TA^3(FN)N - Adaptive feature norm for egocentric action recognition

Matteo Berta
Politecnico di Torino
s295040@studenti.polito.it

Umberto Fontana
Politecnico di Torino
s294992@studenti.polito.it

Enrico Porcelli
Politecnico di Torino
s296649@studenti.polito.it

ABSTRACT

Domain shifts in videos is still not well explored and the objective of the learner is to safely generalize into novel environments by mitigating the aforementioned domain shift. In this paper the Adaptive Feature Norm (AFN) has been implemented into a Temporal Attentive Adversarial Adaptation Network (TA^{3}N) architecture in order to complete a egocentric action recognition task, trying to resolve both the problem of domain alignment with the use of TA^3N and the problem of robustness and of the models against the negative transfer issue. Results have shown a small but significant improvement on a subset of the Epic-Kitchens dataset. Code is available at **GitHub**.

I. Introduction

Deep neural networks are very powerful for object and action recognition tasks, however their performance is very sensible to environmental bias. This prevents neural networks to generalise their results to different domains, thus limiting the possibility to utilize them in a real case scenarios. To overcome this problem, recent studies have focused on Domain Adaptation (DA) [11, 3, 12, 13, 14], which consists in training a classifier on a source domain and applying it to a target domain. Whether the target domain is labeled or unlabeled defines if the DA task is supervised or unsupervised. The source and target domains are required to have different but similar distributions, e.g. recognising kitchen related actions but in different kitchens. In this example the distributions of the two domains are similar in that the actions performed in the two kitchens are the same. On the other hand, the dissimilarity between them presents itself in the form of a different environment. Just to name a few plausible examples: different shapes of the knives, fridges with hinges on opposite sides or a difference in the noise that the running water makes. To try and solve this complex classification task, we propose for the first time a Temporal Attentive Adversarial Adaptation Network (TA^{3}N) [2] equipped with an Adaptive Feature Norm (AFN) [16] loss. The first one is an architecture structured to deal with DA problems by exploiting adversarial learning, while the second one is a loss which aims at increasing the transferability of the learned features from the source to the target domain by gradually increasing the norm of the target domain features.

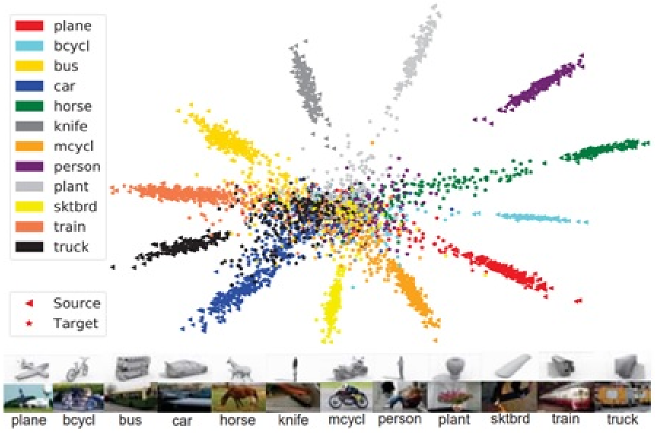

Fig. 1: As illustrated, target samples tend to collide in the small-norm (i.e., low-radius) regions which are vulnerable to slight angular variations of the decision boundaries and lead to erratic discrimination

II. Related Works

A. Egocentric Action Recognition

Recent works have showed the validity of 3D CNN architectures for the egocentric action recognition task [15, 5]. However, these higher performances come with an increased computational cost and sometimes 2D CNNs like TBN [6] and TSN [**TSN**] are preferred, even though they usually coincide with a loss in performance due to the fact that they are not able to keep into account the temporal relations since they process frames individually. Since the temporal relation between frames is a crucial feature in egocentric action recognition tasks, 3D architectures usually have the edge on their 2D counterparts in terms of performances. A popular network that utilizes these CNNs is the Two-Stream Inflated 3D ConvNet (I3D)[1], which is based on 2D ConvNet inflation followed by late fusion of modalities. Another common choice is Temporal Shift Module (TSM)[7], which introduces the possibility to explore relations along the time dimension by shifting the channels, thus favouring the information exchange between neighbouring frames. Temporal Relation Network (TRN)[19] is an aggregation module that allows the network to perform temporal reasoning.

When analysing the results of this field of study, low levels of accuracy should not scare the reader. Advances in action recognition in videos are still very far from the ones made in object recognition on images. This challenge is even harder when dealing with egocentric videos, since the input video is usually defined by sharp movements that increase the complexity of identifying the features.

