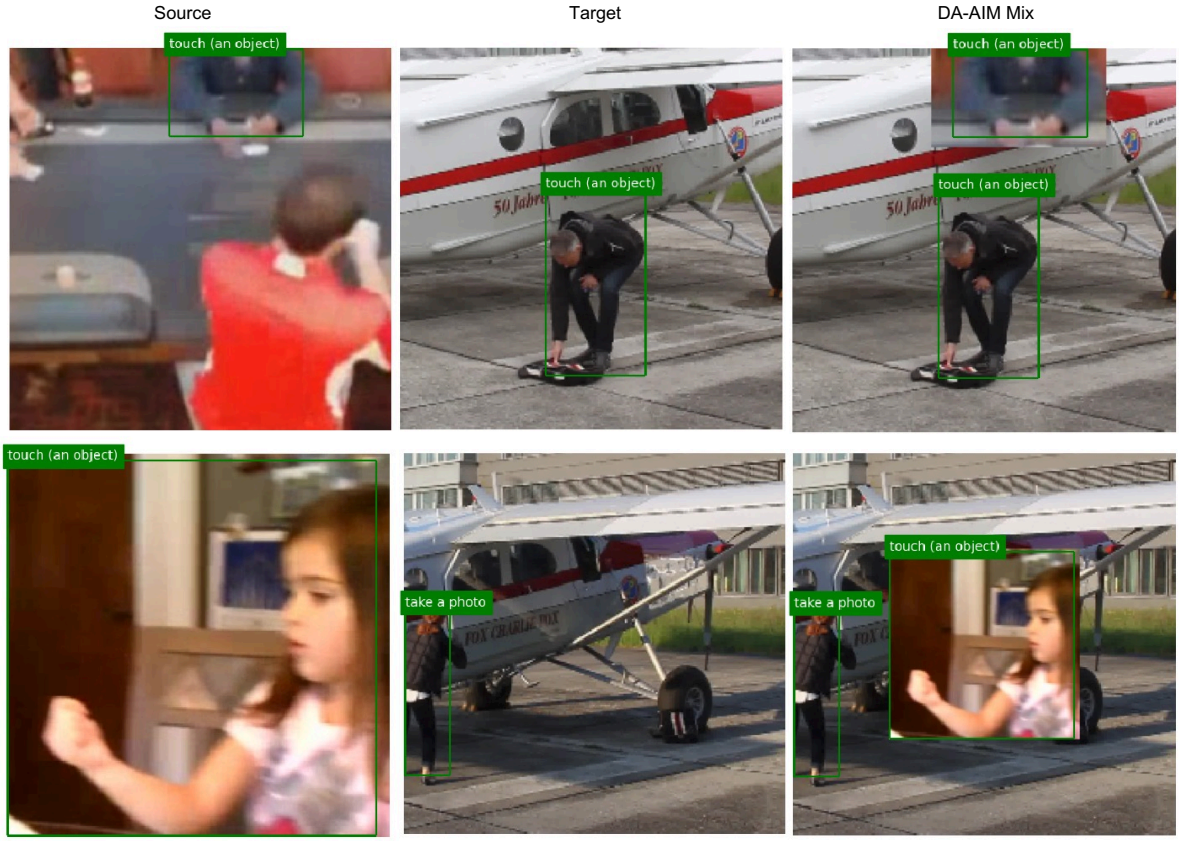


Figure 4.2: Kinetics → Armasuisse: Training samples from source and target domain together with the corresponding augmented samples created by DA-AIM. Only key frames are displayed here.

As can be seen from the figures, resizing can effectively reduce the information loss from target domain. Without resizing, for example, the target domain from the second example of Fig.4.5 will be fully covered by the source domain, because of the single oversized bounding box present in the source domain. The results from Tab.4.2-Tab.4.5 align with our hypothesis. Either mixing or DA-AIM achieves better results on the target domain with resizing, while the performance on the source domain degrades. This can be explained by an increase of information from the target domain at a price of less information from the source domain.

