Domain Adaptation for Egocentric Action Recognition

Elena Di Felice
s303499@studenti.polito.it

Aurora Gensale
s303535@studenti.polito.it

Giuseppe Intilla
s297641@studenti.polito.it

***Abstract*—In this paper we explore different ways to improve the most popular Egocentric Action Recognition (EAR) architectures, combined with some Domain Adaptation strategies, using EPIC-Kitchens [10], the largest dataset available for EAR.**
We start with the implementation of two basic models specific for the task of action recognition, which are Two-Stream Inflated 3D ConvNets (I3D) and Temporal Shift Module (TSM). In particular we use two different sampling strategies for the modalities we are considering, which are RGB and Flow. Then, in order to obtain better results, we introduce different Temporal Aggregation strategies on top of the features extracted from the architectures mentioned above. Finally, we build a robust model able to complete the task even with different source and target domains performing Domain Adaptation with one of the latest innovations in the field, Temporal Attentive Adversarial Adaptation Network (TA^3N). Our main contribution is extending the potential of TA^3N, which was originally implemented to work with single modalities, making the architecture able to process multimodal inputs.
Code available here.

I. Introduction

Egocentric Action Recognition is a computer vision task that aims to recognize an action taking place in a video that has a first person point of view. In recent times there has been a growing interest in this field [3], [6], [9], [11], [12], due to the availability of wearable devices (like GoPros) that can record these kind of videos (Fig. 1). This allowed the collection of large amount of data that can be used to develop new algorithms to perform this task. EAR has many applications in augmented reality, robotics and human-computer interaction, and has also been recently proven to be valuable for human-to-robot imitation learning. Compared to the action recognition task using third-person video input, the egocentric action recognition task has extra challenges: fast camera movement, large occlusions, background clutter and lack of large-scale dataset. In this work, we begin by implementing basic models for the action recognition task (I3D and TSM) and consider them as our baseline. We then proceed to implement specific solutions for the task of action recognition in general (Temporal Aggregation) and also for the problems presented by dealing with videos with a first-person point of view. In this context we focus on solving the problem known as "environmental bias" [13] using Domain Adaptation. The network's strong dependency on the context in which the activities are recorded causes this issue since it makes it difficult for the network to recognize actions when they are carried out in unseen environments. From a practical point of view, when we perform Domain Adaptation, we select a *source* dataset, on which we perform the training, and we want to secure good accuracy when testing on the *target* dataset, that can be significantly different from the source. To do so we implement the basic TA^3N, one of the latest innovations in the field. We compare the performance when using RGB and Flow independently, but we also provide our contribution by implementing multimodal TA^3N. To perform our experiments we exploit EPIC-Kitchens, the largest dataset available. It contains a collection of egocentric videos recorded by 32 participants in their native kitchen environments while performing non-scripted daily activities. This includes actions collected in a variety of environments, i.e., kitchens, thus allowing to study the domain shift across multiple domains. Since the dataset contains several modalities, we were able to test how the results vary when one is used rather than the other, using in particular RGB and Flow.

Fig. 1: An example taken from EPIC-Kitchens dataset, best viewed in color

II. Related work

We divide the related works into three groups: works that implemented specific architectures for Action Recognition, works that introduced strategies for temporal aggregation and works that proposed methods to perform Domain Adaptation.

A. *Action Recognition*

One of the distinctive differences between information in a single image (task of image recognition) and in a video (task of action recognition) is the temporal information. For this reason, it was necessary to implement specific architectures that are able to deal with this new kind of input. The main

architectures utilized in this context, which are generally inherited from third-person literature, divide into two categories: methods based on 2D convolution [9], [15]–[20], [27], and method based on 3D ones [21]–[28]. For our work we take in consideration two models:

- *I3D:* this method was proposed in [2]. The authors propose to add the temporal information by inflating all the filters and pooling kernels of a conventional 2D CNN used for image recognition, going from square filters NxN to cubic filters NxNxN.
- *Temporal Shift Module:* this model was proposed by [1] to overcome the problem of high computational cost that 3D CNN-based methods require. The authors idea is to perform shift operations in order to achieve the performance of a 3D CNN but maintaining 2D CNN's complexity.