B. Domain adaptation and domain discrepancy

Domain Adaptation (DA) aims at generalising the results obtained on a source domain to a target domain which is unexplored by the network. The difference between the two domains is expressed by the so-called domain discrepancy, which can be measured in different ways and can be used as a metric to minimize during the training phase. This helps to align the different domains by finding features that are less domain-specific. Methods based on Maximum Mean Discrepancy (MMD) [8, 9, 17, 18]measure the discrepancy as the average difference between the features from the two domains and they learn transferable features by minimizing the MMD of their kernel embeddings. In our project we implement the Adaptive Feature Norm (AFN) loss proposed in [16] by integrating it into a Temporal Attentive Adversarial Adaptation Network (TA^{3}N). This architecture is suited for domain adaptation because it exploits adversarial learning to learn domain invariant features.

III. The proposed method

As shown in paper [16], as the norm of a feature shrinks, this becomes less and less significant for determining the action that is going on in the video. For this reason, being able to adapt the feature norms of the two domains to a large range of scalars should lead to better performances overall. We implemented two versions of the AFN loss: the Hard AFN (HAFN) and the Stepwise AFN (SAFN).
For the Hard variant of AFN, the mean feature norms of source and target samples are constrained to a shared scalar.

$$C_1(\theta_g, \theta_f, \theta_y) = \frac{1}{n_s} \sum_{(x_i, y_i) \in D_s} L_y(x_i, y_i) + \\ + \lambda (L_d(\frac{1}{n_s} \sum_{x_i \in D_s} h(x_i), R) + L_d(\frac{1}{n_t} \sum_{x_i \in D_t} h(x_i), R))$$

which is composed of two terms:

1) *Source classification term*: L_y is the source classification loss and is introduced to obtain the task discriminative features by minimizing the softmax cross entropy on the source labeled samples.
2) *Feature norm penalty*: introduced in order to obtain the domain-transferable features by minimizing the feature-norm discrepancy between the two domains.

The function $L_d(,)$ represents the L2-distance and λ is a hyperparameter to trade off the two objectives. The parameter R is an arbitrary scalar and, as it increases, the accuracy of the model increases.
On the other hand, HAFN can not deal with very large values of R, since this may lead to an explosion of the gradient during the training phase. This is where the SAFN comes in handy. It is an iterative approach based on the goal of learning high-norm features in a progressive way. It is expressed as:

$$C_2(\theta_g, \theta_f, \theta_y) = \frac{1}{n_s} \sum_{(x_i, y_i) \in D_s} L_y(x_i, y_i) + \\ + \frac{\lambda}{n_s + n_t} \sum_{x_i \in D_s \cup D_t} L_d(h(x_i, \theta_0) + \Delta r, h(x_i), \theta)$$

where θ_0 and θ represent the updated and updating parameters of the model in the last and in the current iterations at the step size of Δr. During each iteration, the second penalty in SAFN encourages a feature-norm enlargement at the step size of Δr with respect to individual examples, based on their feature norms calculated by the past model parameters in the last iteration. In this way, SAFN allows for a better trade-off between the objectives of the loss.

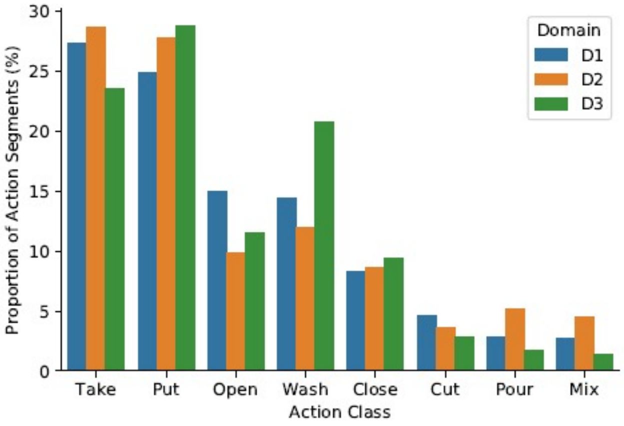

Fig. 2: *Verb distribution in each domain*

IV. Experiments and results

A. Dataset

The dataset used is a split of EPIC-Kitchens [**Epic-Kitchens**] for DA proposed in MM-SADA [4], which comprehends only P01_X, P08_X and P22_X files, which correspond to D1, D2 and D3 domains which represent three different kitchens. The three domains contain an unbalanced number of actions both in the training and test sets. The distribution of the 8 most common verb classes is roughly uniform with respect to the domains, as shown in figure 2

