EPIC-KITCHENS-100- 2021 Challenges Report

Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, Michael Wray
University of Bristol, UK

Antonino Furnari, Giovanni Maria Farinella
University of Catania, Italy

Davide Moltisanti
NTU, Singapore

Abstract

This report summarises the EPIC-KITCHENS-100 2021 challenges, and their findings. It serves as an introduction to all technical reports that were submitted to the EPIC@CVPR2021 workshop, and an official announcement of the winners.

1. EPIC-KITCHENS-100

In July 2020, EPIC-KITCHENS-100 was released as the next version of the EPIC-KITCHENS dataset. EPIC-KITCHENS-100, like its previous version EPIC-KITCHENS-55, has a number of unique features that distinguished its collection, including *non-scripted* and *untrimmed* nature of the footage captured in participants' *native environments*. The dataset was extended in footage, up to 100 hours of annotated egocentric footage. More importantly, the pipeline for annotations was revised and improved on every step including the pause-and-talk narrator, which increased the density and correctness of the annotations. More details and statistics on EPIC-KITCHENS-100 can be found at [5]. Notably, each submission is requested to provide their level of supervision following the proposed Supervision Levels Scale (SLS) [9].

This report details the submissions and winners of the 2021 edition of the five challenges available on CodaLab: Action Recognition, Action Anticipation, Action Detection, Unsupervised Domain Adaptation for Recognition and Multi-Instance Retrieval. For each challenge, submissions were limited per team to a maximum of 50 submissions in total, as well as a maximum daily limit of 1 submission. In Sec. 2, we detail the general statistics of dataset usage. The results for all challenges are provided in Sec. 3-7. The winners of the 2021 edition of these challenges are noted in Sec. 8.

A snapshot of the complete leaderboard, when the 2021 challenge concluded on the 28th of May, is available at `http://epic-kitchens.github.io/2021#results`.

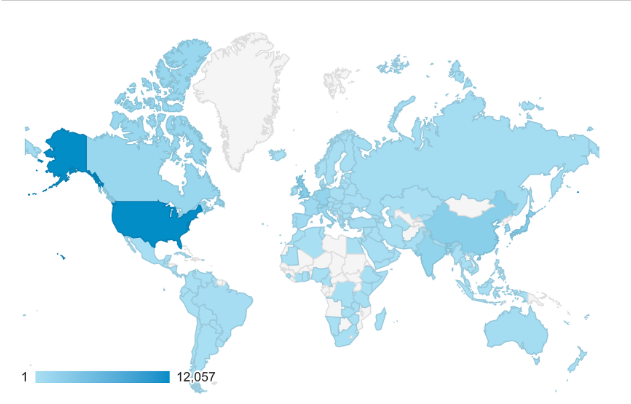

Figure 1: Heatmap of countries based on EPIC-KITCHENS-100 webpage view statistics.

United States	744	China	447	Japan	237
Germany	162	United Kingdom	89	India	82
Unknown	73	France	65	Canada	51
Spain	50	Netherlands	49	Singapore	44
Australia	36	Turkey	33	Italy	33
South Korea	31	Russia	16	Iran	10
Finland	8	Taiwan	7	Austria	7
Malaysia	6	Switzerland	6	Brazil	6
Greece	5	Ireland	4	Thailand	3
Israel	3	Mexico	3	Poland	3
Romania	2	Pakistan	2	Slovakia	2
Vietnam	2	Belgium	2	Croatia	2
Sweden	2	Ukraine	1	Libya	1
Belarus	1	Portugal	1	Algeria	1
Morocco	1	Myanmar	1	Argentina	1

Table 1: Downloads for EPIC-KITCHENS dataset, by country

Details of the 2019 and 2020 challenges are available from the technical reports [8] and [7].

July 6th, 2021
Technical Report

2. Reception and User Statistics

The release of EPIC-KITCHENS-100 was aligned with a 2-hour webinar. The live webinar was attended by 157 researchers, and since then available from https://youtu.be/VraAGAxF9kc[1] and watched more than 1100 times.

We are proud that the dataset's international reach has been further extended over the past year. Fig 1 shows page views of the dataset's website, based on country, while Table 1 lists the number of unique downloads per country, accumulated over the past 3 years and the two versions of the dataset.

In September 2020, we made 5 challenges available in CodaLab and along with each challenge we released codebase with pre-trained models, features and evaluation scripts:

- Action Recognition at https://github.com/epic-kitchens/C1-Action-Recognition: Five pre-trained models were made available using the codebases: TSN, TRN, TBN, TSM and SlowFast, as well as evaluation script.
- Action Detection at https://github.com/epic-kitchens/C2-Action-Detection: with pre-extracted features, a baseline using BMN model and evaluation script.
- Action Anticipation at https://github.com/epic-kitchens/C3-Action-Anticipation with pre-extracted features, RULSTM base model and evaluation script.
- Unsupervised Domain Adaptation for Recognition at https://github.com/epic-kitchens/C4-UDA-for-Action-Recognition with pre-extracted audio-visual features, TA3N model and evaluation script.
- Multi-Instance Retrieval at https://github.com/epic-kitchens/C5-Multi-Instance-Retrieval with features, JPoSE model and evaluation script.

3. Action Recognition Challenge

The Action Recognition challenge has been set similar to previous challenges [3, 16] and has been running since 2019. In both train and test sets, the start and end times of an action are given. Correct recognition of the action includes correctly recognising the 'verb' of the action as well as the main interacting 'noun'. For example, the action 'put plate in sink' would need to be recognised into the verb class 'put' and the noun class 'plate'.

Table 3 shows the entries on the challenge leaderboard for 2021. Methods are ranked based on top-1 action accuracy (noted by arrow), which was used to decide on the overall rank. The top-3 submissions are highlighted in bold. Shaded lines reflect the baseline models. All but one method outperformed the best performing baseline, with the top method improving over the EPIC_TSM_FUSION baseline by +5.32%, +11.43% and +10.80% for VERB, NOUN and ACTION Top-1 Accuracy, respectively. We next describe the contributions of each of the teams, based on their technical reports.

	👥	👥	✉
Action Recognition	32	15	147
Action Anticipation	23	13	64
Action Detection	9	4	10[2]
UDA for Recognition	15	12	160
Multi-Instance Retrieval	6	3	21

Table 2: Number of registered teams, active teams and submissions on CodaLab for the five challenges

3.1. Technical Reports

Technical reports for the Action Recognition challenge, in order of their overall rank on the public leaderboard, are:

SCUT - JD (Rank 1) is the top ranking entry. This work employs an ensemble based approach, consisting of four different SlowFast [10] backbones, where the backbones are trained for predicting nouns, verbs, actions, and noun/verb pairs. The outputs of these models receive a positional encoding, and are then passed through a proposed Transformer [19] based Fusion Block, resulting in six model outputs. During training, each of these outputs are passed through a single fully-connected layer to predict noun, verb and action scores, whilst during testing the scores are averaged across the six outputs.

NUS-HUST-THU-Alibaba (Rank 2) describes an ensemble of Video Vision Transformers (ViViT) [1] and CNNs to achieve a strong performance on the action recognition challenge, coming 2nd place overall. This work focuses on understanding how to best utilise ViViT for the EPIC-KITCHENS-100 dataset, conducting a detailed exploration of augmentations, input resolutions, network initialisation and data quality. As well as ViViT, the work investigates the performance of CNN approaches, CSN [18] and SlowFast [10], observing that both convolution based models outperform ViViT on nouns. Justified by the different strengths of the Transformer and CNN based models, an ensemble is created with multiple ViViT, CSN, and SlowFast models, giving the final predictions.

SAIC-FBK-UB (Rank 3) proposes an ensemble of Gate-Shift-Fuse Networks (GSF), where GSF is an extension of [17], and XViT [2]. GSF is a CNN architecture that ex-

[1] Precut version is also available at https://youtu.be/DXy6lb06DnM

		Submissions		SLS			Overall%			Unseen%			Tail%		
Rank	Team	Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION	VERB	NOUN	ACTION
1	**SCUT - JD**	24	06/01/21	**2.0**	**3.0**	**4.0**	**70.64**	**59.23**	**48.70**	**63.50**	**52.65**	**39.81**	**36.13**	**30.31**	**22.15**
2	**NUS-HUST-THU-Alibaba**	27	05/30/21	**2.0**	**3.0**	**4.0**	**69.25**	**60.31**	**48.49**	**62.92**	**54.11**	**39.49**	**33.95**	**33.06**	**22.70**
3	**SAIC-FBK-UB**	8	005/29/21	**2.0**	**3.0**	**4.0**	**68.16**	**55.49**	**44.82**	**61.97**	**50.56**	**37.47**	**34.58**	**25.92**	**18.96**
4	TCMT	1	06/01/21	2.0	3.0	4.0	68.20	54.97	43.32	61.58	46.93	33.77	37.37	29.83	20.98
5	lyttonhao	15	11/19/20	2.0	3.0	4.0	67.13	53.94	42.08	61.05	49.15	35.18	34.73	24.92	18.19
6	CMU-KLAB	15	05/28/21	2.0	3.0	4.0	63.82	51.12	38.73	57.71	45.62	31.48	36.05	26.26	19.25
7	NUS-UB	12	03/15/21	2.0	3.0	4.0	66.63	48.98	38.59	60.56	43.58	31.63	29.80	15.02	12.97
8	EPIC_TSM_FUSION	2	10/10/21	2.0	3.0	4.0	65.32	47.80	37.39	59.68	42.51	30.61	30.03	16.96	13.45
9	EPIC_SLOWFAST_RGB	3	01/14/21	2.0	3.0	4.0	63.79	48.55	36.81	57.66	42.55	29.27	29.65	17.11	13.45
10	EPIC_TBN_FUSION	7	01/27/21	2.0	3.0	4.0	62.72	47.59	35.48	56.69	43.65	29.27	30.97	19.52	14.10
11	EPIC_TRN_FUSION	2	10/10/20	2.0	3.0	4.0	63.28	46.16	35.28	57.54	41.36	29.68	28.17	13.98	12.18
12	kishore	1	01/15/21	0.0	3.0	3.0	60.56	47.85	34.81	55.64	42.77	29.08	33.10	22.69	15.49
13	EPIC_TSN_FUSION	2	10/10/20	2.0	3.0	4.0	59.03	46.78	33.57	53.11	42.02	27.37	26.23	14.73	11.43

Table 3: Results on EPIC-KITCHENS-100 Action Recognition challenge - 1 June 2021

		Submissions		SLS			Overall%			Unseen%			Tail%		
Rank	Team	Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION	VERB	NOUN	ACTION
1	**AVT-FB-UT**	13	06/01/21	**2.0**	**4.0**	**4.0**	**25.25**	**32.04**	**16.53**	**20.41**	**27.90**	**12.79**	**17.63**	**23.47**	**13.62**
2	**Panasonic_CNSIC_PSNRD**	10	05/27/21	**1.0**	**4.0**	**3.0**	**30.38**	**33.50**	**14.82**	**21.08**	**27.11**	**10.21**	**24.57**	**27.45**	**12.69**
3	**ICL-SJTU**	4	06/01/21	**1.0**	**4.0**	**3.0**	**36.15**	**32.20**	**13.39**	**27.60**	**24.24**	**10.05**	**32.06**	**29.87**	**11.88**
4	NUS_CVML	12	04/16/21	1.0	4.0	3.0	21.76	30.59	12.55	17.86	27.04	10.46	13.59	20.62	8.85
5	RULSTM-FUSION	1	09/30/20	1.0	4.0	3.0	25.25	26.69	11.19	19.36	26.87	9.65	17.56	15.97	7.92
9	EPIC_CHANCE_BASELINE	1	09/30/20	0.0	1.0	3.0	6.17	2.28	0.14	8.14	3.28	0.31	1.87	0.66	0.03

Table 4: Results on EPIC-KITCHENS-100 Action Anticipation challenge - 1 June 2021

tends 2D CNNs to extract spatio-temporal features, whilst XViT is a transformer architecture applied to videos that reduces complexity by only attending to a restricted time window. The authors argue that these two approaches learn distinct, but complimentary features, making them suitable for an ensemble and providing competitive results.

TCMT (Rank 4) proposes an audio-visual transformer of multiple neighbouring actions, in the untrimmed video, with a primary loss that predicts the centre of the temporal window and auxiliary losses for predicting neighbouring actions. Position and modality encodings are incorporate. The reported results combine an ensemble of transformers with different number of actions from 1 to 9 actions. The method showcases the benefit of utilising the untrimmed videos and produces very competitive results in tail classes of verbs and nouns, in line with the approach's motivation.

CMU-KLAB (Rank 6) combine a SlowFast style 2-path network with a Perceiver [13] style transformer. The approach consists of two branches, where the first branch (Main Branch) is a two-path network similar to SlowFast. The third path (Augment Branch) takes the fused output of the first layer of the two paths, providing them as input into a Perceiver model. The outputs from both branches are summed and passed through a softmax layer to produce the final predictions.

NUS-UB (Rank 7) propose an approach to reasoning between mid level features by capturing feature representations corresponding to Regions of Interest (ROI) in the input frames. The proposed approach reasons between these representations through their Transformed Regions of Interest (TROI) module, which localises features corresponding to ROIs, and then uses a transformer architecture to relate the feature representations across space and time.

4. Action Anticipation Challenge

The 2021 edition of the Action Anticipation challenge has been set similarly to the 2019 and 2020 editions. Predictions of upcoming actions (following an observation time) follow the same format as that of the recognition challenge, i.e., the participants provided recognition scores for verbs, nouns and actions. Table 4 shows the results achieved by the participants, along with the public leaderboard rankings. The top-3 submissions are highlighted in bold. Shaded lines reflect the baseline models. All submissions outperformed the baselines. Overall, the submissions have improved over the baselines by $+10.9\%$, $+6.81\%$ and $+5.34\%$ for VERB, NOUN and ACTION Overall Mean Top-5 Accuracy.

We next summarise the contributions of the participants based on their technical reports.

4.1. Technical Reports

Technical reports for the Action Anticipation challenge, in order of their overall rank on the public leaderboard, are:

AVT-FB-UT (Rank 1) The approach uses transformers to model the temporal nature of the video, explicitly considering long-range relationships through attention. The anticipation model is designed to be used with any backbone, but best results are obtained using a transformer-based backbone, which also allows to train the model end-to-end. The model is trained in a multi-task fashion including current action classification and future feature prediction as auxiliary tasks. Final predictions are obtained with an ensemble of different architectural variants of the model based on the proposed approach.

Panasonic_CNSIC_PSNRD (Rank 2) The method adds several optimisations to the Rolling-Unrolling LSTM model. In particular, performance is improved by adding label smoothing based on a pre-trained BERT language model, applying uncertainty modelling loss, class balanced loss, and performing test-time augmentations. The combination of these optimisations and the use of an ensemble model allows to improve the performance of the baseline Rolling-Unrolling LSTM model by a significant margin.

ICL-SJTU (Rank 3) This architecture is based on a Temporal Self-Attention (TSA) model which applies self-attention to features extracted according to different modalities (RGB, optical flow, and object-based features), a Cross-Modality Attention (CMA) module used to aggregate modality-based representations, and a symbiotic attention module used to encourage a coherent prediction of verbs, nouns and actions. The model is trained using equalized cross-entropy to deal with the long-tail distribution of the dataset.

NUS_CVML (Rank 4) The method is based on the analysis of long- and short-term features. Action anticipation is obtained by aggregating such features using computation modules based on non-local blocks. In particular, a coupling block is used to aggregate representations from long- and short-term past representations, whereas a temporal aggregation block combined the obtained information to perform anticipation. Results are further improved by adding Region of Interest (ROI) features extracted from pre-trained TSM and TSN models. The regions of interests are obtained by merging boxes predicted over hands and interacted objects.

5. Action Detection Challenge

The Action Detection challenge follows similar challenges in action detection [11]. Differently from the other challenges, participants have been instructed to consider the test videos as untrimmed, i.e., no temporal segment annotations can be used at test time. The goal is to recognise all action instances within an untrimmed video, as in [12].

Participants provided the detected temporal segments for each test video, along with the predicted verb and noun. Results are reported using mean Average Precision (mAP) considering different Intersection over Union (IoU) thresholds ranging from 0.1 to 0.5. Results are reported on the whole test set. Table 5 shows the results achieved by the participants, along with the public leaderboard ranking. Methods are ranked by Average ACTION mAP. The Top-2 submissions are highlighted in bold. Shaded lines reflect the baseline model. All submissions outperformed the baselines. Overall, the submissions have improved over the baseline by $+11.83\%$, $+15.19\%$ and $+11.71\%$ for VERB, NOUN and ACTION Avarage mAP.

We next summarise the contributions of the participants based on their technical reports.

5.1. Technical Reports

Technical reports for the Action Detection challenge, in order of their overall rank on the public leaderboard, are:

HUST-NUS-THU-Alibaba (Rank 1) The authors train a Video Vision Transformer (ViViT) model for classification and use its features to generate action proposal through Boundary Matching Network (BMN). A sliding window approach is used to generate proposals uniformly, accounting for the varying length of videos. Predicted classification scores are sampled from each proposal and aggregated to output verb and noun labels for each detected action segment. Soft-Non Maximum Suppression is applied to refine the segments.

LocTransformer (Rank 2) The authors propose a single-pass anchor-free method which represent action segments as moments around their centre in the timeline. The model is trained to determine if a given frame t is the centre of an action segment and to predict the start/end offsets from t, which localise the action in the video. The authors adopt and Encoder-Decoder architecture. The Encoder is a Transformer receiving RGB features extracted with a SlowFast backbone. The Decoder is a lightweight MLP. Soft Non Maximum Suppression is utilised to filter the output segments.

6. Unsupervised Domain Adaptation for Recognition Challenge

The Unsupervised Domain Adaptation for Recognition challenge follows the same task as the Action Recognition, however, the labelled videos available during training (source) are collected two years before the videos for testing (target). Due to the different recording times, there is a domain gap between source and target. The different cameras used, the change in location of participants and the differing tools and activities in the domains, are all factors that contribute to the drop in performance when testing on target instead of source. The goal of this challenge is to improve action recognition performance on target with the addition of unlabelled target data during training. This reduces annotation cost as it is assumed unlabelled data is cheap to collect in the target domain compared to annotation.

Participants were given labelled data from 12 participants from EPIC-KITCHENS-55 as the source domain, and unlabelled data from the same 12 participants from the extended dataset collection in EPIC-KITCHENS-100 as the target domain. The narrations, verb, noun and ac-

		Submissions		SLS			Mean Average Precision (mAP)						
Rank	Team	Entries	Date	PT	TL	TD	Task	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.▲
1	**ZiyuanHuang**	7	05/18/21	**2.0**	**3.0**	**3.0**	VERB	**22.77**	**22.01**	**19.63**	**17.81**	**14.65**	**19.37**
							NOUN	**26.44**	**24.55**	**22.30**	**19.82**	**16.25**	**21.87**
							ACTION	**18.76**	**17.73**	**16.26**	**14.91**	**12.87**	**16.11**
2	**LocTransformer**	4	05/22/21	**2.0**	**3.0**	**3.0**	VERB	**18.26**	**17.36**	**16.10**	**12.52**	**10.36**	**14.92**
							NOUN	**15.97**	**14.60**	**13.09**	**10.94**	**8.37**	**12.60**
							ACTION	**8.77**	**8.04**	**7.40**	**6.31**	**5.07**	**7.12**
3	EPIC_BMN_SLOWFAST	1	01/10/21	2.0	3.0	3.0	VERB	11.10	9.40	7.44	5.69	4.09	7.54
							NOUN	11.99	8.49	6.04	4.10	2.80	6.68
							ACTION	6.40	5.37	4.41	3.36	2.47	4.40

Table 5: Results on EPIC-KITCHENS-100 Action Detection challenge - 1 June 2021

		Submissions		SLS			Target Top-1 Accuracy (%)			Target Top-5 Accuracy (%)		
Rank	Team	Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION
1	VI-I2R (chengyi)	20	05/26/21	2.0	4.0	3.0	53.16	34.86	25.00	80.74	59.30	40.75
2	M3EM	33	05/31/21	2.0	3.0	3.0	53.29	35.64	24.76	81.64	59.89	40.73
3	PoliTO-IIT (plnet)	72	06/01/21	2.0	3.0	3.0	55.22	34.83	24.71	81.64	59.89	40.73
4	EPIC_TA3N	4	12/17/20	2.0	3.0	3.0	46.91	27.69	18.95	72.70	50.72	30.53
5	PyKale (xy9)	4	06/01/21	2.0	3.0	3.0	48.45	27.31	18.56	77.31	52.09	33.47
6	EPIC_TA3N_SOURCE_ONLY	2	12/17/20	2.0	3.0	3.0	44.39	25.30	16.79	69.69	48.40	29.06

Table 6: Results on the Unsupervised Domain Adaptation for Recognition challenge

tion labels were not available in the target. Submission included the verb and noun predictions for Target Test and Source Test (optional), which contain videos from the domains not seen during training. Table 6 shows the results achieved from the participants. All submissions outperformed the baseline model trained only on the source domain (EPIC_TA3N_SOURCE_ONLY) and three submissions outperformed the UDA baseline (EPIC_TA3N) by at least 5.76% action accuracy. The majority of submissions did not submit predictions for Source Test, which were optional for submission on CodaLab. This would have provided additional insights into how much each submission improves action recognition in general compared to overcoming the domain gap. We encourage next year's submissions to consider providing the Source Test scores.

6.1. Technical Reports

The technical reports for the Unsupervised Domain Adaptation for Recognition challenge, in order of their overall rank on the public leaderboard, are given in this section. Most solutions exploited multiple modalities for domain adaptation, and the best performing solutions used additional backbone architectures compared to the baselines which used TBN.

VI-I2R (Rank 1) This method is based on the Temporal Attentive Adversarial Adaptation Network (TA3N) [4], augmented with hand-centric features. To locate hands, a Multi-level Entropy Attention Alignment (MEAA) [14] is used to train the detector. Additional hand-labelled hand bounding boxes are used, in comparison to other methods which use pre-trained hand detectors.

M3EM (Rank 2) The main idea is that early fusion between modalities can help improve features across modalities. This is handled by a Multi-Modal Mutual Enhancement Module (M3EM) module. This contains a Semantic Mutual Refinement (SMR) module which finds the most transferable features, and a Cross Modality Consensus (CMC) module which finds the most transferable regions. The best result uses an ensemble which also includes object features, and is guided by a hand feature extractor.

PoliTO-IIT (Rank 3) The approach is based on an audio-visual Relative Norm Alignment Network (RNA-Net) [15] with added flow, applied to source and target separately. Both TA3N and RNA are used for adaptation. Additional losses are incorporated—Temporal Hard Norm Alignment (T-HNA) and Min-Entropy consistency (MEC) to encourage consistency between different modalities.

PyKale (Rank 5) The approach uses transformer encoding on top of the input features to give whole video embeddings. Adversarial domain classification is used after fusing results from each modality, treating each domain separately. Results are not significantly better than TA3N, which shows that just using a standard non-video domain adaptation technique is insufficient.

7. Multi-Instance Retrieval Challenge

The Multi-Instance Retrieval challenge has been introduced as a new challenge of EPIC-KITCHENS-100. Given a query video segment, the goal of video-to-text retrieval is to rank captions in a gallery set, C, such that those with a

		Submissions		SLS			mean Average Precision (mAP)			normalised Discounted Cumulative Gain (nDCG)		
Rank	Team	Entries	Date	PT	TL	TD	T2V	V2T	Avg.	T2V	V2T	Avg.▲
1	**haoxiaoshuai**	6	04/08/21	**2.0**	**3.0**	**3.0**	**38.49**	**49.96**	**44.23**	**51.83**	**55.28**	**53.56**
2	JPoSE	3	01/07/21	2.0	3.0	3.0	38.11	49.91	44.01	51.55	55.51	53.53
3	MLP	6	01/06/21	2.0	3.0	3.0	33.99	42.99	38.49	46.92	50.05	48.49
4	MI-MM	4	05/06/21	2.0	3.0	3.0	23.60	34.83	29.21	42.40	47.18	44.79

Table 7: Results on EPIC-KITCHENS-100 Multi-Instance Retrieval challenge - 1 June 2021

higher rank are more semantically relevant to the action in the query video segment. On the contrary, the goal of text-to-video retrieval is to rank videos given a query caption $c_i \in C$. Differently from the other retrieval challenges, where captions are considered relevant if they were collected for the same video, in this challenge the class knowledge introduced in [6] is used to define caption relevancy (e.g. "put glass" and "place cup" are considered semantically relevant).

Video-to-text and text-to-video results are reported using mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG) on the whole test set.

Table 7 shows the public results achieved by the participants. Methods are ranked by Average nDCG. The Top-1 and only submission is highlighted in bold, whereas shaded lines indicates the baselines models. The submission outperformed the best baselines JPoSE [20] by $+0.38\%$, $+0.05\%$ and $+0.2\%$ for T2V, V2T and Average mAP and $+0.28\%$ and $+0.03\%$ for T2V and Average nDCG.

We next summarise the contribution of the participant based on the technical reports.

7.1. Technical Reports

The technical report of the submission of the Multi-Instance Retrieval challenge is:

haoxiaoshuai (Rank 1) The authors design a loss called Dual Constraint Ranking Loss (DCRL) that simultaneously considers not only the inter-modal ranking constraint, which make semantically similar texts and videos closer, but also the intra-modal structure constraint to preserve both the cross-modal semantic similarity and the modality-specific consistency in the embedding space. The architecture used is based on a video embedding network and a text embedding network that are implemented as a 2 layer perceptron with ReLU. Results are not significantly better than the best baseline JPoSE [20], showing that there is still significant room for improvement in this challenge.

8. 2021 Challenge Winners

Accordingly, Table 8 details the winners of the 2021 EPIC challenges, announced as part of EPIC@CVPR2021 virtual workshop. A zoom capture of the 2021 challenges teams and winners is also in Fig 2.

		Team	Member	Affiliations
Action Recognition	①	SCUT-JD	Zeyu Jiang	South China University of Technology
		(hrgdscs)	Changxing Ding	South China University of Technology
			Canwei Zhang	South China University of Technology
			Dacheng Tao	JD Explore Academy
	②	NUS-HUST-THU-Alibaba	Ziyuan Huang	National University of Singapore
		(ZiyuanHuang)	Zhiwu Qing	Huazhong University of Science and Technology
			Xiang Wang	Huazhong University of Science and Technology
			Yutong Feng	Tsinghua University
			Shiwei Zhang	DAMO Academy, Alibaba Group
			Jianwen Jiang	DAMO Academy, Alibaba Group
			Zhurong Xia	DAMO Academy, Alibaba Group
			Mingqian Tang	DAMO Academy, Alibaba Group
			Nong Sang	Huazhong University of Science and Technology
			Marcelo H. Ang Jr	National University of Singapore
	③	SAIC-FBK-UB	Swathikiran Sudhakaran	Samsung AI Center, Cambridge
		(Sudhakaran)	Adrian Bulat	Samsung AI Center, Cambridge
			Juan-Manuel Perez-Rua	Samsung AI Center, Cambridge
			Alex Falcon	Fondazione Bruno Kessler - FBK, Trento
			Sergio Escalera	Universitat de Barcelona, Spain
			Oswald Lanz	Fondazione Bruno Kessler - FBK, Trento
			Brais Martinez	Samsung AI Center, Cambridge
			Georgios Tzimiropoulos	Samsung AI Center, Cambridge
Action Anticipation	①	AVT-FB-UT	Rohit Girdhar	Facebook AI Research
		(shef)	Kristen Grauman	Facebook AI Research
	②	Panasonic-CNSIC-PSNRD	Yutaro Yamamuro	Panasonic System Networks R&D Lab
		(panasonic)	Kazuki Hanazawa	Panasonic System Networks R&D Lab
			Masahiro Shida	Panasonic System Networks R&D Lab
			Tsuyoshi Kodake	Panasonic System Networks R&D Lab
			Shinji Takenaka	Panasonic System Networks R&D Lab
			Yuji Sato	Connected Solutions Company, Panasonic
			Takeshi Fujimatsu	Connected Solutions Company, Panasonic
	③	ICL-SJTU	Xiao Gu	Imperial College London
		(Shawn0822)	Jianing Qiu	Imperial College London
			Yao Guo	Shanghai Jiao Tong University
			Benny Lo	Imperial College London
			Guang-Zhong Yang	Shanghai Jiao Tong University
Action Detection	①	HUST-NUS-THU-Alibaba	Zhiwu Qing	Huazhong University of Science and Technology
		(ZiyuanHuang)	Ziyuan Huang	National University of Singapore
			Xiang Wang	Huazhong University of Science and Technology
			Yutong Feng	Tsinghua University
			Shiwei Zhang	DAMO Academy, Alibaba Group
			Jianwen Jiang	DAMO Academy, Alibaba Group
			Mingqian Tang	DAMO Academy, Alibaba Group
			Changxin Gao	Huazhong University of Science and Technology
			Marcelo H. Ang Jr	National University of Singapore
			Nong Sang	Huazhong University of Science and Technology
	②	LocTransformer	Chen-Lin Zhang	Nanjing University
		(evangelion)	Jianxin Wu	Nanjing University
			Yin Li	University of Wisconsin-Madison
UDA for Recognition	①	A*STAR	Yi Cheng	A*STAR, Singapore
		(chengyi)	Fen Fang	A*STAR, Singapore
			Ying Sun	A*STAR, Singapore
	②	Tokyo	Lijin Yang	University of Tokyo, Japan
		(M3EM)	Yifei Huang	NUniversity of Tokyo, Japan
			Yusuke Sugano	University of Tokyo, Japan
			Yoichi Sato	University of Tokyo, Japan
	③	Torino	Chiara Plizzari	Politecnico di Torino, Italy
		(plnet)	Mirco Planamente	Politecnico di Torino, Italy
			Emanuele Alberti	Politecnico di Torino, Italy
			Barbara Caputo	Politecnico di Torino, Italy
MI Retrieval	①	IIE-MRG	Xiaoshuai Hao	Institute of Information Engineering, CAS
		(haoxiaoshuai)	Wanqian Zhang	Institute of Information Engineering, CAS
			Dejie Yang	Institute of Information Engineering, CAS
			Shu Zhao	Institute of Information Engineering, CAS
			Dayan Wu	Institute of Information Engineering, CAS
			Bo Li	Institute of Information Engineering, CAS
			Weiping Wang	Institute of Information Engineering, CAS

Table 8: Top-3 Winners - 2021 EPIC-KITCHENS-100 challenges

Figure 2: Organisers and team winners during EPIC@CVPR2021 virtual Workshop, 20 June 2021.

References

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer, 2021. 2

[2] A. Bulat, J.-M. Perez-Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos. Space-time mixing attention for video transformer. *arXiv preprint arXiv:2106.05968*, 2021. 2

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 2

[4] M.-H. Chen, Z. Kira, G. AlRegib, J. Yoo, R. Chen, and J. Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *International Conference on Computer Vision*, 2019. 5

[5] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 1

[6] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proc. ECCV*, 2018. 6

[7] D. Damen, E. Kazakos, W. Price, J. Ma, H. Doughty, A. Furnari, and G. M. Farinella. Epic-kitchens - 2020 challenges report. Technical report, 2020. 1

[8] D. Damen, W. Price, E. Kazakos, A. Furnari, and G. M. Farinella. Epic-kitchens - 2019 challenges report. Technical report, 2019. 1

[9] D. Damen and M. Wray. Supervision levels scale (SLS). *CoRR*, abs/2008.09890, 2020. 1

[10] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019. 2

[11] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Khrisna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. *arXiv preprint arXiv:1710.08011*, 2017. 4

[12] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 4

[13] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver: General perception with iterative attention, 2021. 3

[14] D.-K. Nguyen, W.-L. Tseng, and H.-H. Shuai. Domain-adaptive object detection via uncertainty-aware distribution alignment. In *ACM International Conference on Multimedia*, 2020. 5

[15] M. Planamente, C. Plizzari, E. Alberti, and B. Caputo. Cross-domain first person audio-visual action recognition through relative norm alignment. In *arXiv*, 2021. 5

[16] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 2

[17] S. Sudhakaran, S. Escalera, and O. 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. 2

[18] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017. 2

[20] M. Wray, D. Larlus, G. Csurka, and D. Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 6

Hrgdscs Submission to the EPIC-Kitchens Action Recognition 2021 Challenge

Zeyu Jiang[1] Changxing Ding[1,2*] Canwei Zhang[1] Dacheng Tao[3]
[1] South China University of Technology [2] Pazhou Lab, Guangzhou [3] JD Explore Academy, JD.com
jzy_scut@outlook.com, chxding@scut.edu.cn, cwzhang23@gmail.com, dacheng.tao@gmail.com

Abstract

In this report we briefly describe the technical details of our submission to the EPIC-Kitchens 2021 Action Recognition Challenge. We use a simple fusion architecture to fuse the features extracted from different backbones, different video clips and different recognition tasks. Our method achieves state of the art results on the test set of EPIC-Kitchens 2021 Action Recognition Challenge.

1. Introduction

EPIC-KITCHENS contains action labels structured as verb-noun pairs [1, 2]. 3D CNN employ multi-clip averaging during test-time since the clips should cover the entire video for accurate performance[3].

In this work, we train 4 types of backbone for 4 recognition tasks (only verb, only noun, only action, pairs of verb and noun). Then we extract video clip features from those models. Finally, we devise a simple fusion architecture to fuse the features extracted from different backbones, different video clips and different recognition tasks.

2. Methods

2.1. Feature Extraction

We extract 10 video clip features, which cover the whole temporal length of the video using pretrained backbones. Each backbone is trained for 4 times for different recognition tasks (only verb, only noun, only action, pairs of verb and noun). For each recognition task, we extract 10 video clip features. The shape of the features is number of clips (10) × dimension (2304).

We use the backbones as follow:

Model A SlowFast 16×8, R101+NL[3], which is pretrained on Kinetics-600, is trained with both train and val set. For each recognition task, we denote the features as ModelA_a (indicate action), ModelA_n (indicate noun), ModelA_v (indicate verb) and ModelA_nv (indicate pairs of verb and noun).

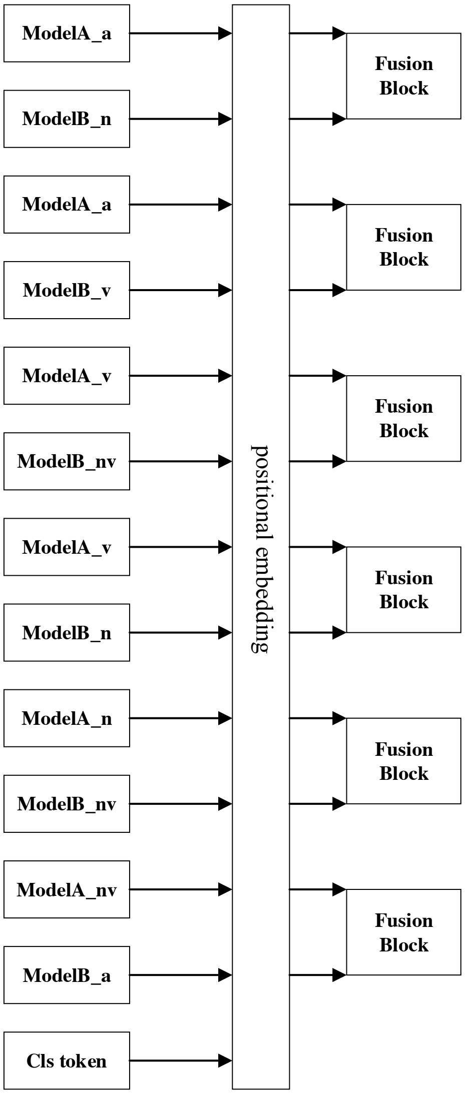

Figure 1. Overview of the fusion architecture.