In this work we implement both these methods with two different kinds of samplings, and compare the results of the two architectures applied to both individual and fused modalities.

B. *Temporal Aggregation*

Recognizing the action being performed in a video based only on frames taken in a scattered manner is difficult if the temporal relationship between them is unknown. For CNNs, reasoning about temporal relations is very challenging but there are several works that implemented strategies to overcome this problem. In particular for our study we consider two strategies for temporal aggregation:

- *Average Pooling (AVG):* aggregates the frame-level feature vectors to form a single video-level feature vector for each video. The pooling is performed along the temporal direction to generate video-level feature vectors.
- *Temporal Relation Network (TRN):* this network module, that enables temporal relational reasoning in neural networks, was proposed by [4]. Instead of sampling dense frames and convolving them, TRN sparsely samples individual frames and then learns their causal relations to efficiently capture temporal relations at multiple time scales.

In this work we implement these two temporal aggregation modalities, adding them on top of I3D and TSM models, and compare the results obtained when RGB or Optical Flow are used.

C. *Domain Adaptation*

The goal of Domain Adaptation (DA) is to train a neural network on a source dataset and secure good performance on a target dataset that is significantly different from the source (Fig. 2). At the state of the art there are three different types of domain adaptation, which are supervised, semi-supervised and unsupervised. We focus on the harder task of Unsupervised Domain Adaptation (UDA), meaning that we do not have labelled sample points in the target domain. We can divide UDA approaches into discrepancy-based methods,

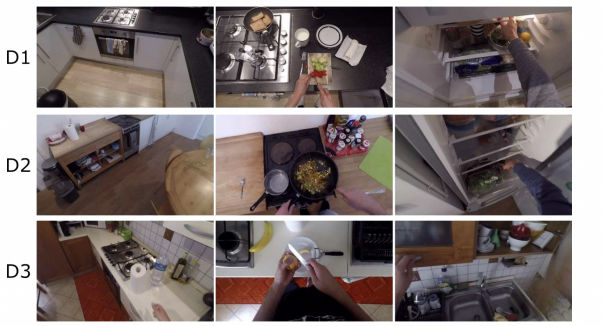

Fig. 2: Three different domains from EPIC-Kitchens dataset that will be used to perform Domain Adaptation, best viewed in color

which explicitly minimize a distance metric among source and target distributions [29]–[31], and adversarial-based methods [32], [33], often leveraging a gradient reversal layer [34]. One of the latest implementations for the task of domain adaptation for action recognition is TA^3N, an adversarial-based approach proposed by [5]. The innovation of TA^3N is the introduction of an attention mechanism that take into account the domain distribution discrepancy and, in this way, it is able to focus on the temporal dynamics that contribute more to the overall domain shift. This leads to a more effective temporal alignment. In this work we compare the performances of the three different adversarial layers (spatial, temporal, relational) applied individually with the performance of TA^3N. We first run the models using only one modality (RGB, Flow) and then extend TA^3N to operate on multiple modalities (RGB and Flow).

III. Method

Here we describe our main contributions step by step.

A. *Basic Architectures*

First we implement, like in [7], two of the main architectures for EAR discussed before: I3D and TSM. We use I3D pre-trained on Kinetics with backbone BNInception and TSM pre-trained on ImageNet with backbone ResNet50. In this stage we test the pre-trained networks onto the split of EPIC-Kitchens proposed in [8]: we compare the results for every modality on three domains that correspond to three different kitchens. This will be useful later in domain adaptation tests. Each subset of the dataset contains 8 verb classes since we perform classification only on verb classes. The fundamental step here is the implementation of the sampling strategy. We develop different samplings for each modality, since, like explained in [9], different time offsets applied to different modalities give the best results. We introduce now two sampling strategies:

- *dense sampling:* taking consecutive samples in order to give the idea of temporal progression of the action. It is used in I3D;