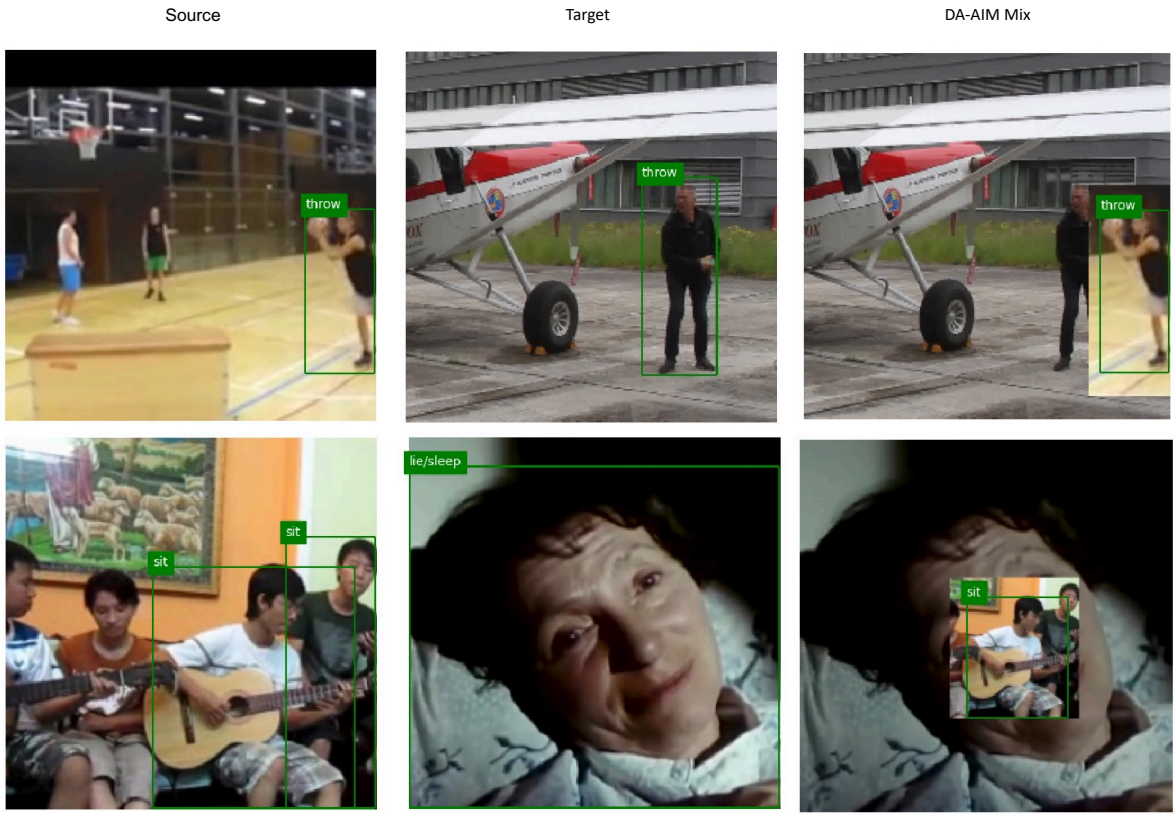


Figure 4.3: Overlapping examples. The upper and bottom rows show examples from Kinetics → Armasuisse and Kinetics → AVA experiments respectively. Only key frames are displayed here.

However, we need to point out that resizing can only solve the overlapping problem to some extent. In our experiments, there are still some examples losing target domain information after mixing (Fig.4.3). Although overlapping can be inevitable under certain circumstances (e.g. the second example in Fig.4.3), it can be further avoided by techniques like random pasting, etc. Currently, we paste the action tubes at exactly the same position as it located in the original video clips. If introducing randomness of the pasted positions, there is chance to avoid overlapping similar to the first example in Fig.4.3. It's a direction for future research to remove this limit.

4.3.2 Evaluation Results

We evaluate various DA techniques and our proposed DA-AIM on two different benchmarks: Kinetics → AVA and Kinetics → Armasuisse. The evaluation results on both source and target domain are exhibited in Tab.4.2-Tab.4.5. The numbers under action names denote average precision in percent and mAP is used as metric for overall performance.

Method	bend/bow	lie/sleep	run/jog	sit	stand	walk	mAP
Baseline w/o DA	**48.12**	74.62	71.19	74.80	80.41	65.76	69.15
Rotation	45.15	74.92	69.75	71.06	78.99	62.16	67.01
Clip-order	45.62	73.52	70.44	72.00	79.05	63.44	67.35
GRL	43.15	75.41	66.88	72.01	79.03	61.06	66.26
Pseudo-labeling	44.60	71.74	70.90	74.38	79.80	63.81	67.54
Mixing (w/ resize)	47.30	72.86	**72.81**	74.40	**81.14**	**65.92**	69.07
DA-AIM (w/ resize)	45.93	73.43	72.00	74.78	80.01	64.38	68.42
Mixing (w/o resize)	47.28	**76.03**	72.05	**76.43**	80.47	62.94	**69.20**
DA-AIM (w/o resize)	44.96	72.06	71.29	75.68	79.61	64.62	67.86

Table 4.2: Kinetics → AVA: Evaluation results on Kinetics (source domain).

Method	bend/bow	lie/sleep	run/jog	sit	stand	walk	mAP
Baseline w/o DA	33.66	54.82	56.82	73.70	80.56	**75.18**	62.46
Rotation	25.53	58.86	55.05	72.42	79.84	68.49	60.03
Clip-order	28.24	57.38	56.90	69.54	77.10	74.68	60.64
GRL	24.99	48.41	59.89	68.68	78.79	71.38	58.69
Pseudo-labeling	30.74	56.20	55.09	73.53	**80.84**	72.44	61.47
Mixing (w/ resize)	**34.65**	56.50	60.19	70.80	79.17	74.75	62.68
DA-AIM (w/ resize)	33.79	**59.27**	**62.16**	71.67	79.90	75.13	**63.65**
Mixing (w/o resize)	33.07	55.87	60.69	72.51	79.43	73.05	62.44
DA-AIM (w/o resize)	32.18	57.70	59.42	**74.03**	80.73	74.38	63.07

Table 4.3: Kinetics → AVA: Evaluation results on AVA (target domain).