Model B SlowFast 16×8, R101+NL[3], which is pretrained on Kinetics-600, is trained with only train set. For

[*] Corresponding author.

each recognition task, we denote the features as ModelB_a (indicate action), ModelB_n (indicate noun), ModelB_v (indicate verb) and ModelB_nv (indicate pairs of verb and noun).

Model C SlowFast 8×8, R101[3], which is pre-trained on Kinetics-600, is trained with both train and val set. For each recognition task, we denote the features as ModelC_a (indicate action), ModelC_n (indicate noun), ModelC_v (indicate verb) and ModelC_nv (indicate pairs of verb and noun).

Model D SlowFast R101 backbone[6], which is pre-trained on Kinetics-700, is trained with both train and val set. For each recognition task, we denote the features as ModelD_a (indicate action), ModelD_n (indicate noun), ModelD_v (indicate verb) and ModelD_nv (indicate pairs of verb and noun).

We use the extracted features as input to our feature fusion architecture. For each input, we use the features of 2 types of backbones. For example, (Model A, Model B) indicate we use the features extracted from Model A and Model B as input.

2.2. Overview

Fig. 1 shows overview of the fusion architecture. In this example, we take features extracted from Model A and Model B as input. We add same extra learnable "classification token" to those features, followed by positional embedding and the fusion blocks. Each of the fusion blocks can fuse pair of the features trained for different recognition tasks (like action feature and noun feature, noun feature and verb feature, etc). During training, each fusion block is followed by a classification layer (linear + softmax) to predict noun, verb and action scores together, respectively. At testing time, we average the scores of all fusion blocks to predict more accurate results.

2.3. Fusion Block

The architecture of the fusion block is presented in Fig. 2. We modify the standard transformer encoder[7] to build both transformer block1 and transformer block2. The fusion block can fuse the features extracted from different backbones, different video clips and different recognition tasks.

2.4. Ensemble

We use an ensemble of a set of 6 models as final result for testing set, which are trained by the combination of extracted features from different backbones as input.

The inputs are as follow:

(Model A, Model B), (Model B, Model A), (Model A, Model C), (Model C, Model A), (Model A, Model D), (Model D, Model A)

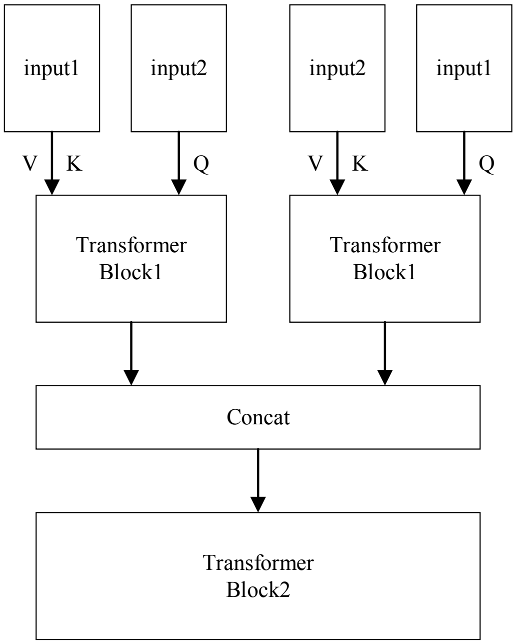

Figure 2. Fusion Block

3. Experiments

3.1. Training Details

During the feature extraction, we make center crop to get a 256×448 region. Before feeding the extracted features into the fusion network, We normalize the extracted features. We train the networks using AdamW[4], using a batch size of 64, mixup[8] of 0.7, label smoothing[5] of 0.4, an l2 weight decay of $5e{-}4$, a dropout of 0.7 before the classifier, and an initial learning rate of $1e-4$. The maximum number of training iterations is set to 35 epochs. A cosine annealing with warm up restart schedule (4 cycles) is used. The first cycles is set to 5 epochs with 1 epochs of linear warmup. Other cycles is set to 10 epochs with 1 epochs of linear warmup. We use the last checkpoint for inference. All 6 models which we use an ensemble for testing set are trained on the same hyper parameters with different random seed.

3.2. Results

The final ensemble result on test set are presented in Table 1. Our algorithm achieved the first place, in terms of top-1 action recognition accuracy.

4. Conclusion

In this paper, we propose a novel fusion architecture. The testing results show that our proposed method can

Table 1. Action recognition results on test set.

	Overall						Unseen Participants			Tail Classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Ensemble	70.64	59.23	48.70	90.97	80.83	68.59	63.50	52.65	39.81	36.13	30.31	22.15

achieve excellent performance.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 1

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1

[3] 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. 1, 2

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[6] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, pages 71–87. Springer, 2020. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2

[8] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

Towards Training Stronger Video Vision Transformers for EPIC-KITCHENS-100 Action Recognition

Ziyuan Huang[1,4†] Zhiwu Qing[2,4†] Xiang Wang[2,4] Yutong Feng[3,4] Shiwei Zhang[4*]
Jianwen Jiang[4] Zhurong Xia[4] Mingqian Tang[4] Nong Sang[2] Marcelo H. Ang Jr[1*]
[1]ARC, National University of Singapore
[2] AIA, Huazhong University of Science and Technology
[3]Tsinghua University [4]Alibaba Group
ziyuan.huang@u.nus.edu, mpeangh@nus.edu.sg
{qzw, wxiang, nsang}@hust.edu.cn
fyt19@mails.tsinghua.edu.cn
{zhangjin.zsw, jianwen.jjw, zhurong.xzr, mingqian.tmq}@alibaba-inc.com

Abstract

With the recent surge in the research of vision transformers, they have demonstrated remarkable potential for various challenging computer vision applications, such as image recognition, point cloud classification as well as video understanding. In this paper, we present empirical results for training a stronger video vision transformer on the EPIC-KITCHENS-100 Action Recognition dataset. Specifically, we explore training techniques for video vision transformers, such as augmentations, resolutions as well as initialization, etc. With our training recipe, a single ViViT model achieves the performance of 47.4% on the validation set of EPIC-KITCHENS-100 dataset, outperforming what is reported in the original paper [1] by 3.4%. We found that video transformers are especially good at predicting the noun in the verb-noun action prediction task. This makes the overall action prediction accuracy of video transformers notably higher than convolutional ones. Surprisingly, even the best video transformers underperform the convolutional networks on the verb prediction. Therefore, we combine the video vision transformers and some of the convolutional video networks and present our solution to the EPIC-KITCHENS-100 Action Recognition competition.

1. Introduction

Recent developments in the computer vision field have witnessed rapid expansion of transformer based model family, which has demonstrated remarkable potential in various computer vision applications, such as image recognition [6, 25], point cloud classification [23] as well as video understanding [1, 2]. They are shown to supersede the performance of convolutional networks when given proper combinations of augmentation strategies [16].

In this paper, we report our recent exploration on the training techniques for the video vision transformers. Specifically, we employ ViViT [1] as our base model, and explored the influence of the quality of the data source, augmentations, input resolutions as well as the initialization of the network. The resultant training techniques enable ViViT to achieve 47.4% on the action recognition accuracy of Epic-Kitchen-100 dataset. Additionally, it is noticed that although ViViT performs better than convolutional networks by a notable margin on the action classification, it underperforms convolutional ones on verb classification. This means that the ensemble of them can be beneficial to increasing the final accuracy. By combining video transformers with the convolutional ones, this paper finally presents our solution to the Epic-Kitchen-100 Action Recognition challenge.

2. Training video vision transformers

We use the ViViT-B/16x2 with factorized encoder as our base model. Two classification heads are connected to the same class token to predict the verb and the noun for the input video clip respectively. We first pre-train the networks on large video datasets that are available publicly, and then fine-tune the ViViT on the epic-kitchen dataset.

† Equal Contribution.

∗ Corresponding authors.

This work is done when X. Wang , Z. Qing, Z. Huang and Y. Feng are interns at Alibaba Group.

Model	Dataset	Resolution	Top 1	Views
ViViT-B/16x2 Fact. encoder	K400	224	78.6	4×3
		320	80.6	4×3
	K700	224	69.7	4×3
		320	71.5	4×3
	SSV2	224	63.6	1×1
		320	-	1×1
	K400-Raft	224	60.5	4×3
	K400-Tvl1	224	65.4	4×3

Table 1. **Pre-training ViViT on Kinetics 400, 700 and SSV2.** The pre-training using respective dataset X with an input resolution Y is denoted further as X-Y. *E.g.,* K400-224 for initialization weights trained on K400 with an input resolution of 224.

2.1. Initialization preparation

There are multiple ways to prepare the pre-trained model, such as supervised pre-training [17, 7, 1, 4] as is used in [14, 13, 18] as well as unsupervised ones [10, 9]. Here we adopt the supervised pre-training as it yields a better downstream performance. The model is firstly trained on Kinetics 400 [11], Kinetics 700 [3] and Something-Something-V2 [8]. respectively. For this part, we mostly follow the training recipe in DeepViT [25] to boost the performance. Specifically, we use AdamW as our optimizer [12] and set the base learning rate to 0.0001. The weight decay is set to 0.1. We initialize the ViViT model with the ViT weight pre-trained on ImageNet21k following the initialization approach in [1], and train the model for 30 epochs with cosine learning rate schedule. The training is warmed up with 2.5 epochs, with the start learning rate as 1e-6. We enable color augmentation, mixup and label smoothing. The model is additionally regularized with a droppath rate of 0.2. The results on the Kinetics and SSV2 are as in Table 1. We also trained ViViT on the optical flow of Kinetics 400, which we extract using Raft [15] and TVL1 [21].

2.2. Training video transformers on Epic-Kitchen

For training video transformers on Epic-Kitchen, we ablate on the training recipes in terms of initialization, the quality of data source, augmentations, input resolutions, the action calculation strategy, as well as the temporal sampling stride. The training parameters including the optimizer, the base learning rate. The training schedule is set to be overall 50 epochs and warmup for 10 epochs. The results can be observed in Table 2. If not otherwise specified, we sample frames with one as the interval.

Initialization: we ablate initialization by pre-training from ImageNet-21K, Kinetics400, Kinetics700 as well as SSV2. The reason that we also ablated the SSV2 initialization is that SSV2 is also egocentric action recognition datasets with complex spatio-temporal interactions. It can be observed that using a strong initialization (from ImageNet21K to Kinetics400, and further to Kinetics700) lead to a notable improvement on the action recognition accuracy. If we decompose the improvement on verb and noun predictions, we can see that stronger initialization model brings the most improvements on noun predictions. However, although higher verb prediction accuracy can be observed by replacing the initialization from K700 to SSV2 (0.1%), there is a notable decrease on the noun prediction (1.4%). Therefore, in the final submission, we did not include models initialized with SSV2.

ID	Init.	Qual.	Res.	Aug.	Top1			
					A	V	N	A*
A	IN21K	256	224	CJ	36.1	62.4	48.2	-
B	**K400-224**	256	224	CJ	37.2	61.7	50.9	-
C	K400-224	**512**	224	CJ	38.4	62.7	52.2	-
D	**K700-224**	512	224	CJ	39.6	63.5	53.3	-
E	K700-224	512	224	**CJ+**	42.8	65.2	56.2	-
F	K700-224	512	**320**	CJ+	45.2 46.3†	67.4	58.9	42.4 43.4
G	**SSV2-224**	512	320	CJ+	44.5 45.7†	**67.5**	57.5	- -
I	K700-224	512	**384**	CJ+	45.8 **47.0†**	67.2	**59.0**	42.5 -
[1]	-	-	224	CJ*	44.0	66.4	56.8	-

Table 2. **Fine-tuning ViViT on EPIC-KITCHENS-100.** *Init* indicates the pre-training dataset. *Qual* indicates the length of the short side of the input video before any transformation is performed. We resize the original video. *Res* indicates the resolution of the input video to the model. *Aug* indicates the augmentation strategy besides random cropping and random flipping. *A, V* and *N* denotes respectively the action, verb and noun prediction accuracies. *A** denotes the action prediction accuracy on the test set. *CJ+* and *CJ* respectively denote random color jitter with and without mixup, cutmix with random erasing. *CJ** indicates a different augmentation strategy used in [1]. †indicates that the action prediction is calculated for each view first before aggregating them together. **Blue font** highlights the change in the respective experiment. **Bold font** in the performance columns indicate the best performing model.

Quality of data source: to mitigate the pressure on the hard drive i/o and thus to speed up training, we resize the short side of the videos to 256 and 512 respectively. It can be observed that raising the quality of the input data source can have an improvement of 1.2%, 1.0% and 1.3% on the action, verb and noun predictions respectively.

Augmentations: we observe the benefit of utilizing stronger augmentations (mixup [22], cutmix [20] and random erasing [24]). Compared to only using random color jittering, stronger augmentations brings an improvement of 3.2% on the action prediction, and 1.7% as well as 2.9% on the verb and noun predictions respectively.

Input resolutions: we further alter the input resolutions. Raising input resolution from 224 to 320 brings about 2.4% improvement on the action prediction, 2.2% on verb prediction and 2.7% on the noun prediction. A saturation of the

ID	Temp Sampling Rate	Top1		
		A	V	N
F	2	45.2 46.3†	67.4	58.9
I	3	46.4 47.4†	68.4	59.6
[1]	2	44.0	66.4	56.8

Table 3. **Altering the temporal sampling rate of ViViT.** *A*, *V* and *N* denotes respectively the action, verb and noun prediction accuracies. †indicates that the action prediction is calculated for each view first before aggregating them together.

Model	Optical Flow	Top1		
		A	V	N
ViViT-B/16x2-Flow-A	Raft	34.6 35.4†	66.8	43.5
ViViT-B/16x2-Flow-B	TVL1	34.5 35.1†	66.4	43.3

Table 4. **Training ViViT with optical flow.** *A*, *V* and *N* denotes respectively the action, verb and noun prediction accuracies. †indicates that the action prediction is calculated for each view first before aggregating them together.

prediction accuracy is observed when we further raise the input resolution from 320 to 384, where only an improvement of 0.6% on the action prediction is observed.

Action score calculation: as indicated in the table as numbers with †, calculating action scores differently could lead to different action prediction results. As two predictions are made for each video clip, there are two ways of aggregating action predictions for multiple views. Suppose we have predictions for verb $P_v^i \in \mathbb{R}^{1\times N_v}$ and noun $P_n^i \in \mathbb{R}^{1\times N_n}$ respectively, where N_v and N_n denotes the number of class for verbs and nouns and i denotes the index for a view, the first way of aggregating the predictions are:

$$P_a = (\sum_i P_v^\top)(\sum_i P_n) \,, \tag{1}$$

where $P_a \in \mathbb{R}^{N_v \times N_n}$ denotes the prediction for actions. This approach aggregates the verb and noun predictions for multiple views first, before calculating the action predictions directly. The second approach calculate the action prediction for each view respectively, before aggregating them:

$$P_a = \sum_i (P_v^\top P_n) \,. \tag{2}$$

As can be seen from the Table 2, aggregating action scores for each view can outperform the other variant by around 1%. What's more important is that this improvement can be reflected in the test set as well.

Temporal sampling stride. Since Epic-Kitchen dataset has a relatively higher FPS, sampling frames with one frame as the interval (which means the temporal sampling rate is 2) can be insufficient for the temporal coverage. When sampling 32 frames, only one second is covered. Therefore, we also ablated on the temporal sampling rate, and the result is presented in Table 5. As can be seen, a minor modification on the temporal sampling rate can have notable improvement on the performance. One reason for this may be the longer temporal coverage. Another possible reason is that the resultant FPS genearted by the sampling rate of 3 is closer to the pretraining FPS. Our final single model performance of ViViT-B/16x2-I outperforms the reported performance in [1] by 3.4%.

Model	Training	Top1		
		A	V	N
TimeSformer 8 × 32	original	34.4	57.1	51.3
	ours-224	39.4	63.9	51.7
	ours-320	**42.5**	**65.2**	**55.0**

Table 5. **Results of TimeSformer on EPIC-KITCHENS-100.** *A*, *V* and *N* denotes respectively the action, verb and noun prediction accuracies. All action accuracies are obtained by calculating action predictions before aggregating them together.

2.3. Training video transformers with optical flow

In order to capture better motion features, optical flow is utilized as another data source. The video transformers with optical flow as the data source are trained using the same training recipe as mentioned before. We trained two optical flow models, whose inputs are respectively optical flow extracted using Raft [15] and TVL1 [21]. The results are presented in

2.4. Other transformer based models

Another transformer based video classification model that we use is the TimeSformer [2]. For TimeSformer, we directly use the open-sourced pretrained model on K600 and first kept its default settings and trained for 15 epochs. Then we used our training recipe in comparison. It shows that our training recipe improves the original one by 5% on the action prediction accuracy. Further increasing the input resolution gives an improvement of 3% on the action prediction accuracy.

3. Training convolutional video networks

Although video vision transformers can have a strong performance, complementary predicions are also needed from the convolutional networks. As we will shown in the following parts, the convolutional networks such as CSN [17] and SlowFast [7] are relatively stronger at predicting verbs.

We use the ir-CSN-152 and SlowFast-16 × 8-101 as our base model. Similar to the training process in ViViT model, we obtain the pre-trained weights by training these

ID	Model	F. BN	Res.	Aug.	Top1			
					A	V	N	A*
A	ir-CSN-152	×	224	CJ	41.0 42.4†	66.4	52.4	37.8 -
B	ir-CSN-152	✓	224	CJ	42.7 43.9†	67.6	55.1	- 40.9
C	ir-CSN-152	✓	224	**CJ+**	43.5 44.5†	68.4	55.9	- **42.5**
D	ir-CSN-152	✓	**320**	CJ+	45.1 **46.2**†	**69.0**	**57.2**	- 42.4
-	SlowFast-16×8-101	✓	224	CJ+	43.0 43.9†	68.2	55.1	- -

Table 6. **Fine-tuning ir-CSN-152 and SlowFast-16×8-101 on EPIC-KITCHENS-100.** *F.BN* denotes frozen batch norm mean and variance. *Res* indicates the resolution of the input video to the model. *Aug* indicates the augmentation strategy besides random cropping and random flipping. *A, V* and *N* denotes respectively the action, verb and noun prediction accuracies. *A** denotes the action prediction accuracy on the test set. †indicates that the action prediction is calculated for each view first before aggregating them together. **Blue font** highlights the change in the respective experiment. **Bold font** in the performance columns indicate the best performing model.

two models on Kinetics 700 [3]. For training on the EPIC-KITCHENS-100 dataset, we use the same training parameters as the ViViT, including the optimizer and the learning rate schedule, etc. We follow [5] and freeze the batch norm mean and variance during training. The results can be seen in Table 6. As can be seen, freezing batch norm mean and variance gives about 1.3% improvement on the action recognition accuracy. Applying mixup, cutmix as well as random erasing yields further improvements both on the validation and the test set. However, different from the experimental result in ViViT, although increasing the input resolution indeed increases the performance on the validation set, the accuracy on the test set is not improved. Therefore, we keep the training resolution as 224 for SlowFast-16×8-101 as well. It is interesting to see that the convolutional models can outperform most ViViT in terms of verb prediction even when the input resolution is only 224×224.

In order to cover a longer period for one video clip, we additionally employ the long-term feature banks (LFB) [19] for the CSN models. For these experiments, we initialize the model with Epic-Kitchen trained ir-CSN-152s, and further train the model for 10 epochs with the same base learning rate as before, with 2 warm-up epochs. The results are shown in Table 7. When using the features extracted by the original model that is used for initializing the training, we observe an improvement for ir-CSN-152-C on the noun prediction. When using the ViViT feature as the feature bank, the noun predictions are further improved, thus notably improving the final action prediction accuracy. In comparison, the verb accuracy is hardly affected.

ID	Model	LFB Feature	Top1		
			A	V	N
-	ir-CSN-152-B	-	43.9†	67.6	55.1
E	ir-CSN-152-B	ir-CSN-152-B	42.9†	66.9	54.7
F	ir-CSN-152-B	ViViT-B/16x2-F	47.3†	67.6	60.1
-	ir-CSN-152-C	-	44.5†	68.4	55.9
G	ir-CSN-152-C	ir-CSN-152-C	44.8†	68.1	56.8
H	ir-CSN-152-C	ViViT-B/16x2-F	**47.3**†	**68.1**	**60.3**

Table 7. **Applying long-term feature banks for ir-CSN-152.** †indicates that the action prediction is calculated for each view first before aggregating them together. **Bold font** in the performance columns indicate the best performing model.

Model Name	Top1		
	A	V	N
ir-CSN-152-B	43.9	67.6	55.1
ir-CSN-152-C	44.5	68.4	55.9
ir-CSN-152-F	47.2	67.6	60.1
ir-CSN-152-G	44.8	68.1	56.8
ir-CSN-152-H	47.3	68.1	60.3
SlowFast-16×8-101	43.9	68.2	55.1
ViViT-B/16x2-Flow-A	35.4	66.8	43.5
ViViT-B/16x2-Flow-B	35.1	66.4	43.3
ViViT-B/16x2-F	46.3	67.4	58.9
ViViT-B/16x2-H	47.0	67.2	59.0
ViViT-B/16x2-I	47.4	68.4	59.6
TimeSformer-320	42.5	65.2	55.0
Overall (Val)	51.7	72.4	62.6
Overall (Test)	48.5	69.2	60.3

Table 8. **List of ensembled models.** All the performance listed in the table are to calculate action for each view before aggregation.

4. Ensembling models

To utilize the complementary predictions of different models, we ensembled a selected subset of the presented models. The selected model set is presented in Table 8. The ensemble of models boost the performance of the best performing one by 4.3% on the action prediction. The final test accuracy that we obtained is 48.5% on action prediction, 69.2% on verb prediction and 60.3% on noun prediction.

5. Conclusion

This paper presents our solution for the EPIC-KITCHENS-100 action recognition challenge. We set out to train a stronger video vision transformer, and reinforce its performance by ensembling multiple video vision transformers as well as convolutional video recognition models.

Acknowledgement. This work is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project A18A2b0046), the National Natural Science Foundation of China under grant 61871435 and the Fundamental Research Funds for the Central Universities no. 2019kfyXKJC024 and by Alibaba Group through Alibaba Research Intern Program.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 1, 2, 3

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 1, 3

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 2, 4

[4] 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. 2

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 4

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[7] 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. 2, 3

[8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 2

[9] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020. 2

[10] Ziyuan Huang, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Rong Jin, and Marcelo Ang. Self-supervised motion learning from static images. *arXiv preprint arXiv:2104.00240*, 2021. 2

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[13] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. *arXiv preprint arXiv:2103.13141*, 2021. 2

[14] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019. 2

[15] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 2, 3

[16] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1

[17] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 2, 3

[18] Xiang Wang, Baiteng Ma, Zhiwu Qing, Yongpeng Sang, Changxin Gao, Shiwei Zhang, and Nong Sang. Cbr-net: Cascade boundary refinement network for action detection: Submission to activitynet challenge 2020 (task 1). *arXiv preprint arXiv:2006.07526*, 2020. 2

[19] 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*, pages 284–293, 2019. 4

[20] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 2

[21] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 2, 3

[22] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[23] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020. 1

[24] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 2

[25] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 1, 2

SAIC_Cambridge-HuPBA-FBK Submission to the EPIC-Kitchens-100 Action Recognition Challenge 2021

Swathikiran Sudhakaran[1] Adrian Bulat[1] Juan-Manuel Perez-Rua[1] Alex Falcon[2]
Sergio Escalera[3,4] Oswald Lanz[2] Brais Martinez[1] Georgios Tzimiropoulos[1]

[1]Samsung AI Center, Cambridge, UK
[2]Fondazione Bruno Kessler - FBK, Trento, Italy
[3]Computer Vision Center, Barcelona, Spain
[4]Universitat de Barcelona, Barcelona, Spain

Abstract

This report presents the technical details of our submission to the EPIC-Kitchens-100 Action Recognition Challenge 2021. To participate in the challenge we deployed spatio-temporal feature extraction and aggregation models we have developed recently: GSF and XViT. GSF is an efficient spatio-temporal feature extracting module that can be plugged into 2D CNNs for video action recognition. XViT is a convolution free video feature extractor based on transformer architecture. We design an ensemble of GSF and XViT model families with different backbones and pretraining to generate the prediction scores. Our submission, visible on the public leaderboard, achieved a top-1 action recognition accuracy of 44.82%, using only RGB.

1. Introduction

Video content understanding is one of the widely researched areas in computer vision with several applications ranging from automated surveillance to robotics, human computer interaction, video indexing and retrieval, *etc*., to name a few. Egocentric action recognition is a particularly challenging sub-task of video content understanding. Egocentric videos are captured using wearable cameras and are often characterized by the presence of a cluttered environment containing several objects and egomotion caused by movement of the camera wearer. Recognizing the action present in a video requires extraction of fine-grained spatio-temporal features that can discriminate one action from another. EPIC-Kitchens-100 [2] is the largest egocentric action recognition dataset with 90K video segments composed of 97 verb and 300 noun categories. The verb and noun labels of a video segment is combined to form its action label.

To participate in the challenge, we used two different video action recognition models composed of entirely different building blocks and feature aggregation strategy.

- GSF[5]: A plug and play module that can transform 2D CNNs into a high performing spatio-temporal feature extractor;
- XViT[1]: A convolution free transformer based architecture for efficient video representation learning

Gate-Shift-Fuse (GSF) is a CNN based architecture that captures local relationship which introduces an inductive bias about the 3D structure of video frames within a small spatio-temporal receptive field. On the other hand XViT, a transformer based model, captures global information and learns geometric relationship between the pixels. While GSF relies on the inductive bias owing to the locally connected convolution layers for feature extraction, XViT disregards any prior about the data and learns the relevant patterns in it that are suitable for addressing the end task. Thus the two models used in the challenge extract different features and are highly complementary to each other. We deployed an ensemble of the two model families to participate in the challenge. The final score is obtained by averaging the prediction scores from the individual members in the ensemble.

2. Models

We describe details of the two model families in this section.

2.1. GSF

GSF, an extension of GSM [4], is a light weight feature encoding module capable of converting a 2D CNN into an

Method	Backbone	Pre-training	Verb	Noun	Action
Validation set					
GSF	IncV3	Kinetics400	68.89 (90)	51.42 (75.49)	43.11 (64.19)
	Res-50	Kinetics400	68.88 (90.44)	52.73 (76.37)	43.84 (64.95
	Res-101	ImageNet	69.06 (90.33)	53.18 (75.81)	44.48 (64.68)
XViT	ViT-B/16	Kinetics400	68 (90.08)	55.63 (78.86)	44.91 (65.97)
Ensemble	-		70.86 (91.67)	56.7 (79.9)	46.88 (68.18)
Test set					
Ensemble	-		68.16 (90.01)	55.49 (78.98)	44.82 (65.45)

Table 1. Performance of the models on the validation set (top) and test set (bottom) of EPIC-Kitchens 100 dataset. Ensemble score is generated by averaging the scores of individual models.

efficient and effective spatio-temporal feature extractor. The output features from a spatial convolution layer of the 2D backbone is first applied to a gating module, composed of a light-weight 3D convolution kernel, to generate grouped spatial gating. The spatial gating is then applied to the input features to obtain group-gated features and residual. Forward and backward shifting in time is then applied to the group-gated features. In GSM, the time-shifted features are combined with the residual using addition operation. GSF extends this simple fusion with a data dependent weighted channel fusion mechanism using a convolution layer. The resulting spatio-temporal features are then propagated to the next layer of the backbone CNN for further processing.

2.2. XViT

Vision transformers [3] can be extended for video recognition by extending the self attention mechanism between tokens within a frame to tokens from other frames as well. However, this will increase the complexity quadratically with the increase in the number of frames. To make the model tractable XViT [1] proposes efficient space-time mixing attention as follows. Let $\mathbf{q}_{s,t} \in \mathbf{R}^{1\times d_h}$, $\mathbf{k}_{s,t} \in \mathbf{R}^{1\times d_h}$ and $\mathbf{v}_{s,t} \in \mathbf{R}^{1\times d_h}$ be the query, key and value at a spatial location s and temporal location t. Then the self-attention $\mathbf{y}_{s,t}$ is computed as

$$\mathbf{y}_{s,t} = \sum_{s'=0}^{S-1} \text{softmax}\{(\mathbf{q}_{s,t} \cdot \tilde{\mathbf{k}}_{s',-t_w:t_w})/\sqrt{d_h}\}\tilde{\mathbf{v}}_{s',-t_w:t_w} \tag{1}$$

with

$$\tilde{\mathbf{k}}_{s',-t_w:t_w} = [\mathbf{k}_{s',t-t_w}(d_h^{t-t_w}), \dots, \mathbf{k}_{s',t+t_w}(d_h^{t+t_w})] \tag{2}$$

$$\tilde{\mathbf{v}}_{s',-t_w:t_w} = [\mathbf{v}_{s',t-t_w}(d_h^{t-t_w}), \dots, \mathbf{v}_{s',t+t_w}(d_h^{t+t_w})] \tag{3}$$

where, $\mathbf{k}_{s',t'}(d_h^{t'})$ and $\mathbf{v}_{s',t'}(d_h^{t'})$ denotes the operator for indexing the $d_h^{t'}$ channels from $\mathbf{k}_{s',t'}$ and $\mathbf{v}_{s',t'}$, respectively.

The video transformer model used in this challenge is constructed by replacing the self-attention in [3] with Eqn. 1

3. Experiments

We describe the implementation details of the two model families along with their training and testing settings in this section.

3.1. Implementation Details

GSF. Gate-Shift-Fuse Networks are instantiated by plugging in GSF to the backbone layers of a 2D CNN. For the challenge, we instantiated three different models by changing the backbone CNNs. This includes InceptionV3, ResNet50 and ResNet101. The GSF variant of InceptionV3 and ResNet50 are first pre-trained on Kinetics400 dataset while for ResNet101, we used the ImageNet pretrained weights and directly trained the model on EPIC-Kitchens-100 dataset.

XViT. Backbone used is the base architecture ViT-B/16 from [3] with 12 transformer layers each with 12 attention heads and an embedding dimension of 768. Each frame from the video is first divided into non-overlapping patches of size 16×16 and are then applied to a linear layer for vectorization. The temporal window t_w is set as 1.

Training. We trained all our models using SGD with momentum (0.9) and a cosine scheduler with linear warmup. The base learning rate for GSF models are set as 0.01 for a batch size of 32 while XViT is trained with a base learning rate of 0.05 and a batch size of 128. GSF models are trained for 60 epochs and XViT is trained for 50 epochs. 16 frames uniformly sampled from the input video clip are applied as input to all the models. We also applied temporal jittering during training, as done in [6]. All models are trained in a multi-task classification setting using three classification layers to predict verb, noun and action labels. We generated the action labels by combining the verb and noun label of the video provided with the dataset to obtain a total of 3806 action categories in the training set. More details regarding training can be found in [5] and [1].

Testing. We sample 2 clips consisting of 16 frames during testing. From each frame, 3 spatial crops are generated. Thus, from each video, we generate 6 clips. The prediction score from each of the 6 clips are averaged to obtain the video prediction.

3.2. Results

Tab. 1 lists the performance of the various models used for the challenge. The top part of the table shows the results on the validation set. From the validation set results, one can see that GSF is strong on verb prediction while XViT results in a better performance on noun prediction. This shows that GSF is a powerful model for temporal reasoning. On the other hand, the presence of global spatial receptive field of XViT enables it to perform as a strong object recognition model. Combining the prediction scores obtained from both model families improves the performance considerably, showing their complementarity in extracting spatio-temporal features. The bottom part of the table shows the performance on the test set, which is visible on the leaderboard. Note that all model developments have been done on the validation set and evaluation of individual models is not done on the test set to tune the models' performance. This shows that our ensemble generalizes well to the test data.

4. Conclusion

In this report, we summarized the details of the two model families used for participating in the EPIC-Kitchens-100 Action Recognition Challenge 2021. The improved performance of the ensemble consisting of the two model families shows that the two models are complementary to each other. This resulted in achieving a top-3 action recognition performance on the leaderboard.

Acknowledgements

FBK gratefully acknowledge the support from Amazon AWS Machine Learning Research Awards (MLRA).

References

[1] A. Bulat, J. Perez-Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos. Space-time Mixing Attention for Video Transformer. *arXiv preprint arXiv:2106.05968*, 2021. 1, 2

[2] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, Georg Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[4] S. Sudhakaran, S. Escalera, and O. Lanz. Gate-Shift Networks for Video Action Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1102–1111, 2020. 1

[5] S. Sudhakaran, S. Escalera, and O. Lanz. Gate-Shift-Fuse Networks for Video Action Recognition. *arXiv*, 2021. 1, 2

[6] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal Segment Networks for Action Recognition in Videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755, 2019. 2

TCMT: Temporal Context with Multimodal Transformers

Evangelos Kazakos[1] Jaesung Huh[2] Arsha Nagrani[3] Andrew Zisserman[2] Dima Damen[1]
[1]University of Bristol [2]University of Oxford [3]Google Research

Abstract

We propose a method that leverages temporal context in untrimmed videos. We formulate temporal context as a sequence of consecutive actions, and the goal is to recognise the action at the centre of the sequence. A transformer attends at audio-visual temporal context to identify relevant auditory and visual action segments of nearby actions.

Our methods ranks 4^{th} *on the test set of EPIC-KITCHENS-100.*

1. Introduction

Action recognition is challenging in EPIC-KITCHENS, as actions are fine-grained (e.g. 'open bottle') and noticeably short, often one second or shorter. Along with the challenge, the dataset offers an under-explored opportunity, as actions are captured in long untrimmed videos of well-defined and at-times predictable sequences. For example the action 'wash aubergine' can be part of the following sequence – you first 'take the aubergine', 'turn on the tap', 'wash the aubergine' and finally 'turn off the tap' (Fig. 1). Furthermore, the objects (the aubergine and tap in this case) are persistent over some of the neighbouring actions. This opportunity allows us to design models able to attend to the *temporal context* that is relevant and useful in recognising ongoing actions.

In this work, we define temporal context of an action as a sequence of actions surrounding an ongoing action to be recognised, by utilising the annotated start/end times of neighbouring actions, and excluding background information. We investigate not only using the action's temporal context in the data stream, but also the labels of neighbouring actions as additional supervision. Furthermore, we are motivated by previous works that demonstrate the significance of audio in recognising egocentric actions [3, 5, 6], and thus include the auditory temporal context in addition to the visual clips. Concretely, we use the attention mechanism of a multimodal transformer architecture to take account of the context in the data stream, using vision and audio, and introduce an auxiliary loss function by predicting actions in the temporal context window, in addition to the action of interest, which is the action at the centre of the window.

2. Temporal Context with Multimodal Transformers

We define temporal context of an action as the sequence of neighbouring actions that precede and succeed the action in untrimmed videos. Our goal is to design an architecture able to attend to relevant audio-visual temporal context around actions to improve the recognition of the ongoing action. To address that, we introduce an audio-visual transformer that ingests a temporally-ordered sequence of visual inputs, along with the corresponding sequence of auditory inputs of the same video segments. We use modality-independent positional encodings as well as modality-specific encodings.