This project exploits the RGB and optical flow modalities for the standard Egocentric Action Recognition section, while for the domain adaptation one it only relies on the RGB

		Verb Accuracy (%)								
Model	**Sampling**	**RGB**			**Flow**			**RGB & Flow**		
		D1	***D2***	***D3***	***D1***	***D2***	***D3***	***D1***	***D2***	**D3**
I3D	**5 clips** **16 frames**	45.29	55.70	57.91	44.14	45.14	44.15	48.74	56.25	56.88
TSM	**5 clips** **5 frames**	54.02	65.38	67.45	58.39	59.56	63.86	58.39	73.55	64.37

TABLE I: *Verb accuracy results for I3D and TSM tested on one single modality at a time and without any temporal aggregations module*

Model	**Sampling**	**Strategy**	**Accuracy RGB (%)**			**Accuracy Optical Flow (%)**		
			D1	***D2***	***D3***	***D1***	***D2***	***D3***
I3D	**5 clips** **16 frames**	**AvgPool**	54.02	61.47	60.88	56.55	69.73	64.68
TSM	**25 frames**	**TRN**	59.31	71.60	73.20	64.37	73.33	72.59

TABLE II: *Verb accuracy results for I3D and TSM together with a temporal aggregation module. One single modality has been used*

TA^{3}N component	**AvgPool**		**TRN**	
	Avg Accuracy	***Avg Gain***	***Avg Accuracy***	***Avg Gain***
Source Only	35.25	-	35.09	-
G_{sd}	35.57	+0.32	35.59	+0.50
G_{td}	35.57	+0.32	35.97	+0.88
G_{rd}	-	-	35.85	+0.76
All G_d **(TA^{2}N)**	35.94	+0.69	36.43	+1.34
All G_d + **domain attention**	36.05	+0.80	36.44	+1.35

TABLE III: *comparison between the average contribution of each component of TA^3*

modality. The study ”EPIC-Fusion: Audio-visual Temporal Binding for action Recognition”[6] shows the importance of audio information for accuracy improvement. In this paper, however, this modality is not explored and is left as a possible improvement for future work.

B. Baseline architectures

In the egocentric action recognition part, the performances of TSM and I3D architectures have been compared. Training was not necessary since pre-trained weights were available for these two models. TSM relies on a ResNet-50 backbone pretrained on ImageNet for image recognition and is able to model temporal relations by shifting the input features along the time dimension, thus giving the model a wider view of what is happening in the video. The second model, I3D, has a BNinception backbone that has been pre-trained on Kinetics.

Two different sampling strategies have been implemented for the two architectures:

- **TSM**: every action sequence has been split into 5 clips and, from each clip, 5 frames have been uniformly sampled.
- **I3D**: a dense sampling strategy has been implemented, meaning that in each of the 5 clips 16 consecutive frames have been selected.

Both models have been tested using first RGB and optical flow separately and then using both modalities together in order to access the importance of multimodal inputs. The results of the first experiment highlight the importance of temporal relation modelling. As shown in table I, the TSM is able to outperform by quite a margin the I3D on every domain and with any of the modalities. Worthy to note is also the high performance of the models when using only the optical flow modality. This result confirms the ones obtained in other papers such as [10] and is due to the crucial role that this modality holds in real life when defining an action. For example the deciding factor when deciding whether we are ”opening a fridge” or ”closing a fridge” is the direction that our hand and the fridge door are following. This information can be understood by the network through the optical flow modality.

C. Temporal aggregation

To improve the performances of the models, two temporal aggregation strategies have been implemented:

- **Average Pooling (AvgPool)**: pooling mechanism that computes the average along the time dimension, applied to the I3D;
- **Temporal Relational Network (TRN)**: temporal aggregation method which enables temporal relational reasoning in neural networks for videos. Applied to TSM.