Mixing is the only DA technique that can boost performance on source domain. In the augmented mixed video clips, some action instances are from source domain but other objects and background are from target domain. Hence, the action labels of those action instances should not correlate with other objects or background. In this way, the model is forced to focus more on the foreground action content rather than extracting cues from specific discriminative objects or background scene, which often helps to overcome overfitting on the source domain. However, mixing itself can barely promote the model to learn from target domain. The improvements on the target domain are restricted. Since mixing only utilizes the ground-truth labels to compute final loss,

Method	touch	throw	take a photo	mAP
Baseline w/o DA	62.80	**70.06**	98.29	77.05
Rotation	61.60	68.07	98.54	76.07
Clip-order	63.20	68.58	97.92	76.57
GRL	62.74	67.99	98.75	76.49
Pseudo-labeling	58.94	60.96	97.91	72.60
Mixing (w/ resize)	63.29	69.40	98.74	77.14
DA-AIM (w/ resize)	61.31	66.33	97.99	75.21
Mixing (w/o resize)	**63.80**	69.06	**98.83**	**77.23**
DA-AIM (w/o resize)	59.14	65.78	98.14	74.35

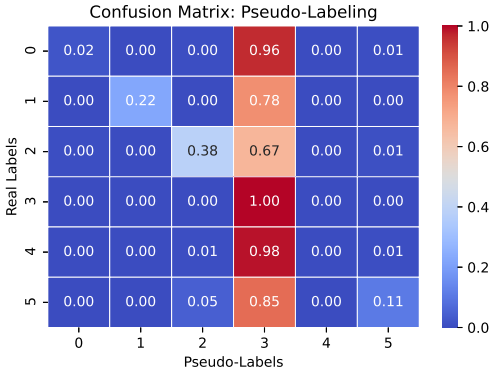
Table 4.4: Kinetics → Armasuisse: Evaluation results on Kinetics (source domain).

Method	touch	throw	take a photo	mAP
Baseline w/o DA	34.12	32.91	27.42	31.48
Rotation	30.12	34.58	25.39	30.03
Clip-order	28.28	32.30	29.93	30.17
GRL	25.79	**39.71**	28.90	31.46
Pseudo-labeling	29.97	28.10	29.82	29.30
Mixing (w/ resize)	32.27	32.48	30.37	31.71
DA-AIM (w/ resize)	**34.38**	35.65	**39.84**	**36.62**
Mixing (w/o resize)	33.00	29.79	29.26	30.68
DA-AIM (w/o resize)	33.67	38.06	32.83	34.85

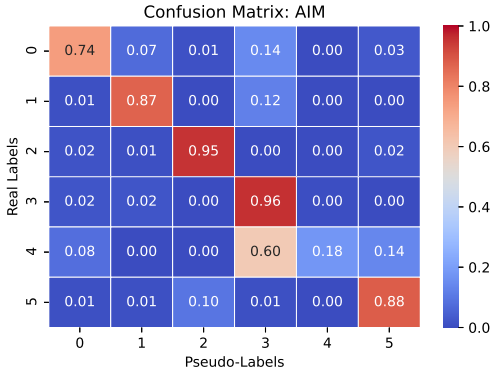
Table 4.5: Kinetics → Armasuisse: Evaluation results on Armasuisse (target domain).

which makes the loss rely heavily on the contents from source domain while contents from target domain only have few impact.

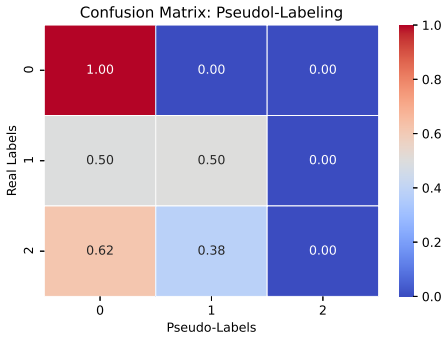
Conversely, pseudo-labeling worsen the performance on both source and target domain compared to baseline experiment without any DA techniques. We observe that the pseudo-labels created by the teacher network tend to bias towards easy-to-predict classes. Fig.4.4 (a) and (c) illustrate the confusion matrices of pseudo-labels created during the last epoch of training. In Kinetics → AVA experiment pseudo-labels bias towards class *sit* and in Kinetics → Armasuisse experiment pseudo-labels bias towards class *touch*. Similar phenomenon is identified in earlier works applying pseudo-labelling to UDA for semantic segmentation tasks [68, 87].