The audio-visual transformer utilises two separate summary embeddings to attend to the action (i.e. verb) class and the object (i.e. noun) class. This allows the model to attend independently to the temporal context of verbs vs objects. Each summary embedding uses a different learnt classification token, and the classifier predicts a verb and a noun from the summary embeddings. We refer to our method as Temporal Context with Multimodal Transformers (TCMT).

2.1. Formulation of audio-visual temporal context

Let $X_v \in \mathbb{R}^{w \times d_v}$ be the sequence of inputs from an untrimmed video and $X_a \in \mathbb{R}^{w \times d_a}$ the corresponding audio inputs, for w consecutive actions in the video (i.e. the temporal context window), with d_v, d_a being the input dimensions of the two modalities respectively. Our temporal window of w action segments is centred around an action b_i with surrounding action segments, excluding any background frames. That is, each action b_j within the window, $i - \frac{w-1}{2} \leq j \leq i + \frac{w-1}{2}$ is part of the transformer's input.

2.2. Audio-Visual Transformer

Encoding layer. Our model first projects the inputs X_v, X_a to a lower dimension D and then tags each with positional and modality encodings. Then, an audio-visual encoder performs self-attention on the sequence to aggregate relevant audio-visual temporal context from neighbouring actions.

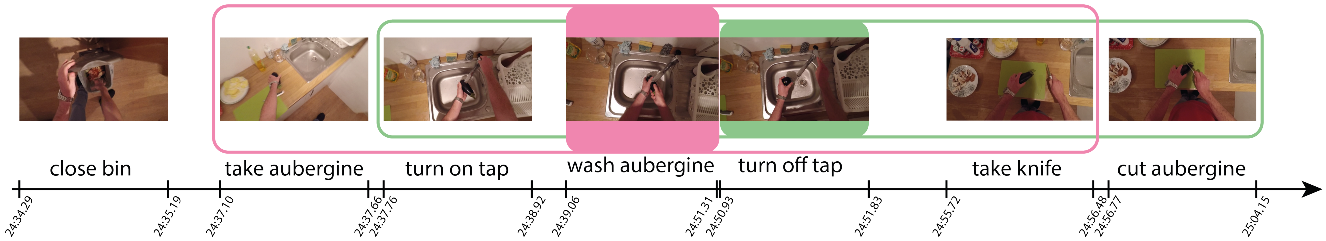

Figure 1. Our approach uses temporal context windows around actions to be recognised ('wash aubergine' and 'turn off tap'), leveraging nearby actions from the untrimmed video. Prediction of 'wash aubergine' is enhanced by observing audio-visually that the tap was turned on before and turned off afterwards.

We use positional encodings to retain information about the ordering of actions in the sequence. We use w positional encodings, shared between audio-visual features to model corresponding inputs from the two modalities. Modality encodings, $m_v, m_a \in \mathbb{R}^D$, are learnt vectors added to discriminate between audio and visual tokens. Visual and auditory features encoded with positional and modality information are denoted as X_v^e and X_a^e.

A classifier predicts the action b_i, using two summary embeddings, acting on the learnt verb/noun tokens. We use the standard approach of appending learnable classification tokens to the end of the sequence but use two tokens, one for verbs and one for nouns, denoted as $\text{CLS}_\text{V}, \text{CLS}_\text{N} \in \mathbb{R}^D$, with unique positional encodings. Classification tokens with encoded positional information are denoted as CLS_V^e and CLS_N^e. The inputs to the transformer are $X^e = [X_v^e; X_a^e; \text{CLS}_\text{V}^e; \text{CLS}_\text{N}^e]$, where $[;]$ denotes input concatenation.

Transformer and classifier. We use a transformer encoder to process sequential audio-visual inputs X^e. A two-head classifier then predicts the sequence of w actions from both the transformed visual and audio tokens, and the action b_i from the summary embeddings.

Loss function. We can leverage the ground-truth of neighbouring actions within w for additional supervision to train the audio-visual transformer. Our loss is composed of two terms, the main loss for training the model to classify the action at the centre of our temporal context using the predictions of the summary embeddings, and an auxiliary loss applied to predicting all actions in the sequence using the predictions from the transformed visual and auditory inputs.

3. Experiments

Feature extraction. We extract visual features with SlowFast [2], using model and code we made available from [1]. We first train the model with slightly different hyperparameters, where we sample a clip of 2s from an action segment, do not freeze batch normalisation layers, and warm-up training during the first epoch starting from a learning rate of 0.001. We noted that this gave us better performance. All unspecified hyperparameters remain unchanged. We use the trained model to extract features from EPIC-KITCHENS train, val and test sets. For feature extraction, 10 clips of 1s each are uniformly sampled for each action segment, with a center crop per clip. The resulting features have a dimensionality of $d_v = 2304$. We use our proposed Auditory SlowFast [4] for audio feature extraction. Similarly to the visual features, we extract 10 clips of 1s each uniformly spaced for each action segment, with average pooling and concatenation of the features from the Slow and Fast streams, and the resulting features have the same dimensionality, $d_a = 2304$.

Architecture details. The audio-visual transformer encoder consists of 4 layers that share weights, 8 attention heads and a hidden unit dimension of 512. In the encoding layer, the fully-connected layer projects the inputs to a lower dimension $D = 512$. Positional/modality encodings as well as verb/noun tokens have also dimensionality $D = 512$ and are initialised to $\mathcal{N}(0, 0.001)$. In the encoding layer, dropout is applied at both the inputs X_v, X_a and the outputs X^e.

Training details. TCMT is trained using SGD, a batch size of 32 and a learning rate of 0.01 for 100 epochs. Learning rate is decayed by a factor of 0.1 at epochs 50 and 75. For regularisation, a weight decay of 0.0005 is used and a dropout 0.5 and 0.1 for the encoding layers and transformer layers respectively. We use mixup data augmentation [7] with $\alpha = 0.2$.

Ensemble. We report results of an ensemble where each model in the ensemble is trained with different temporal context length, $w = \{1, 3, 5, 7, 9\}$.

Results are shown in Table 1. We show results for different temporal context lengths to showcase its importance. At the closing of the challenge, TCMT is ranked 4^{th} on the leaderboard. Table 1 shows the reported results on all metrics.

Challenge entry: As two authors are prime contributors to EPIC-Kitchens collection and running the challenge, we are not officially competing in the challenge. However, we wish to note that we did not use any of the test set annotations in optimising our results.

		Overall						Unseen Participants			Tail-classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
	w	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Val.	1	67.88	52.29	41.31	90.52	76.47	61.52	61.03	44.60	32.58	42.05	27.42	21.48
	3	69.12	54.85	43.42	91.26	78.92	64.19	60.85	46.48	34.08	40.68	31.89	23.41
	5	69.80	**55.96**	44.75	**91.65**	**79.48**	64.94	61.78	46.57	34.65	**43.18**	**32.68**	**24.51**
	7	69.91	55.83	44.77	91.23	79.13	**65.03**	61.60	**46.95**	34.93	41.99	32.47	24.12
	9	**70.23**	55.82	**45.00**	91.13	79.06	64.58	**63.29**	46.38	**35.02**	41.76	32.26	24.41
	Ensemble	**70.91**	**56.21**	**45.25**	**91.89**	**80.34**	**65.96**	63.19	**47.23**	34.65	42.78	32.42	24.44
Test	Ensemble	68.20	54.97	43.32	89.64	78.70	63.09	61.58	46.93	33.77	37.37	29.83	20.98

Table 1. We showcase the importance of temporal context on the validation set; as temporal context increases, action accuracy improves. Ensembling further improves the results. On the leaderboard (**Test**), we submitted the ensemble.

4. Conclusion

We have shown that TCMT successfully utilises both auditory and visual temporal context through learnt attention to improve the recognition performance of ongoing actions. TCMT obtains results comparable with the state of the art in the final leaderboard.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 2

[2] 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 (ICCV)*, October 2019. 2

[3] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[4] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-fast auditory streams for audio recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 855–859, 2021. 2

[5] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[6] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *CoRR*, abs/2001.08740, 2020. 1

[7] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR*, 2018. 2

EgoAugment: CMU-KLAB Submission to the EPIC-Kitchens Action Recognition 2021 Challenge

Xuhua Huang Ye Yuan Xingyu Liu Qichen Fu Kris M. Kitani
Robotics Institute
Carnegie Mellon University

Abstract

In this report, we describe the technical details of our submission to the EPIC-Kitchens Action Recognition Challenge 2021, by Team "CMU-KLAB" (username: xhking). Egocentric videos are captured by a wearable camera in first-person perspective, which are different from classical videos in that they usually involve rapid scene change, object distortion and limited visual range. Therefore, it requires a much more efficient and stronger architecture to recognize objects appeared in different frames as well as to understand hidden relationships among human-object interactions. Attention-type methods have demonstrated their capabilities in learning such relationships, which, nevertheless, suffer from high computation cost, stopping them from being applied to large inputs (e.g. videos). We propose EgoAugment, which combines an efficient transformer with classic video architecture, aiming to augment the information captured by our network and boost performance in egocentric video analysis. Our method demonstrates better performance than the most popular architectures in video action recognition.

1. Introduction

Egocentric video analysis is gaining its popularity with the development of different human-computer interaction applications such as Virtual Reality/Augmented Reality (VR/AR), but it is also challenging due to (1) rapid movement: because a wearable device is usually worn on the head of a camera wearer, where a small turn could result in large movement for both background and foreground objects, leading to frequent occlusion and motion blur effect; (2) distortion by the wide-angle lens design; (3) limited visual range: the perspectives of egocentric videos are always restricted to the working area around hands, which makes it hard to utilize the surrounding environment for thorough analysis.

Many state-of-the-art methods targeted for egocentric videos such as [18, 14, 13] are trying to integrate attention mechanism [16] thanks to its promising ability to capture the relationship across frames under challenging settings. Motivated by a novel design of transformer introduced in [10], we propose a new architecture named EgoAugment, by adding a computation-efficient Augment Branch to enhance the learning ability under egocentric setting.

2. Methodology

In this section, we introduce our proposed framework by parts. Figure 1 summarizes the overall pipeline of our proposed method.

2.1. Main Branch (Path 1 + Path 2)

Two-pathway design is a common schema used for video models, and previous work [2, 7, 15] presents its advantage in extracting spatial and temporal features from videos simultaneously. For the Main Branch, two sets of video frames of different number are sampled randomly from the entire video sequence as inputs. We adopt four residual stages (i.e. Res-Stage) following the settings in [7] with 3D convolution and bottleneck residual blocks. After each Res-Stage, we apply lateral connections between these two paths to enable information fusion. Specifically, outputs from Path 1 will be fused to Path 2 by time-strided convolution [7] and concatenation.

2.2. Augment Branch (Path 3)

Despite that many two-pathway models have achieved state-of-the-art performance on major video recognition benchmarks such as AVA, Charades and Kinetics, they might fail to exhibit satisfying results on egocentric videos even after fine tuning. Transformers have been justified to be a powerful module in both image tasks [6, 11, 1, 9] and EPIC-55 challenge [4], but the biggest obstacle of applying classical transformers into video tasks is their quadratic scaling computational complexity of all-to-all attention.

Inspired by the latest work in bottleneck transformer [10], we aim to design an efficient transformer mod-

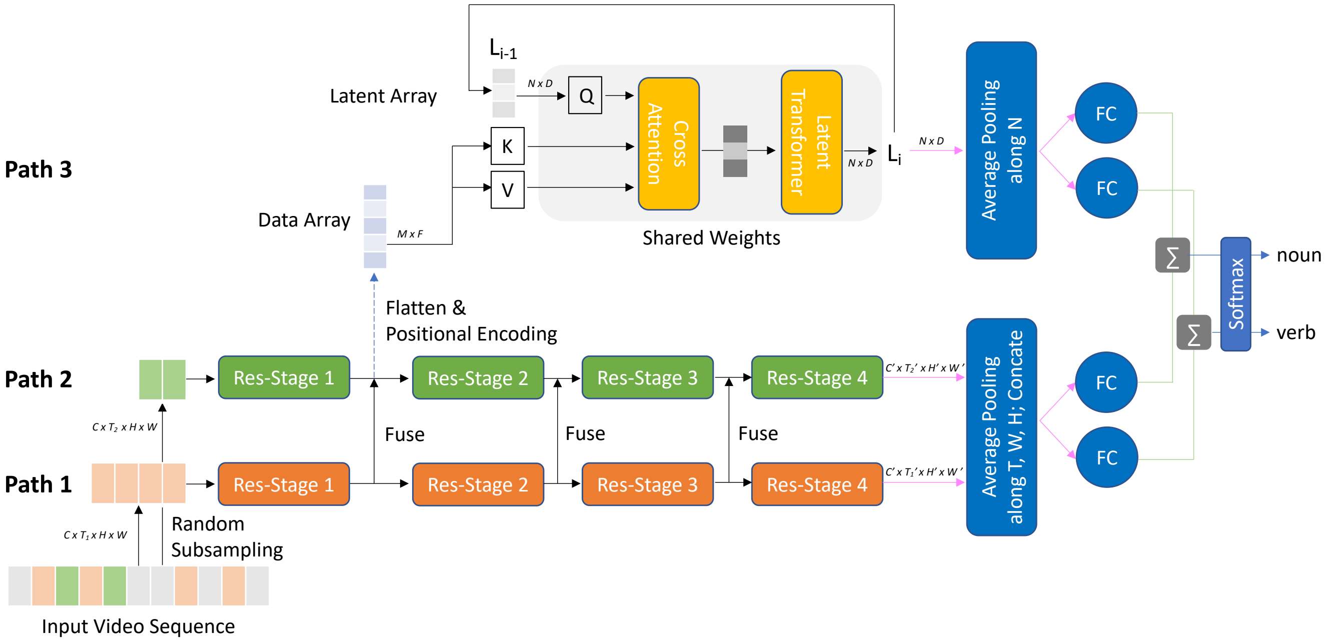

Figure 1: Overview of EgoAugment: given an input video sequence, we will randomly sample T_1 and T_2 frames as the input to the Path 1 and Path 2 respectively, where $T_1 = 2T_2$. Main Branch (Path 1 + Path 2) takes in two sets of video frames and calculates the logits for noun and verb via a data layer, 4 residual stages, average pooling and FC layer. The Augment Branch (Path 3) adopts an iterative design, and derives the logits using a pre-processed input taken from Res-Stage 1 by cross-attention and self-attention. Logits from both branches are summed and passed through softmax layer to make the final prediction. Best viewed in color.

ule which can augment the features extracted from Main Branch, while keeping the process as computational-efficient as possible even when the input is a large data array (i.e. a 3D feature map from Main Branch). Figure 1 shows the detail of our Augment Branch, and we elucidate the components in the following paragraphs.

Iterative Design We adopt a shared-weight design for the Augment Branch, where the cross-attention and latent transformer blocks are iteratively used.

Data Array Denote the size of Data Array as $M \times F$. Data Array comes from the fused feature maps after Res-Stage 1 in our Main Branch, and arrays are the same for all iterations. However, before it is fed into the cross-attention module, we need three pre-processing steps:

(1) *Positional Encoding*. Following the positional encoding method introduced in [10], we parametrize the frequency encoding and take values within range of $[\sin(f_k \pi x_d), \cos(f_k \pi x_d)]$, where f_k is the frequencies of the k^{th} band of a bank of frequencies, and x_d is the value of input position along d^{th} dimension (for video we have $d = 3$). We also concatenate the original positional value x_d to the encoding, so we have $d(2K + 1)$-dim positional encoding vector for each pixel and we denote the result array as D_1 with shape of $(T \times W \times H) \times d(2K + 1)$

(2) *Flatten*. The $T \times H \times W \times C_1$ feature map from Res-Stage 1 will be flatten along spatial and temporal dimension into $(T \times W \times H) \times C_1$, we denote the result array as D_2 with shape of $(T \times W \times H) \times C_1$

(3) *Concatenation*. We concatenate D_1 and D_2 along the feature dimension to generate the $M \times F$ Data Array, where $M = T \times W \times H$ and $F = d(2K + 1) + C_1$.

Latent Array Denote the size of Latent Array as $N \times D$. As shown in Figure 1, the core idea is to introduce N low-dimensional latent units to play the role of *query*. Since N is designed to be small ($M \gg N$), it will form an attention bottleneck during the cross-attention operation with high-dimensional data array. Note that Latent Array can be viewed as a trainable module, whose values are initialized randomly at the beginning of training, and are updated by gradient descent during training. The input Latent Array comes from the output of last iteration. For the first iteration the Latent Array is initialized with random values.

Linear projection layers are applied before Q, K, V to project the input onto the same low-d latent space before attention. The shared weights and bottleneck design allow our model to handle very large video domain input, while keeping low computation cost, and is proved to be a performance booster on egocentric videos in Section 3.

Methods	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Unseen Top-1 (%)			Tail Classes Top-1 (%)		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TSN	59.03	46.78	33.57	87.55	72.10	53.89	53.11	42.02	27.37	26.23	14.73	11.43
TRN	63.28	46.16	35.28	88.33	72.32	55.26	57.54	41.36	29.68	28.17	13.98	12.18
TBN	63.02	47.12	35.55	89.00	73.01	56.19	57.42	41.39	29.25	30.46	18.67	13.97
TSM	65.32	47.80	37.39	89.16	73.95	57.89	59.68	42.51	30.61	30.03	16.96	13.45
SlowFast	63.79	48.55	36.81	88.84	74.49	56.39	57.66	42.55	29.27	29.65	17.11	13.45
Ours	**63.82**	**51.12**	**38.73**	88.42	**75.02**	57.02	**57.71**	**45.62**	**31.48**	**36.05**	**26.26**	**19.25**

Table 1: Action recognition results on EPIC-100 TEST sets

2.3. Prediction

An average pooling operation will be applied to the outputs of two branches, generating a 1D vector, and the vector will go through two fully-connected (FC) layers at the end. These two FC layers correspond to nouns and verbs activation and can be viewed as logits. In order to fully explore the information from all paths, we sum the logits from Main Branch and Augment Branch for each category, i.e. noun and verb. After the summation, a softmax layer is applied to output the final prediction scores for each class.

2.4. Loss

We use a variant of cross entropy loss as our training loss,

$$\mathcal{L} = CrossEntropy(\tilde{y}, \hat{y})$$

where $\tilde{y}$ is our predicted label. However, during experiments we find that since Epic Kitchens has a smaller scale compared with those large-scale video dataset (e.g. Kinetics), it is beneficial to introduce label smoothing proposed in [17]. The ground truth label used during training, $\hat{y}$, becomes a mixture of one-hot ground-truth label, y, and a uniform distribution μ to regularize our model to make less confident predictions during training stage,

$$\hat{y} = (1 - \lambda)y + \lambda\mu$$

The mixture is controled by a hyperparameter $\lambda \in [0, 1]$.

2.5. Implementation Details

We train our model for 50 epochs using SGD optimizer, with batch size 8, initial learning rate 10^{-3}, dropout rate 0.5 and momentum 0.9. The learning rate is set in a cosine annealing schedule [12]. The mixture scalar λ is set to 0.2. Every frame is randomly cropped to 224 × 224 before feeding into our pipeline. Random crop, flip and random augment [3] are used during training.

Main Branch The number of input frames sampled for Path 1 and Path 2 of Main Branch are 32 and 16 respectively. The instantiations of the network architectures are same as the ResNet-50 backbone in [7, 8]. The weights of Main Branch are pre-trained on Kinetics.

Augment Branch Random initialization are applied to the weights of Latent Array and linear projection layers. We set $N = 256, D = 512$, $K = 6$ bands, 1 head for Cross Attention, 8 heads for Latent Transformer and the number of iterations is 3. The inner dimension for Q, K, V is 64.

Methods	Top-1 Accuracy (%)		Top-5 Accuracy (%)	
	Verb	Noun	Verb	Noun
TSN	60.18	46.03	89.59	72.90
TRN	65.88	45.43	90.42	71.88
TBN	66.00	47.23	90.46	73.76
TSM	67.86	49.01	90.98	74.97
SlowFast	65.56	50.02	90.00	75.62
Ours	**67.90**	**51.82**	**91.50**	**76.70**

Table 2: Action recognition results on EPIC-100 VAL sets

3. Experiments

3.1. Ablation Study

Table 3 demonstrates that our three-pathway design with Augment Branch can make obvious improvement on egocentric benchmark such as Epic Kitchens.

Methods	Top-1 Accuracy (%)		Top-5 Accuracy (%)	
	Verb	Noun	Verb	Noun
w/o Aug	65.24	50.10	89.46	74.63
w/ Aug	$67.90_{\uparrow 2.7}$	$51.82_{\uparrow 1.7}$	$91.50_{\uparrow 2.0}$	$76.70_{\uparrow 2.0}$

Table 3: Ablation of a model trained without Augment Branch compared with a model trained with Augment Branch. Results are reported on EPIC-100 VAL sets.

3.2. Results

Table 1 presents our submitted results on EPIC-100 test sets (i.e. results on the leaderboard). Table 2 compares our method with all baselines results provided in [5] on validation set. It is noticeable that our method outperforms all those highly-performed methods which are widely used for general video action recognition tasks, under egocentric vision setting.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[2] 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. 1

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[7] 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. 1, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 1

[10] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021. 1, 2

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[12] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3

[13] Juan-Manuel Perez-Rua, Brais Martinez, Xiatian Zhu, Antoine Toisoul, Victor Escorcia, and Tao Xiang. Knowing what, where and when to look: Efficient video action modeling with attention. *arXiv preprint arXiv:2004.01278*, 2020. 1

[14] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long term video understanding. *arXiv preprint arXiv:2006.00830*, 2020. 1

[15] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 1

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1

[17] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020. 3

[18] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12249–12256, 2020. 1

EPIC-KITCHENS-100 Action Recognition Challenge 2021
User “tsm_transformer” Technical Report

Abhinav Rai[1], Fadime Sener[2], Angela Yao[1]
[1]National University of Singapore
[2]University of Bonn, Germany
e0403959@u.nus.edu, sener@cs.uni-bonn.de, ayao@comp.nus.edu.sg

Abstract

This technical report describes the approach of user “tsm_transformer” (team “NUS_CVML”) for the EPIC-KITCHENS-100 Action Recognition Challenge 2021. Our submission is ranked 7th in the leaderboard of the challenge. Modeling the visual changes that an action brings to a scene is critical for video understanding. Currently, CNNs process one local neighbourhood at a time, so contextual relationships over longer ranges, while still learnable, are indirect. We present TROI, a plug-and-play module for CNNs to reason between mid-level feature representations that are otherwise separated in space and time. The module relates localized visual entities such as hands and interacting objects and transforms their corresponding regions of interest directly in the feature maps of convolutional layers. We refer the reader to [11] for further details and evaluations on other datasets.

1. Introduction

In this work, we target the task of recognizing human-object interactions found in daily activities. These actions are fine-grained and inherently compositional between the movement (verb) and the interacting object. Throughout the interaction, the objects are often transformed visually and or physically, and over longer ranges of time. Correctly identifying these actions requires the ability to relate the object appearance from beginning to end throughout the transformation. State-of-the-art convolutional neural networks (CNNs), armed with 3D convolutions, are designed to learn such relationships in space and time. However, convolution operations are by design, locally limited, and are therefore inefficient in capturing relationships over a long range.

One way to expand network’s receptive field over time and space would be to increase network depth. However, adding more layers naïvely also increases memory requirements and computations [7], and runs risk of over-fitting. Instead, architectural additions in the form of skip connections[8], attention [15] and graphs [3] has been proposed, all with the aim of capturing either long-range or global dependencies more directly.

With the recent advances in object detectors and trackers, a common strategy that has been adopted is to first localize key entities such as body parts and or interacting objects. The action is then recognized based on the detections or tracks, with additional reasoning through graphs [16] or visual relations [1]. To represent the detections and tracks, features are drawn from the output of a CNN’s last convolutional layer [16, 1].

In learning global dependencies in video, purely feature-based and purely entity-based approaches form two extremes. On one side, feature-based approaches like non-local blocks and GloRe [15, 3] exhaustively relate locations in feature maps. This results in a combinatorially large number of operations even though the majority of these computations are redundant, as most of the locations have little contribution to others. Furthermore, because these methods capture global relationships from the entire feature map indiscriminately, they cannot ensure that learned relationships correspond to causal entities such as the hands and interacting object. This in turn may lead to undesirable biases to the scene [9] or other dataset latencies. On the other hand, purely entity-based approaches reason only between detections or tracks. They miss useful cues from the background and lack the ability to reason between entities and the scene itself. Therefore, to boost performance, most approaches [1, 16] add a separate appearance stream and do a late fusion to merge this information.

We aim to strike a middle ground between abstract visual features and concrete visual entities. Relying solely on high-level visual entities may not capture the dynamic changes in objects’ appearance; lower-level visual features can capture these differences, but at the same time may lack sufficient semantic context. Therefore, we propose a module for regions of interest (ROIs) associated with visual entities on mid-level feature maps. Representations of the

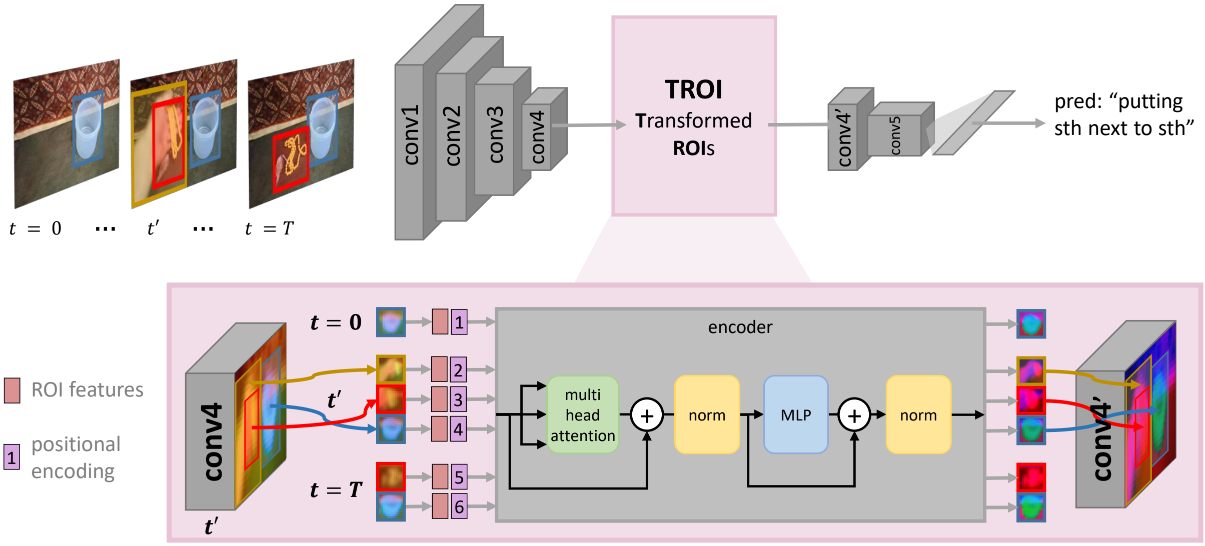

Figure 1. **Overview of TROI.** Our module can be integrated between convolution layers of standard CNNs; above, TROI is inserted after the *conv4* layer. Within TROI, an RoIAlign operation localizes the features for each visual entity and a transformer-style encoder relates these mid-level representations in space and time. To retain spatial and temporal information from the input ROIs. The transformed features from hand and object regions are updated in place in the feature maps. The updated feature maps, *i.e.* **conv4'** above, is followed by the last convolutional and a classification layer.

ROIs are transformed in-place, directly within the feature map. Accordingly, we name our proposed module as TROI (**T**ransformed **ROI**s). Based on detections, the module associated ROI features across an entire video via an attention mechanism. This localized form of attention bypasses the locality constraints of CNN backbones and allows the ROIs to be related to each other in a manner that is global and or long-range in time. The transformed ROIs continue to be processed within their feature map context by the original CNN backbone to capture global information about the entire video content. An overview on is presented in Figure 1.

Our work is novel in several regards. First and foremost, our method is the first that focuses on relating mid-level features over space and time in middle CNN layers from an entity point of view. This allows us to bypass the large number of operations on irrelevant regions contrary to feature-based approaches [15, 3]. Our in-place transformations in the ROIs unifies entity localization and action recognition in CNNs contrary to other approaches which separate the two in a CNN's last convolutional layer. Finally, our approach is flexible. Any state-of-the-art relational model can be used to represent spatio-temporal inputs; we chose transformers.

2. Method

2.1. Preliminaries

Given a video, we select T evenly interspersed frames as is standard in CNNs for action recognition [10, 5]. Let $\mathbf{X} \in \mathbb{R}^{T \times W \times H \times C}$ denote a feature representation of the video, where W, H and C are the width, height, and number of channels of the feature map respectively. $\mathbf{X}$ can come from any convolutional layer in a neural network; as we are interested in relating mid-level features, $\mathbf{X}$ is the feature map after several convolution operations, *e.g.* at conv3 or conv4 of a ResNet architecture.

On a selected frame t, there are N_t regions of interest (ROIs) around fixed entities such as the hands or interacting objects. We denote the collection of ROIs across the T selected frames as $\mathcal{R} = \{r_1, ..., r_N\}$, with N ROIs in total. The objective of our approach is to learn a mapping g, parameterized by Θ, to map $\mathbf{X}$ to $\mathbf{X}'$, *i.e.*

$$\mathbf{X}' = g(\mathbf{X}, \mathcal{R}, \Theta). \quad (1)$$

$\mathbf{X}'$ has the same dimensions as $\mathbf{X}$ but is transformed in that the N ROI regions are modified according to the learned relationships between all the entities in the video. To do this, we propose a transformer-based self-attention module which we name TROI (**T**ransformed **ROI**s).

We denote with $f_i \in \mathbb{R}^C$ the visual feature within the ROI r_i, where $f_i = h(X_i, r_i)$. Here, h is a function that localizes the features corresponding to ROI r_i, and $X_i \in \mathbb{R}^{1 \times W \times H \times C}$ is the part of the feature map selected from the frame to which r_i corresponds. In practice, we estimate the ROI r_i based on object detections and localize the associated feature from X_i by applying ROIalign [6] followed by a spatial average pooling for h. To retain positional information from the bounding boxes in space and time, we add a positional encoding [13] to each f_i. For the positional encoding, we follow a left to right order for the bounding boxes across the frames by labelling them sequentially, as shown in Fig. 1. We found similar results when ordering from right to left, which indicates that TROI consistently learns patterns as long as some temporal and spatial ordering is preserved. We then use this as the positional value and use the sine and cosine formula used in [13] to generate the final encoding. The entire set of features associated with all the ROIs extracted from the video's feature maps $\mathbf{X}$ is then denoted as $\mathbf{F} \in \mathbb{R}^{N \times C}$.

2.2. Relational Model

For the mapping g and its parameters Θ, we choose a transformer-style encoding architecture and leverage self-attention to learn the spatio-temporal relationships between visual entities. Transformers compute sequence representations by attending to different input positions via stacks of self-attention layers. In this context, we adapt them to model the set of mid-level ROI features in a video as a natural spatio-temporal sequence.

We follow standard transformer notation and use **qkv** self-attention [13], with query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$. Conceptually, we perform a series of in-place transformations dependent on the other ROIs from the entire video through self-attention. A **query** is an input ROI feature, and the **keys** are the sequence of features from all the other ROIs in a video. The self-attention mechanism estimates attention weights that correspond to the importance of the ROIs keys for a particular query ROI. According to this, we first compute the linear projections for $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$ features :

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{F}\mathbf{W}_{qkv}, \tag{2}$$

where $\mathbf{W}_{qkv} \in \mathbb{R}^{C \times (2d_k + d_v)}$ corresponds to the parameter matrices $\mathbf{W}_q \in \mathbb{R}^{C \times d_k}, \mathbf{W}_k \in \mathbb{R}^{C \times d_k}, \mathbf{W}_v \in \mathbb{R}^{C \times d_v}$ of the projected versions of query, key and value respectively. Here $d_k = d_v = C/m$ and m is the number of attention heads. We refer the reader to [13] for more details on the transformer architecture.

We compute the attention weights, A_{ij}, as scaled pairwise similarities between two ROI features from $\mathbf{F}$ and their query and key representations, $\mathbf{q}^i$ and $\mathbf{k}^j$,

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_k}}\right), \qquad \mathbf{A} \in \mathbb{R}^{N \times N}. \tag{3}$$

For computing the self-attention, SA, for each ROI in $\mathbf{F} \in \mathbb{R}^{N \times C}$, we compute a weighted sum over all *values* $\mathbf{v}$, *i.e.*

$$\text{SA}(\mathbf{F}) = \mathbf{A}\mathbf{v}. \tag{4}$$

The self-attention operation calculates the response at each position, f_i, by attending over all other positions in $\mathbf{F}$. Multi-head attention extends self-attention by running the same procedure m times; each self-attention procedure is referred to as a *head*. Multi-head attention has an implicit ensembling effect and generally improves performance by concatenating multiple self-attention outputs:

$$\text{MHA}(\mathbf{F}) = [\text{SA}_1(\mathbf{F}), \text{SA}_2(\mathbf{F}), \ldots, \text{SA}_m(\mathbf{F})]\mathbf{W}_m, \tag{5}$$

where the query, value and key projections are found in the parameter matrix $\mathbf{W}_m \in \mathbb{R}^{m \cdot d_v \times C}$.

Let ℓ be the index of one encoder layer. Then $f'_{i,\ell} \in \mathbb{R}^C$ denotes visual information updated with self attention, and $\mathbf{F}'_\ell \in \mathbb{R}^{N \times C}$ the entire set of updated features in layer ℓ. We compute these features as

$$\begin{aligned} \mathbf{G}'_\ell &= \text{LN}\left(\text{MHA}(\mathbf{F}'_{\ell-1}) + \mathbf{F}'_{\ell-1}\right), \\ \mathbf{F}'_\ell &= \text{LN}\left(\text{MLP}(\mathbf{G}'_\ell) + \mathbf{G}'_\ell\right) \end{aligned} \tag{6}$$

where $\mathbf{F}'_\ell$ corresponds to updated region features after every encoder-layer. The updated features from the previous layers, $\mathbf{F}'_{\ell-1}$, are used as input in the following layers. The multi-layer perceptron (MLP) contains two linear transformations with ReLU activations in between. LN corresponds to layer normalization. We update the original feature map $\mathbf{X}$ with the updated features, $\mathbf{F}'$ with each f_i in place, so that the relative spatial positioning of the features from h is retained in the transformed $\mathbf{X}'$. As a result, we only modify the features for the ROI regions; the rest remains unchanged.

The relational module in TROI was designed with similar motivations as other relational frameworks such as non-local networks [15] and GloRe [3]. We share some basic components such as self-attention. However, we differ in a key aspect in that non-local networks and GloRe compute relations of between every feature position with all others in the feature maps, $\mathbf{X}$ exhaustively. They do not leverage any explicit entity-aware localization, $\mathbf{F}$, so they perform many more redundant relations and computations.