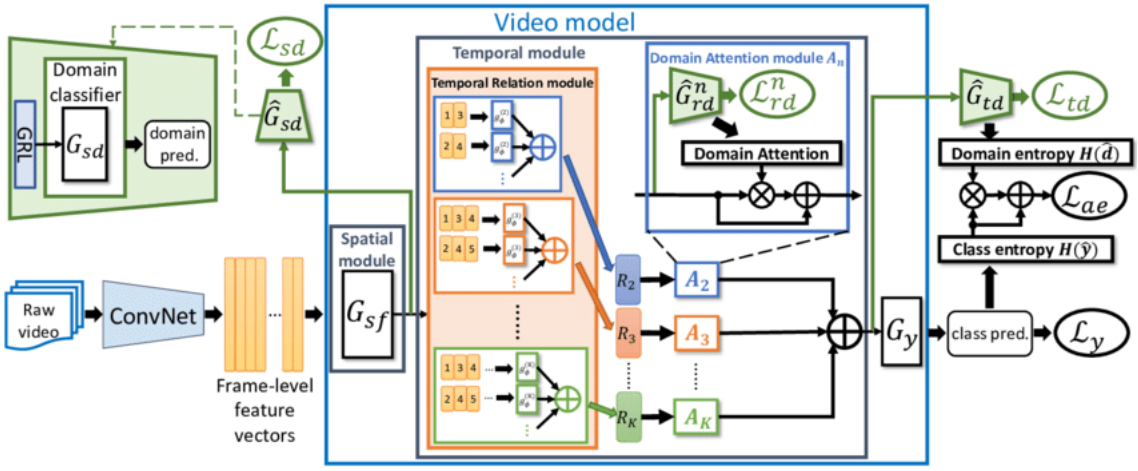

Fig. 3: The detailed architecture of TA3N proposed in [5], best viewed in color

- *uniform sampling:* selecting samples uniformly along the video in order to render the global meaning of the segment. It is used in TSM.

To better explain how samplings work let $\mathcal{V}$ be an action video with n frames and consider one sampling at the time.

Dense sampling: We select k frames randomly within $\mathcal{V}$ and we use them as starting points. Now, beginning with one of these points, we perform dense sampling selecting 16 frames. We refer to these sequences of frames as *clips*. So we end up with k clips by 16 frames. We used two different variations of dense sampling based on the modality:

- *Flow:* dense sampling is performed selecting 16 consecutive frames for each clip;
- *RGB:* the frames within each clip are not selected consecutive but with a fixed step of 2.

We have this difference because we have seen that Optical Flow perform better selecting consecutive frames since it capture the progressive sense of an action.

Uniform sampling: following the strategy of [1] we divide $\mathcal{V}$ in k intervals. Within each interval we select the mid frame. In this way the whole video segment is explored and we end up with k frames.

In the second part of this step we perform *multimodal fusion*, testing the networks with both input from RGB and Optical Flow. Here we perform late fusion: every stream is trained independently and then the two predictions are averaged in order to obtain the final label. The sampling strategies remain the same.

B. Temporal Aggregation

In this part we take advantage of the pre-extracted features obtained by convolutional layers of the architectures in subsection A. We use them to train and test a classifier using different temporal aggregation strategies:

- average pooling;
- temporal relation network [4].

In particular we use Average Pooling on top of the features obtained by I3D and TRN on the features of TSM. Our classifier is a simple linear layer with batch normalization. Here we train and test only on the single modalities. We then perform a basic grid search on our model's parameters using bayesian optimization tools. The details will be explained in section 4.

C. Domain Adaptation

In this phase we introduce Domain Adaptation for EAR. In particular we implement the architecture TA^3N [5], discussed in section 2, capable of recognizing actions independently from the domain, and then we train and test it on the features of subsection A but with different domain shifts. TA^3N is composed of two main components, adversarial layers and the domain attention layer (Fig. 3).