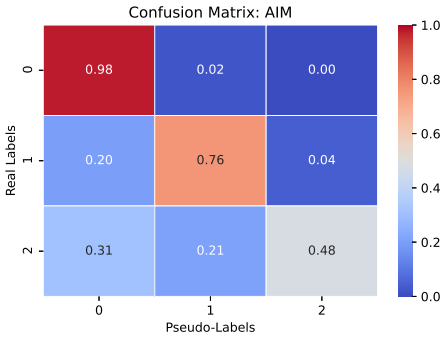
(a) Kinetics → AVA: Pseudo-labeling



(b) Kinetics → AVA: DA-AIM



(c) Kinetics → Armasuisse: Pseudo-labeling



(d) Kinetics → Armasuisse: DA-AIM

Figure 4.4: Confusion matrix of pseudo-labels during last epoch training in (a) Kinetics → AVA: Pseudo-labeling (b) Kinetics → AVA: DA-AIM (c) Kinetics → Armasuisse: Pseudo-labeling (d) Kinetics → Armasuisse: DA-AIM experiments.

DA-AIM outperforms other DA techniques on target domain for both Kinetics → AVA and Kinetics → Armasuisse benchmarks. The above mentioned drawbacks of mixing and pseudo-labeling can be redressed by integration. Taking pseudo-labels into consideration during loss computation push the network to learn domain-invariant features that apply to target domain classification as well. On the other hand, replacing parts of the pseudo-labels by parts of the ground-truth

labels incredibly addresses the bias issue of pseudo-labels. The confusion matrices of pseudo-labels created by DA-AIM are present in Fig.4.4 (b) and (d). DA-AIM achieves 63.65 mAP on target domain Kinetics $\rightarrow$ AVA benchmark compared with 62.46 mAP of baseline experiment. The improvements of average precision for class *lie/sleep* and class *run/jog* are more than 5%. Meanwhile on Kinetics $\rightarrow$ Armasuisse benchmark, DA-AIM increases the mAP from 31.48 of baseline experiment to 36.62. The improvements of average precision for class *take a photo* exceeds 10%.

4.3.3 Qualitative Results

Apart from the quantitative results above, we provide some qualitative examples from the experiments. Fig.4.5 and Fig.4.6 show examples where DA-AIM can identify difficult classes that baseline fails to do or DA-AIM obtains much better confidence scores.