2.3. Implementation Details

Training: Our models are trained for 80 epochs with a batch-size of 64 over two 48GB GPUs. We use SGD optimization with momentum 0.9 and weight decay of 5e-4. The learning rate is set at 0.01 for the initial 20 epochs, decreased to 0.001 for the following 20 epochs, and finally set to 10e-4 for the remaining epochs. The backbone network is initialized with pre-trained Kinetics [2] weights as provided by [10] and fine-tuned on the respective datasets. We then add our module and train using the strategy outlined above. For the transformer, we use two attention heads. We found conv4 as the optimal location for their relational method.

To sample images from the video, we follow [14] and TSM [10]. A fixed number of frames (8 or 16) are selected evenly in time from each video clip. Using more frames, *i.e.*, 16 instead of 8 typically improves the performance at the expense of larger models. Unless otherwise indicated,

		Overall						Unseen Participants			Tail Classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Set	Method	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	SlowFast [5]	65.56	**50.02**	38.54	90.00	**75.62**	58.60	56.43	**41.50**	29.67	36.19	23.26	**18.81**
	TSM [10]	67.86	49.01	38.27	90.98	74.97	**60.41**	58.69	39.62	29.48	36.59	**23.37**	17.62
	TROI	**68.80**	49.22	**38.90**	**91.18**	72.46	58.86	**60.85**	39.53	**30.05**	**37.61**	21.58	17.84
	TSM + TSM	68.24	50.03	39.06	91.18	75.80	61.67	58.59	40.84	29.67	36.02	22.73	17.52
	TSM + TROI	68.98	50.10	39.59	91.09	75.94	61.64	59.91	41.03	30.70	36.99	22.42	17.87
Test	SlowFast [5]	63.79	**48.55**	36.81	88.84	**74.49**	56.39	57.66	**42.55**	29.27	29.65	**17.11**	**13.45**
	TSM [10]	65.32	47.80	**37.39**	89.16	73.95	**57.89**	**59.68**	42.51	**30.61**	30.03	16.96	**13.45**
	TROI	**65.98**	47.19	37.33	**89.58**	72.67	57.29	59.42	41.09	30.02	**30.19**	15.06	13.31
	TSM + TROI	66.63	48.98	38.59	89.94	73.84	58.62	60.56	43.58	31.63	29.80	15.02	12.97

Table 1. EPIC-Kitchens validation and test server results. All methods use RGB and optical flow images. The best non-ensembled results are indicated in **bold**; best ensembled results are underlined. The results of SlowFast [5] and TSM [10] are computed by [4].

we use 8 frames in our experiments. For data augmentation, we use the same strategy as suggested in [10]: scale jittering, corner cropping, and random horizontal flipping. The augmented image is then resized to 224 × 224.

Hand and Object Bounding Boxes: The Epic-Kitchens-100 [4] dataset provides RGB and optical flow images, as well as bounding boxes extracted by a hand-object detection framework [12] that returns class labels left/right hand and left/right interacting object. For each frame, there are up to four bounding boxes for these classes. On average, 70% and 76% of the frames have left and right hand/object detections

3. Experiments

Visual Entities	Epic Top1
All hands & objects	36.40
Only objects	35.83
Only hands	33.41
Right hand & object	35.18
Left hand & object	34.61
No entities	32.04
All entities, IOU@0.05	30.45
All entities, IOU@0.25	34.97
All entities, IOU@0.50	35.73

Table 2. Influence of object and hand detection quality the validation set of Epic using a TSM backbone on RGB.

3.1. Detection Quality

We first evaluate how much impact does the quality of the detections have during inference. To check this, we test our models with either missing or corrupted ROIs and show results in Table 2. Using *“only hands”* regions decreases the performance more compared to using *“only objects”* by 5.7%. Selecting only the right hand and right object, *“r hand & obj”*, performs better than selecting the entities on the left side, *“l hand & obj”*, with a difference of 0.6%. This is likely due to the imbalance of right-handed participants in this dataset. When no ROIs are used during inference, *i.e.* our module is bypassed completely, our performance decreases by more than 4%. This decrease is unsurprising, as our model is trained with the module in place for all samples and the backbone gets adapted to the transformed ROIs in the feature map after the conv4 layer.

Finally, we corrupt the ROIs by shifting the ROIs to decrease the IoU with the original bounding boxes, leaving an IoU of $\alpha = \{50\%, 25\%\ 5\%\}$. This experiment corresponds to explicitly putting emphasis (attending) in wrong ROIs, *i.e.* the background in the middle layers of the CNN. Our model simply transforms the wrong regions and places it back into the overall feature map; this overall feature map, however, still contains visual cues of the object and hand. Our performance does not decrease significantly, even with 50% corruption. However, decreasing the overlap to 5%, which corresponds to an intentional focus on the background regions, decreases our performance by 6%.

3.2. Results on EPIC-KITCHENS-100

We compare with action recognition methods on Epic in Table 1. Like TSM [10] and SlowFast [5], we train two models on RGB and optical flow images individually to predict both verbs and nouns. During inference, we average pre-softmax predictions from the two models. TROI with a TSM backbone outperforms SlowFast for overall and unseen action accuracies on both the validation and test sets. It achieves higher verb scores than SlowFast (+3.2% and +2.2% in overall Top-1 accuracy) but noun scores degrade (-0.4% and -1.4%). On the validation set, we improve the action accuracy over TSM [10] on overall, unseen and classes by 0.6%, 0.6% and 0.2%. However, on the test set, we decrease on the unseen and tail action accuracy by 0.6% and 0.1%. Similar to our comparison to SlowFast, our verb ac-

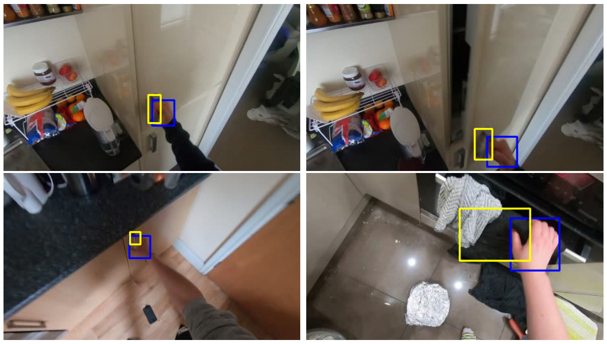

Figure 2. Qualitative examples of misleading object localizations in *"open fridge"* (top row), *"open cupboard"* (bottom left) and *"open oven"* (bottom right).

curacy outperforms TSM in almost all cases, but noun accuracy is generally lower.

TROI's poor performance on the noun accuracy, especially in the tail classes, is puzzling at first glance, since it is supplied with interacting object ROIs. We further break down the classes of Epic and observe that we improve over TSM on actions involving small objects like spoon or knife and mid-sized objects like pan or bowl. But on large objects like fridge or cupboard we do worse. Visualizing the bounding boxes for these classes, we observe that the detection algorithm of [12] has a strong size prior and localizes object parts such as the fridge or oven handle (see Figure 2). We speculate that in these cases, the ROI for the interacting object does not provide sufficient visual context and that one needs to localize the entire object or appliance to identify the noun and therefore action.

To compensate for this weakness, we test an ensemble (TSM + TROI), fusing our predictions with TROI with a standard TSM. The two complement each other well, as the ensemble provides a considerable gain over both our own work and TSM individually. It also edges out an ensemble of two TSMs (TSM + TSM).

4. Conclusion

The relations and transformations of human-centered entities such as hands or interacting objects are critical cues for video recognition. In this work, we introduced a relation module that can be integrated into standard CNNs to model both short and long-range interactions of such entities. Based on these interactions, we transform localized regions of interest within the feature-map via self-attention. Our framework's gains over the state of the art highlights the importance of integrating temporal and relational information for action recognition from videos. It also opens up a new avenue of exploration for relational models via mid-level features.

References

[1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, pages 105–121, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[3] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[7] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *CVPR*, pages 5353–5360, 2015.

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[9] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, pages 513–528, 2018.

[10] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019.

[11] Abhinav Rai, Fadime Sener, and Angela Yao. Transformed rois for capturing visual transformations in videos. *arXiv preprint arXiv:2106.03162*, 2021.

[12] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, pages 9869–9878, 2020.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[14] 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 *ECCV*, pages 20–36, 2016.

[15] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[16] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pages 399–417, 2018.

Anticipative Video Transformer @ EPIC-Kitchens Action Anticipation Challenge 2021

Rohit Girdhar[1]   Kristen Grauman[1,2]
[1]Facebook AI Research   [2]University of Texas, Austin
http://facebookresearch.github.io/AVT

Abstract

In this report, we describe an Anticipative Video Transformer (AVT) [11] based solution for the EPIC-Kitchens-100 anticipation challenge. AVT leverages a vision transformer based backbone architecture followed by causal attention based transformer decoder to model the sequential nature of videos. For the challenge, we aggregate predictions from multiple variants of AVT, applied to different input modalities and backbone architectures, along with prior work. Our final model obtains strong performance on the challenge test set with 16.5% mean top-5 recall in predicting future actions.

1. Introduction

Anticipating actions that a person might do in the future is an important task in egocentric computer vision. It forms the basis for various downstream applications on wearable devices, from safety systems that warn the user before potentially dangerous actions, to an assistive systems that help a user to perform actions by suggesting next steps. Compared to traditional action recognition, anticipation tends to be significantly more challenging. It requires going beyond classifying current spatiotemporal visual patterns into a single action category—a task nicely suited to today's well-honed discriminative models—to instead predict the multi-modal distribution of future activities. Moreover, while action recognition can often side-step temporal reasoning by leveraging instantaneous contextual cues [12], anticipation inherently requires modeling the progression of past actions to predict the future. For instance, the presence of a plate of food with a fork may be sufficient to indicate the action of eating, whereas anticipating that same action would require recognizing and reasoning over the sequence of actions that precede it, such as chopping, cooking, serving, *etc*. Indeed, recent work [9, 18] finds that modeling long temporal context is often important for anticipation, unlike action recognition where frame-level modeling is often enough [14, 21].

To that end, there have been attempts to use sequential modeling architectures for action anticipation. While recurrent models like LSTMs have been explored for anticipation [1, 9, 23], they are known to struggle with modeling long-range temporal dependencies due to their sequential (non-parallel) nature. Recent work mitigates this limitation using attention-based aggregation over different amounts of the context to produce short-term ('recent') and long-term ('spanning') features [18]. However, it still reduces the video to multiple aggregate representations and loses its sequential nature.

Hence, we introduce *Anticipative Video Transformer* (AVT), an alternate video modeling architecture that replaces "aggregation" based temporal modeling with a *anticipative* architecture. Aiming to overcome the tradeoffs described above, the proposed model naturally embraces the sequential nature of videos, while minimizing the limitations that arise with recurrent architectures. Similar to recurrent models, AVT can be rolled out indefinitely to predict further into the future (*i.e*. generate future predictions), yet it does so while processing the input in parallel with long-range attention, which is often lost in recurrent architectures. Furthermore, while it is compatible with various backbone architectures, we leverage the recently proposed vision transformer based architectures [7] as the frame encoder, resulting in an end-to-end attention based architecture.

2. Our Approach

We now describe AVT briefly as illustrated in Figure 1, and refer the readers to the full paper [11] for details.

2.1. Backbone Network

Given a video clip with T frames, $V = \{\mathbf{X}_1, \cdots, \mathbf{X}_T\}$ the backbone network, $\mathcal{B}$, extracts a feature representation for each frame, $\{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ where $\mathbf{z}_t = \mathcal{B}(\mathbf{X}_t)$. While various video base architectures have been proposed [4, 8, 20, 21] and can be used with AVT as we demonstrate later, in this work we propose an alternate architec-

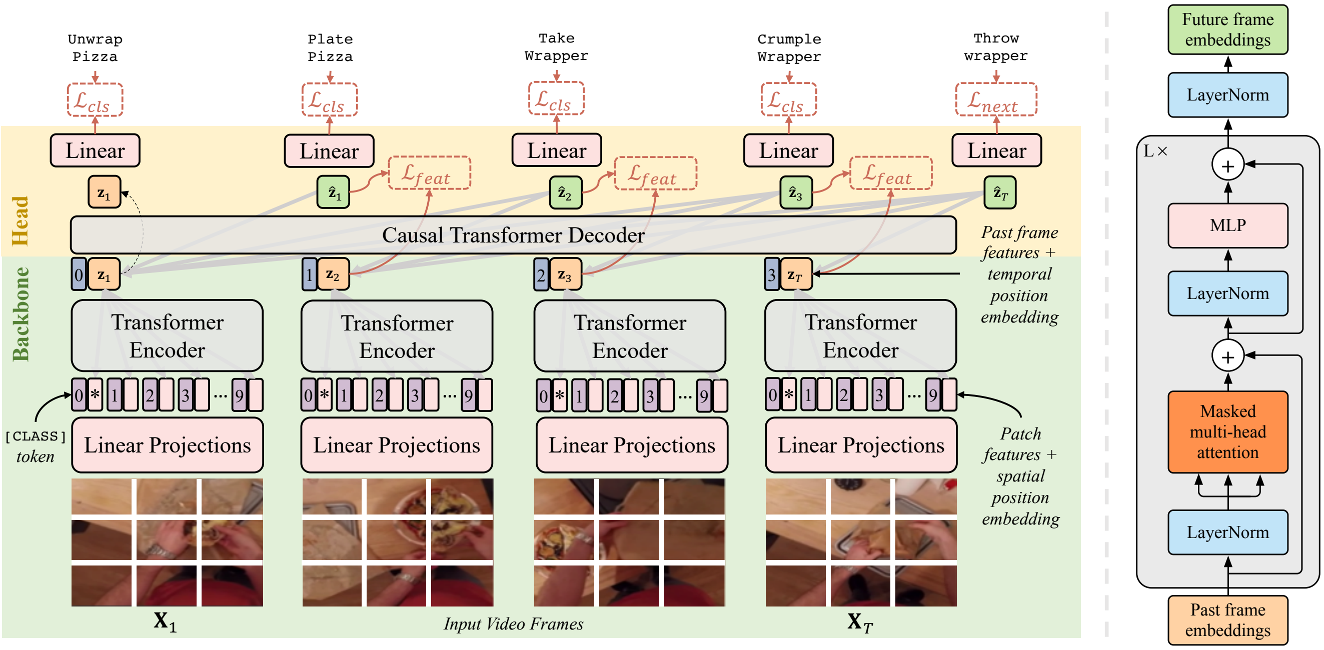

Figure 1: *(Left)* **AVT architecture.** We split the T input frames into non-overlapping patches that are linearly projected. We add a learned `[CLASS]` token, along with spatial position embeddings, and the resulting features are passed through multiple layers of multi-head attention, with shared weights across the transformers applied to all frames. We take the resulting features corresponding to the `[CLASS]` token, append a temporal position encoding and pass it through the Causal Transformer Decoder that predicts the future feature at frame t, after attending to all features from $1 \cdots t$. The resulting feature is trained to regress to the true future feature ($\mathcal{L}_{feat}$) and predict the action at that time point if labeled ($\mathcal{L}_{cls}$), and the last prediction is trained to predict the future action ($\mathcal{L}_{next}$). *(Right)* **Causal Transformer Decoder.** It follows the Transformer architecture with pre-norm [22], causal masking in attention, and a final LayerNorm [16].

ture for video understanding based purely on attention. This backbone, which we refer to as AVT-b, adopts the recently proposed Vision Transformer (ViT) [7] architecture, which has shown impressive results for static image classification. Specifically, we adopt the ViT-B/16 architecture.

AVT-b is an attractive backbone design because it makes our architecture purely attentional. Nonetheless, in addition to AVT-b, AVT is compatible with other video backbones, including those based on 2D CNNs [19, 21], 3D CNNs [4, 8, 20], or fixed feature representations based on detected objects [2, 3] or visual attributes [15]. In § 3 we provide experiments testing several such alternatives. For the case of spatiotemporal backbones, which operate on clips as opposed to frames, we extract features as $\mathbf{z}_t = \mathcal{B}(\mathbf{X}_{t-L}, \cdots, \mathbf{X}_t)$, where the model is trained on L-length clips. This ensures the features at frame t do not incorporate any information from the future, which is not allowed in the anticipation problem setting.

2.2. Head Network

Given the features extracted by the backbone, the head network, referred to as AVT-h, is used to predict the future features for each input frame using a Causal Transformer Decoder, $\mathcal{D}$:

$$\hat{\mathbf{z}}_1, \cdots, \hat{\mathbf{z}}_T = \mathcal{D}(\mathbf{z}_1, \cdots, \mathbf{z}_T). \qquad (1)$$

Here $\hat{\mathbf{z}}_t$ is the predicted future feature corresponding to frame feature $\mathbf{z}_t$, after attending to all features before and including it. The predicted features are then decoded into a distribution over the semantic action classes using a linear classifier θ, *i.e.* $\hat{\mathbf{y}}_t = \theta(\hat{\mathbf{z}}_t)$. The final prediction, $\hat{\mathbf{y}}_T$, is used as the model's output for the next-action anticipation task. Note that since the next action segment $(T+1)$ is τ_a seconds from the last observed frame (T) as per the problem setup, we typically sample frames at a stride of τ_a so that the model learns to predict future features/actions at that frame rate. However, empirically we find the model is robust to other frame rate values as well.

We implement $\mathcal{D}$ using a masked transformer decoder inspired from popular approaches in generative language modeling, such as GPT-2 [16]. We start by adding a temporal position encoding to the frame features implemented as a learned embedding of the absolute frame position within the clip. The embedded features are then passed through multiple decoder layers, each consisting of masked multi-head attention, LayerNorm (LN) and a multi-layer perceptron (MLP). The final output is then passed through another

LN, akin to GPT-2 [16], to obtain the future frame embeddings.

2.3. Training Details

The models are then trained with a combination of three objectives that include next action anticipation, future feature prediction, and current action classification. We refer the reader to the main paper [11] for details.

3. Experiments

3.1. Implementation Details

We preprocess the input video clips by randomly scaling the height between 248 and 280px, and take a 224px crops at training time. We sample 10 frames at 1FPS by default. We adopt network architecture details from [7] for the AVT-b backbone. Specifically, we use a 12-head, 12-layer transformer encoder model that operates on 768D representations. We initialize the weights from a model pre-trained on ImageNet-1K (IN1k), ImageNet-21K (IN21k) or ImageNet-1K finetuned from ImageNet-21K (IN21+1k), and finetune end-to-end for the anticipation tasks. For AVT-h, we use a 4-head, 6-layer model that operates on a 2048D representation, initialized from scratch. We employ a linear layer between the backbone and head to project the features to match the feature dimensions used in the head. We train AVT end-to-end with SGD+momentum using 10^{-6} weight decay and 10^{-4} learning rate for 50 epochs by default, with a 20 epoch warmup [13] and 30 epochs of cosine annealed decay. In all experiments, we train the model to predict the future actions, and verbs/nouns are inferred from the action prediction by marginalizing over the other. At test time, we employ 3-crop testing, where we compute three 224px spatial crops from 248px input frames, and average the predictions over the corresponding three clips.

The default backbone for AVT is AVT-b, based on the ViT-B/16 architecture. However, we also experiment with only our head model operating on fixed features from 1) a frame-level TSN [21] backbone pre-trained for action classification, or 2) a recent spatiotemporal convolutional architecture irCSN-152 [20] pre-trained on a large weakly labeled video dataset [10], which has shown strong results when finetuned for action recognition. We finetune that model for action classification on the anticipation dataset and extract features that are used by the head for anticipation. In these cases, we only train the AVT-h layers. We use the validation set to optimize the hyperparameters for each setting, and use that setup on the held out test sets.

3.2. Ablations

In Table 1, we experimentally compare AVT to prior work and variants of itself with different backbones and modalities on the validation set. We find AVT-h over features from prior work [9] already outperforms prior work. We are able to further improve results with the AVT-b backbone and training jointly, especially with the IN21k initial-

	#	Head	Backbone	Init	Context	Verb	Noun	Action
RGB	1	RULSTM [5]	TSN	IN1k	2.8s	27.5	29.0	13.3
	2	AVT-h	TSN	IN1k	10s	27.2	30.7	13.6
	3	AVT-h	irCSN152	IG65M	10s	25.5	28.1	12.8
	4	AVT-h	AVT-b	IN1k	10s	28.2	29.3	13.4
	5	AVT-h	AVT-b	IN21+1k	10s	28.7	32.3	14.4
	6	AVT-h	AVT-b	IN21k	10s	**30.2**	31.7	14.9
	7	AVT-h	AVT-b	IN21k	15s	30.1	**33.8**	**15.7**
OBJ	8	RULSTM [5]	Faster R-CNN	IN1k	2.8s	17.9	23.3	7.8
	9	AVT-h	Faster R-CNN	IN1k	10s	**18.0**	**24.3**	**8.7**
Flow	10	RULSTM [5]	TSN	IN1k	2.8s	19.1	16.7	**7.2**
	11	AVT-h	TSN	IN1k	10s	**20.9**	**16.9**	6.6

Table 1: EK100 (val) using individual modalities. AVT outperforms prior work using the exact same features, and further improves with our AVT-b backbone. The 15s model (row 7) was also trained for longer (70 epochs as opposed to 50 default). Performance reported using overall class-mean recall@5.

Models fused	Weights	Action
2 + 9	1.5:0.5	14.8
6 + 9	2.5:0.5	15.9
1 + 6 + 9	1.0:1.0:0.5	16.9
1 + 2 + 3 + 6 + 9	1.0:1.0:1.0:1.0:0.5	18.2
1 + 2 + 3 + 6 + 7 + 9 + 11	1.0:1.0:1.0:0.5:1.5:0.5:0.5	**19.2**

Table 2: EK100 (val) late fusing predictions from different architectures. The numbers refer to the model in the corresponding row in Table 1. Performance reported using overall class-mean recall@5 for action prediction.

		Overall			Unseen Kitchen			Tail Classes		
Split	Method	Verb	Noun	Act	Verb	Noun	Act	Verb	Noun	Act
Val	chance	6.4	2.0	0.2	14.4	2.9	0.5	1.6	0.2	0.1
	RULSTM [5]	27.8	30.8	14.0	28.8	**27.2**	**14.2**	19.8	22.0	11.1
	AVT+ (TSN)	25.5	31.8	14.8	25.5	23.6	11.5	18.5	25.8	12.6
	AVT+	**28.2**	**32.0**	**15.9**	**29.5**	23.9	11.9	**21.1**	**25.8**	**14.1**
Test	chance	6.2	2.3	0.1	8.1	3.3	0.3	1.9	0.7	0.0
	RULSTM [5]	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	TBN [24]	21.5	26.8	11.0	20.8	**28.3**	**12.2**	13.2	15.4	7.2
	AVT+	**25.6**	**28.8**	**12.6**	**20.9**	22.3	8.8	**19.0**	**22.0**	**10.1**
Challenge	IIE_MRG	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	NUS_CVML [17]	21.8	30.6	12.6	17.9	27.0	10.5	13.6	20.6	8.9
	ICL+SJTU	**36.2**	32.2	13.4	**27.6**	24.2	10.1	**32.1**	**29.9**	11.9
	Panasonic	30.4	**33.5**	14.8	21.1	27.1	10.2	24.6	27.5	12.7
	AVT++	25.2	32.0	**16.5**	20.4	**27.9**	**12.8**	17.6	23.5	**13.6**

Table 3: EK100 val and test sets using all modalities. We split the test comparisons between published work and CVPR'21 challenge submission. We outperform prior work including all challenge submissions, with especially significant gains on tail classes. Performance reported using class-mean recall@5. AVT+ and AVT++ late fuse predictions from multiple modalities, please see text for details.

ization. Finally, by using additional frames of context and training for longer, we obtained the best RGB-only performance of 15.7%, showing AVT is effective in incorporating long-term context.

Next, to further improve the performance, we aggregate predictions across modalities and models by simple weighted averaging of L_2 normalized predictions, as shown in Table 2. The model numbers refer to the model in the corresponding row in Table 1. We find that combining multiple RGB models, based on fixed features and end-to-end trained, as well as ones using other architectures [9], and AVT-h applied on obj and flow features gave the best results on val set. We use a similar model on the test set as described next.

3.3. Final Model

For the test submission, we first train our models on the train+val set, and test those models as well as the models trained only on train set, on the test set. Then, we late fuse predictions using similar weights as the best combination in Table 2, and for each case where we use both train+val and train-only models, we use the same weight on predictions from both. Specifically, we use both train+val and train-only models for 2, 3, 6, 7 and 9; and train-only models for 1 and 11. This model obtains 16.53% mean top-5 recall for actions, as reported in our challenge submission on the leaderboard. We show the full comparison to existing state-of-the-art as well as challenge submissions in Table 3. Our RGB+Obj (6 + 9) late fused model is referred to as AVT+, and final late fused model is referred to as AVT++. It was submitted to the challenge using CodaLab username "shef" with team name "AVT-FB-UT".

In terms of the supervision scales [6], our pre-training scale is 2 since we use publicly available models pre-trained on public weakly supervised videos [10]. The full available supervision in Epic Kitchens is used for training, leading to supervision level of 4. The training data used is train + val sets, leading to training data scale of 4.

4. Conclusion

We have presented the Anticipative Video Transformer (AVT) architecture as used in the EPIC-Kitchens 2021 challenge. We propose a end-to-end Transformer based architecture for predictive video tasks such as anticipation, and show that it improves over prior work. Our best model, that aggregates predictions across modalities and models, obtains strong performance of 16.5% mean top-5 recall in predicting future actions on the test set.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, 2018.

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020.

[3] Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video. In *NeurIPS*, 2020.

[4] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017.

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[6] Dima Damen and Michael Wray. Supervision levels scale (sls). *arXiv preprint arXiv:2008.09890*, 2020.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[9] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *ICCV*, 2019.

[10] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[11] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. *arXiv preprint arXiv:2106.02036*, 2021.

[12] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

[13] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[14] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[15] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPR Workshop*, 2019.

[16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Fadime Sener, Dibyadip Chatterjee, and Angela Yao. Technical report: Temporal aggregate representations. *arXiv preprint arXiv:2106.03152*, 2021.

[18] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020.

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[20] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolu-

tional networks. In *ICCV*, 2019.

[21] 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 *ECCV*, 2016.

[22] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. In *ACL*, 2019.

[23] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *TIP*, 2021.

[24] Olga Zatsarynna, Yazan Abu Farha, and Juergen Gall. Multimodal temporal convolutional network for anticipating actions in egocentric videos. In *CVPR Workshop*, 2021.

Submission to EPIC-Kitchens Action Anticipation Challenge 2021

Yutaro Yamamuro[1], Kazuki Hanazawa[1], Masahiro Shida[1], Tsuyoshi Kodake[1], Shinji Takenaka[1]
Yuji Sato[2], Takeshi Fujimatsu[2]
{yamamuro.yutaro, hanazawa.kazuki, shida.masahiro, kodake.tsuyoshi,
takenaka.shinji-k, sato.yuji, fujimatsu.takeshi}@jp.panasonic.com

[1]Panasonic System Networks R&D Lab. Co., Ltd.
[2]Innovation Center, Connected Solutions Company, Panasonic Corporation

Abstract

This report describes our approach to the EPIC-Kitchens Action Anticipation task. As a base model, we adopted RULSTM, which has achieved excellent results in past EPIC-Kitchens competitions. In this competition, we tried the following three approaches. 1) Label Smoothing using the semantic expression of words; 2) Uncertainty modeling loss and Background entropy loss for building a model that separates the action frame and the background frame; 3) Weighting by Class-Balanced loss for imbalanced data sets and postponement of weighting by Deferred Re-Weight (DRW). All three methods resulted in improvements over the baseline method. Finally, we submitted the predictions made by the ensemble of Model 3) and recorded an accuracy of 14.82% in the test data.

1. Introduction

EPIC-Kitchens [1] is a large data set that includes videos of working in kitchens taken with head-mounted cameras. The features of EPIC-Kitchens are that the number of Action labels is very large, and the number of training data items is an imbalanced data set for each class.

We took on the Action Anticipation task in this competition. Our contributions are the following.

1) Applied Label Smoothing based on the similarity of language vectors [3] to Action labels that consist of two words: a verb and a noun.

2) Applied Uncertainty modeling loss and Background entropy loss [4] to separate the action frame and background frame and focus on training the action frame.

3) Applied Class Balanced Loss [5], which applies weighting based on the actual number of data items per class, to improve the accuracy of predictions for a small number of classes in an imbalanced dataset. Also applied DRW [6], which applies weighting after training has stabilized, and confirmed its effects.

Our report is structured as follows: the models and methods used in this task are outlined in Section 2. The experiments performed and results are described in Section 3. Section 4 consists of our conclusions and the discussion.

2. Our approach

2.1. Base model

As a base model, we used RULSTM [2], which has achieved excellent results in the Action Anticipation task in past EPIC-Kitchens competitions. The input of RULSTM is learned by each of the three modalities of rgb, flow, and obj. During training, the fusion modality learns the weightings for the three modalities at the same time in a module called MATT [2]. This makes it possible to prioritize the modality that contributes to prediction in each scene.

For the input data of RULSTM, rgb and flow used a 1024-dimensional vector extracted from Temporal Segment Networks [9] that had been pre-trained in the task of action recognition. Obj used a 352-dimensional vector extracted from Faster-RCNN [10] that had been pre-trained in the object detection task.

2.2. Label smoothing

Label Smoothing is normally used to improve the generalization performance of the model via the regularization effect. In past competitions, it was reported that when Label Smoothing is applied to Action labels of the EPIC-Kitchens dataset [1], the prediction accuracy improves. Also, in past competitions, not only simple Label Smoothing, but also a method has been proposed of assigning a higher value to "incorrect answers that are close to the correct answer" according to the EPIC-Kitchens domain.

Among the proposed methods [3], Label Smoothing using GloVe [7] is effective: this method makes the label assignment proportional to the similarity of word vectors. We tried Label Smoothing with BERT [8], a pre-trained language model, to obtain better word vectors. GloVe [7] obtains a word-by-word vector, while BERT [8] can obtain a word vector for a set of verbs and nouns that make up an

Table 1: Results for the EPIC-Kitchens validation data. Bold red indicates the maximum value for each column. LS: Label Smoothing, Um & Be: Uncertainty modeling loss & Background entropy loss, CB: Class-Balanced loss, DRW: Deferred Re-Weight, TTA: Test-Time Augmentation

Algo	**Modality**	**Mean Top-5 Recall % (1.0 sec)**								
		Overall			Unseen			Tail		
		verb	noun	action	verb	noun	action	verb	noun	action
Base Model [2]	fusion	27.76	30.76	14.04	28.78	27.22	14.15	19.77	22.02	11.14
LS BERT	fusion	25.91	32.64	15.19	28.44	27.88	14.49	17.59	24.25	12.26
Um & Be	fusion	29.57	32.35	15.15	29.48	25.08	13.79	22.39	25.09	12.56
Um & Be + LS BERT	fusion	27.59	32.75	15.89	30.42	24.77	14.34	19.82	25.23	13.41
Um & Be + DRW	fusion	**32.83**	33.92	17.07	29.03	26.28	12.15	**27.19**	29.43	16.12
CB	fusion	28.81	32.09	16.02	26.96	24.92	10.50	23.30	31.02	16.21
CB + LS BERT	fusion	27.61	32.45	15.72	24.76	25.09	10.52	21.81	28.61	15.29
CB + DRW	fusion	32.36	**36.58**	17.95	**33.21**	**27.89**	14.34	26.27	**31.68**	16.68
CB + DRW + TTA	fusion	32.53	36.42	**18.27**	32.93	26.91	**15.37**	26.51	31.41	**17.09**

action label. To express closeness between labels, the cosine similarity of the acquired word vector is calculated and the square is assigned to each label.

2.3. Uncertainty modeling loss & Background entropy loss

Uncertainty modeling loss & Background entropy loss is a method proposed by Lee et al. [4]. Uncertainty modeling loss can model uncertainty without frame-by-frame labeling, allowing the action frame and the background frame to be better separated. Background entropy loss can also evenly distribute the probability distribution of actions across all action classes to better distinguish background frames. The overall loss function is as follows.

$$L_{total} = L_{cls} + \alpha L_{um} + \beta L_{be} \quad (1)$$

$$L_{um} = \frac{1}{N}\sum_{n=1}^{N}\left(\max(0, m - \|f_n^{act}\|) + \left\|f_n^{bkg}\right\|\right)^2 \quad (2)$$

$$L_{be} = \frac{1}{NC}\sum_{n=1}^{N}\sum_{c=1}^{C} - \log\left(p_c(s_n^{\sim bkg})\right) \quad (3)$$

L_{cls} is the classification error for action labels. We adopted cross entropy. L_{um} and L_{be} respectively indicate Uncertainty modeling loss and Background entropy loss. α and β are hyper-parameters. f_n^{act} *and* f_n^{bkg} are the mean feature of the pseudo action and background segments of the n-th video. m is the maximum feature magnitude. $p_c(s_n^{\sim bkg})$ is the averaged action probability for the c-th class of background segments.

We have adopted Uncertainty modeling loss & Background entropy loss as the loss function of RULSTM to be able to separate the action frame and the background frame and focus on training the action frame in the scene before the action to be predicted occurs.

2.4. Class-Balanced loss & DRW

EPIC-Kitchens is an imbalanced dataset [1] with a large bias in the number of samples per class.

Therefore, when training without taking measures against imbalanced data sets, the prediction of the "major" class with a large number of samples works well, but the training does not proceed in the "tail" class with a small number of samples, so prediction does not work well.

As a countermeasure to imbalanced data sets, Cui et al. [5] proposed a weighting method for each class. Weights are calculated by the effective number of data items per class. The effective number [5] E_n is calculated as follows.

$$E_n = \frac{1 - \beta^n}{1 - \beta} \quad (4)$$

β is hyper-parameter $\beta \in [0,1)$. n is the number of samples. Class-Balanced loss (CB) is defined as follows.

$$\mathrm{CB}(\mathbf{p}, y) = \frac{1}{E_{n_y}} L(\boldsymbol{p}, y) = \frac{1 - \beta}{1 - \beta^{n_y}} L(\boldsymbol{p}, y) \quad (5)$$

p is the model's estimated class probabilities. n_y is the number of samples in the ground-truth class y. We adopted binary cross entropy as loss function L.

There are several classes with no samples in the EPIC-Kitchens training data. Such classes replace the number of samples with 1 when calculating the effective number of data.

As a technique for improving accuracy, we also adopted Deferred Re-Weight (DRW) as proposed by Cao et al. [6]. One of the challenges of weighting each class is training instability [6]. As a countermeasure to this problem, DRW weights not from the beginning of training, but after training has stabilized.

2.5. Test-Time Augmentation

Test-Time Augmentation (TTA) was adopted as a method of improving the score [12]. TTA is a method of augmenting input data at the test time to obtain multiple predictions from the same model. The obtained multiple predictions are blended based on the following equations to obtain the final prediction. $\alpha \in [0,1]$ is the hyper-parameter.

$$pred_{total} = \alpha * pred_{noTTA} + (1 - \alpha) * pred_{TTA} \quad (6)$$

3. Experiment

3.1. Base model

The input of RULSTM was 14 frames before the action occurred. The first six frames were used for encoding, and the subsequent eight frames were used as anticipation frames. The input data is the EPIC-Kitchens video converted to 30 fps and acquired at 0.25-second intervals.

In each modality (rgb, flow, obj), RULSTM trained 100 epochs of pre-training called "sequence completion" [2], and after the pre-training was complete, 100 epochs of action anticipation training. The fusion modality trained 100 epochs of training with the parameters of the three modalities that had completed the training of action anticipation as initial weights. We used SGD as the optimizer, with a learning rate of 1e-2 and a momentum of 0.9.