Adversarial layers: Domain classifiers that are used to discriminate whether the data is from the source or from the target domain. The purpose is to make the domain classifier unable to distinguish between source and target domains, so ideally we want to maximize the loss that indicates the ability in distinguishing between domains. In practice we do something that is the equivalent: we insert a Gradient Reversal Layer (GRL) that invert the direction of the gradient, and during adversarial training the parameters of the domain classifier are learned so that the domain loss is minimized. We refer to adversarial layers as the combination of a GRL and a domain classifier. There are three layers of domain classifiers:

- spatial: uses the first level features obtained by the ConvNet;
- video: performs domain classification on the aggregated video-level features;
- relation: it's applied on the relation-level features obtained by the TRN.

Each one of these classifiers is linked to a different domain loss.

Domain attention layer: The attention mechanism is needed to give more importance to features that are more informative for the task of domain adaptation. In fact, not all the features are equally important to align and in order

	RGB				Flow				Fusion			
	D1	D2	D3	Mean	D1	D2	D3	Mean	D1	D2	D3	Mean
I3D	51.49	61.33	60.57	57.80	54.02	60.27	56.06	56.78	54.25	63.20	65.81	61.09
TSM	52.64	69.87	67.56	63.36	58.16	69.07	66.32	64.52	58.62	75.60	74.23	69.48

TABLE I: The comparison of accuracy (%) between I3D and TSM using single modalities and multimodal fusion in the different domains.

to effectively align overall temporal dynamics we want to focus more on aligning the local temporal features which have larger domain discrepancy. In order to focus on aligning the features that are more domain discriminative, the domain attention mechanism assigns different weights to the features. It's introduced the domain attention loss that compose, together with the classification loss and the domain losses, the overall objective loss function. Let the subscript S denote the source data and $S \cup T$ the union between source and target data, the overall loss function can be written as:

$$\mathcal{L} = \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{L}_c^i + \frac{1}{N_{S \cup T}} \sum_{i=1}^{N_{S \cup T}} \gamma \mathcal{L}_a^i - \frac{1}{N_{S \cup T}} \sum_{i=1}^{N_{S \cup T}} \left(\beta^s \mathcal{L}_{sd}^i + \beta^t \mathcal{L}_{td}^i + \beta^r \mathcal{L}_{rd}^i \right)$$

where we define

$$\mathcal{L}_a^i = \left(1 + \mathcal{H}(\hat{d}_i)\right) \mathcal{H}(\hat{y}_i)$$

as the attention entropy loss and $\mathcal{H}(\cdot)$ is the entropy function applied to the predicted domain and classification label. The values β^i and γ represent respectively the weight of the loss for the discriminative layer i and for the attention loss.

We test the architecture on all the domain-shifts comparing results obtained with temporal pooling and with TRN using only RGB features.

D. Multimodal Fusion

In the last step we extend the previous architecture to be multi-modal in order to improve the results obtained before. In particular, taking clue from [9], we perform a mid-level fusion by concatenating the features from RGB and Optical Flow modality. This allows the model to learn the combination of the two modalities together boosting the overall performances.

IV. Experiments and results

A. Dataset

To carry out all the experiments we use the largest dataset available for Egocentric Action Recognition: EPIC-Kitchens [10]. As already stated in the Introduction section, it contains 39,596 action segments recorded by 32 participants performing non-scripted daily activities in their native kitchen environments. All of the sequences have been captured using a head-mounted GoPro. An action is defined by a combination of a verb and a noun and in total there are 125 verb classes and 331 noun classes (heavily-unbalanced).

B. Modalities

In all our experiments we focus on using only two modalities, RGB and Optical Flow, that are publicly available with the dataset. We use them both individually and combined.

C. Training details

Action Recognition: We do not perform any training on the I3D and TSM networks implemented in this step as we are provided with the pre-trained model weights.

Temporal Aggregation: In this second step we use the pre-extracted features obtained by convolutional layers of the architectures in the previous step to train and test a classifier using two different temporal aggregation strategies: Average Pooling and TRN. To find the best parameters to use in the training we perform a basic grid search using bayesian optimization tools. In particular we search for learning rate, batch size, dropout rate and optimizer type for every combination of temporal aggregation strategy and modality. We choose the parameters that give us the best average accuracy among all the domains.