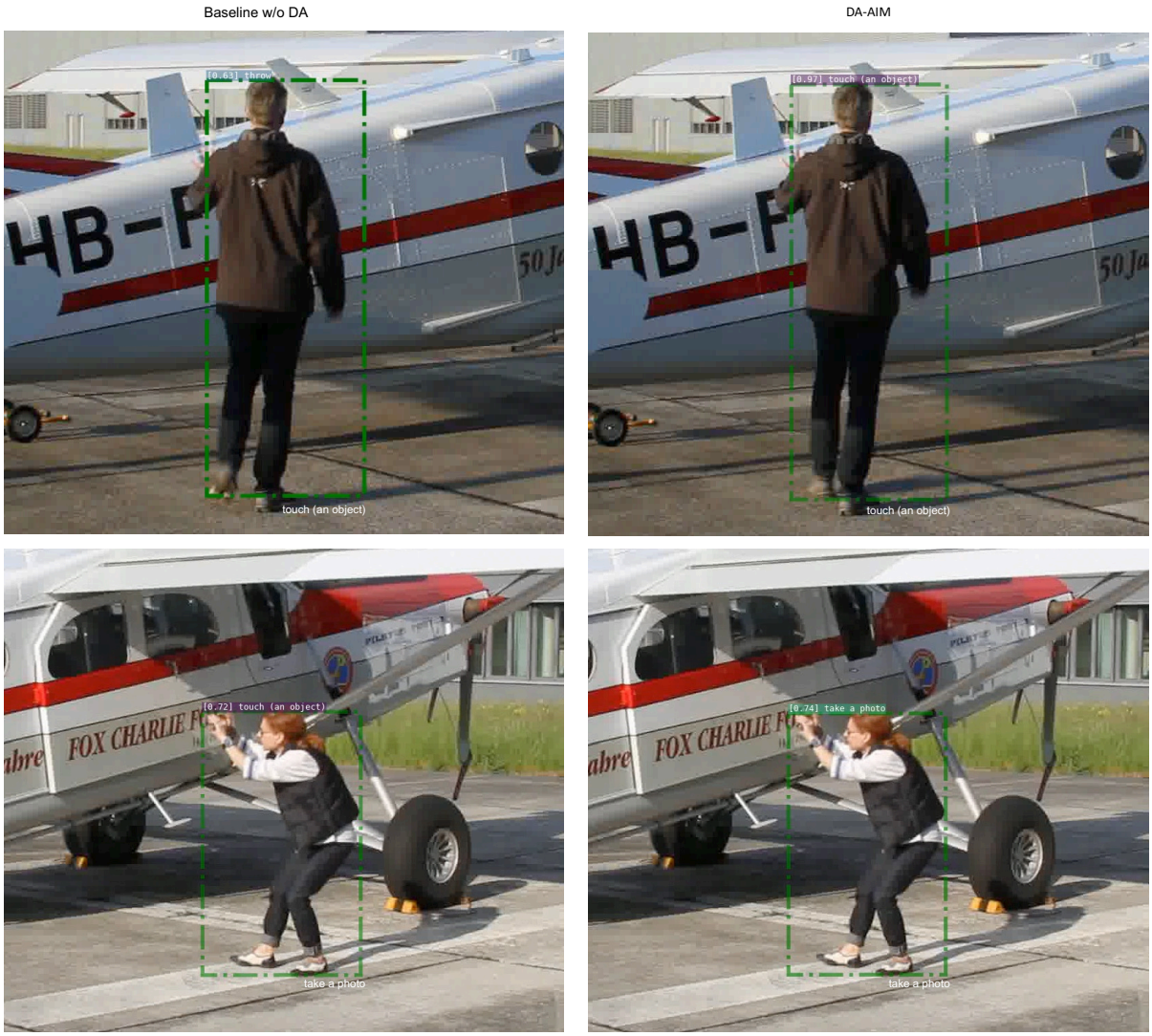


Figure 4.5: Kinetics $\rightarrow$ Armasuisse: Comparison of qualitative results. Only key-frames are illustrated.

Figure 4.6: Kinetics → AVA: Comparison of qualitative results. Only key-frames are illustrated.

Chapter 5

Conclusion and Outlook

We are the first to address unsupervised domain adaptation for action detection. We implemented and systematically analyzed the efficacy of various domain adaptation strategies including self-supervised learning, adversarial learning, self-training and naive cross-domain video mixing. More importantly, we proposed DA-AIM, a novel algorithm tailored for unsupervised domain adaptive action detection. DA-AIM considers the inherent characteristics of action detection and mixes 3D video clips, bounding boxes and labels (ground-truth or pseudo-labels) from source and target domain reasonably. We empirically demonstrated DA-AIM beat other DA techniques on two challenging benchmarks: Kinetics $\rightarrow$ AVA and Kinetics $\rightarrow$ Armasuisse. Compared with baseline experiment without DA techniques, DA-AIM gives rise to an increase of mAP by 1.2% on Kinetics $\rightarrow$ AVA benchmark and 5.2% on Kinetics $\rightarrow$ Armasuisse benchmark. Average precision of class *take a photo* improves over 10%.

Nevertheless, there remain limits to be removed and open questions to be answered. We restricted ourselves in single-label classification setting. When adopted to multi-label classification, DA-AIM demands for a number of new hyper-parameters. We didn't finish the fine-tuning procedure due to time limit. In the meantime, we didn't consider action classes involving more than one action instances, such as class *talking*. This limit can be removed by treating those action classes particularly during mixing. Moreover, there is still great potential to improve the current performance of DA-AIM. Overlapping can be further avoided by introducing randomness into pasted positions of action tubes. The condition to examine overlapping in Eq.3.15 can be replaced by intersection area, so that different level of overlapping can be handled flexibly. Oversampling minority classes during mixing may also enhance the performance, especially when datasets are imbalanced.

Appendix A

Statistics of Datasets

We list some overall statistics of reduced datasets in Tab.4.1, here more detailed sample distributions are displayed in Fig.A.1 - Fig.A.4. The reduction is proceeded only on the training set. After reduction, sample distributions of AVA-6-5000 and Kinetics-6-5000 training sets are almost uniform . AVA-6-5000 contains 5000 training samples for all classes except class *run/jog*, which has only 3280 samples. Likewise, *lie/sleep* is the only class in Kinetics-6-5000 that possesses less than 5000 samples. The original class imbalance of large-scale dataset has been greatly mitigated by the training set reduction.

On the contrary, the class imbalance still exists in Kinetics-3-5000 and Armasuisse-3-5000 after reduction. Samples from classes *take a photo* and *throw* are really rare compared with class *touch*. The difference between the number of samples is especially huge in Kinetics-3-5000.