3.2. Label Smoothing

When training with Label Smoothing based on the word vector obtained from BERT [8], the hyper-parameter α of Label Smoothing was set from 0.1 to 0.6 at intervals of 0.1. The training was conducted under the same conditions as the Base Model. As a result, when α = 0.3, a score of 15.19% was recorded for the validation data, which was an improvement of 1.15% from the base model (Table 1).

We also compared Label Smoothing methods used in past competitions. The BERT-based method proved to be the best (Table 2).

Table 2. The results of comparing Label Smoothing methods

Method	**α**	verb%	noun%	action%
Uniform [3]	0.1	**28.61**	31.92	14.90
VerbNoun [3]	0.1	27.47	29.95	13.92
GloVe [3]	0.2	26.89	**32.73**	15.08
VerbNoun+GloVe [3]	0.5	28.06	31.20	14.12
BERT	0.3	25.91	32.64	**15.19**

3.3. Uncertainty modeling loss & Background entropy loss

There are four types of hyper-parameters α, β, act, bkg for Uncertainty modeling loss and Background entropy loss (Eq. 1, 2, 3). These four types of hyper-parameters and the ratio of Dropout inside RULSTM were tuned using Optuna [11]. As a result of training with tuned parameters, 15.15% was recorded for validation data, an improvement of 1.11% from the base model. (Table 1)

Next, we examined the combination with Label Smoothing. We carried out training by simultaneously applying BERT-based Label Smoothing, which was the best in the experiment in 3.2. As a result, we recorded 15.89% for the validation data, an improvement of 1.85% from the base model (Table 1).

3.4. Class-Balanced loss & DRW

We trained by setting β = 0.999 and 0.9999 as the hyper-parameters in Equation 5. As a result, when β = 0.999, 16.02% was recorded for the validation data, an improvement of 1.98% from the base model. (Table 1)

In DRW, we tuned the timing of the start of weighting. We compared four patterns of weighting, starting from 50 epochs, 60 epochs, 70 epochs, and 80 epochs. DRW was applied to all the training processes, including pre-training. When weighting was started from 60 epochs, 17.95% was recorded for the validation data, an improvement of 3.91% over the base model (Table 3).

Table 3. The results of comparing DRW start epoch

Start epoch	verb%	noun%	action%
50	31.84	36.16	17.44
60	32.36	36.58	**17.95**
70	31.41	35.75	17.68
80	**32.37**	**36.85**	17.69

We also verified the combination of Class-Balance loss, DRW and other verification items.

Class-Balanced loss × Label Smoothing

BERT-based Label Smoothing was applied and trained in three patterns from α = 0.1 to 0.3; however, none of them improved. As α is increased, the prediction accuracy falls, so it appears that the effect of weighting for each class is weakened by the regularization of Label Smoothing.

DRW × Uncertainty modeling loss & Background entropy loss

The classification error L_{cls} of Eq. 1 was weighted as calculated by Class Balanced Loss. DRW was applied from 60 epochs. It improved by recording 17.05% of the validation data compared to the result in 3.3, but it did not update the best score.

3.5. Test-Time Augment

We adopted TTA, which randomly replaces the feature vector of up to two out of six frames for encoding with 0. The final prediction was a blend of action scores without TTA and action scores with TTA at a ratio of 6: 4. Our final submitted score was the application of this TTA to the result in 3.4, recording 18.27% of the validation data (Table 1).

4. Conclusion & Discussion

In this report, to improve the accuracy of the Action Anticipation task using the RULSTM model, we used Label Smoothing employing the meaning expression of words, and Uncertainty modeling loss & Background entropy loss to focus on the action frame. Furthermore, we verified Class Balanced Loss & DRW as a countermeasure to unbalanced data sets. Of these, Class-Balanced loss & DRW proved the most effective in this competition, with a score of 14.82% on the leaderboard.

Class-Balanced Loss & DRW, which recorded the best score, improved the accuracy of the Tail class compared to other methods. This result is in line with the purpose of introducing Class-Balanced Loss, which is to improve the prediction of classes with a small number of samples. It can therefore be said that the effect has been confirmed. However, for the Unseen class, it did not improve as expected. This result means a decrease in generalization performance for videos of domains not included in the training data. Improving the Unseen class is therefore a challenge for the future.

References

[1] Damen, Dima and Doughty, Hazel and Farinella, Giovanni Maria and Furnari, Antonino and Ma, Jian and Kazakos, Evangelos and Moltisanti, Davide and Munro, Jonathan and Perrett, Toby and Price, Will and Wray, Michael. Rescaling Egocentric Vision arXiv:2006.31256, 2020.

[2] Antonino Furnari and Giovanni Maria Farinella. Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video. arXiv:2005.02190, 2020.

[3] Guglielmo Camporese, Pasquale Coscia, Antonino Furnari, Giovanni Maria Farinella, Lamberto Ballan. Knowledge Distillation for Action Anticipation via Label Smoothing. arXiv:2004.07711, 2020.

[4] Pilhyeon Lee, Jinglu Wang, Yan Lu, Hyeran Byun. Weakly-supervised Temporal Action Localization by Uncertainty Modeling. arXiv:2006.07006, 2020.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. arXiv:1901.05555, 2019.

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, Tengyu Ma. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. arXiv:1906.07413, 2019

[7] Jeffrey Pennington, Richard Socher, Christopher D. Manning. *Empirical Methods in Natural Language Processing (EMNLP)*. GloVe: Global Vectors for Word Representation. pages 1532-1543, 2014.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, 2018.

[9] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. arXiv:1608.00859, 2016.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497, 2015.

[11] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902, 2019.

[12] Wang, Guotai and Li, Wenqi and Aertsen, Michael and Deprest, Jan and Ourselin, Sebastien and Vercauteren, Tom. Test-time augmentation with uncertainty estimation for deep learning-based medical image segmentation. 2018

TransAction: ICL-SJTU Submission to EPIC-Kitchens Action Anticipation Challenge 2021

Xiao Gu[1], Jianing Qiu[1], Yao Guo[2], Benny Lo[1], Guang-Zhong Yang[2]
[1]Imperial College London, UK
[2]Shanghai Jiao Tong University, China
{xiao.gu17,jianing.qiu17,benny.lo}@imperial.ac.uk, {yao.guo, gzyang}@sjtu.edu.cn

Abstract

In this report, the technical details of our submission to the EPIC-Kitchens Action Anticipation Challenge 2021 are given. We developed a hierarchical attention model for action anticipation, which leverages Transformer-based attention mechanism to aggregate features across temporal dimension, modalities, symbiotic branches respectively. In terms of Mean Top-5 Recall of action, our submission with team name ICL-SJTU achieved 13.39% *for overall testing set,* 10.05% *for unseen subsets and* 11.88% *for tailed subsets. Additionally, it is noteworthy that our submission ranked 1st in terms of verb class in all three (sub)sets.*

1. Introduction

Egocentric action anticipation [1] is receiving increasing attention recently, which aims to anticipate what the subject to do next based on the recordings from egocentric cameras. Different from the third-person action anticipation, it actually records what the subject observes and performs high-level perception of in the brain. Associating past sensory input with future actions is a fundamental step for understanding human cognition mechanisms.

It is a challenging problem since future events are highly uncertain, and there exist several possible diverse predictions based on the observation of the past [3]. It is difficult to establish an explicit model between the past and the future, as the sensory input (e.g. visual observation) may have asynchronous casual effect on the next action and the future is of multi-modality in nature. Directly arranging the sensory input as a sequential order and feeding it to some conventional temporal modelling architectures (e.g. RNN) may tend to ignore the effects contributed by some relatively old experiences. In our submission, we adopted the Transformer to dynamically fuse information across time, modalities, and *verb* & *noun* branches.

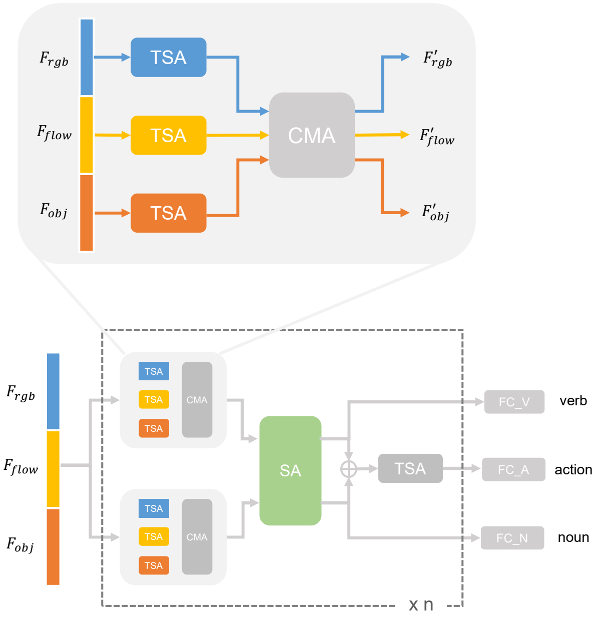

Figure 1. Overview of our hierarchical Transformer-based fusion framework. Our framework is a cascade of several singular blocks. In each block, the temporal self-attention (TSA) module aims to model long-range temporal information, capturing asynchronous effect for the action anticipation. The cross-modality attention (CMA) module aims to fuse information across modalities via Transformer-based attention mechanism. The symbiotic attention (SA) module serves for the mutual interaction between *verb* and *noun* branches with the goal of benefiting each other.

On the other hand, each label of egocentric actions in Epic-Kitchen is formulated as a {*verb, noun*} pair. The combination of different *verbs* and *nouns* would lead to thousands of candidates [3]. Similar to the "long-tailed" distribution in many real-world applications, the majority of actions only occur very few times. Such imbalanced distribution would decrease the generalization capability of trained model on rare classes. In this report, we adopted a state-of-the-art method, Equalization Loss [7], to handle the long-tailed distribution problem.

2. Methods

We directly adopted the multi-modality feature provided by RULSTM [2, 4], which consists of features from three modalities, rgb F_{rgb}, flow F_{flow}, and object F_{obj}. F_{rgb} and F_{flow} were extracted from pretrained TSN models [9] on the action recognition task. F_{obj} was formed by the object probability score predicted by pretrained FasterRCNN model [6]. Each input $F \in \mathbb{R}^{N \times D_f}$ denotes the feature vector with a dimensionality of D_f extracted from N frames, (3.5-1)s before the beginning of the actions.

Our key idea is to exploit Transformer based attention mechanisms to fuse information from temporal dimension, different modalities, as well as verb/noun branches. The overall framework is illustrated in Fig. 1 and the details of each basic component are given below.

2.1. Temporal Self-Attention (TSA)

Instead of applying conventional network architectures for temporally modelling like LSTM/GRU, we applied Transformer [8] to better model the long-range temporal relationship by attention mechanisms. The input feature vector is added by sinusoidal positional embedding to incorporate the positional information. It transforms the input feature to a set of queries ($\mathbf{Q}$), keys ($\mathbf{K}$) and values ($\mathbf{V}$) via linear projection. Subsequently, the attention weights computed from the normalized dot product of $\mathbf{Q}$ and $\mathbf{K}$ are applied to aggregate values, as formulated in Eq. 2. It subsequently applies add & norm operations to enable residual connections, as formulated in Eq. 3. Subsequently, nonlinear feedforward MLPs followed by add & norm residual connections are applied, as in Eq. 4.

$$\mathbf{Q} = \mathbf{F}\mathbf{W}^q, \mathbf{K} = \mathbf{F}\mathbf{W}^k, \mathbf{V} = \mathbf{F}\mathbf{W}^v \tag{1}$$

where $\mathbf{W}^q \in \mathbb{R}^{D_f \times D_q}$, $\mathbf{W}^k \in \mathbb{R}^{D_f \times D_k}$, $\mathbf{W}^v \in \mathbb{R}^{D_f \times D_v}$ denote corresponding linear projection matrices.

$$\mathbf{A} = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)\mathbf{V} \tag{2}$$

$$\mathbf{F}^{'} = layer\_norm(\mathbf{A} + \mathbf{F}^{in}) \tag{3}$$

$$\mathbf{F}^{out} = layer\_norm(\mathbf{MLP}(\mathbf{F}') + \mathbf{F}^{'}) \tag{4}$$

2.2. Cross-Modality Attention (CMA)

To make use of the complementary information encoded in different modalities, we introduced a cross-modality attention (CMA) mechanism, which is expected to capture asynchronous yet relevant information across modalities. Inspired by the fusion method proposed in [5], we concatenate F_{rgb} F_{flow} F_{obj} into a feature with a shape of $N \times \sum D_f$, and then apply the CMA module to aggregate features across time.

2.3. Symbiotic Attention (SA)

Similar to previous action recognition/anticipation works, we utilized two branches to predict *verb* and *noun* separately. However, it is not appropriate to consider *verb* and *noun* as two independent variables to be predicted by two independent branches, since they share mutual contextual information [10]. The awareness of the next active object provides the prior probability for predicting the next verb, whereas predicting the next verb would help recognize the next object to be manipulated. Therefore, we incorporated another Transformer module for the interaction between *verb* and *noun* branches. This module, referred to as Symbiotic Attention (SA) module, applied Transformer network to process concatenated feature input with a shape of $2N \times \sum D_f$.

2.4. Cascaded Architecture

Based on the TSA, CMA, and SA modules, the illustration of our network architecture is given in Fig. 1. It firstly processes the input of each modality by their corresponding TSA modules. Subsequently, the CMA modules in both branches fuse features across multiple modalities, followed by a SA module performing interactions between both branches. Finally, the features extracted from two branches are concatenated together and fed into another TSA module to predict the action. We developed a cascaded architecture with the repetition of the same block, whereas the output of each block is extracted for prediction. In practice, the block number n is set as 2.

2.5. Equalization Loss

To deal with the long tailed distribution, we adopted the Equalization Loss proposed in [7]. It proposed a simple yet effective loss aimed at protecting the learning of rare classes by randomly neglecting the updating of rare classes when the target is a majority class. The loss function is modified from cross-entropy loss, and its formulation is shown as below,

$$L_{SEQL} = -\sum_{j=1}^{c} y_j \log(\tilde{p}_j) \tag{5}$$

$$\tilde{p}_j = \frac{e^{z_j}}{\sum_{k=1}^{c} \tilde{w}_k e^{z_k}} \tag{6}$$

$$\tilde{w}_k = 1 - \beta T_\lambda(y_k)(1 - y_k) \tag{7}$$

where β is random binary variable with a probability of γ to be 1 and otherwise 0. $T_\lambda(y_k)$ is a threshold function determining whether y_k is a majority class by predefined occurrence frequency threshold.

3. Implementation Details

The whole model was implemented with Pytorch and trained on a single RTX 2080 Ti GPU. The batch size was

Table 1. Results of Ablation Studies on Validation Set.

Method	Overall (%)			Unseen (%)			Tail (%)		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
RULTSM[2]	27.76	30.76	14.04	28.78	27.22	**14.15**	19.77	22.02	11.14
TSA-RGB	33.23	32.65	13.71	28.65	20.61	10.23	29.12	31.41	13.34
TSA-Flow	24.19	17.02	6.74	30.61	15.74	6.01	19.33	15.46	5.72
TSA-Obj	25.37	29.51	9.93	28.39	22.19	7.06	21.26	28.09	9.51
w/o CMA	31.46	31.92	14.90	34.10	23.47	10.22	26.37	30.14	14.56
w/o SA	**35.78**	32.18	12.93	29.79	17.56	10.51	**32.08**	31.01	12.43
w/o Equal	27.65	31.34	14.16	27.49	25.25	12.61	20.92	25.60	11.98
Proposed-Single	33.60	32.54	15.05	33.05	25.43	11.96	29.04	31.03	14.39
Proposed-Ensemble	35.04	**35.49**	**16.60**	**34.64**	**27.26**	13.83	30.08	**33.64**	**15.53**

Table 2. Results of Testing Set on LeaderBoard.

Method	Overall (%)			Unseen (%)			Tail (%)		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
RULSTM-RGB	24.69	26.38	10.45	17.88	23.16	9.13	17.32	16.79	7.39
RULSTM-Flow	21.24	18.12	7.36	17.27	18.95	6.86	13.54	9.44	4.97
RULSTM-OBJ	13.93	15.17	3.96	14.05	20.41	5.79	6.18	5.37	1.85
RULSTM-Fusion	25.25	26.69	11.19	19.36	26.87	9.65	17.56	15.97	7.92
Proposed-Single	37.13	30.19	12.44	29.72	20.87	10.57	34.53	28.42	9.74
Proposed-Ensemble	36.15	32.20	13.39	27.60	24.24	10.05	32.06	29.87	11.88

set as 128 and we applied SGD optimizer with a learning rate of 0.01 and a momentum of 0.9. The implementation details can be found in https://github.com/guxiao0822/trans_action.

To participate in the challenge, we developed an ensemble of three trained models based on our proposed method together with the baseline RULSTM-Fusion to achieve performance gains from their complementary information.

4. Results and Discussion

Following the evaluation guideline of this challenge[1], the Mean Top-5 Recall Metric is used. First of all, to demonstrate the effectiveness of different modules proposed, we conducted ablation study on the validation subset with the results shown in Table 1. The TSA-RGB/Flow/Obj refers to the variant only applying TSA with their corresponding single-modality feature as input. w/o CMA, SA denote the variants with CMA, SA module removed respectively. w/o Equal replaces the Equalization Loss by the conventional cross-entropy loss. It can be observed that overall the complete method performs well.

For the test set, The final results of our single model and the ensemble version are given in Table 2, together with the results of the baseline method RULSTM [4]. As shown in Table 2, for our single model, our method competes against the baseline methods regarding most metrics. Especially for the tail classes, a significant improvement can be observed. The ensemble of our models and RULSTM_Fusion leads to slight improvement in terms of some metrics, especially for the result of Tail *action*. It is also noteworthy that our proposed method ranked 1st for *verb* in all three (sub)sets.

We noticed marginally preferable results reported by some other teams in terms of *action* as shown in the Leaderboard. Future work should be targeted at further exploring the symbiotic relationship between *verb* and *noun* for the improvement of *action* classification. Modelling the temporal transition of different actions as well as the label distribution to handle label uncertainty should also be taken into consideration.

[1] https://competitions.codalab.org/competitions/25925

References

[1] D. Damen, H. Doughty, G. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[2] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 2, 3

[3] A. Furnari, S. Battiato, and G. Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[4] A. Furnari and G. Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2, 3

[5] A. Prakash, K. Chitta, and A. Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[7] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11662–11671, 2020. 1, 2

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. 2

[9] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[10] X. Wang, L. Zhu, Y. Wu, and Y. Yang. Symbiotic attention for egocentric action recognition with object-centric alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

EPIC-KITCHENS-100 Action Anticipation Challenge 2021
User “temporalAgg” Technical Report

Dibyadip Chatterjee[1], Fadime Sener[2], Angela Yao[1]
[1]National University of Singapore
[2]University of Bonn, Germany
dibyadip@comp.nus.edu.sg, sener@cs.uni-bonn.de, ayao@comp.nus.edu.sg

Abstract

This technical report describes the approach of user “temporalAgg” (team “NUS_CVML”) for the EPIC-KITCHENS-100 Action Anticipation Challenge 2021. In this report, we predict upcoming actions in long videos of daily activities. Future prediction requires reasoning from current and past observations. We employ the multi-granular temporal aggregation framework from Sener et al. *[9], for action anticipation. This method achieved competitive results in EPIC-KITCHENS-55 Action Anticipation Challenge 2020. We train this method on the EPIC-KITCHENS-100 videos and further boost its performance with additional features. Our submission is ranked 4th on the leaderboard of the challenge. Our code and models can be found at* `https://github.com/dibschat/tempAgg`

1. Introduction

We tackle long-term video understanding, specifically anticipating not-yet-observed but upcoming actions. The anticipation task of EPIC-KITCHENS-100 requires anticipating the future action $\tau_\alpha = 1$s before it starts. We use a general framework from [9, 8] for encoding long-term videos. We split video streams into snippets of equal length and max-pool the frame features within the snippets. We then create ensembles of multi-scale feature representations that are aggregated bottom-up based on scaling and temporal extent. The model is described in detail in [9], and we refer the reader to this paper for further detail.

An overview of the building blocks of this framework can be found in Fig. 1. Based on different start and end frames i and j and number of snippets K, we define two types of snippet features: *‘recent’* features $\{\mathcal{R}\}$ from recent observations and *“spanning”* features $\{\mathcal{S}\}$ drawn from the long-term video. The recent snippets cover the couple of seconds (or up to a minute, depending on the temporal granularity) before the current time point, while spanning snippets refer to the long-term past and may last up to ten minutes. In Fig. 1 we use two starting points to compute the “recent past” snippet features and represent each with $K_R = 3$ number of snippets (🟨🟩🟥 & 🟩🟥🟥). In Fig. 1 we use three scales to compute the “spanning past” snippet features with $K = \{7, 5, 3\}$ (🟦🟧🟨🟨🟨🟩🟥, 🟦🟧🟨🟩🟥 & 🟦🟨🟥). Key to both types of representations is the ensemble of snippet features from multiple scales.

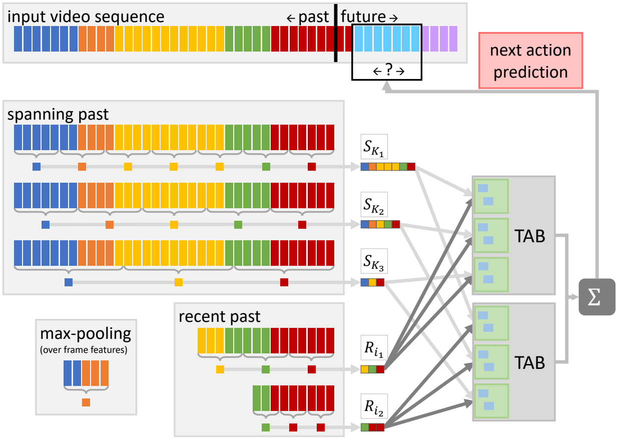

Figure 1. Model overview: In this example we use 3 scales for computing the “spanning past” snippet features $\mathbf{S}_{K_1}, \mathbf{S}_{K_2}, \mathbf{S}_{K_3}$, and 2 starting points to compute the “recent past” snippet features, $\mathbf{R}_{i_1}, \mathbf{R}_{i_2}$, by max-pooling over the frame features in each snippet. Each recent snippet is coupled with all the spanning snippets in the Temporal Aggregation Blocks (TAB). An ensemble of TAB outputs is used for next action anticipation. Best viewed in color.

The framework is built in a bottom-up manner, starting with the recent and spanning features $\mathcal{R}$ and $\mathcal{S}$, which are coupled with non-local blocks (NLB) within coupling blocks (CB). Non-local operations [12] are applied to capture relationships amongst the spanning snippets and between spanning and recent snippets. Two such NLBs are combined in a Coupling Block (CB) which calculates

# segments	$\{i\}$(in seconds (s))	spanning scope (s)	K_R	$\{K\}$
90K	$\{t-1.6, t-1.2, t-0.8, t-0.4\}$	6	2	$\{2, 3, 5\}$

Table 1. Our model parameters.

		Overall			Unseen Participants			Tail Classes		
Split	Modality	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	RGB (TSM)	23.41	29.28	12.91	$\underline{\textbf{29.44}}$	23.92	$\underline{13.33}$	15.56	22.00	10.24
	RGB(TSN)	$\underline{\textbf{24.22}}$	$\underline{29.76}$	$\underline{13.02}$	27.04	22.95	12.21	$\underline{\textbf{16.23}}$	$\underline{22.93}$	$\underline{10.41}$
	Flow (TSN)	18.90	18.68	7.27	26.53	18.86	9.54	10.65	12.53	5.25
	Obj	20.45	27.64	10.45	24.17	$\underline{24.71}$	11.45	12.55	19.31	7.36
	ROI (TSN)	21.22	26.61	11.62	25.49	19.16	10.10	13.36	19.91	9.10
	Late Fusion	23.15	**31.37**	**14.73**	28.01	**26.23**	**14.47**	14.50	**22.47**	**11.75**

Table 2. Action **anticipation** results (reported in class-mean top-5 recall (%)) on EPIC-KITCHENS-100 validation set. Here, late fusion refers to the average voting of the outputs from the modalities RGB (TSN), Flow (TSN), Obj and ROI (TSN).

		SLS			Overall			Unseen Participants			Tail Classes		
Split	Modality	PT	TL	TD	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Test	Late Fusion	1.0	4.0	4.0	21.76	30.59	12.55	17.86	27.04	10.46	13.59	20.62	8.85

Table 3. Action **anticipation** results (reported in class-mean top-5 recall (%)) on EPIC-KITCHENS-100 test set. SLS (Supervision Levels Scale) [3] for our results are: SLS-Pretraining (PT) = 1.0 (pre-trained on public image datasets), SLS Training Labels (TL) = 4.0 (full-supervision i.e. spatio-temporal), SLS Training Data (TD) = 4.0 (trained on Train+Val set). Late fusion refers to the average voting of the outputs from the modalities RGB (TSN), Flow (TSN), Obj and ROI (TSN).

attention-reweighted recent and spanning context representations. Each recent with all spanning representations are coupled via individual CBs, and their outputs are combined in a Temporal Aggregation Block (TAB). Outputs of different TABs are then chained together for action anticipation.

2. Implementation Details

We train our model using the Adam optimizer [6] with batch size 10, learning rate 10^{-4} and dropout rate 0.3. We train for 15 epochs and decrease the learning rate by a factor of 10 every 10^{th} epoch. We use 512 dimensions for all non-classification linear layers.

Parameters: The spanning scales $\{K\}$, recent scale K_R and recent starting points $\{i\}$ are given in Table 1. In our work, we anticipate the action classes directly rather than anticipating the verbs and nouns independently [2] which is shown to outperform the latter [4].

3. Features

We use the appearance (RGB), motion (optical flow), and object-based features provided by Furnari and Farinella [5] for reporting the baseline results on EPIC-100. They independently train two CNNs using the TSN [11] framework on RGB and flow images for action recognition on EPIC-Kitchens-100. They also train object detectors to recognize the 352 object classes of the EPIC-KITCHENS-100 dataset.

We also extract RGB features from TSM [7] using a model pre-trained on EPIC-Kitchens-100 provided by [1]. In addition to this, we also extract regions of interest (ROI) features from pre-trained TSN and TSM models for the hand-object interaction regions in frames. We use the interacting hand-object bounding boxes provided by [10] and consider the union of these boxes to be our ROI for the frame. The RGB features from this ROI help our model to ignore the background clutter that adversely affects our performance and focus primarily on the interacting regions. We observed that ROI features extracted from TSN perform better than the TSM-based features. We think this is because TSM is a segment-based model, whereas we extract ROI features for each frame. The feature dimensions are 1024/2048, 1024 and 352, 1024 for appearance (TSN/TSM), motion, object, and ROI features, respectively.

4. Results on EPIC-KITCHENS-100

The **anticipation** task of EPIC-KITCHENS-100 requires anticipating the future action $\tau_\alpha = 1$s before it starts. We train our model separately for each feature modality (appearance, motion, object and ROI) with the parameters described in Table 1.

During inference, we apply a late fusion of the predictions from the different modalities by average voting. We report our results (class-mean top-5 recall (%)) for validation in Table 2 for different modalities and late fusion. We report our late fusion-based results on the hold-out test data on EPIC-KITCHENS-100 Action Anticipation Challenge (2021) in Table 3 for the entire set (overall), unseen participants not present in the training set and tail classes.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[3] Dima Damen and Michael Wray. Supervision levels scale (sls). *CoRR*, abs/2008.09890, 2020.

[4] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *ECCV, Workshops*, 2018.

[5] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *ICCV*, 2019.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014.

[7] 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.

[8] Fadime Sener, Dibyadip Chatterjee, and Angela Yao. Technical report: Temporal aggregate representations. *arXiv preprint arXiv:2106.03152*, 2021.

[9] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020.

[10] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[11] 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.

[12] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

A Stronger Baseline for Ego-Centric Action Detection

Zhiwu Qing[1,4†] Ziyuan Huang[2,4†] Xiang Wang[1,4] Yutong Feng[3,4] Shiwei Zhang[4∗]
Jianwen Jiang[4] Mingqian Tang[4] Changxin Gao[1] Marcelo H. Ang Jr[2] Nong Sang[1∗]
[1]Key Laboratory of Image Processing and Intelligent Control
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]ARC, National University of Singapore
[3]Tsinghua University [4]Alibaba Group
{qzw, wxiang, cgao, nsang}@hust.edu.cn
ziyuan.huang@u.nus.edu, mpeangh@nus.edu.sg
fyt19@mails.tsinghua.edu.cn
{zhangjin.zsw, jianwen.jjw, mingqian.tmq}@alibaba-inc.com

Abstract

This technical report analyzes an egocentric video action detection method we used in the 2021 EPIC-KITCHENS-100 competition hosted in CVPR2021 Workshop. The goal of our task is to locate the start time and the end time of the action in the long untrimmed video, and predict action category. We adopt sliding window strategy to generate proposals, which can better adapt to short-duration actions. In addition, we show that classification and proposals are conflict in the same network. The separation of the two tasks boost the detection performance with high efficiency. By simply employing these strategy, we achieved 16.10% performance on the test set of EPIC-KITCHENS-100 Action Detection challenge using a single model, surpassing the baseline method by 11.7% in terms of average mAP. Finally, we achieve ***Rank 1st*** *in this challenge.*

1. Introduction

Temporal action detection is a challenging task, especially for the EPIC-KITCHENS-100 dataset [9], where (a) most actions spans a short period, compared to the duration of the original untrimmed videos and (b) consistently altering action categories under the same background environment requires the network to have the ability to look for fine-grained features and discriminate complicated spatio-temporal interactions. To alleviate these issues, we propose the following strategies: (a) We use sliding windows to restrict the length of the input untrimmed videos for each video clip that is to be evaluated. This ensures that enough features are assigned to the short action segment candidates, which will be otherwise overwhelmed by the features from other segments in a long video that is simply normalized. The possibility is also increased that the length of the potential action segments can be matched to the pre-defined temporal anchors. (b) For more accurate verb and noun classifications, pre-trained backbones are employed for the classification of each video clip in the long videos. Additionally, we noticed an optimization conflict for proposal evaluation and classification, where the performances of both tasks drop drastically when a joint head is used to perform both tasks. Hence, we propose to use separate heads for evaluating the proposals as well as performing the classifications.

2. Our Approach

The overall architecture of our approach is visualized in Figure 1. The general process can be divided into four steps, respectively, **(i)** the pre-training of the classification models, **(ii)** feature extraction process, **(iii)** proposal generation process as well as the **(iv)** detection result generation process. We will discuss all the four stages one by one in the following sections.

2.1. Pre-train of Classification Models

Transfer learning is an important measure to improve the generalization ability of the model. Supervised training [24, 26, 8, 29, 25, 11] as well as unsupervised ones [15, 13, 21] are two mainstream pre-training strategy. Although the latter strategy can leverage a larger set of data, leading to a more generalized representation, supervised pre-training

† Equal Contribution.

∗ Corresponding authors.

This work is done when Z. Qing, Z. Huang, X. Wang and Y. Feng are interns at Alibaba Group.

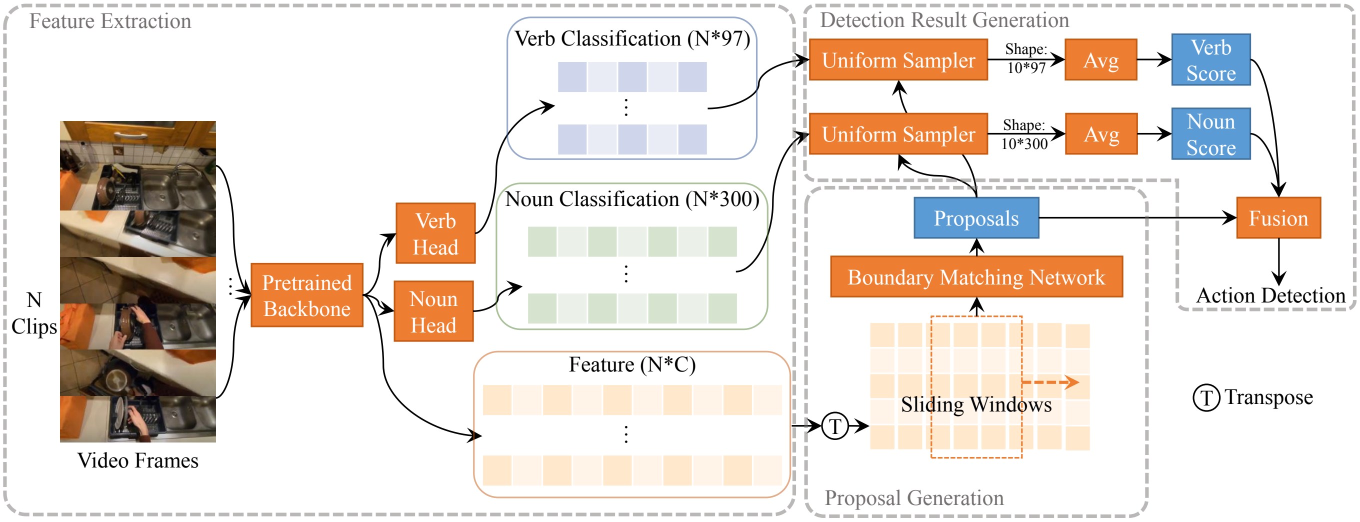

Figure 1: **The overall framework of our approach.** In feature extraction process, the input videos are divided into N clips, which are fed to the pre-trained backbone to extract features as well as verb and noun predictions. In the proposal generation stage, the sliding windows are uniformly distributed along temporal dimension, and clip-level features covered by each sliding window are fed to the Boundary Matching Network to generate proposals. In detection generation stage, the classification results for each proposal are sampled from classification scores yielded by the pre-trained backbone. Finally, the verb and noun predictions are fused with proposals to generate detection result with action predictions.

utilizes training data more efficiently and effectively. Therefore, we adopt the former strategy. Recently, Transformer-Based methods have shown great potential in image recognition [10, 33] and video understanding [1, 3]. We employ ViViT [1] and CSN [25] as our backbone for comparison and first pre-train it on Kinetics700 [7] dataset, mostly following the training recipes in DeepViT [33]. And then the pre-trained backbone is fine-tuned on EPIC-KITCHENS-100 dataset with verb head and noun head. In the fine-tuning stage, the FPS of the videos are normalized to 60, we sample 32 frames with sampling rate of 2 for each clip. The training details of the backbone models can be referred to our Action Recognition report [14].

2.2. Feature Extractor

Limited by the GPU memory, the raw frames cannot be directly fed to the backbone. Because of the limited GPU memory, it is impossible to put the whole video to the computing device. Therefore, multiple high-dimensional feature vectors are extracted from the untrimmed video using the pre-trained backbones, which will be further used in the later stage to generate action proposals. For the feature extraction process, we mostly follow the common setting in the temporal action detection community [18, 19, 17, 30, 12, 5, 2, 31, 22, 23, 27, 16, 28]. Specifically, given the number of frames l in a video, we split the video according to a fixed stride δ between consecutive video clips. Hence, the whole video is split into N clips, where $N = l/\delta$. In our experiments, the value of δ is set to 16. It is worth noting that, besides the feature vectors, the verb and noun predictions are saved at the same time for each clip when extracting its features.