The results are summarised in table II. Even though the TSM analyzes much less frames due to the sampling strategy associated with this architecture, it is able to systematically outperform the I3D with AvgPool. This is due to the shifting mechanism of the TSM which enables it to learn relations between frames that are further apart with respect to the

Component	Aggregation	Accuracy RGB (%)					
		D1 ->D2	*D1 ->D3*	*D2 ->D1*	*D2 ->D3*	*D3 ->D1*	*D3 ->D2*
Source Only	**AvgPool**	37.44	31.51	32.86	39.76	34.02	35.90
	TRN	38.05	32.24	30.90	39.81	34.28	35.25
G$_{sd}$	**AvgPool**	37.61	31.62	32.86	40.25	34.04	37.04
	TRN	38.63	32.41	30.74	40.76	34.03	36.95
G$_{td}$	**AvgPool**	37.68	31.59	32.86	40.14	33.96	37.15
	TRN	38.01	32.01	33.12	40.97	34.36	37.37
G$_{rd}$	**AvgPool**	/	/	/	/	/	/
	TRN	37.96	32.11	31.57	41.17	34.22	38.05
All G$_d$ (TA^{2}N)	**AvgPool**	37.51	31.78	32.65	40.24	34.02	39.45
	TRN	38.07	32.39	33.02	40.87	34.71	39.54
All G$_d$ + Domain Attention (TA^{3}N)	**AvgPool**	37.66	31.65	32.93	40.25	34.80	39.00
	TRN	37.80	32.41	33.03	40.23	35.24	39.99

TABLE IV: *Results of the TA^{3}N ablation study about the impact of its components*

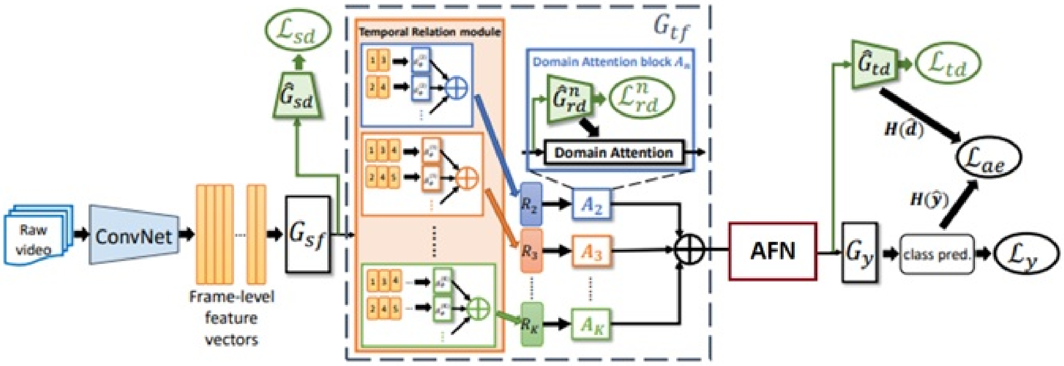

Fig. 3: *TA^{3}N with the implementation of the AFN module*

Hyper-parameters	Values
Initial learning rate	**0.0003**, 0.003
Learning rate decay	DANN
γ	**0.0003**, 0.003, 0.3
β_s	**0.5**, 0.75, 1
β_t	0.5, **0.75**
β_r	0.5, **0.75**
Optimiser	**SGD**, ADA

TABLE V: *Grid search results for TA^{3}N hyperparameters*

ones learned by the I3D. The TSM proves once again to be superior in this task with respect to the I3D. Moreover, the temporal aggregation modules gives to both models a significant increase in performance as the comparison of these results with table I can show.

D. Domain Adaptation

For the domain adaptation section, an ablation study has been performed over the TA^{3}N components. In particular, the different components tested are:

- **Source Only**: first, the TA^{3}N has been tested directly from the souce domain to the target domain
- **Adversarial Discriminators**: used for learning features that are more transferable
- **Domain Attention mechanism**: it utilizes the entropy criterion to generate the domain attention value for each n-frame relation feature as below. This is the component that transforms a TA^{2}N into a TA^{3}N.

The temporal aggregation strategies tested in this experiment are, again, AvgPool and TRN. The performance obtained with training performed only on the source domain is considered the lower bound for the performances of the other components. A worse performance would indicate that the domain adaptation strategy is not working properly. To keep the comparison meaningful, we decided to fix the hyperparameters of the model throughout the experiment. The values of the hyperparameters have been selected through a grid search, and its results are summarised in table V. We decided to tweak the hyperparameters that we thought would have had a bigger impact on the performance of the model. In particular, we tested: the learning rate, the weight of the attentive entropy loss (γ), the trade-off weight for the spatial domain loss (β_s), the trade-off weight for the temporal domain loss (β_t), the