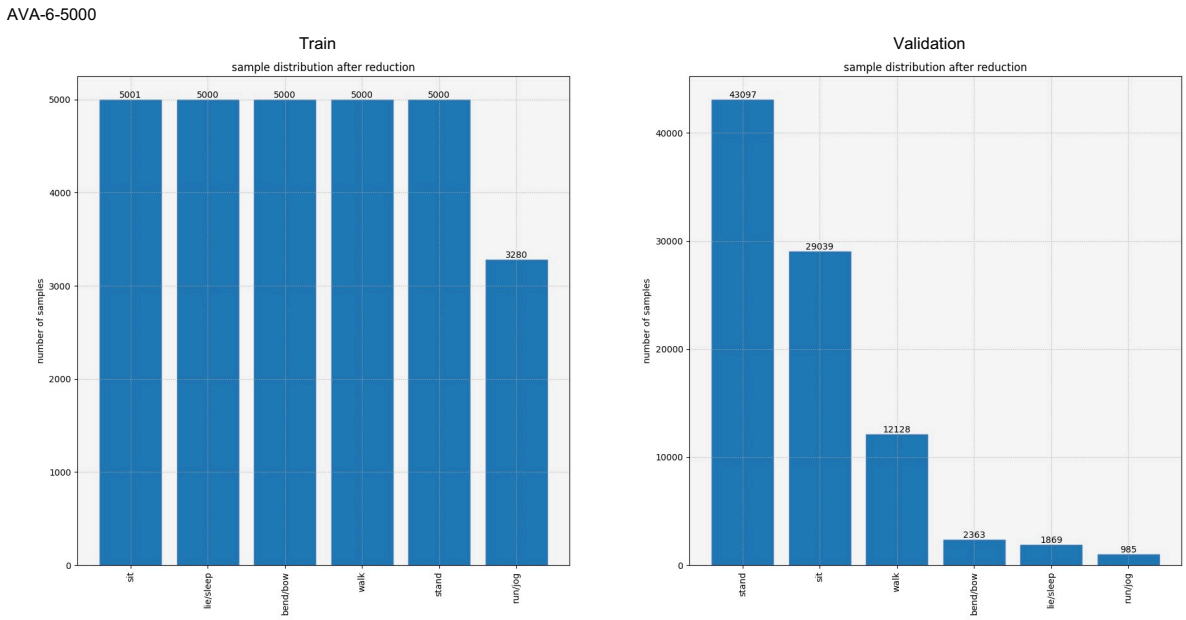


Figure A.1: Statistics of AVA-6-5000.

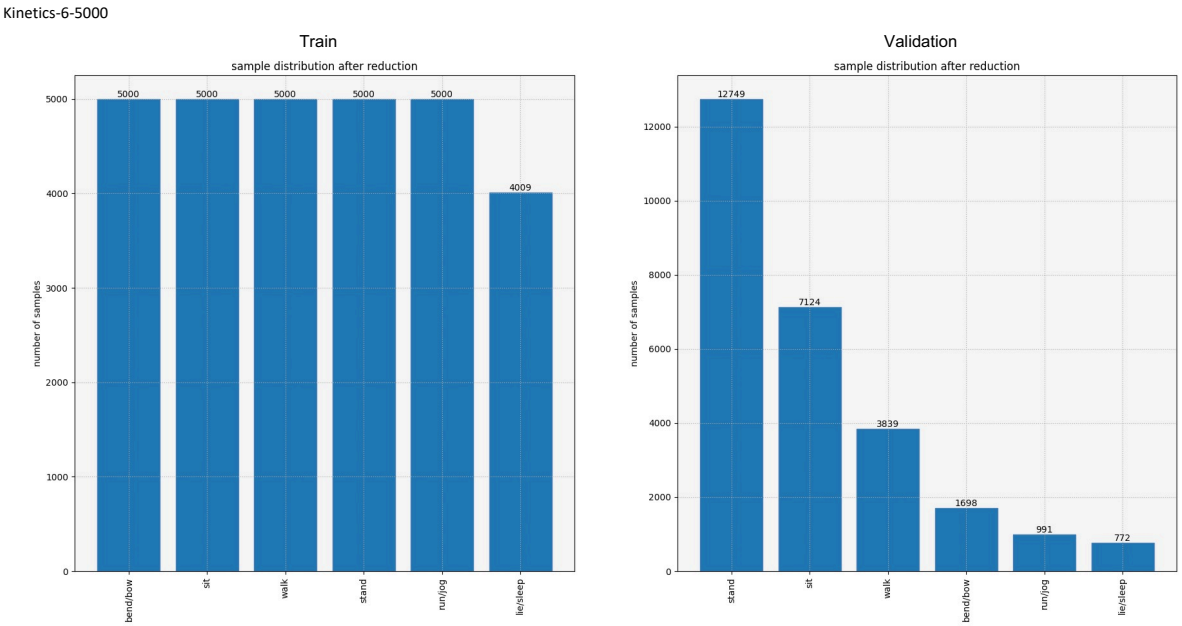


Figure A.2: Statistics of Kinetics-6-5000.