2.3. Generation of Proposals

In our observations, different from mainstream action detection datasets [6, 32], 98.15% of the duration of the ground truth action segments are less than 20s. However, the average duration of the videos is up to 512.43s, which results in an extremely low percentage that the action segment candidate accounts for in the entire video. Therefore, we propose to generate action proposals within sliding windows. For each sliding window, we include features for 200 video clips. Because the interval between two consecutive video clips are 16 frames, which can be converted to 0.2667s in a 60-fps video, each sliding window contains contextual information lasting around 53.33s. The time interval between each sliding window is half of the size of one sliding window, which is 26.67 seconds. To ensure that at least one sliding window will cover the whole action segment candidate, we limit the maximum length of the potential action segment to be 26.67 seconds.

With sliding windows, Boundary Matching Network(BMN) [18] is employed to generate accurate proposals. Given the clip-level features $\mathbf{x} \in \mathbb{R}^{N \times C}$, BMN ex-

Feature Backbone	Classification	mAP(Val) for Action						mAP(Test) for Action					
		@0.1	@0.2	@0.3	@0.4	@0.5	Avg	@0.1	@0.2	@0.3	@0.4	@0.5	Avg
ViViT [1]	BMN [18]	6.84	6.01	5.28	4.42	3.26	5.16	5.86	5.33	4.67	4.03	3.03	4.59
CSN [25]	BMN [18]	7.30	7.01	6.56	6.05	5.08	6.40	-	-	-	-	-	-
ViViT [1]	CSN [25]	13.90	13.23	11.98	10.48	8.80	11.68	13.08	11.97	10.84	9.56	8.00	10.69
ViViT [1]	ViViT [1]	**21.14**	**20.10**	**19.02**	**17.32**	**15.11**	**18.53**	**18.76**	**17.73**	**16.26**	**14.91**	**12.87**	**16.11**

(a) **Action detection results for Action.**

Feature Backbone	Classification	mAP(Val) for Verb						mAP(Test) for Verb					
		@0.1	@0.2	@0.3	@0.4	@0.5	Avg	@0.1	@0.2	@0.3	@0.4	@0.5	Avg
ViViT [1]	BMN [18]	11.32	10.07	8.64	6.73	5.00	8.35	10.07	9.41	8.29	6.63	4.80	7.84
CSN [25]	BMN [18]	12.89	12.39	11.55	10.42	8.09	11.06	-	-	-	-	-	-
ViViT [1]	CSN [25]	16.57	15.56	14.10	12.12	10.21	13.71	17.58	15.91	14.21	12.23	9.73	13.93
ViViT [1]	ViViT [1]	**22.92**	**21.86**	**20.89**	**18.33**	**15.66**	**19.93**	**22.77**	**22.01**	**19.63**	**17.81**	**14.65**	**19.37**

(b) **Action detection results for Verb.**

Feature Backbone	Classification	mAP(Val) for Noun						mAP(Test) for Noun					
		@0.1	@0.2	@0.3	@0.4	@0.5	Avg	@0.1	@0.2	@0.3	@0.4	@0.5	Avg
ViViT [1]	BMN [18]	9.70	8.35	7.21	5.77	4.08	7.02	9.76	8.67	7.43	6.02	4.19	7.22
CSN [25]	BMN [18]	11.00	10.34	9.46	8.29	6.71	9.16	-	-	-	-	-	-
ViViT [1]	CSN [25]	18.47	17.21	15.56	13.38	10.58	15.04	19.46	17.79	15.87	13.62	10.90	15.53
ViViT [1]	ViViT [1]	**30.09**	**27.59**	**25.81**	**22.80**	**19.26**	**25.11**	**26.44**	**24.55**	**22.30**	**19.82**	**16.25**	**21.87**

(c) **Action detection results for Noun.**

Table 1: **Action detection results on EPIC-KITCHENS-100 dataset.** Features are extracted by the backbone in the feature backbone column. BMN in the classification column indicates that two classification heads are added upon the BMN feature to perform verb and noun predictions, while CSN and ViViT in the classification column indicates that we directly sample the prediction results temporally to obtain the final classification. The predictions results of CSN and ViViT are saved during the feature extraction process.

tracts candicate-level features for each potential action segments by matrix multiplication. The resultant features are then scored according to their IoU with the ground truth. We discard the scores from the Temporal Evaluation Module(TEM) and only retain the scores from the regression map and the classification map in Proposal Evaluation Module(PEM), as the inclusion of TEM actually hurt the performance. For more details obout PEM and TEM, please refer to BMN [18]. Since there is redundancy in the proposals generated by BMN, Soft-NMS [4] is applied to remove the redundant proposals. The hyperparameters in Soft-NMS are set to 0.25, 0.9 and 0.4 for low threshold, high threshold and alpha, respectively. In training, we utilize AdamW [20] as optimizer and set learning rate to 0.002. The model is trained for 10 epochs with cosine learning rate schedule.

2.4. Generation of Detection Results

To detect actions, it would be convenient if the BMN [18] can directly output both scores for the candidate proposals as well as the predictions of the verb and noun category for the corresponding proposals. However, we observe that when we use the candidate-level features extracted by the BMN network to perform both proposal evaluation task and classification, the performance is terrible. We suspect that there is some optimization conflict in the classification and the proposal evaluation tasks.

Since the accuracy of the feature extraction backbone ViViT (pre-trained on Kinetics700 and fine-tuned for EPIC-KITCHENS-100 action recognition task) can achieve 47.4% with 3×10 views in the validation set, we directly use its classification predictions. Specifically, we save all the predictions during the feature extraction process as we have mentioned before. Empirically, we show in Table 1 that, when we use ViViT [1] or CSN [25] features as the clip-level features and the candidate-level features of BMN to perform classification, the performance is worse than simply using the classification results generated directly by ViViT or CSN. Furthermore, when ViViT classification is used, a 6.85% performance improvement is observed over the CSN classification results. This is mainly due to the higher accuracy of ViViT in action recognition task. In our experiments, we do not use any ensemble strategy, which provides a simple and strong baseline for EPIC-KITCHENS-100 dataset.

To generate detection results, for each proposal generated by BMN, we sample the classification results in time range covered by the proposal with 10 uniform temporal location. The sampled classification results are averaged to get the prediction of respectively verb and noun for the proposal. Finally, the action detection results are obtained by fusing verb, noun scores and proposals.

3. Conclusion

In this report, we propose a stronger baseline for Ego-Centric Action Detection. We adopt a sliding window strategy to alleviate the problem that the temporal duration of proposals is too short to be detected difficultly. In addition, we also found that the conflict when the classification task and the proposal task coexist in the same network. Separating the two has significantly improved the performance. These two problems are inevitable in EPIC-KITCHENS-100 temporal action detection, how to solve these problems elegantly is still worthy of further study.

4. Acknowledgment

This work is supported by the National Natural Science Foundation of China under grant 61871435 and the Fundamental Research Funds for the Central Universities no. 2019kfyXKJC024, the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project A18A2b0046) and by Alibaba Group through Alibaba Research Intern Program.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 2, 3

[2] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. *arXiv preprint arXiv:2008.01432*, 2020. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 3

[5] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. 2

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 2

[7] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 2

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6299–6308, 2017. 1

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[11] 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. 1

[12] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017. 2

[13] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020. 1

[14] Ziyuan Huang, Zhiwu Qing, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Zhurong Xia, Mingqian Tang, Nong Sang, and Marcelo Ang. Towards training stronger video vision transformers for epic-kitchens-100 action recognition. *arXiv preprint arXiv:2106.05058*, 2021. 2

[15] Ziyuan Huang, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Rong Jin, and Marcelo Ang. Self-supervised motion learning from static images. *arXiv preprint arXiv:2104.00240*, 2021. 1

[16] Jianwen Jiang, Yu Cao, Lin Song, SZY Li, Z Xu, C Gan Q Wu, C Zhang, and G Yu. Human centric spatio-temporal action localization. *ActivityNet Workshop on CVPR*, 2018. 2

[17] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI*, pages 11499–11506, 2020. 2

[18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019. 2, 3

[19] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[21] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end

learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 1

[22] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. *arXiv preprint arXiv:2103.13141*, 2021. 2

[23] Zhiwu Qing, Xiang Wang, Yongpeng Sang, Changxin Gao, Shiwei Zhang, and Nong Sang. Temporal fusion network for temporal action localization: Submission to activitynet challenge 2020 (task e). *arXiv preprint arXiv:2006.07520*, 2020. 2

[24] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Int. Conf. Comput. Vis.*, pages 5533–5541, 2017. 1

[25] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 1, 2, 3

[26] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6450–6459, 2018. 1

[27] Xiang Wang, Baiteng Ma, Zhiwu Qing, Yongpeng Sang, Changxin Gao, Shiwei Zhang, and Nong Sang. Cbr-net: Cascade boundary refinement network for action detection: Submission to activitynet challenge 2020 (task 1). *arXiv preprint arXiv:2006.07526*, 2020. 2

[28] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. *arXiv preprint arXiv:2104.03214*, 2021. 2

[29] S Xie, C Sun, J Huang, Z Tu, and K Murphy. Rethinking spatiotemporal feature learning for video understanding (2017). arxiv preprint. *arXiv preprint arXiv:1712.04851*, 2017. 1

[30] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 2

[31] Shiwei Zhang, Lin Song, Changxin Gao, and Nong Sang. Glnet: Global local network for weakly supervised action localization. *IEEE Transactions on Multimedia*, 22(10):2610–2622, 2019. 2

[32] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8668–8678, 2019. 2

[33] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 2

LocTransformer: Submission to EPIC Kitchens 100 Action Detection Challenge

Chen-Lin Zhang[1], Jianxin Wu[1], and Yin Li[2,3]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison
[3]Department of Computer Sciences, University of Wisconsin-Madison

{zclnjucs, wujx2001}@gmail.com, yin.li@wisc.edu

Abstract

This report describes our submission to EPIC Kitchens 100 action detection challenge 2021. The key to our submission is a novel action localization model. Specifically, our model, built on a Transformer network, considers an action as a moment around the center of its timeline, and further estimates its temporal boundaries. Our final model, equipped with SlowFast features provided by [7], achieves 7.2 mAP on the validation set and 7.1 mAP on the test set, outperforming previous methods.

1. Introduction

Identifying action instances in time and recognizing their categories, known as temporal action localization, is a critical task for video understanding. Inspired by the success of object detection, almost all previous approaches represent an action instance as a temporal *segment*. These segments are modeled either by using sampled sliding windows *i.e.*, anchors [4, 21, 14, 6, 26, 17, 25], or by detecting their temporal boundaries, *e.g.*, the onsets and offsets [15, 13, 27, 2]. Unfortunately, actions in videos can have drastically different lengths that are hard to capture using a fixed set of anchors. Moreover, there is considerable ambiguity in the exact onsets and offsets of actions [19], making them difficult to localize in videos.

We consider representing an action instance as a *moment* around the center of its timeline, from which its onset and offset can be further estimated. This alternative design stems from recent works on point-based object detection [28, 10, 23]. Without using pre-defined anchors, our representation is more flexible to cover actions with disparate lengths. Further, detecting action moments around the center is arguably less ambiguous than finding the onsets and offsets [22, 1]. On the other hand, the detection of an action moment and the subsequent estimation of temporal boundaries, require the information about the full action instance within its temporal context. Therefore, a major challenge of our representation is the modeling of long-range temporal information in videos.

To address this challenge, we explore a Transformer-based model [24] for temporal action localization, replacing the widely used convolutional and recurrent networks. A Transformer network adopts self-attention to aggregate contextual information from a full input sequence, thereby offering an ideal approach to model long-range temporal dependencies in videos. Transformer-based models have been primarily used for natural language processing [8, 24], recently explored in vision tasks [5, 9] including action understanding [12], yet not considered for action localization.

Our model, dubbed "LocTransformer", offers a *single-stage anchor-free* model for temporal action localization. We report results on the action detection task of EPIC Kitchens 100 dataset using pretrained SlowFast features from [7]. Our method achieves 7.2 mAP on the validation set and 7.1 mAP on the test set, surpassing the strong baseline of BMN [13] by a significant margin.

2. Localizing Moments of Actions with Transformers

Given an input video $\mathbf{X}$, we assume that $\mathbf{X}$ can be represented using a set of vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ defined on discretized time steps $t = \{1, 2, \ldots, T\}$, where T varies across videos. For example, $\mathbf{x}_t$ can be the feature vector of a clip at time step t extracted from a 3D convolutional network. The goal of temporal action localization is to predict the action label $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ based on the input video sequence $\mathbf{X}$. $\mathbf{Y}$ consists of N action instances $\mathbf{y}_i$, where N also varies across videos. Each instance $\mathbf{y}_i = (s_i, e_i, a_i)$ is defined by its onset s_i, offset e_i and action label a_i, where $s_i \in [1, T]$, $e_i \in [1, T]$,

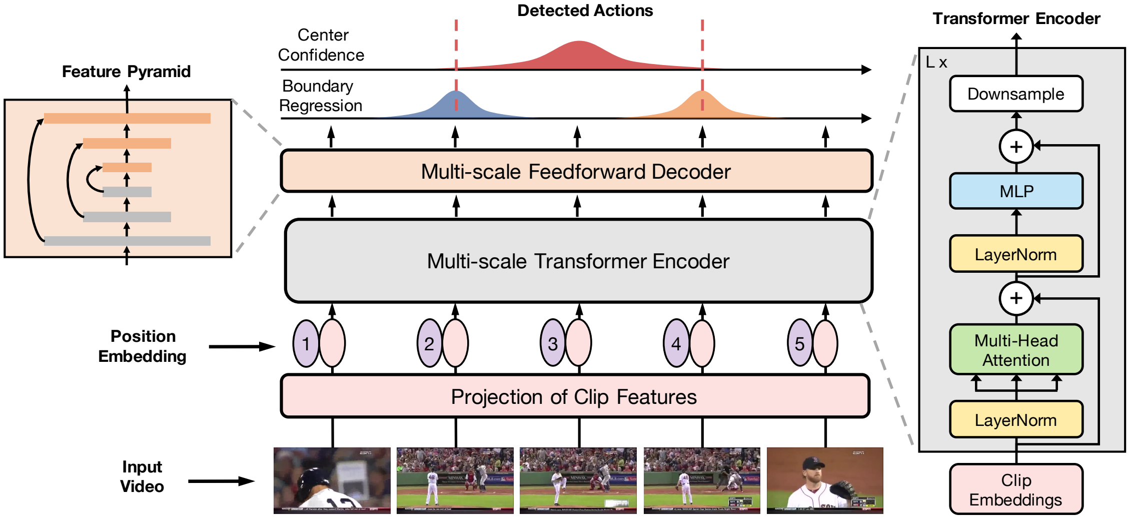

Figure 1. Overview of our method. Our method detects an action instance as a moment at the center of its timeline, and further estimates the distances from the center to the action's onset and offset (top middle). Specifically, our model first extracts a sequence of video clip features, embeds each of these features, and adds position embedding. The embedded features are further encoded into a feature pyramid using a multi-scale transformer (right). The feature pyramid is further aggregated (top left) and examined by shared classification and regression networks, producing an action candidate at every time step. Our method provides a single-stage anchor-free model for temporal action localization with strong performance across several datasets.

$a_i \in \{1, 2, \ldots, C\}$ (C pre-defined categories) and $s_i < e_i$. The task of temporal action localization is thus a challenging structured output prediction problem.

Actions as Moments. Our key idea is to represent an action as a moment at the center of its timeline, plus the distances between this center point and the action's onset and offset. This is equivalent to reparameterize an action $\mathbf{y}_i = (s_i, e_i, a_i)$ using its center point $c_i = (e_i + s_i)/2$, such that $\hat{\mathbf{y}}_i = (c_i, d_i^s = c_i - s_i, d_i^e = e_i - c_i, a_i)$. Based on this parameterization, we convert the structured output prediction problem ($\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\} \rightarrow \mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$) into a more approachable sequence labeling problem

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\} \rightarrow \hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_T\}. \quad (1)$$

The output $\hat{\mathbf{y}}_t$ at time step t is defined as $\hat{\mathbf{y}}_t = (p(c_t), d_t^s, d_t^e, p(a_t))$.

- $p(c_t)$ is the probability of a binomial variable c_t (with slight abuse of notation). $p(c_t) \rightarrow 1$ means that current time step t is (close to) a center point of an action.
- $d_t^s > 0$ and $d_t^e > 0$ are the distance between the current center point to the starting and ending points, respectively. d_t^s and d_t^e are undefined if t is not a center point.
- $p(a_t)$ is the probability of a multinomial variable indicating the current action category a_t at time t with $a_t \in \{1, 2, \ldots, C\}$.

This formulation, inspired by recent developments in object detection [23, 10, 28], considers *every time step* t in the video $\mathbf{X}$ as an action candidate centered around t, recognizes the action's category a_t, and estimates the distances to the action's onset and offset (d_t^s and d_t^e). Finally, action localization results can be decoded from $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_T\}$ by selecting the top action candidates.

Overview of Our Approach. The core of our method lies in the learning of $f(\mathbf{X}) \rightarrow \hat{\mathbf{Y}}$ for sequence labeling. Specifically, f is realized using a deep model. Our model follows an encoder-decoder architecture proven successful in many vision tasks, and decomposes f as $h \circ g$. Here $g \colon \mathbf{X} \rightarrow \mathbf{Z}$ encodes the input into a latent vector $\mathbf{Z}$, and $h \colon \mathbf{Z} \rightarrow \hat{\mathbf{Y}}$ subsequently decodes $\mathbf{Z}$ into the sequence label $\hat{\mathbf{Y}}$.

Fig. 1 presents an overview of our model. Importantly, our encoder g is parameterized by a Transformer network [24], originally designed for sequence modeling in NLP. Our decoder h adopts a lightweight feedforward network. To capture actions at disparate temporal scales, we design a multi-scale feature representation $\mathbf{Z} = \{\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^L\}$ forming a feature pyramid with varying resolutions. Details of our model was described in a separate paper submission that is currently under review.

3. Implementation

This section describes implementation details of our model tailored for the EPIC Kitchens dataset.

Feature Extraction. We used a SlowFast network [11] from [7] pre-trained on EPIC Kitchens 100 action recognition challenge for feature extraction. Only RGB frames were considered. We fed 32 frames into the model with the stride of the sliding window set to 16, leading to a 2304-d

Split	Method	Task	mAP@tIoU					
			0.1	0.2	0.3	0.4	0.5	mean
Val	BMN [13] + SlowFast [11]	Verb	10.8	9.8	8.4	7.1	5.6	8.4
		Noun	10.3	8.3	6.2	4.5	3.4	6.5
		Action	7.0	6.1	5.2	4.4	3.4	5.2
	Ours (Single) +SlowFast [11]	Verb	15.9	14.9	13.7	12.2	10.0	13.4
		Noun	14.7	13.6	12.5	10.9	8.8	12.1
		Action	8.2	7.8	7.2	6.3	5.5	7.0
	Ours (Ensembled) + SlowFast [11]	Verb	16.1	15.2	14.2	12.7	10.3	13.7
		Noun	15.1	14.1	12.9	10.9	8.7	12.3
		Action	8.5	8.1	7.5	6.5	5.5	7.2
Test	BMN [13] + SlowFast [11]	Verb	11.1	9.4	7.4	5.7	4.1	7.5
		Noun	12.0	8.5	6.0	4.1	2.8	6.7
		Action	6.4	5.4	4.4	3.4	2.5	4.4
	Ours (Single) + SlowFast [11]	Verb	16.4	15.5	14.0	12.3	10.0	13.6
		Noun	15.4	13.9	12.7	10.7	8.0	12.1
		Action	8.1	7.5	6.9	5.9	5.0	6.7
	Ours (Ensembled) + SlowFast [11]	Verb	18.3	17.4	16.1	12.5	10.4	**14.9**
		Noun	15.3	14.3	12.8	10.9	8.4	**12.6**
		Action	8.8	8.0	7.4	6.3	5.1	**7.1**

Table 1. Results on EPIC Kitchens 100 action detection challenge 2021. All results on the test set were evaluated on the test server. Single: our single model that uses multi-task learning for verbs and nouns; Ensemble: ensembling of two separate models for nouns and verbs, respectively. We also include BMN results as our baseline. Both our models significantly outperform the strong baseline results.

feature vector for each timestamp. A feature sequence with variable lengths for different videos was extracted for each video. Besides the pre-trained action recognition model, we did not use any other external data or any other external models.

Network Architecture. We used 2-layer MLP for projection, 5 Transformer units for the encoder, 1-layer MLP in the feature pyramid network, and 3 layers of 1D convolutions with kernel size=5 for classification / regression networks. Position embeddings were added to every level of the transformer encoder. The regression range on each pyramid level is limited and normalized by the stride of the features: The regression range for each layer is $[0, 4]$, $[4, 8]$, $[8, 16]$, $[16, 32]$, $[32, 64]$, $[64, 128]$ (feature steps). We used the center sampling technique in FCOS [23] with the center sampling ratio of 2.5.

Training and Inference. During training, we capped the input length to 2304 (around 25 minutes). A shorter video was zero-padded to 2304 time steps, while a longer video was randomly truncated without cutting any action boundaries. At inference time, we again zero-padded shorter sequences (less than 2304 steps), yet fed the full longer sequences (larger than 2304 steps) by interpolating the position embeddings, similar to [9]. We used the Focal loss [16] for classification task and GIoU loss [20] for regression task. Our model was trained for 15 epochs using AdamW [18] with learning rate 1e-4, and another 15 epochs with learning rate 1e-5. The mini-batch size was 8, and a weight decay of 1e-4 was used.

Single vs. Ensembled Model. For EPIC Kitchens dataset, we used two separate heads for the classification of nouns and verbs, respectively. A single regression head was shared to predict action onsets and offsets. This model serves as our starting point. Moreover, to further boost the results, we found it helpful to build an ensembled model. More concretely, we also trained two separate models to localize verbs and nouns individually. Their results were further merged for the final recognition. We took the union of the detected actions for the two models, computed the verb and noun scores from individual models, and multiplied these scores for final action scores. This ensembled model provides slightly better results on the test set.

Post Processing. After training, we obtained the predictions for every time step across all pyramid levels. These results were further multiplied with classification scores from the same SlowFast network used for feature extraction. SoftNMS [3] with a threshold of 0.7 was used to keep top-1000 action predictions (2000 on test set with ensembled models).

4. Action Localization Results

This section presents the results of our model on EPIC Kitchens dataset. We describe the dataset and the evaluation protocol and metric, followed by a summary of our results.

Dataset. Our results are reported on EPIC Kitchens 100 action detection dataset [7]. EPIC Kitchens 100 is the largest egocentric action dataset with more than 100 hours of videos from 700 sessions capturing cooking activities across several kitchen environments. The dataset has an average 128 actions from a large array of categories per session. Each action is defined as a combination of a verb (action) and a noun (object).

Evaluation Protocol and Metrics. We followed the official splits of train, validation and test set. When reporting results on validation set, we trained our model on the training set. For the results on test set, we combined both training and validation sets for training. Our results are reported for noun, verb and action, respectively. The metrics include the mean average precision (mAP) at different tIoU thresholds [0.1:0.1:0.5], as well as the average mAP , following [7]. As this dataset was recently released, we only compared our methods to BMN [13] from [7], which uses the same SlowFast network for classification.

Results. Table 1 summarizes our results on on the validation and test set. On validation set, our method reaches an average mAP of 13.4%, 12.1% and 7.0% for verb, noun and action, respectively, largely outperforms the strong baseline of BMN [13, 7] by 5.0%, 5.6% and 1.8%. Importantly, our results are significantly better at all tIoU. For example when tIoU=0.5, our method achieves 10.0% mAP on verb detection tasks, beating BMN by 4.4%. On test set, Our ensembled model has a final mAP of 14.9/12.6/7.1 for noun/verb/action, respectively, slightly beating our single model. Our final model uses the same feature as the baseline BMN, yet outperforms BMN by a very large margin of +7.4/+5.9/+2.7 in mAP (+99%/+88%/+61%) for noun/verb/action, respectively.

Our method also demonstrates strong results across several public action localization benchmarks, including THUMOS14 and ActivityNet-1.3. Those results were described in our other paper submission. We hypothesis that our results on EPIC Kitchens can be further improved by incorporating stronger backbones for video features, as well as object detection results for object information.

5. Conclusion

In this report, we presented a new representation that considers actions as moments at the center of the timeline, and a novel model using Transformer network for temporal action localization. Our method has demonstrated strong performance on the EPIC Kitchens dataset. We hope that our action representation and our model can shed light on the task of temporal action localization, and the more broader problem of video understanding.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *Eur. Conf. Comput. Vis.*, volume 11207 of *LNCS*, pages 256–272, 2018. 1

[2] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Eur. Conf. Comput. Vis.*, volume 12373 of *LNCS*, pages 121–137, 2020. 1

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *Int. Conf. Comput. Vis.*, pages 5561–5569, 2017. 3

[4] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-stream temporal action proposals. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2911–2920, 2017. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Eur. Conf. Comput. Vis.*, volume 12346 of *LNCS*, pages 213–229, 2020. 1

[6] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the Faster-RCNN architecture for temporal action localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1130–1139, 2018. 1

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1, 2, 4

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Asso. Comput. Lin.*, pages 4171–4186, 2019. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 1, 3

[10] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. CenterNet: Keypoint triplets for object detection. In *Int. Conf. Comput. Vis.*, pages 6569–6578, 2019. 1, 2

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Int. Conf. Comput. Vis.*, pages 6202–6211, 2019. 2, 3

[12] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 244–253, 2019. 1

[13] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *Int. Conf. Comput. Vis.*, pages 3889–3898, 2019. 1, 3, 4

[14] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM Int. Conf. Multimedia*, pages 988–996, 2017. 1

[15] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *Eur. Conf. Comput. Vis.*, volume 11208 of *LNCS*, pages 3–19, 2018. 1

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, pages 2980–2988, 2017. 3

[17] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 344–353, 2019. 1

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019. 3

[19] Davide Moltisanti, Michael Wray, Walterio Mayol-Cuevas, and Dima Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2886–2894, 2017. 1

[20] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 658–666, 2019. 3

[21] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5734–5743, 2017. 1

[22] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *Int. Conf. Comput. Vis.*, pages 2137–2146, 2017. 1

[23] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Int. Conf. Comput. Vis.*, pages 9627–9636, 2019. 1, 2, 3

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 1, 2

[25] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-TAD: Sub-graph localization for temporal action detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10156–10165, 2020. 1

[26] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Int. Conf. Comput. Vis.*, pages 7094–7103, 2019. 1

[27] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *Eur. Conf. Comput. Vis.*, volume 12353 of *LNCS*, pages 539–555, 2020. 1

[28] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 850–859, 2019. 1, 2

Team VI-I2R Technical Report on EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition

Yi Cheng, Fen Fang, Ying Sun
Institute for Infocomm Research, A*STAR, Singapore
{cheng_yi, fang_fen, suny}@i2r.a-star.edu.sg

Abstract

In this report, we present the technical details of our approach to the EPIC-KITCHENS-100 Unsupervised Domain Adaptation (UDA) Challenge for Action Recognition. The EPIC-KITCHENS-100 dataset consists of daily kitchen activities focusing on the interaction between human hands and their surrounding objects. It is very challenging to accurately recognize these fine-grained activities, due to the presence of distracting objects and visually similar action classes, especially in the unlabelled target domain. Based on an existing method for video domain adaptation, i.e.*, TA3N, we propose to learn hand-centric features by leveraging the hand bounding box information for UDA on fine-grained action recognition. This helps reduce the distraction from background as well as facilitate the learning of domain-invariant features. To achieve high quality hand localization, we adopt an uncertainty-aware domain adaptation network,* i.e.*, MEAA, to train a domain-adaptive hand detector, which only uses very limited hand bounding box annotations in the source domain but can generalize well to the unlabelled target domain. Our submission achieved the 1st place in terms of top-1 action recognition accuracy, using only RGB and optical flow modalities as input.*

1. Introduction

The EPIC-KITCHENS-100 dataset is a large-scale video benchmark, capturing daily cooking activities from egocentric perspective [2]. It mainly contains fine-grained actions which reflect the interaction between human hands and their surrounding objects, and each action class is defined by a verb and a noun class. The EPIC-KITCHENS-100 Unsupervised Domain Adaptation (UDA) Challenge for Action Recognition aims to adapt an action recognition model trained on a labelled source domain to an unlabelled target domain. In this challenge, the source domain contains egocentric videos captured in 2018, while the target domain contains egocentric videos captured in 2020 with the same subjects but different cameras and potential change of kitchens. This is a challenging task, as the source and target domains have different data distributions, due to the changes of environments and camera settings. Solutions successfully addressing this challenge can help save much time and efforts when applying the model trained on an existing labelled dataset to a newly collected dataset without annotation.

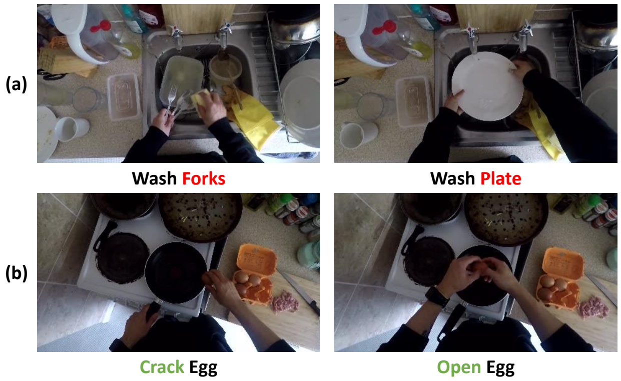

Figure 1. Illustration of the challenges in fine-grained action recognition on EPIC-KITCHENS-100 dataset. (a) Presence of distracting objects: there are many distracting objects in the scene, *e.g.*, sponge and tap, which makes it difficult to identify the active objects. (b) Visually similar actions, *e.g.*, "crack egg" versus "open egg", which requires to capture the subtle difference between hand motions. To handle both of these two challenges, it is important to enhance the features around hand regions.

In Fig 1, we present some samples from the EPIC-KITCHENS-100 dataset. As shown in the figure, activities in this dataset mainly focus on the interactions between human hands and their surrounding objects. This brings two challenges for fine-grained action recognition. The first challenge is the presence of distracting objects in the scene, which makes it difficult to identify the active objects. Based on the observation that active objects are generally located around the hand regions, we find that it is important to enhance the features around the hand regions [3]. The other

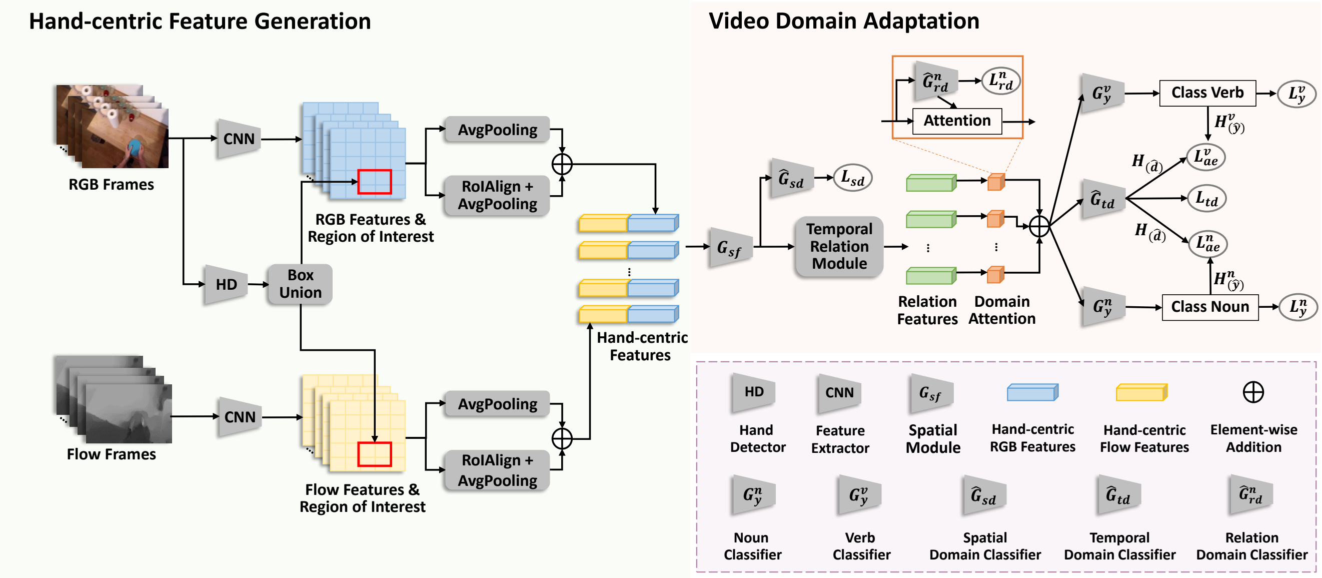

Figure 2. Overall architecture of the proposed framework. L_{sd}, L_{td} and L^n_{rd} denote the spatial, temporal and relation domain classification loss, respectively. L^n_y and L^v_y denote the noun and verb classification loss, respectively. $H(\hat{d})$, $H(\hat{y}^n)$ and $H(\hat{y}^v)$ denote the entropy of predictions from temporal domain classifier, noun classifier and verb classifier, respectively. L^n_{ae} and L^v_{ae} denote the attentive entropy loss for noun and verb, respectively. This is best viewed in color.

challenge is that some action classes are very similar visually, where capturing the subtle difference among hand motions are essential to accurately recognize the target actions. Therefore, enhancing the features around hand regions can help to reduce the distraction from background and thus improving the action recognition accuracy. Moreover, this may provide further benefits in the context of UDA for fine-grained action recognition by facilitating the learning of domain-invariant features. To the best of our knowledge, MM-SADA [6] is the first attempt on UDA for fine-grained action recognition. It leverages the multi-modal nature of video data to adapt fine-grained action recognition models to unlabelled target domain, which provides the first benchmark on this task. However, it does not consider the aforementioned challenges.

To address these two challenges, we propose to learn hand-centric features by leveraging the hand bounding box information for UDA on fine-grained action recognition. Specifically, we adapt the TA3N [1], an existing method for video domain adaptation, to learn a fine-grained action recognition model that can be adapted to the unlabeled target domain. To achieve high quality hand localization, we apply an uncertainty-aware domain adaptation network, *i.e.*, MEAA, to train a hand detector, which only uses very limited hand bounding box annotations in the source domain but can generalize well to the unlabelled target domain. The experimental results on the EPIC-KITCHENS-100 dataset demonstrate the effectiveness of our approach for UDA on fine-grained action recognition.

2. Our Approach

In this section, we present the technical details of our proposed approach. As illustrated in Fig. 2, the overall architecture has two stages: hand-centric feature generation and video domain adaptation.

2.1. Hand-centric Feature Generation

The hand-centric feature generation stage consists of three key components: feature extractors, domain-adaptive hand detector, and hand-centric feature generator. Next, we will describe each component in details.

Feature extractors. To learn discriminative feature representations, we investigate two pre-trained action recognition models for feature extraction, *i.e*. TBN [4] and TSM [5]. The extracted features are used to generate hand-centric RGB and flow features which serve as the input to the video domain adaptation model. We empirically find that features extracted with TSM model can lead to better domain adaptation results, which is consistent with the action recognition results on EPIC-KITCHENS-100 dataset reported in [2]. This indicates that compared with TBN model, TSM model tends to learn more discriminative features for action recognition. Therefore, we employ the features extracted with TSM model in our final submission.