Component	**Parameters**	**Accuragy RGB%**							
		D1->D2	***D1->D3***	***D2->D1***	***D2->D3***	***D3->D1***	***D3->D2***	**Average**	**Average Gain**
TA^{2}N		38.07	32.39	33.02	40.87	34.71	39.54	36.43	/
TA^{2}N + HAFN	R = 0.8, λ = 0.01	38.74	32.49	34.17	41.08	33.56	41.01	36.85	**+0.42**
TA^{2}N + SAFN	Δr = 0.3, λ = 0.1	38.75	32.60	34.17	40.77	34.25	41.41	36.99	**+0.56**

TABLE VI: *Results of the TA^{2}N and impact of AFN*

Component	**Parameters**	**Accuracy RGB%**							
		D1->D2	***D1->D3***	***D2->D1***	***D2->D3***	***D3->D1***	***D3->D2***	**Average**	**Average Gain**
TA^{3}N		37.80	32.41	33.03	39.93	35.24	40.23	36.44	/
TA^{3}N + HAFN	R = 1, λ = 0.01	38.33	32.29	34.01	39.94	34.90	40.42	36.65	**+0.21**
TA^{3}N + SAFN	Δr = 0.3, λ = 0.01	38.41	32.43	33.96	39.86	34.91	40.80	36.73	**+0.29**

TABLE VII: *Results of the TA^{3}N and impact of AFN*

one for the relational domain loss (β_r). Some of the results corroborate the ones found by the authors of TA^{3}N, such as the optimal learning rate being equal to $3x10^{-4}$.

The results obtained with this optimal configuration are summarised in table IV and they follow what we expected before starting the experiment. The addition of any of the adversarial discriminators into the architecture gives it a boost in performance. Of particular interest is the impact of the spatial discriminator, which is much lower with respect to one of the other two discriminators, with the implementation of all three of them simultaneously, the average increase is even bigger.

Among the temporal aggregation modules tested, TRN confirms itself as the best performing one, giving again an average improvement which is superior with respect to the one given by AvgPool. It must be noted that the difference in average accuracy between these two models is much lower with respect to the one obtained in table II. A further improvement is obtained with the domain attention module, which in this case is represented by a cross entropy loss. The best-performing configuration is composed of all the aforementioned components of the TA^{3}N and the average gain with respect to the vanilla architecture is +1.35. This confirms the importance of all of the components of the TA^{3}N architecture and corroborates the results of paper [2].

E. Improvement results

Although the components of TA^{3}N and the temporal aggregation modules revealed to have a meaningful impact on the performance of the network, the best accuracy that we managed to obtain was around 40%. Since we suspect that part of the problem is due to the low separability of verbs when feature norms are low, we implemented an AFN loss into the architecture in order to obtain features that are more peculiar in this egocentric action recognition task. To find the best configuration for this loss, we executed a grid search on its hyperparameters for both the Hard and Step-wise versions. Starting from the HAFN implementation, we experimented with the following choices of parameters:

- **R**: this is the value of the radius. Tested values comprehend R = 0.1, 0.8, 1, 5. The authors of the AFN paper believe that large values of R should lead to improved accuracy. The downside is that in the HAFN setting, R can not be set to an arbitrarily large number since gradient explosion may ruin the weight update process.
- λ: this is the hyperparameter that handles the trade-off between source classification and domain adaptation. Tested values comprehend λ = 0.01, 0.1, 1, 1.5. A very low value of this parameter will result in a model which is unable to generalize its results on a new domain. On the other hand, if λ is too high, the features extracted will be generalizable across domains but they may not be useful for egocentric action recognition, thus leading to poor performances of the network.

The best configuration of the hyperparameters turned out to be the one having R=0.8 and λ = 0.01 for the TA^{2}N architecture and R=1, λ = 0.01 for the TA^{3}N architecture. The accuracy (for TA^{3}N) of the model peaks at R=1, then gradient explosion kicks in, and a drop in performances start to show with bigger values of R. We tested with values of R lower than 1 because the AFN could also succeed by working as a dropout for the features. By, for example, halving all of the norms the AFN could reduce to basically zero some nonessential features.

For the SAFN, the procedure is the same but there is only one hyperparameter to tune. This is Δr and it controls the step size of the feature norm enlargement encouraged by the loss. Tested values are Δr=0.05, 0.1, 0.3, 0.5. The best choice for Δr for both TA^{2}N and TA^{3}N architectures turned out to be 0.3. This parameter usually depends on the specific dataset on which AFN is being applied. For domains that are easy to converge, a value of Δr smaller than 1 is preferable.