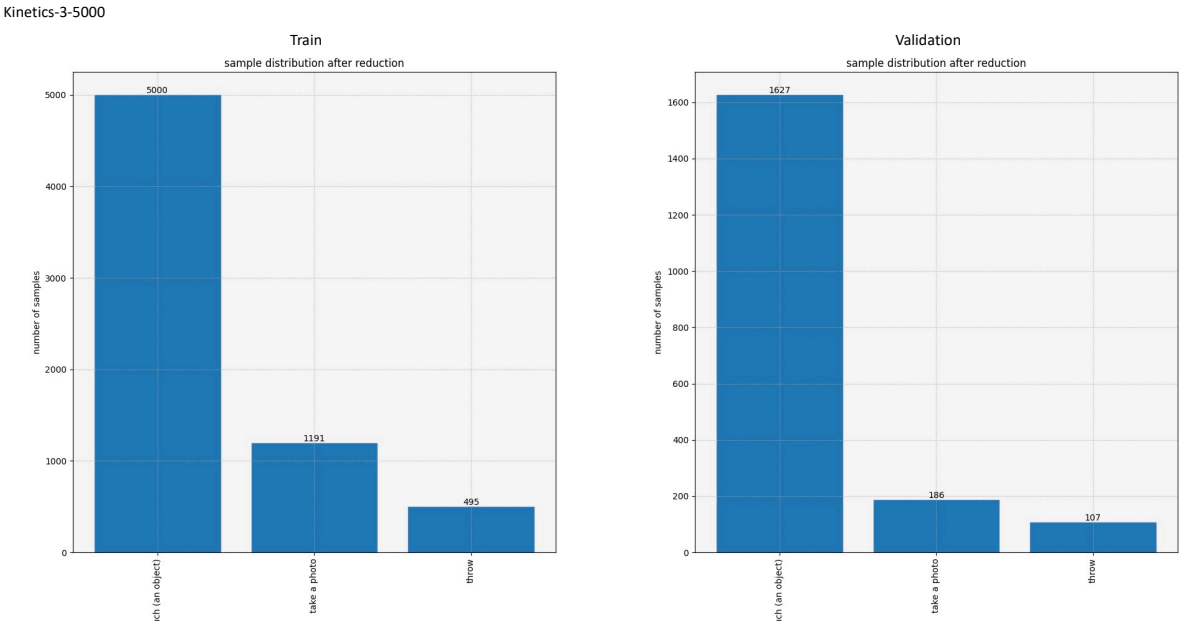


Figure A.3: Statistics of Kinetics-3-5000.

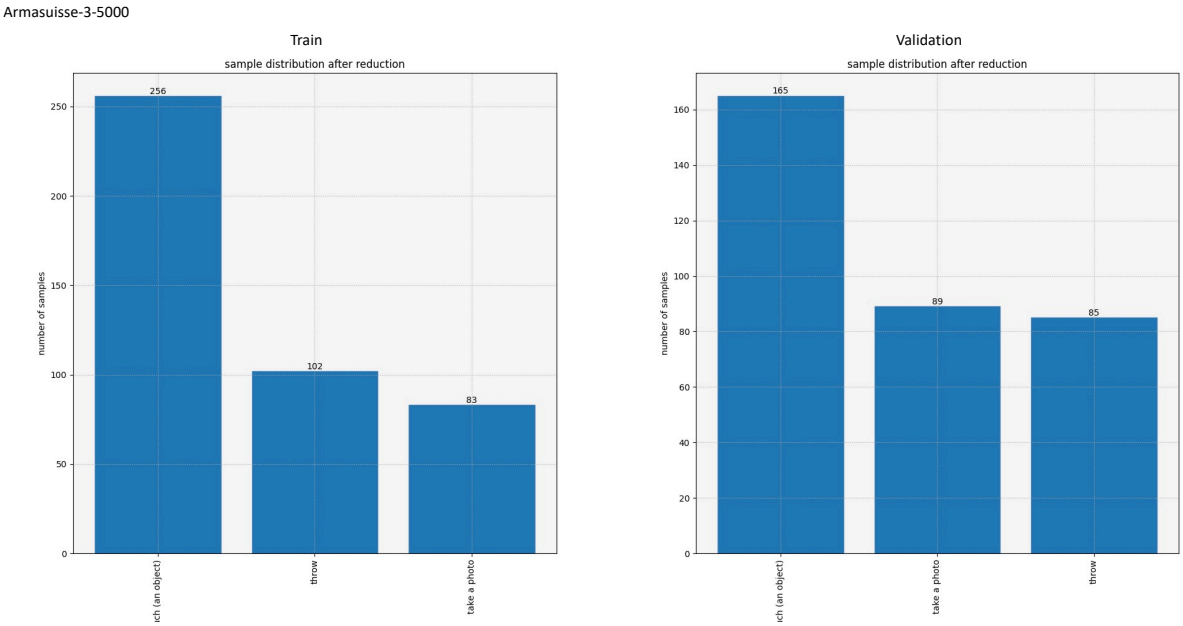


Figure A.4: Statistics of Armasuisse-3-5000.

Bibliography

[1] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *CoRR*, abs/1905.02249, 2019.

[2] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. *CoRR*, abs/1904.11245, 2019.

[3] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. *CoRR*, abs/1903.06864, 2019.

[4] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[5] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. *CoRR*, abs/1808.09347, 2018.

[6] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. *CoRR*, abs/1907.12743, 2019.

[7] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. *CoRR*, abs/1909.13589, 2019.

[8] Minmin Chen, Kilian Q. Weinberger, and John C. Blitzer. Co-training for domain adaptation. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2456–2464, Red Hook, NY, USA, 2011. Curran Associates Inc.

[9] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. *CoRR*, abs/1704.08509, 2017.

[10] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster R-CNN for object detection in the wild. *CoRR*, abs/1803.03243, 2018.

[11] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. *CoRR*, abs/2001.03182, 2020.

[12] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1706–1715, 2020.

[13] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, page 678–695, Berlin, Heidelberg, 2020. Springer-Verlag.

[14] A M Derrington and P Lennie. Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque. *The Journal of Physiology*, 357(1):219–240, 1984.