Domain-adaptive hand detector. To achieve high quality hand localization, it is necessary to train a hand detector in labelled source domain and adapt it to unlabelled target domain. This makes the hand detection in our task falls in the area of domain-adaptive object detection which aims

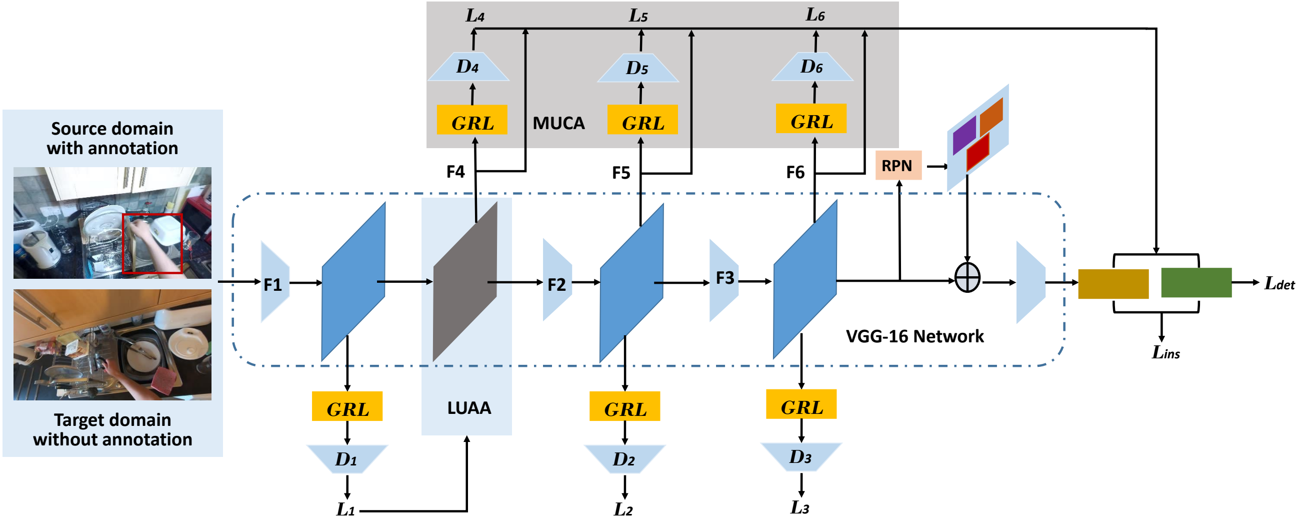

Figure 3. Overall architecture of the domain-adaptive hand detector (MEAA). It consists of two modules, namely LUAA and MUCA. D_i, F_i and L_i, where $i = 1, 2, ..., 6$, denote multi-level domain classifiers, feature extractors and losses, respectively. GRL denotes Gradient Reverse Layer. This is best viewed in color.

to transfer knowledge from labelled source domain to unlabelled target domain. To this end, we adapt the existing method MEAA [7] to train a domain-adaptive hand detector, and the detailed structure is presented in Fig. 3. It investigates (1) uncertainty measurement of input pairs (images in source and target domain) by utilizing domain classifiers at multiple levels of the backbone network as well as (2) uncertainty at image and instance levels to guide the model to pay more attention to hard-to-align instances and images. Specifically, it designs the Local Uncertainty Attentional Alignment (LUAA) module to align high-level features and low-level features by perceiving structure invariant regions of objects and a Multi-level Uncertainty-Aware Context Alignment (MUCA) module to enrich the model with uncertainty-weighted context vectors.

Hand-centric feature generator. After obtaining the extracted feature maps and the predicted hand bounding boxes, we generate the hand-centric features to train the domain-adaptive action recognition model. As the RGB and flow features are generated in the same manner, we only describe the steps to generate RGB features. As illustrated in Fig. 2, for each frame, we first extract the RGB features with TSM model pretrained in the source domain and obtain the region of interest (ROI) by applying union operation on all the hand bounding boxes in the frame. Then, the context features are generated by applying the average pooling operation on the extracted RGB feature, while the hand-related features are generated by applying the RoIAlign operation followed by the average pooling operation. By combining the context features and hand-related features with the element-wise addition, we obtain the final hand-centric features. Finally, the hand-centric RGB and flow features are concatenated to serve as the input to the domain-adaptive action recognition model.

2.2. Video Domain Adaptation

Following the baseline provided by the organizers, we adapt an existing method for video domain adaptation, i.e., TA3N [1], to train the domain-adapted action recognition model. As illustrated in Fig. 2, TA3N designs the temporal relation module to model the n-frame temporal relation by taking n temporal-ordered sampled frames as input and output n-frame relation features. These relation features are then aggregated to generate the video-level features. Similar to other approaches for video domain adaptation, TA3N applies the adversarial discriminator $\hat{G}_{sd}$ to align the spatial (frame-level) features and the adversarial discriminator $\hat{G}_{td}$ to align the video-level features from different domains. Differently, it designs a set of adversarial discriminators $\hat{G}^n_{rd}$ to align the n-frame relation features from different domains. In our solution, we modify the code of TA3N by designing two classifiers for the video-level features, with G^v_y for verb classification and G^n_y for noun classification.

3. Experiments

3.1. Datasets.

The EPIC-KITCHENS-100 dataset [2] contains a source domain and a target domain. The source domain contains labelled videos collected in 2018 and the target domain contains unlabelled videos collected in 2020. Videos from both domains are further split into train, valuation and test sets.

Table 1. The performance of different models on the EPIC-KITCHENS-100 validation set. "FeatDim" and "NumSeg" are hyper-parameters in TA3N, which denote the dimension of shared features and number of input frames, respectively. "Raw features" denote features extracted with backbone models, while "Hand-centric features" denote features generated by incorporating the hand bounding box information.

Method	Backbone	Input Type	FeatDim	NumSeg	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
					Verb	Noun	Action	Verb	Noun	Action
TA3N	TBN	Raw features	512	5	42.97	27.17	16.63	74.34	49.01	41.31
TA3N	TSM	Raw features	512	5	46.31	33.17	20.02	80.58	56.44	48.94
TA3N	TSM	Hand-centric features	512	5	48.62	35.14	21.73	80.50	57.94	50.25
TA3N	TSM	Hand-centric features	1024	20	52.37	37.00	24.48	81.13	59.18	51.75

Table 2. The performance of different models on the EPIC-KITCHENS-100 test set. "Ensemble" denotes whether model ensemble is used to generate the testing results. Other definitions are the same as in Table 1.

Method	Backbone	Input Type	FeatDim	NumSeg	Ensemble	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
						Verb	Noun	Action	Verb	Noun	Action
TA3N	TSM	Hand-centric features	1024	20	No	52.99	34.76	24.71	80.05	58.66	40.23
TA3N	TSM	Hand-centric features	1024	20	Yes	53.16	34.86	25.00	80.74	59.30	40.75

3.2. Implementation Details

Following the baseline method in [2], we train our model using a two-stage optimization scheme. Specifically, we first train the TSM [5] model and hand detection model on source domain data to generate the hand-centric features. Subsequently, we train the modified TA3N [1] for domain-adaptive action recognition.

Feature extractors. As the organizers provide the RGB and flow features extracted with TBN [4] model pretrained in the source domain, we only need to train TSM [5] in the source domain for feature extraction. The network parameters are learned with SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. We train the models for 60 epochs, where the learning rate is initialized at 0.01 and multiplied by 0.1 for every 20 epochs. The batch size is set at 16 and the input size is set at 256×256. During training, we first resize the shorter edge of each frame to 256 while keeping the aspect ratio, and then randomly crop the frame to 256×256 to feed it into the backbone model. During testing, we take the same resizing strategy but use center crop to generate the input of size 256×256. After applying the Average Pooling, the dimension of generated feature is 2048.

Domain-adapted hand detector. As shown in Fig. 3, the training inputs are image pairs (source image with annotation, target image without annotation). As no hand bounding box annotation is provided in the source domain, we randomly select a very limited number of frames, *i.e.*, 3100 images, and annotate the hand bounding boxes manually. Meanwhile, we select double size images with hand from target videos. In this case, one source image will appear in two image pairs, and totally we have 6200 image pairs for training the hand detector. During training, the parameters are learned with Adam optimizer. We train the model for 10 epochs, and the leaning rate is set as 0.001 with the decay step as 4.

Video domain adaptation. We follow the guidelines given by the organizers to train the domain-adaptive action recognition model. First, we train the model using the source validation and target validation splits to select the best hyper-parameters. Then, we retrain the model using the source train and target train splits, where the retrained model is evaluated on the target test split to generate the action predictions for this challenge. We adapt TA3N to train our model. During training, the parameters in the feature extractors and the hand detector are freezed. The parameters in domain-adaptive action recognition model are learned using SGD optimizer, where the initial learning rate are set as 3×10^{-3}. We train the model for 30 epochs and the learning rate is multiplied by 0.1 for every 10 epochs. In our submission, the number of input frames and the shared feature dimension are empirically set as 20 and 1024, respectively.

3.3. Results

UDA for action recognition. We employ two backbones (TBN and TSM) trained on the labelled source domain as feature extractors, and train TA3N with different features as inputs. The performance of different models on the validation set are summarized in Table 1. All the models use both RGB and optical flow as input. As shown in the table, by replacing the TBN with TSM as feature extractors, the top-1 action accuracy can be improved by 3.30%. By leveraging the hand bounding box information to generate hand-centric RGB and flow features can further improve the top-1 action accuracy by 1.71%, where the top-1 noun and verb branches achieve similar performance gains. This demonstrates the effectiveness of the hand-centric features in recognizing active objects as well as understanding the hand motions. More importantly, it also helps to capture the domain-invariant features for accurate action recognition in the target domain. By adapting the hyper-parameters, *i.e.*, shared feature dimension and number of input frames, in

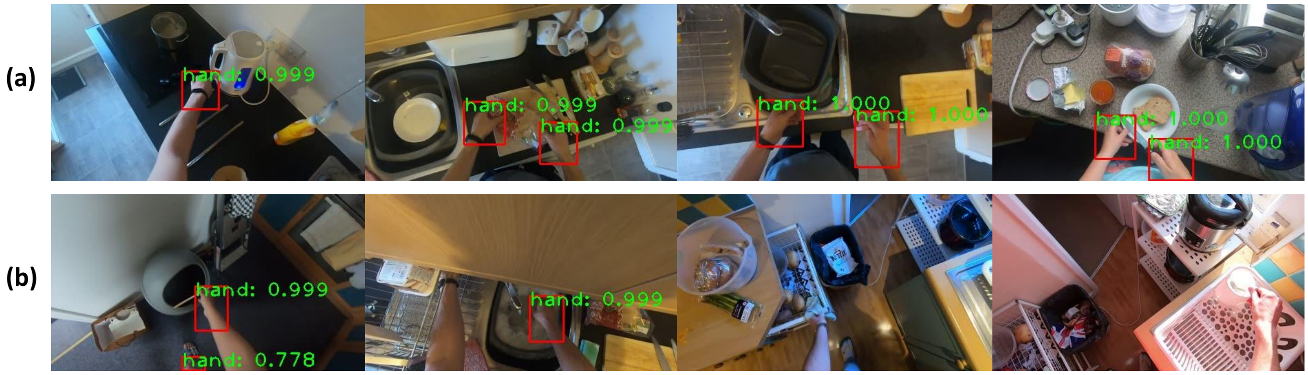

Figure 4. Visualization of hand detection results on unlabelled target domain. The red boxes denote predicted hand bounding boxes generated by the domain-adaptive hand detector. The green characters illustrate the predicted object class of each bounding box and its confidence score.

TA3N, our final model can achieve 24.48% in terms of top-1 action accuracy on the validation set. For better robustness, we adopt the same model ensemble strategy as in [8] to generate our final submission to the challenge based on the best model in Table 1. The results on the test set are presented in Table 2. The best model in Table 2 ranks first in terms of the top-1 action accuracy in the EPIC-KITCHENS-100 UDA Challenge for Action Recognition.

Visualization results of domain-adaptive hand detector. We present the predicted hand bounding boxes and their confidence scores on selected samples from the unlabeled target domain in Fig. 4. The results demonstrate that the domain-adaptive hand detector trained with very limited labelled samples in the source domain can generalize well to most of the cases in the unlabelled target domain. Fig. 4 (a) shows that hands can be correctly detected with high confidence scores under normal view angle and lighting condition. Fig. 4 (b) shows selected hard samples where there are some false or missed hand detection results. Specifically, false detection may happen when some objects are visually similar as human hands (the first image), while miss detection may happen under heavy occlusions (left hand in the second image), extreme view angle (the third image), or extreme lighting condition (the last image).

4. Conclusion

In this report, we describe the technical details of our approach to the EPIC-KITCHENS-100 UDA Challenge for Action Recognition. Specifically, we propose to learn hand-centric features by leveraging the hand bounding box information for UDA on fine-grained action recognition. To obtain high-quality hand localization, we apply MEAA to train a domain-adaptive hand detector with very limited hand bounding boxes annotations in the source domain. The experimental results on the EPIC-KITCHENS-100 dataset demonstrate the effectiveness of our proposed method. With further performance increase from the model ensemble, our final submission ranks first on the leaderboard in terms of top-1 action recognition accuracy.

Acknowledgments

We would like to thank Dr. Joo Hwee Lim for his support and guidance. This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

[1] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019. 2, 3, 4

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1, 2, 3, 4

[3] Hehe Fan, Tao Zhuo, Xin Yu, Yi Yang, and Mohan Kankanhalli. Understanding atomic hand-object interaction with human intention. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1

[4] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 4

[5] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2, 4

[6] 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*, pages 122–132, 2020. 2

[7] Dang-Khoa Nguyen, Wei-Lun Tseng, and Hong-Han Shuai. Domain-adaptive object detection via uncertainty-aware distribution alignment. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2499–2507, 2020. 3

[8] Y. Sun, Yi Cheng, M. Leong, Hui Li Tan, and Kenan E. Ak. Team VI-I2R technical report on epic-kitchens action anticipation challenge 2020. 2020. 5

EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition 2021: Team M^{3}EM Technical Report

Lijin Yang, Yifei Huang, Yusuke Sugano, Yoichi Sato
Institute of Industrial Science, the University of Tokyo
Tokyo, Japan
{yang-lj,hyf,sugano,ysato}@iis.u-tokyo.ac.jp

Abstract

In this report, we describe the technical details of our submission to the 2021 EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition. Leveraging multiple modalities has been proved to benefit the Unsupervised Domain Adaptation (UDA) task. In this work, we present Multi-Modal Mutual Enhancement Module (M^{3}EM), a deep module for jointly considering information from multiple modalities to find most transferable representations across domains. We achieve this by implementing two sub-modules for enhancing each modality using the context of other modalities. The first sub-module exchanges information across modalities through the semantic space, while the second sub-module finds the most transferable spatial region based on the consensus of all modalities.

1. Introduction

EPIC-KITCHENS-100 dataset contains fine-grained actions performed in different kitchens [3]. How to make a model learned on a subset of kitchens (the source domain) to perform well on other unseen kitchens (the target domain) is challenging, since not only the verbs but also the associated objects can be dissimilar across domains.

Previous works [7, 9] have shown that using multiple modalities can improve the performance of UDA on action recognition, but none of them considered using early fusion on the modalities to enhance the transferability of the generated features. Intuitively, with the guidance of RGB, Flow can give more focus on the correct object, whereas by using knowledge from the motion, the RGB modality would concentrate more on the moving parts. In this paper, we argue that leveraging the information exchange across modalities before the final decision can significantly improve the transferability of features. Based on this intuitive, we propose a novel Multi-Modal Mutual Enhancement Module (M^{3}EM) for domain adaptive action recognition by enhancing the features across modalities.

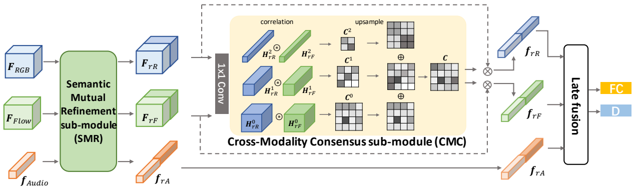

Figure 1. Overview of the proposed M^{3}EM. We showcase three modalities RGB, Flow and Audio as input but it can be easily extended to add other modalities such as depth or hand. In the figure, $\oplus$ denotes element-wise summation, $\otimes$ is element-wise multiplication, and $\circledast$ means the correlation operation that calculates the Pearson correlation coefficient on each spatial position. *FC* and *D* are short for *classifier* and *discriminator*, respectively.

The proposed M^{3}EM consist of two sub-modules: a Semantic Mutual Refinement sub-module (SMR) and a Cross Modality Consensus sub-module (CMC). To leverage the strength of each modality, the SMR enables information exchange across modalities through the semantic space. With SMR, a modality M can receive recommendations from other modalities about transferable components that are easily ignored by the modality itself. The CMC highlights the spatial regions that are transferable consistently in all the modalities. This sub-module complements SMR by preventing the SMR from emphasizing similar but irrelevant background regions that harm the action recognition. With the two proposed simple yet effective sub-modules, our module can be built on top of most existing domain adaptive action recognition models and improve their performance by integrating multi-modality signals.

2. Method

Figure 1 depicts the overview of the proposed M^{3}EM. For each modality of RGB, Flow and Audio, backbone (omitted in the figure) networks encode the input into frame-level features $\boldsymbol{F}_{RGB}, \boldsymbol{F}_{Flow}$ and $\boldsymbol{f}_{Audio}$, respectively. The features are then enhanced by information from other modalities using our proposed Semantic Mutual Refinement sub-module (SMR). For modality M, SMR summarizes information of M and receive information from other modalities. Two gating functions are proposed to enhance the feature transferability by re-evaluating and re-mapping the transferable channels based on the summarized and received information. We then use a Cross-Modality Consensus sub-module (CMC) to get the most transferable spatial region. CMC finds the transferable region by calculating bit-wise correlation from different scales of the features. Finally, we adopt the adversarial learning framework by adding a discriminator to differentiate whether the input is from the source domain or not. We will introduce each component in detail in the following part of this section.

2.1. The Semantic Mutual Refinement sub-module

In this report, we propose a Semantic Mutual Refinement sub-module (SMR) to bridge the gap that prevents modality information exchange by channel re-evaluation and re-mapping.

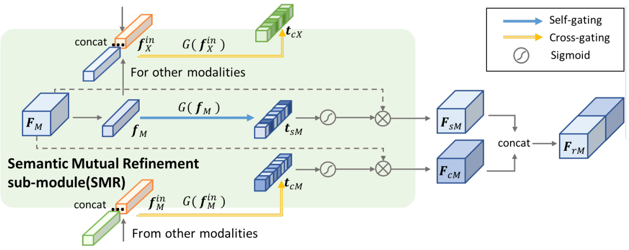

Figure 2. The Semantic Mutual Refinement sub-module (SMR) showcased using modality M. M could be any modalities of RGB, Flow and Audio, also can be extended to other modalities if available, *e.g*., object.

Figure 2 depicts the proposed SMR by showcasing the workflow of modality M. With SMR, the feature $\boldsymbol{F}_M \in \mathbb{R}^{c \times h \times w}$ encoded by the backbone will be enhanced to $\boldsymbol{F}_{rM} \in \mathbb{R}^{2c \times h \times w}$. $\boldsymbol{F}_{rM}$ is the concatenation of a self-refined feature $\boldsymbol{F}_{sM}$ and a cross-refined feature $\boldsymbol{F}_{cM}$. For getting $\boldsymbol{F}_{sM}$, a global average pooling is first conducted on $\boldsymbol{F}_M$ to obtain a global information embedding $\boldsymbol{f}_M \in \mathbb{R}^c$.

$\boldsymbol{f}_M$ will re-evaluate the semantic transferability of modality M itself by a self-gating function [4]:

$$\boldsymbol{t}_{sM} = G(\boldsymbol{f}_M) = \sigma \boldsymbol{W}_2^M(\delta(\boldsymbol{W}_1^M \boldsymbol{f}_M)), \tag{1}$$

where $\boldsymbol{W}_1^M, \boldsymbol{W}_2^M$ are weight matrices, σ and δ denotes the sigmoid and ReLU activations, respectively. Here $\boldsymbol{t}_{sM}$ is the re-evaluation of semantic transferability, and is used to emphasize the channels of $\boldsymbol{F}_M$ by element-wise multiplication on each spatial location (i, j):

$$\boldsymbol{F}_{sM}^{(i,j)} = \boldsymbol{F}_M^{(i,j)} \cdot \boldsymbol{t}_{sM}, \tag{2}$$

For getting $\boldsymbol{F}_{cM}$, we use a similar gating operation but with $\boldsymbol{f}_M^{in}$, the concatenation of features after global average pooling of all other modalities, as input. $\boldsymbol{f}_M^{in}$ serves as the recommendation information provided to modality M by other modalities. We call this step cross-gating and represent it by:

$$\boldsymbol{t}_{cM} = \sigma \boldsymbol{W}_2^{in}(\delta(\boldsymbol{W}_1^{in} \boldsymbol{f}_M^{in})); \qquad \boldsymbol{F}_{cM}^{(i,j)} = \boldsymbol{F}_M^{(i,j)} \cdot \boldsymbol{t}_{cM}, \tag{3}$$

Thus, $\boldsymbol{F}_{cM}$ is the M modality feature refined by other modalities via the cross-gating operation.

It is important to prevent domain adaptation models from overfitting on the source domain. The SMR only introduces a small amount of model parameters by leveraging bottleneck during gating, *i.e*., we reduce the dimension by a ratio r via making $\boldsymbol{W}_1 \in \mathbb{R}^{\frac{c}{r} \times c}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{c \times \frac{c}{r}}$. Finally, we get the refined feature of modality M by fusing the two refined features $\boldsymbol{F}_{sM}$ and $\boldsymbol{F}_{cM}$ via concatenation:

$$\boldsymbol{F}_{rM} = Concat(\boldsymbol{F}_{sM}, \boldsymbol{F}_{cM}). \tag{4}$$

2.2. The Cross-Modality Consensus sub-module

The structure of CMC is shown in Figure 1. This module first uses a 1x1 convolution layer on $\boldsymbol{F}_{rR}$ and $\boldsymbol{F}_{rF}$ for mapping the two modalities into a same latent space, formulating two features $\boldsymbol{H}_{rR}$ and $\boldsymbol{H}_{rF}$. Since the transferable regions vary in size in different samples, we compute the correlation of the feature maps at different scales [6]: the features $\boldsymbol{H}_{rR}$ and $\boldsymbol{H}_{rF}$ are first downsampled a factor of 2 k times, resulting two groups of feature maps $\{\boldsymbol{H}_{rR}^0, \boldsymbol{H}_{rR}^1, ...\boldsymbol{H}_{rR}^k,\}$ and $\{\boldsymbol{H}_{rF}^0, \boldsymbol{H}_{rF}^1, ..., \boldsymbol{H}_{rF}^k,\}$. For each scale k, we compute the Pearson correlation coefficient on each spatial position (i, j) as:

$$\boldsymbol{C}^{k,(i,j)} = \frac{\boldsymbol{H}_{rR}^{k,(i,j)} * \boldsymbol{H}_{rF}^{k,(i,j)}}{\|\boldsymbol{H}_{rR}^{k,(i,j)}\|^2 \cdot \|\boldsymbol{H}_{rF}^{k,(i,j)}\|^2}, \quad \boldsymbol{C}^k \in \mathbb{R}^{\frac{w}{2^k} \times \frac{h}{2^k}} \tag{5}$$

where $*$ indicate dot product. It is important that CMC contains fewest number of parameters so that most of the representation is learned in the SMR, so we choose to use correlation instead of spatial attention [10]. Finally, all the correlation maps $\{\boldsymbol{C}^0, \boldsymbol{C}^1, ..., \boldsymbol{C}^k\}$ are upsampled to match the same size w, h as $\boldsymbol{F}_{rR}$ and summed together to form a consensus map $\boldsymbol{C}$.

The consensus map $\boldsymbol{C}$ is then used as a spatial weight map for the weighted average of feature maps $\boldsymbol{F}_{rR}$ and $\boldsymbol{F}_{rF}$. For generating more robust consensus map, we add a residual connection following [10], forming feature vectors $\boldsymbol{f}_{rR}$ and $\boldsymbol{f}_{rF}$.

2.3. Late fusion and adversarial training

After processed by SMR and CMC, for each modality a refined feature $\boldsymbol{f}_{rR}$, $\boldsymbol{f}_{rF}$ and $\boldsymbol{f}_{rA}$ is acquired, where the fusion of modalities can be adopted. For fusion these features, we concatenate $\boldsymbol{f}_{rR}$, $\boldsymbol{f}_{rF}$ to get a prediction score s_1, then fuse with the score s_2 generated by $\boldsymbol{f}_{rA}$ using weighted average. The weight for s_1 is 1 and for s_2 is 0.5.

Our full loss function is a combination of classification loss $\mathcal{L}_y$ and adversarial loss $\mathcal{L}_d$:

$$\mathcal{L} = \lambda_y \mathcal{L}_y + \lambda_d \mathcal{L}_d \tag{6}$$

3. Experiments

We follow the experiment setup and use the train-val-test split as required by the challenge.

3.1. Feature extraction

We use two backbones for feature extraction: pretrained I3D [1] and pretrained TBN [5], both of them are fine-tuned on the source training set. Additionally, object features extracted by Faster R-CNN object detector trained on EPIC-object-detection dataset is used as the object modality. We also use hand-object bounding boxes to crop the input images, and extract features using TBN without further fine-tuning as the cropped-RGB modality and cropped-Flow modality. The bounding boxes are generated by a hand-object detector trained on 100 DOH dataset [8]. We take the maximum of all detected boxes as the crop area.

3.2. Implementation Details

The SMR processes the feature with dimension $c = 1024$, and the ratio for gating bottleneck is $r = 16$. We empirically choose $\lambda_y = 1$ in all experiments, $\lambda_d = 3$ for experiments with I3D backbone and $\lambda_d = 1$ otherwise. For all experiments, we train the model 30 epochs on 4 NVIDIA-V100 GPUs.

3.3. Result

Table 1 demonstrates the recognition performance on target test set. Using RGB, Flow and Audio modalities and the same backbone TBN, our proposed method performs favorably against TA^{3}N [2] by 1.46% in terms of the accuracy of action.

Module	Top-1			Top-5		
	Verb	Noun	Action	Verb	Noun	Action
TA^{3}N	46.91	27.69	18.95	72.70	50.72	30.53
TA^{3}N+Ours	49.99	30.45	20.41	77.97	54.58	35.20

Table 1. Comparison of action recognition result on the target test set.

3.4. Model ensemble

To take advantages of models trained with different inputs or different backbones, we explore model ensemble technique to fuse the following models:

- Model A: taking RGB, Flow and Audio modalities as inputs, and using TBN as the feature extraction backbone.
- Model B: compared with Model A, adding object features as an additional modality.
- Model C: taking cropped-RGB, cropping-Flow and Audio modalities as inputs, and using TBN as the feature extraction backbone.
- Model D: taking RGB and Flow modalities as inputs, and using I3D as the feature extraction backbone.

All of the four models use TA^{3}N+Ours as the domain adaptation module.

Models	Top-1			Top-5		
	Verb	Noun	Action	Verb	Noun	Action
A+B	51.45	34.07	22.93	80.88	59.03	38.69
A+B+C	52.60	35.32	24.13	81.30	59.57	39.96
A+B+C+D	53.29	35.64	24.76	81.64	59.89	40.73

Table 2. Model ensemble results on the target test set.

4. Conclusion

In this report, we introduce a novel Multi-Modal Mutual Enhancement module, which enables the mutual refinement between multiple modalities. The experimental result validates that our M^{3}EM can significantly improve the domain adaptive action recognition performance. With model ensemble technique, we achieve competitive results on the leaderboard of the 2021 EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge.

References

[1] 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. 3

[2] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019. 3

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[5] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 5492–5501, 2019. 3

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[7] 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*, pages 122–132, 2020. 1

[8] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. 2020. 3

[9] Sijie Song, Jiaying Liu, Yanghao Li, and Zongming Guo. Modality compensation network: Cross-modal adaptation for action recognition. *IEEE Transactions on Image Processing*, 29:3957–3969, 2020. 1

[10] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 2

PoliTO-IIT Submission to the EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition

Chiara Plizzari[*,1] Mirco Planamente[*,1,2] Emanuele Alberti [1] Barbara Caputo[1,2]

[1] Politecnico di Torino
name.surname@polito.it

[2] Istituto Italiano di Tecnologia
name.surname@iit.it

Abstract

In this report, we describe the technical details of our submission to the EPIC-Kitchens-100 Unsupervised Domain Adaptation (UDA) Challenge in Action Recognition. To tackle the domain-shift which exists under the UDA setting, we first exploited a recent Domain Generalization (DG) technique, called Relative Norm Alignment (RNA). It consists in designing a model able to generalize well to any unseen domain, regardless of the possibility to access target data at training time. Then, in a second phase, we extended the approach to work on unlabelled target data, allowing the model to adapt to the target distribution in an unsupervised fashion. For this purpose, we included in our framework existing UDA algorithms, such as Temporal Attentive Adversarial Adaptation Network (TA^{3}N), jointly with new multi-stream consistency losses, namely Temporal Hard Norm Alignment (T-HNA) and Min-Entropy Consistency (MEC). Our submission (entry 'plnet') is visible on the leaderboard and it achieved the 1st position for 'verb', and the 3rd position for both 'noun' and 'action'.

1. Introduction

First person action recognition offers a wide range of opportunities which arise from the use of wearable devices. In fact, since it intrinsically comes with rich sound information, due to the strong hand-object interactions and the closeness of the sensors to the sound source, it encourages the use of auditory information. Moreover, the continuous movement of the camera, which moves around with the observer, strongly motivates the use of secondary modalities capturing the motion in the scene, such as optical flow.

Our idea is that exploiting the intrinsic peculiarities of all these modalities is of crucial importance, especially in cross-domain scenarios. In fact, these modalities suffer from a domain shift which is not of the same nature. For instance, the optical flow modality, by focusing on the motion in the scene rather than on the appearance, is less sensitive to environmental changes, and thus potentially more robust than the visual modality when changing environment [10] (Figure 1). On the other side, the domain shift of auditory information is very different from the visual one (e.g., the sound of 'cut' will differ from a plastic to a wooden cutting board). For all those reasons, the classifier should be able to measure and understand which modality is informative and should rely on in the final prediction, and which is not.

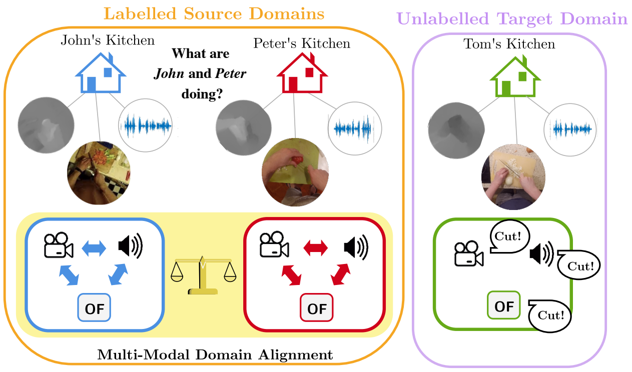

Figure 1. The correlation between the distinctive sound of an action and its corresponding visual information or motion is not always guaranteed across different domains. Thus, effectively combining multi-modal information from *multiple sources* is fundamental to increase the capability to recognize daily actions.

To this purpose, authors of [11] recently proposed a multi-modal framework, called Relative Norm Alignment network (RNA-Net), which aims to progressively align the feature norms of audio and visual (RGB) modalities among multiple sources in a Domain Generalization (DG) setting, where target data are not available during training. In that work, they bring to light that *simply feeding all*

[*] The authors equally contributed to this work. This paper is partially supported by the ERC project RoboExNovo. We also acknowledge that the research activity herein was carried out using the IIT HPC infrastructure.

the source domains to the network without applying any adaptive techniques leads to sub-optimal performance. Indeed, a multi-source domain alignment allows the network to promote domain-agnostic features.

Interestingly, the availability of multiple sources in the official challenge dataset make it perfect to tackle the problem under a DG setting. To this purpose, we extended RNA-Net to the Flow modality, obtaining remarkable results without accessing target data. In a second stage, we further adapted it to work with unlabelled target data under the standard Unsupervised Domain Adaptation (UDA) setting. Finally, our final submission was obtained by ensembling different model streams by means of DA-based consistency losses, namely Temporal Hard Norm Alignment (T-HNA) and Min-Entropy Consistency (MEC).

2. Our Approach

In this section, we first describe the DG approach we used. Then, we illustrate its extension to unlabelled target data under the standard UDA framework. Finally, we repurpose existing DA-based losses to induce consistency between different architectures.

2.1. Domain Generalization

The multi-source nature of the proposed challenge setting makes it perfect to deal with the domain shift using DG techniques. Thus, we first exploited a method which has been recently proposed to operate in this context, called Relative Norm Alignment (RNA) [11]. This methods consists in performing an *audio-visual domain alignment* at feature-level by minimizing a cross-modal loss function ($\mathcal{L}_{RNA}$). The latter aims at minimizing the *mean-feature-norm distance* between the audio and visual features norms among all the source domains, and it is defined as

$$\mathcal{L}_{RNA} = \left(\frac{\mathbb{E}[h(X^v)]}{\mathbb{E}[h(X^a)]} - 1 \right)^2, \tag{1}$$

where $h(x_i^m) = (\|\cdot\|_2 \circ f^m)(x_i^m)$ indicates the L_2-norm of the features f^m of the m-th modality, $\mathbb{E}[h(X^m)] = \frac{1}{N}\sum_{x_i^m \in \mathcal{X}^m} h(x_i^m)$ for the m-th modality and N denotes the number of samples of the set $\mathcal{X}^m = \{x_1^m, ..., x_N^m\}$.

Authors of [11] proved that the norm unbalance between different modalities might cause the model to be biased towards the source domain that generate features with greater norm and thus causing a wrong prediction. Indeed, by simultaneously solving the problem of classification and relative norm alignment on different domains, the network extracts a shared knowledge between the different sources, resulting in a domain-agnostic model.

In our submission to the EPIC-Kitchen UDA challenge, we extended the RNA-Net framework to the optical flow modality, and we exploited the multiple sources available from the official training splits to show the effectiveness of RNA loss in a multi-source DG setting.

2.2. Domain Adaptation

In this section, we describe the UDA techniques that are integrated in our approach.

Relative Norm Alignment Network. We followed the extension towards the UDA setting proposed in [11], which is possible thanks to the unsupervised nature of RNA. In order to consider the contribution of both source and target data during training, we redefined $\mathcal{L}_{RNA}$ under the UDA setting as

$$\mathcal{L}_{RNA} = \mathcal{L}_{RNA}^s + \mathcal{L}_{RNA}^t, \tag{2}$$

where $\mathcal{L}_{RNA}^s$ and $\mathcal{L}_{RNA}^t$ correspond to the RNA formulation in Equation 1 illustrated above, when applied to source and target data respectively.