Domain Adaptation: Also in this phase we perform a grid search to find the best parameter configuration; in particular we choose the best parameters configuration based on the average accuracy among all the domain shifts. We perform a different grid search for each combination of temporal aggregation method and layer of domain attention. In this case, in order to optimize the contribution of the discriminative layers, we fix the training parameters to the default values and search for the best configuration of:

- β: a vector containing the weights of the domain losses at different levels;
- γ: the weight of the domain attention loss.

In Fig. 4 it is shown how the performance changes when we vary the value of beta for each one of the three layers.

D. Results

For our study we always use accuracy as evaluation metric.

I3D vs TSM: In table I we report the results obtained for I3D and TSM when using RGB and Optical Flow modalities individually, with the sampling strategies explained in the Method paragraph (subsection A). For what concerns I3D, we can notice that the results that we get when using RGB are better on average than the one obtained with Optical Flow, in contrast to TSM. In general, however, we can see that TSM outperform I3D in all the cases considered. As expected, when we perform multimodal fusion the results improve on both

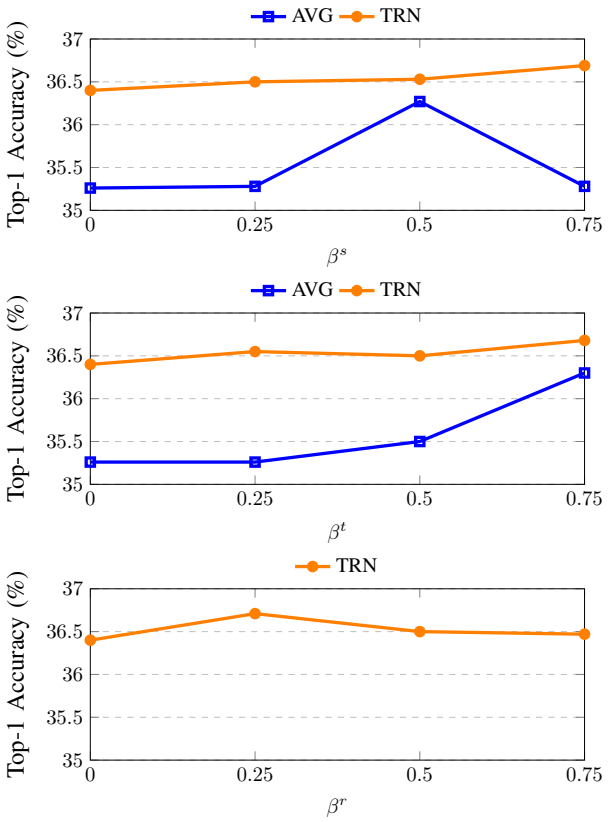

Fig. 4: Difference in terms of performance (average Top-1 Accuracy (%)) based on the value of β with respect to the different discriminative layers, respectively Spatial (β^s), Temporal (β^t) and Relational (β^r).

models of almost 4%, comparing the averages of the three domains considered.

AVG vs TRN: We report in table II the results that we get when performing the methods considered for temporal aggregation, applied on single modalities. It can be seen that TRN has a better average performance and its advantages are more evident on the RGB modality.

	RGB				Flow			
	D1	D2	D3	Mean	D1	D2	D3	Mean
AVG	56.11	61.95	60.72	59.59	60.89	68.94	65.42	65.08
TRN	60.53	73.41	74.50	69.48	60.40	68.76	66.30	65.15

TABLE II: The comparison of accuracy (%) between Average Pooling and Temporal Relation Network in the different domains.

Single layers vs TA^3N: For this step we compare the results of the baseline (no domain adaptation) with the ones obtained when different strategies for domain adaptation are applied. For these results we only consider RGB modality. As before, TRN always outperform Average Pooling when considering the averaged result over the domain shifts. It's clear that adding single domain adaptation layers improves the results with respect to the baseline, and, as expected, when we add all the three discriminators together with the domain attention layer the numbers obtained are even better. Results can be seen at table III.

	AVG	TRN
Source only	35.26	36.47
Spatial discriminator	36.27	36.69
Temporal discriminator	36.31	36.68
Relational discriminator	–	36.71
All discriminators	36.43	36.81
All + Domain attention	36.46	37.39

TABLE III: The comparison of accuracy (%) between single discriminative layers and the complete TA^3N. We report only the average over the domains.