[15] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *CoRR*, abs/1812.03982, 2018.

[17] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *CoRR*, abs/1604.06573, 2016.

[18] D. J. Felleman and D C van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1 1:1–47, 1991.

[19] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. *CoRR*, abs/1611.06646, 2016.

[20] Geoffrey French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham D. Finlayson. Consistency regularization and cutmix for semi-supervised semantic segmentation. *CoRR*, abs/1906.01916, 2019.

[21] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[22] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[23] Muhammad Ghifary, David Balduzzi, W. Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *CoRR*, abs/1510.04373, 2015.

[24] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[25] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

[26] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[27] Georgia Gkioxari and Jitendra Malik. Finding action tubes. *CoRR*, abs/1411.6031, 2014.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[29] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample problem. *CoRR*, abs/0805.2368, 2008.

[30] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. *CoRR*, abs/1705.08421, 2017.

[31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[33] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. *CoRR*, abs/1907.10343, 2019.

[34] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017.

[35] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[36] Rui Hou, Chen Chen, and Mubarak Shah. An end-to-end 3d convolutional neural network for action detection and segmentation in videos. *CoRR*, abs/1712.01111, 2017.

[37] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. *CoRR*, abs/1910.11319, 2019.

[38] Arshad Jamal, Vinay P. Namboodiri, Dipti Deodhare, and K. S. Venkatesh. Deep domain adaptation in action space. In *BMVC*, 2018.

[39] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *CoRR*, abs/1811.11387, 2018.

[40] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[41] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G. Macready. A robust learning approach to domain adaptive object detection. *CoRR*, abs/1904.02361, 2019.

[42] Jongmok Kim, Jooyoung Jang, and Hyunwoo Park. Structured consistency loss for semi-supervised semantic segmentation. *CoRR*, abs/2001.04647, 2020.

[43] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. *CoRR*, abs/1703.04044, 2017.

[44] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. *CoRR*, abs/1708.01246, 2017.

[45] Ang Li, Meghana Thotakuri, David A. Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *CoRR*, abs/2005.00214, 2020.

[46] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *CoRR*, abs/1811.08383, 2018.

[47] Margaret Livingstone and David Hubel. Segregation of form, color, movement, and depth: Anatomy, physiology, and perception. *Science*, 240(4853):740–749, 1988.

[48] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Domain adaptation with randomized multilinear adversarial networks. *CoRR*, abs/1705.10667, 2017.

[49] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. *Advances in neural information processing systems*, 30, 2017.

[50] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. *CoRR*, abs/2008.12197, 2020.

[51] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016.

[52] Mathew Monfort, Kandan Ramakrishnan, Alex Andonian, Barry A. McNamara, Alex Lascelles, Bowen Pan, Quanfu Fan, Dan Gutfreund, Rogério Schmidt Feris, and Aude Oliva. Multi-moments in time: Learning and interpreting models for multi-action video understanding. *CoRR*, abs/1911.00232, 2019.

[53] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[54] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. *CoRR*, abs/2007.07936, 2020.

[55] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[56] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. *CoRR*, abs/1809.02176, 2018.

[57] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 744–759, Cham, 2016. Springer International Publishing.

[58] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[59] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[61] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *CoRR*, abs/1712.02560, 2017.

[62] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[63] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[64] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision. *CoRR*, abs/1909.11825, 2019.

[65] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017.

[66] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2014.

[67] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition, 2017.

[68] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: domain adaptation via cross-domain mixed sampling. *CoRR*, abs/2007.08702, 2020.

[69] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. *CoRR*, abs/1901.05427, 2019.

[70] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. *CoRR*, abs/1510.02192, 2015.

[71] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *CoRR*, abs/1702.05464, 2017.

[72] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[73] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. DADA: depth-aware domain adaptation in semantic segmentation. *CoRR*, abs/1904.01886, 2019.

[74] Qin Wang, Dengxin Dai, Lukas Hoyer, Olga Fink, and Luc Van Gool. Domain adaptive semantic segmentation with self-supervised depth estimation. *CoRR*, abs/2104.13613, 2021.

[75] Xiaofu Wu, Suofei hang, Quan Zhou, Zhen Yang, Chunming Zhao, and Longin Jan Latecki. Entropy minimization vs. diversity maximization for domain adaptation, 2020.

[76] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017.

[77] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

[79] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.

[80] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. *CoRR*, abs/1705.05498, 2017.

[81] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *CoRR*, abs/1910.13049, 2019.

[82] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. *CoRR*, abs/1904.04663, 2019.

[83] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. *CoRR*, abs/1904.05801, 2019.

[84] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *CoRR*, abs/2003.03773, 2020.

[85] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[86] Junbao Zhuo, Shuhui Wang, Shuhao Cui, and Qingming Huang. Unsupervised open domain recognition by semantic discrepancy minimization. *CoRR*, abs/1904.08631, 2019.

[87] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *CoRR*, abs/1810.07911, 2018.

[88] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training, 2019.