With the refined configuration of the AFN losses parameters, we performed a comparison with a standard TA^{3}N architecture to see if the results from [16] about norm manipulation can find confirmation on different architectures. The results are collected in table VII and they highlight the benefits that the AFN implementation can bring to the TA^{3}N architecture. With an average gain across all possible source-target pairs of +0.21% for HAFN and +0.29% for SAFN, we can conclude that the implementation of an AFN loss leads to small but significative improvements in the accuracy of the predictions. To understand whether these results translate to

Parameters	**TA^{2}N**		**TA^{3}N**	
	HAFN	**SAFN**	**HAFN**	**SAFN**
R	0.1, **0.8**, 1, 5	/	0.1, 0.8, **1**, 5	/
Δr	/	0.05, 0.1, **0.3**, 0.5	/	0.05, 0.1, **0.3**, 0.5
λ	**0.01**, 0.1, 1, 1.5	0.01, **0.1**, 1, 1.5	**0.01**, 0.1, 1, 1.5	**0.01**, 0.1, 1, 1.5

TABLE VIII: *Hyperparameters chosen after gridsearch*

simpler architectures, we decided to test both AFN losses also on TA^{2}N and on a ”source only” TA^{3}N. This led to an interesting result which is summarised in tables VI and VII. On simpler networks the optimal results for HAFN are obtained with R=0.8, meaning that the loss is reducing the feature norms instead of increasing them. We think that the reason behind this is that in TA^{2}N and source only TA^{3}N, the loss is computed over a set of features that contains more irrelevant features with respect to the set used at the end of TA^{3}N. This is due to the lack of domain attention and in these cases, the AFN benefits the predictions by working as a dropout layer and reducing to a value close to zero some of the norms of these irrelevant features.

V. Conclusion

Adjusting the results from [16], this paper confirms the superiority of SAFN with respect to HAFN but it also that in some cases HAFN losses can lead to improvements even with small values of the hyperparameter R. When a domain is already well separated, big values for R may only lead to problems during the training phase. Moreover, when using simpler architectures, reducing the norm of the features by setting R to a value below 1 can actually lead to improvements in performance. The AFN in these cases acts like a dropout layer and reduces the norms of the irrelevant features. Future work may include extensive research on the effect of the positioning of the AFN loss with respect to the layers of TA^{3}N.

References

[1] Joao Carreira and Andrew Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In: (2018).

[2] Min-Hung Chen et al. “Temporal Attentive Alignment for Large-Scale Video Domain Adaptation”. In: (2019).

[3] Gabriela Csurka. *Domain Adaptation for Visual Applications: A Comprehensive Survey*. 2017.

[4] Dima Damen et al. “Scaling Egocentric Vision: The EPIC-KITCHENS Dataset”. In: (2018).

[5] Shuiwang Ji et al. “3D Convolutional Neural Networks for Human Action Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013), pp. 221–231.

[6] Evangelos Kazakos et al. “EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition”. In: (2019).

[7] Ji Lin, Chuang Gan, and Song Han. “TSM: Temporal Shift Module for Efficient Video Understanding”. In: (2018).

[8] Mingsheng Long et al. “Learning Transferable Features with Deep Adaptation Networks”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Lille, France: PMLR, July 2015, pp. 97–105.

[9] Mingsheng Long et al. *Unsupervised Domain Adaptation with Residual Transfer Networks*. 2016.

[10] Jonathan Munro and Dima Damen. “Multi-Modal Domain Adaptation for Fine-Grained Action Recognition”. In: (2020).

[11] Sinno Jialin Pan and Qiang Yang. “A Survey on Transfer Learning”. In: *IEEE Trans. on Knowl. and Data Eng.* 22.10 (Oct. 2010), pp. 1345–1359. ISSN: 1041-4347.

[12] Fan Qi, Xiaoshan Yang, and Changsheng Xu. “A Unified Framework for Multimodal Domain Adaptation”. In: *Proceedings of the 26th ACM International Conference on Multimedia*. Association for Computing Machinery, 2018, pp. 429–437.

[13] Kuniaki Saito et al. *Maximum Classifier Discrepancy for Unsupervised Domain Adaptation*. 2017.

[14] Kihyuk Sohn et al. *Unsupervised Domain Adaptation for Face Recognition in Unlabeled Videos*. 2017.

[15] Du Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4489–4497.

[16] Ruijia Xu et al. “Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation”. In: (2019).

[17] Hongliang Yan et al. “Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 945–954.

[18] Werner Zellinger et al. “Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning”. In: (2017).

[19] Bolei Zhou et al. “Temporal Relational Reasoning in Videos”. In: (2018).