Single modalities vs multimodal fusion with Domain Adaptation: In table IV we can see the comparison between our model with single modalities and its extension to multimodal fusion. As expected, when we use fused modalities there is a general increase. However, in this case the difference between the baseline results and the results related to the application of the complete TA^3N is less significant.

	RGB		Flow		Fusion	
	AVG	TRN	AVG	TRN	AVG	TRN
Source	35.26	36.47	45.62	44.39	49.59	49.79
All + Attention	36.46	37.39	45.96	46.85	49.75	50.42

TABLE IV: The comparison of accuracy (%) between single modalities and multimodal fusion using TA^3N with different temporal aggregation strategies. We report only the average over the domains.

V. Conclusion

We explored the task of Egocentric Action Recognition, starting with elementary architectures and adding tools to improve the results obtained. We demonstrated the importance of Temporal Aggregation when dealing with videos as input, needed to capture the temporal link between frames. We also delved into the task of Domain Adaptation, trying to apply one of the latest innovations in the field (TA^3N) and comparing it with less complex methods such as applying individual adversarial layers. We gave our contribution by making TA^3N multimodal. Looking at the results we can state that the multimodal fusion for Egocentric Action Recognition is necessary to achieve a performance improvement. In this process we have seen the model become increasingly robust and capable of reaching better results, but there are for sure improvements that can be done. Further avenues for exploration may include the introduction of audio as a third modality.

REFERENCES

[1] Lin, J., Gan, C., & Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7083-7093).

[2] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6299-6308).

[3] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9954–9963, 2019.

[4] Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In Proceedings of the European conference on computer vision (ECCV) (pp. 803-818).

[5] Chen, M. H., Kira, Z., AlRegib, G., Yoo, J., Chen, R., & Zheng, J. (2019). Temporal attentive alignment for large-scale video domain adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6321-6330).

[6] Antonino Furnari and Giovanni Farinella. Rolling-unrolling lstms for action anticipation from first-person video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[7] Plizzari, C., Planamente, M., Alberti, E., Caputo, B. (2021). PoliTO-IIT Submission to the EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition

[8] Munro, J., & Damen, D. (2020). Multi-modal domain adaptation for fine-grained action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 122-132).

[9] Kazakos, E., Nagrani, A., Zisserman, A., & Damen, D. (2019). Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5492-5501).

[10] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., ... & Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 720-736).

[11] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. arXiv preprint arXiv:1807.11794, 2018.

[12] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Largescale weakly-supervised pre-training for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 12046–12055, 2019.

[13] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In CVPR 2011, pages 1521–1528. IEEE, 2011.

[14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS'14, page 568–576, Cambridge, MA, USA, 2014. MIT Press.

[15] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision, pages 20–36. Springer, 2016.

[16] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1894–1903, 2016.

[17] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE International Conference on Computer Vision, pages 7083–7093, 2019.

[18] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In Proceedings of the European Conference on Computer Vision (ECCV), pages 803–818, 2018.

[19] Alejandro Cartas, Jordi Luque, Petia Radeva, Carlos Segura, and Mariella Dimiccoli. Seeing and hearing egocentric actions: How much can we learn? In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.

[20] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1102–1111, 2020.

[21] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.

[22] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2620–2628, 2016.

[23] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[24] Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. Multitask learning to improve egocentric action recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.

[25] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 4489–4497, 2015.

[26] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6202–6211, 2019.

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS'14, page 568–576, Cambridge, MA, USA, 2014. MIT Press.

[28] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[29] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In Proceedings of the IEEE International Conference on Computer Vision, pages 1426–1435, 2019.

[30] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3723–3732, 2018.

[31] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In International conference on machine learning, pages 97–105. PMLR, 2015.

[32] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In Proceedings of the IEEE International Conference on Computer Vision, pages 9944–9953, 2019.

[33] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In AAAI, pages 5940–5947, 2020.

[34] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. volume 37 of Proceedings of Machine Learning Research, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.