Temporal Attentive Adversarial Adaptation Network (TA^{3}N). Authors of [2] proposed an UDA technique based on three components. The first one, called *Temporal Adversarial Adaptation Network (TA^{2}N)*, consists in an extension of DANN [5], aiming to align the temporal features on a multi-scale Temporal Relation Module (TRM) [14] through a gradient reversal layer (GRL). The second component is based on a domain attention mechanism which guides the temporal alignment towards features where the domain discrepancy is larger. Finally, the third component uses a minimum entropy regularization (attentive entropy) to refine the classifier adaptation.

2.3. Ensemble UDA losses

For our final submission, different models are used in order to exploit the potentiality of popular video architectures. Training individually each backbone with standard UDA protocols results in an adapted feature representation which varies from stream to stream. Our intuition is that this aspect could impact negatively the training process and the performance on target data. In fact, since the domain adaption process acts on each architecture independently, different prediction logits are obtained on target data. When combining them, this could cause a mismatch between the final scores, increasing the level of uncertainty of the model. Thus, we impose a consistency constraint between feature representations from different models, by repurposing existing UDA loss functions to operate between multiple streams. Those are:

Temporal Hard Norm Alignment (T-HNA). It rebalances the contribution of each model during training by extending HNA [11] to align the norms of features coming from the different streams towards the same value R. This is applied on features extracted from multiple scales of each TRN module. The resulting $\mathcal{L}_{\textit{T-HNA}}$ is defined as

Unsupervised Domain Adaptation Leaderboard

	Rank	Verb Top-1	Noun Top-1	Action Top-1	Verb Top-5	Noun Top-5	Action Top-5
chengyi	1	53.16	34.86	**25.00**	80.74	59.30	40.75
M3EM	2	53.29	**35.64**	24.76	81.64	59.89	40.73
plnet	3	**55.22**	34.83	24.71	**81.93**	**60.48**	**41.41**
EPIC_TA3N [3]	6	46.91	27.69	18.95	72.70	50.72	30.53
EPIC_TA3N_SOURCE_ONLY [3]	12	44.39	25.30	16.79	69.69	48.40	29.06

Table 1. Leaderboard results of EPIC-Kitchens Unsupervised Domain Adaptation Challenge. The results obtained by the top-3 participants and the provided baseline methods are reported. **Bold:** highest result; Green: our final submission.

Ensemble UDA losses

	Top-1			Top-5		
	Verb	Noun	Action	Verb	Noun	Action
Ensemble	52.83	30.82	21.96	**81.04**	52.67	46.66
Ensemble+T-HNA	53.84	32.54	22.65	80.63	54.86	48.03
Ensemble+T-HNA+MEC	**54.02**	**33.53**	**23.58**	81.00	**55.03**	**48.27**

Domain Generalization

	Target	Verb Top-1	Verb Top-5
Source Only	✗	44.39	69.69
EPIC_TA3N [3]	✓	46.91	72.70
RNA-Net [11]	✗	47.96	79.54
EPIC_TA3N+RNA-Net	✓	**50.40**	**80.47**

Table 2. **Left.** Results on the EPIC-Kitchen validation set with different ensembling UDA losses. **Right.** Results on EPIC-Kitchen test set under the DG setting. **Bold** highest result.

$$\mathcal{L}_{T\text{-}HNA} = \sum_{b} \left(\mathbb{E}[h_t(X^b)] - R\right)^2, \tag{3}$$

where h_t denotes the L_2-norm of features extracted from the t-th multi-scale level of the b-th backbone network.

Min Entropy Consensus (MEC loss). We extended the loss proposed in [12] to encourage coherent predictions between different models. The resulting loss is defined as:

$$\mathcal{L}_{MEC} = -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{b}\max_{y\in\mathcal{Y}}\sum_{b}\text{log}p_b(y|x_i^t) \tag{4}$$

where m is the cardinality of the batch size of the target set, y is the predicted class, and $\text{log}p_b(y|x_i^t)$ is the prediction probability of the b-th backbone network. The intuitive idea behind the proposed approach is to encourage different backbones to have a similar predictions.

3. Framework

In this section, we describe the architectures of the feature extractors used to produce suitable multi-modal video embeddings, and the fusion stategies adopted to combine them. We complete this section with the description of the hyper-parameters used for the training.

3.1. Architecture

Backbone. For our submission, we adopted different network configurations. In the first one, corresponding to the RNA-Net framework in [11], we used the Inflated 3D ConvNet (I3D), pre-trained on Kinetics [1], for RGB and Flow streams, and a BN-Inception model [7] pre-trained on ImageNet [4] for the auditory information. Each feature extractor produces a 1024-dimensional representation which is fed to an action classifier. In the second configuration, we used BNInception for all the three streams, using pre-extracted features from a TBN [10] model trained on EPIC-Kitchens-55. In the last configurations, we used standard ResNet50 [6] for all the streams using TSN [13] and TSM [9] models pre-trained on Epic-Kitchen55[1].

λ_{RNA}	λ_{HNA}	R	λ_{MEC}	γ	β
1	0.0006	40	0.01	0.003	0.75, 0.75, 0.5

Table 3. UDA losses hyper-parameters used during training.

Multi-modal fusion strategies. In all the above mentioned configurations, each modality is processed by its own backbone, and the corresponding extracted representations are then fused following different strategies. For RNA-Net, we followed a standard late fusion strategy, consisting in averaging the final score predictions obtained from two different fully-connected layers (verb, noun) from each modality. In the other configurations, we adopted the mid-fusion strategy proposed in [8], to generate a common frame-embedding among the modalities and used a Temporal Relation Module (TRM) [14] to aggregate features from different frames before feeding the final embeddings to the verb and noun classifiers.

3.2. Implementation Details

We trained I3D and BNInception models with SGD optimizer, with an initial learning rate of 0.001, dropout 0.7,

[1] https://github.com/epic-kitchens/epic-kitchens-55-action-models

and using a batch size of 128, following [11]. Instead, when using pre-extracted features from ResNet50 or BN-Inception, we trained the TRM modules on top of them for 100 epochs with an initial learning rate of 0.03, decayed after epochs 30 and 60 by a factor of 0.1. We used a batch size of 128 with SGD optimizer. In Table 3 we report the other hyper-parameter used. Specifically, we indicate with λ_{RNA}, λ_{T-HNA} and λ_{MEC} the weights of RNA, T-HNA and MEC losses respectively, and with R the values of the radius of T-HNA (see Equation 4). In addition, we report the values used in TA^{3}N to weight the attentive entropy loss (γ) and the domain losses at different levels (β).

4. Results and Discussion

In Table 1 we report our best performing model on the target test, achieving the **1st** position on 'verb', **3rd** on 'noun' and 'action', and **1st** position on Top-5 accuracy on all categories. In Table 2 (left) we show an ablation on the contribution of the proposed ensemble UDA losses, T-HNA and MEC respectively, on the official validation set. As it can be seen, they improve Top-1 accuracy on all categories by up to 2%, proving the effectiveness of imposing a consistency between features from different streams.

How well do DG approaches perform? We show in Table 2 (right) the results obtained under the multi-source DG setting, when target data are not available during training. Noticeably, RNA outperforms the baseline Source Only by up to 3% on Top-1 and 10% on Top-5, remarking the importance of using ad-hoc alignment techniques to deal with multiple sources in order to effectively extract a domain-agnostic model. Moreover, it outperforms the very recent UDA technique TA^{3}N without accessing to target data. Interestingly, when combined with EPIC_TA3N, it further improves performance, proving the complementarity of RNA to other existing UDA approaches.

References

[1] 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.

[2] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6321–6330, 2019.

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[4] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] 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.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456. PMLR, 2015.

[8] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[9] 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.

[10] 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*, pages 122–132, 2020.

[11] Mirco Planamente, Chiara Plizzari, Emanuele Alberti, and Barbara Caputo. Cross-domain first person audio-visual action recognition through relative norm alignment. *arXiv preprint arXiv:2106.01689*, 2021.

[12] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9471–9480, 2019.

[13] 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.

[14] 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.

Team PyKale (xy9) Submission to the EPIC-Kitchens 2021 Unsupervised Domain Adaptation Challenge for Action Recognition

Xianyuan Liu[1,2,3], Raivo Koot[3], Shuo Zhou[3], Tao Lei[1,2], Haiping Lu[3]
[1]Institute of Optical and Electronics, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China [3]University of Sheffield, United Kingdom
{xianyuan.liu, szhou20, h.lu}@sheffield.ac.uk, raivokoot@gmail.com, taoleiyan@ioe.ac.cn

Abstract

This report describes the technical details of our submission to the EPIC-Kitchens 2021 Unsupervised Domain Adaptation Challenge for Action Recognition. The EPIC-Kitchens dataset is more difficult than other video domain adaptation datasets due to multi-tasks with more modalities. Firstly, to participate in the challenge, we employ a transformer to capture the spatial information from each modality. Secondly, we employ a temporal attention module to model temporal-wise inter-dependency. Thirdly, we employ the adversarial domain adaptation network to learn the general features between labeled source and unlabeled target domain. Finally, we incorporate multiple modalities to improve the performance by a three-stream network with late fusion. Our network achieves the comparable performance with the state-of-the-art baseline TA^3N and outperforms the baseline on top-1 accuracy for verb class and top-5 accuracies for all three tasks which are verb, noun and action. Under the team name xy9, our submission achieved 5th place in terms of top-1 accuracy for verb class and all top-5 accuracies.

1. Introduction

The EPIC-Kitchens dataset (EPIC-Kitchens) [2] is the largest dataset in the first-person viewpoint, with daily activities captured in the kitchen, and provides some benchmarking challenges for researchers to explore including Unsupervised Domain Adaptation (UDA) for first-person action recognition. UDA for first-person action recognition is a challenging problem that aims to minimize the distribution distance between domains (domain shift) in both spatial and temporal feature spaces. Action can be divided into two parts, verb and noun. The verb refers to temporal feature spaces, while the noun is for spatial.

Most current UDA networks for action recognition focus on only verb class [4] or directly on action class [1] without exploring noun class. However, in first-person videos, actions tend to occur in some local areas, particularly where the hands and objects interact, and more than one object is usually visible in these local areas. Therefore, it will benefit the network learning that networks pay more attention to these areas and the object interacting with the hands. In this challenge, EPIC-kitchens provides annotations with the verb and noun classes, requires researchers to evaluate their networks on these two tasks individually and utilizes these two task results to calculate the action classification accuracy. The deficiencies of each will affect the final action performance. This setting is more clear for us to evaluate the model learning on spatial and temporal feature spaces.

To participate in the challenge, we propose to apply a transformer to locate the more important local areas to capture the spatial information and temporal-wise attention to extract the temporal information. We then construct a three-stream network to extract the spatio-temporal information from three modalities (RGB, Flow and Audio). Finally, we employ a domain adversarial neural network (DANN) [3] to tackle the domain shift problem. Our submission shows our network has achieved comparable performance with the state-of-the-art baseline TA^3N.

2. Proposed Method

In this section, we briefly introduce our method utilized for participating in the challenge.

2.1. Transformer Module

To capture better spatial information, we employ the transformer which can generate the attention-based representation to locate a specific piece of information from context in [5]. For efficiency, we employ a simple version of the transformer, containing a multi-head self-attention layer and a fully connected feed-forward network, to capture more important spatial information from all frames in an action segment.

For multi-head self-attention layer construction, We

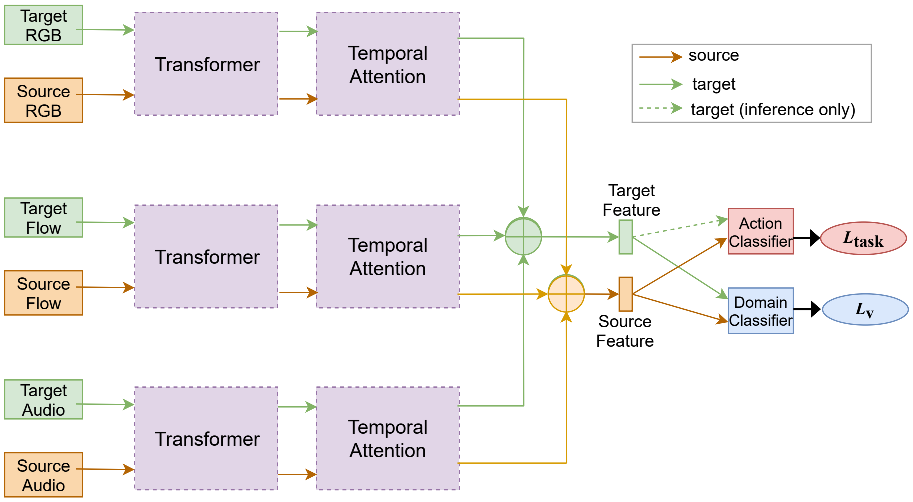

Figure 1. The proposed network for first-person action recognition. For each modality, source and target domains share the same transformers and temporal-wise attention. In training, transformers and temporal-wise attention take labeled source and unlabeled target features as the input and generates source and target spatio-temporal features as the output. All features for three modalities from source and target are fused into source features and target features. Source features are fed into both action and domain classifiers, while target features are only fed to domain classifier. In test, only target features are the input to the transforms and attention and then the action classifier.

firstly build scaled dot-product attention for the input feature. We obtain key $\mathbf{K} \in \mathbb{R}^{d_k}$, query $\mathbf{Q} \in \mathbb{R}^{d_K}$ and value $\mathbf{V} \in \mathbb{R}^{d_v}$ from the input feature by fully connected layers with weights of $\mathbf{W^Q}$, $\mathbf{W^K}$ and $\mathbf{W^V}$. We compute the dot product of $\mathbf{Q}$ and $\mathbf{K}$, and re-scale the dot product outputs with the scaling factor of $\frac{1}{\sqrt{d_k}}$. We apply the softmax function to generate the attention weights and compute another dot product of these weights and $\mathbf{V}$, as shown in Eq. (1).

$$A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V} \tag{1}$$

As shown in Eq. (2), we secondly utilize several scaled dot-product attentions to build multi-head attention, which can learn the information from different representation subspaces from different areas. We generate different $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ by different fully connected layers. We then apply attentions to different $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ individually and concatenate all the outputs.

$$M = concat(A(\mathbf{QW_n^Q}, \mathbf{KW_n^K}, \mathbf{VW_n^V})), n \in (1, N) \tag{2}$$

where N refers to the number of attentions used in the multi-head self-attention layer.

We finally combine the multi-head self-attention layer with a fully connected feed-forward network to finish our transformer construction for spatial information extraction.

2.2. Temporal-wise Attention Module

To improve the temporal feature extraction, we build a temporal-wise attention module that excites informative temporal features in the 3D input features. In this 3D feature $\mathbf{X} \in \mathbb{R}^{N \times T \times F}$, N, T and F denote batch size, temporal dimension and feature size. First, we utilize average pooling on feature size to capture the temporal-wise feature $\mathbf{X}_t \in \mathbb{R}^{N \times T \times 1}$.

$$\mathbf{X}_t = \frac{1}{F}\sum_{f=1}^{F}\mathbf{X} \tag{3}$$

Second, we reduce the feature through the temporal dimension by a linear layer with weights $\mathbf{W}_{t1}$ with a reduction ratio r for efficiency. Here is the temporal-reduced feature $\mathbf{X}_{tr} \in \mathbb{R}^{N \times T/r \times 1}$.

$$\mathbf{X}_{tr} = \mathrm{ReLU}(\mathbf{W}_{t1}\mathbf{X}_t) \tag{4}$$

Third, we utilize another linear layer with weights $\mathbf{W}_{t2}$ to restore the temporal dimension and a sigmoid function σ to capture the temporal-wise attentive weights. Finally, we compute a Hadamard product between the weights and the input feature, and utilize skip connection to prevent temporal attention from suppressing other information. The excited output $\mathbf{X}_t^o \in \mathbb{R}^{N \times T \times F}$ is as below.

$$\mathbf{X}_t^o = \mathbf{X} + \mathbf{A}_t \odot \mathbf{X} = \mathbf{X} + \sigma(\mathbf{W}_{t2}\mathbf{X}_{tr}) \odot \mathbf{X} \tag{5}$$

2.3. Adversarial UDA

After feature extraction, the network needs to learn common spatio-temporal features across domains for classification tasks. We utilize a discriminator G_f and a domain classifier G_d to form the DANN, in which a two-player mini-

	TA^3N [1]	MLP	T	T+A
Verb	48.92	48.54	<u>49.18</u>	**49.93**
Noun	<u>29.39</u>	25.98	28.76	**30.59**

Table 1. The comparison of top-1 accuracy (%) with other approaches on validation dataset. MLP refers to 2 fully connected layers without transformer and attention. T refers to MLP with transformer and T+A refers to MLP with transformer and attention. The best result for each task is in **bold**, and the second best is <u>underlined</u>.

max game is constructed considering the limited computation resources. The domain loss L_v is defined for each input x_i as:

$$L_v = -\frac{1}{n} \sum_{x_i \in D_s \cup D_t} L_d(G_d(G_f(x_i)), d_i) \tag{6}$$

where D_s and D_t are source and target domains respectively, n is the number of samples from both domains, and d_i is the domain label of x_i. If x_i is from the source (target) domain, d_i is set as 1 (0).

2.4. Late Fusion

After constructing spatial and temporal feature extractors, we employ the late fusion to combine extracted features from these three modalities by concatenation. We finally integrate transformer, temporal-wise attention, adversarial UDA and late fusion into a three-stream network and also build multiple classifiers for each task to generate classification loss L_{ci} for task i. The total loss is shown in the Eq. (7).

$$\begin{aligned} L =& \lambda_v L_v + L_{task} \\ =& \lambda_v L_v + \sum_{i=1}^{I} L_{ci} \end{aligned} \tag{7}$$

where λ_v is a hyper-parameter to trade-off domain adaptation with classification respectively and I refers to the total number of tasks.

3. Experiments

In this section, we introduce the experimental details for our proposed method.

3.1. Experimental Setup

Dataset. We test our proposed network on the challenge provided dataset following the challenge guidelines. The usage of the dataset can be divided into two steps. The first step is using provided validation dataset for model building and debugging because no label provided for train and test datasets. We split the validation dataset into train and test subsets by random sampling with a ratio of 8:2. We apply the same strategy to both source validation and target validation datasets to build new source and target subsets. We then train and evaluate our network on these subsets and select the best hyperparameters. The second step is to train our network directly on the provided train and test datasets with selected hyper-parameter and then submit the results to the public leader board.

Implementation details. We utilize four transformers for spatial feature extraction with a hidden layer size of 512 and 8 heads and one temporal-wise attention for temporal with a reduction ratio of 2. Between transformers and attentions, we utilize two fully connected layers to reduce the feature dimension from 1024 to 512 for efficiency. We also utilize one domain discriminator and two classifiers composed of 2 fully connected layers. The dimension of the discriminator is 100, while the verb classifier is 128 and the noun classifier is 512 due to different class number. In the training process, we utilize labeled source data and unlabeled target data and only use the unlabeled target data in the testing process. In the first 20 epochs, we only train the transformer, attention and classifiers, while train the whole network in the rest 80 epochs. The optimisation is performed using SGD with a momentum of 0.9 and batch size of 128. A weight decay with 5e-4 is applied for all parameters.

3.2. Experimental Results

Results on the validation dataset. We evaluate our network on the validation dataset first and the results are shown in Table 1. Our network outperforms the baseline TA^3N in top-1 accuracy for verb by 1.01% and noun by 1.20%. We also explore the performance of different structures. We first test the network with 2 fully connected layers (MLP). We find the accuracy decreases slightly for verb but sharply for the noun. It shows a very simple network can still learn the temporal information from these multi-modalities. One reason is Flow and Audio containing more temporal information than spatial information. TA^3N utilizes the early fusion to combine the three modality features and extract the spatio-temporal information from these combined features. However, we apply individual stream to each modality and utilize late fusion to keep their respective characteristics. We then apply transformers to improve the spatial information extraction and the result shows transformers increase the top-1 accuracy for noun by 2.78%. Finally, we add temporal-wise attention, which benefits both verb and noun top-1 accuracy of 0.75% and 1.83%.

Results for submission. We then train and evaluate our network on the training and test dataset and the results are shown in Table 2. Our network outperforms TA^3N in target top-1 accuracy for verb by 1.54% and achieves comparable

Team	Target Top-1			Target Top-5			Source Top-1			Source Top-5		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
plnet	55.22 (1)	34.83 (3)	24.71 (3)	81.93 (1)	60.48 (1)	41.41 (1)	63.26 (1)	46.37 (1)	36.27 (1)	88.32 (1)	69.99 (1)	53.96 (1)
M3EM	53.29 (2)	35.64 (1)	24.76 (2)	81.64 (2)	59.89 (2)	40.73 (3)	-	-	-	-	-	-
chengyi	53.16 (3)	34.86 (2)	25.00 (1)	80.74 (3)	59.30 (3)	40.75 (2)	-	-	-	-	-	-
tackgeun	51.09 (4)	29.60 (4)	21.19 (4)	75.44 (7)	52.34 (4)	35.12 (4)	-	-	-	-	-	-
PyKale (xy9)	48.45 (5)	27.31 (8)	18.56 (7)	77.31 (5)	52.09 (5)	33.47 (5)	60.66 (3)	40.34 (4)	30.41 (4)	85.67 (2)	66.89 (4)	50.91 (4)
...	...	...	...	...	...	...	...	...	...	...	...	...
TA^3N	46.91 (11)	27.69 (7)	18.95 (6)	72.70 (10)	50.72 (8)	30.53 (11)	60.52 (4)	44.42 (2)	34.04 (2)	84.81 (3)	67.73 (2)	51.54 (2)
TA^3N SOURCE ONLY	44.39 (12)	25.30 (12)	16.79 (12)	69.69 (12)	48.40 (11)	29.06 (12)	60.98 (2)	44.07 (3)	33.67 (3)	84.76 (4)	67.50 (3)	51.40 (3)

Table 2. The comparison of accuracy (%) with submission approaches. Numbers in brackets refer to the ranking. Our network xy9 ranks 5th in four target accuracies.

results on other top-1 accuracies. Our network improves the top-5 accuracy significantly by 4.61% for verb, 1.37% for noun and 2.94% for action. It's worth noting that TA^3N outperforms our network on the source accuracy which is the upper limit accuracy for the target domain. It means even with the lower upper limit, our network can achieve better results than TA^3N. The limit can be increased by adjusting the training strategy or selecting better hyper-parameters.

4. Conclusion and Future Work

In this report, we explore to utilize transformer and attention to improve the UDA performance for first-person action recognition. The results show our network achieve comparable performance with the state-of-art networks. However, the performance of our network on the noun is not good enough. It means our network needs some improvements in spatial information extraction. In the future, we will focus on spatial feature extraction and explore new strategy to capture spatial information.

References

[1] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, 2019. 1, 3

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE TPAMI*, 2020. 1

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

[4] Jonathan Munro and Dima Damen. Multi-modal Domain Adaptation for Fine-grained Action Recognition. In *CVPR*, 2020. 1

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1

Team IIE-MRG: Technical Report for EPIC-KITCHENS-100 2021 Multi-Instance Retrieval Challenge

Xiaoshuai Hao, Wanqian Zhang, Dejie Yang, Shu Zhao, Dayan Wu, Bo Li, Weiping Wang
Institute of Information Engineering, Chinese Academy of Sciences
{haoxiaoshuai, zhangwanqian, yangdejie, zhaoshu, wudayan, libo, wangweiping}@iie.ac.cn

Abstract

In this report, we present a solution to the EPIC-KITCHENS-100 2021 Multi-Instance Retrieval Challenge. The task of retrieving relevant videos with natural language queries plays a critical role in effectively indexing large-scale video data. At present, the current dominant for cross-modal video-text retrieval is commonly achieved through learning a shared embedding space, that can indifferently embed modalities. However,all existing works train the embedding network by considering the inter-modal constraint to make the semantically-similar texts and videos much closer to each other and vice versa. Ideally, a good embedding space should also satisfy the requirement that similar videos/texts should stay closer. Thus, we argue that preserving this modality-specific characteristic is essential for learning the embedding space. In this report, we elaborately design a novel Dual Constraint Ranking Loss (DCRL), which simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint to preserve both the cross-modal semantic similarity and the modality-specific consistency in the embedding space. This novel method allowed us to achieve the 1st place in the CVPR 2021 workshop of EPIC KITCHENS-100 Multi-Instance Retrieval Challenge.

1. Multi-Instance Retrieval Challenge

In this report, we present the method that we implemented for the EPIC-KITCHENS-100 2021 Multi-Instance Retrieval Challenge. This challenge tackles the task of caption-to-video retrieval. Specifically, given a query action segment, the aim of video-to-text retrieval is to rank captions in a gallery set, C, such that those with a higher rank are more semantically relevant to the action in the video. Conversely, text-to-video retrieval uses a query caption $c_i \in C$ to rank videos. The challenge uses EPIC-KITCHENS-100 dataset [4]. The EPIC-KITCHENS-100 dataset is an unscripted egocentric action dataset collected from 45 kitchens from 4 cities across the world. Submissions are evaluated on the test set for action retrieval. This Challenge uses two evaluation metrics: mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG).

2. Motivation

With the rapid growth of user-generated videos, cross-modal retrieval between video data and natural language descriptions, known as video-text retrieval, has attracted much attention. Most existing methods [3, 5] adopt the visual feature to represent videos. However, other rich information in the videos which is effective for video-text retrieval is ignored. The video pentathlon challenge 2020 [1] recently defined a retrieval challenge across five datasets, which considers multi-modal features in the video. Recently, feature aggregation methods [9, 8, 10, 7] greatly boost the benchmark of video-text retrieval, which make use of different features in videos like object, motion, audio, and caption on the screen. Moreover, Wray et al. [12] propose to enrich the embedding by disentangling parts-of-speech (PoS) in the accompanying captions.

However, all existing works train the embedding network by considering the inter-modal constraint to make the semantically-similar texts and videos much closer to each other and vice versa. Ideally, a good embedding space should also satisfy the requirement that similar videos/texts should stay closer. Thus, we argue that preserving this modality-specific characteristic is essential for learning the embedding space. In this report, we elaborately design a novel Dual Constraint Ranking Loss (DCRL) that simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint. In light of the proposed DCRL, we can preserve the modality-specific characteristics in the embedding space to further improve retrieval performance. With our method, not only more target videos can be retrieved, but also similar videos are ranked higher than other irrelevant videos as they are mapped closer in the embedding space.

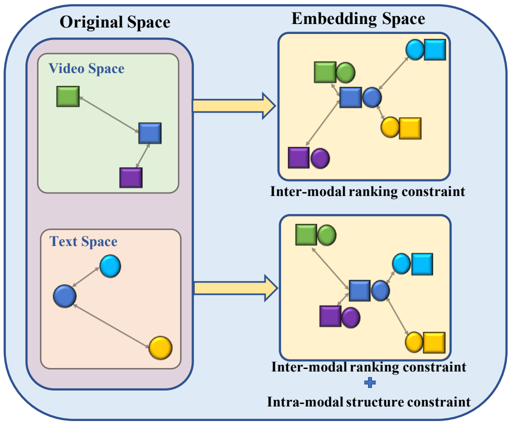
Original Space
Embedding Space
Video Space
Text Space
Inter-modal ranking constraint
Inter-modal ranking constraint
+
Intra-modal structure constraint

Figure 1. Embedding space with (top) and without (bottom) intra-modal structure constraint. By leveraging the intra-modal structure constraint, we can preserve modality-specific characteristics in the joint embedding space **(best viewed in color)**.

3. Method

Our aim is to learn representations suitable for cross-modal search where the query modality is different from the target modality. Specifically, we use video sequences with textual captions/descriptions and perform video-to-text (vt) or text-to-video (tv) retrieval tasks.

At present, given a video V and a query text T, we try to create a pair of functions $\phi(V)$ and $\psi(T)$ mapping videos and texts into a joint embedding space, in which embeddings for matched texts and videos should lie close together, while embeddings for mismatched texts and videos should lie far apart. Additionally, a good embedding space should also satisfy the requirement that similar videos/texts should stay closer. Thus, we argue that preserving this modality-specific characteristic is essential for learning the embedding space. In this report, we elaborately design a novel Dual Constraint Ranking Loss (DCRL), which simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint to preserve both the cross-modal semantic similarity and the modality-specific consistency in the embedding space.

3.1. Video Embedding

We extract flow and appearance features using the TSN BNInception model [11] pre-trained on Kinetics and fine-tuned on our training set. TSN averages the features from 25 uniformly sampled snippets within the video. We then concatenate appearance and flow features to create a 2048 dimensional vector per action segment.

3.2. Text Embedding

We map each lemmatised word to its feature vector using a 100-dimension Word2Vec model, trained on the Wikipedia corpus. Multiple word vectors with the same part of speech were aggregated by averaging. Motivated by [12], we break down the text caption into different PoS tags. In our experiments, we focus on the most relevant ones for fine- grained action recognition: verbs and nouns. We extract all words from a caption for a given PoS tag and obtain one PoS tags embeds of these words.

3.3. Joint Embedding Learning

In this subsection, we introduce the Dual Constraint Ranking Loss (DCRL) in detail, which simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint.

Inter-modal ranking constraint: Existing works train the embedding network with the only consideration of the ranking constraints between modals, which makes the semantically similar texts and videos become closer and vice versa. While bridging the gap between an anchor and a positive sample, inter-modal ranking constraint can also maximize the distance between an anchor and a negative sample. The expression of the inter-modal ranking constraint of a video is as follows:

$$d\left(V_i, T_i\right) + m < d\left(V_i, T_j\right), \tag{1}$$

where, V_i (anchor) and T_i (positive sample) are the feature embeddings in the joint embedding space for the i-th video and text. T_j (negative sample) refers to the j-th text. $d\left(V, T\right)$ indicates the distance between two feature embeddings in the joint embedding space, and m indicates a margin constant. Analogously, given a text input, we set the inter-modal ranking constraint as follows:

$$d\left(T_i, V_i\right) + m < d\left(T_i, V_j\right). \tag{2}$$

In this triplet selection, there are two methods: the bi-directional max-margin ranking loss (Bi-MMRL), which calculates for all negatives; and the bi-directional hard-negatives ranking loss (Bi-HNRL) only the penalty incurred by the hardest negatives is considered. We adopt the Bi-HNRL as it has been proved more effective [5].

Intra-modal structure constraint: During the whole training procedure, if we only utilize the inter-modal ranking constraint, inherent characteristics within each modality (i.e., modality-specific characteristics) will be lost. To solve this problem, we devise a novel intra-modal structure constraint.

Suppose there are three samples (videos or texts), we can extract features using the process described in Section. 3.1 or Section. 3.2. Since deep features are not been fed into the joint embedding network, they can be used measure the modality-specific similarities. As Fig. 1 show, in the video space, blue is more similar to purple than green. In the text space, blue is more similar to sky-blue than yellow. By

Table 1. Video-to-Text and Text-to-Video retrieval results on the EPIC-KITCHENS-100 dataset.

Method	**mean Average Precision (%)**			**normalised Discounted Cumulative Gain (%)**		
	Avg	T2V	V2T	Avg	T2V	V2T
MLP	38.49	33.99	42.99	48.49	46.92	50.05
JPoSE	44.01	38.11	49.91	53.53	51.55	**55.51**
DCRL(Our)	**44.23**	**38.49**	**49.96**	**53.56**	**51.83**	55.28

leveraging the intra-modal structure constraint in the embedding space, we can preserve modality-specific characteristics after the joint embedding process. The intra-modal structure constraint between samples is a soft relationship. When defining the intra-modal structure constrain, we do not use the margin constant. The expression of our proposed intra-modal structure constraint for a video is as follows:

$$d\left(V_i, V_j\right) < d\left(V_i, V_k\right), \; if \; d\left(\widetilde{V_i}, \widetilde{V_j}\right) < d\left(\widetilde{V_i}, \widetilde{V_k}\right), \tag{3}$$

where V_i, V_j, V_k are the video embeddings in the joint embedding space from i-th, j-th and k-th video, respectively. $\widetilde{V_i}$, $\widetilde{V_j}$, $\widetilde{V_k}$ are the video features from i-th, j-th and k-th in the original video space. Analogously, given a text input, we set the intra-modal structure constraint as follows:

$$d\left(T_i, T_j\right) < d\left(T_i, T_k\right), \; if \; d\left(\widetilde{T_i}, \widetilde{T_j}\right) < d\left(\widetilde{T_i}, \widetilde{T_k}\right), \tag{4}$$

where T_i, T_j, T_k are the text embeddings in the joint embedding space from i-th, j-th and k-th text, respectively. $\widetilde{T_i}$, $\widetilde{T_j}$, $\widetilde{T_k}$ are the text features from i-th, j-th and k-th text in the original text space.

Dual Constraint Ranking Loss (DCRL): Here, we can propose a simple yet effective ranking loss by the combination of the inter-modal ranking constraint and the proposed intra-modal structure constraint.

Assume there are one batch of text-video pairs, we have N pairs of embedded features (V_i, T_i). Here, V_i and T_i are the feature embeddings for the video and text in the i-th text-video pair in the joint embedding space. In light of the inter-modal ranking constraint, two difference types of triplets (V_i, T_i, T_j) and (T_i, V_i, V_j) can be constructed, where $i \neq j$. For the intra-modal structure constraint, we adopt two difference types of triplets (V_i, V_j, V_k) and (T_i, T_j, T_k), where $i \neq j \neq k$. Taking all these triplets into consideration, the Dual Constraint Ranking Loss (DCRL) can be written as:

$$\begin{aligned} L = & \sum_{i \neq j} max\left(0, \; V_i^T T_j - V_i^T T_i + m\right) \\ & + \sum_{i \neq j} max\left(0, \; T_i^T V_j - T_i^T V_i + m\right) \\ & + \lambda \left[\sum_{i \neq j \neq k} C_{ijk}\left(V\right)\left(V_i^T V_j - V_i^T V_k\right) \right. \\ & \left. + \sum_{i \neq j \neq k} C_{ijk}\left(T\right)\left(T_i^T T_j - T_i^T T_k\right) \right], \end{aligned} \tag{5}$$

where, λ balance the impact of intra-modal structure constraint. The function $C\left(\cdot\right)$ in Eq. 5 can be written as::

$$C_{ijk}\left(x\right) = sign\left(x_i^T x_k - x_i^T x_j\right) - sign\left(\widetilde{x}_i^T \widetilde{x}_k - \widetilde{x}_i^T \widetilde{x}_j\right), \tag{6}$$

where x_i, x_j and x_k are the feature embeddings in the joint embedding space and $\widetilde{x}_i$, $\widetilde{x}_j$ and $\widetilde{x}_k$ are intra-modal features in the original space. As stated above, the intra-modal structure constraint is soft. Hence, we replace real distance values with the $sign$ function when introducing the intra-modal structure constrain Eq. 3 and Eq. 4 to the final loss funtion.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets. To show the effectiveness of the proposed method, we conduct experiments on EPIC-KITCHENS-100 dataset [4]. As access to the captions are required for both video-to-text and text-to-video retrieval, the Val set is used for evaluating this challenge to allow the held-out Test set for all other challenges to remain intact. We consider all the videos in Val, and all unique captions, removing repeats. Moreover, we also conduct experiments on MSR-VTT dataset and MSVD dataset in [8].

Evaluation Metrics. We uses two evaluation metrics: mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG) in the CVPR 2021 workshop of EPIC KITCHENS-100 Multi-Instance Retrieval Challenge. Mean Average Precision (mAP) has also been used for retrieval baselines [13, 5] as it allows for the full ranking to be evaluated. nDCG has been used previously for information retrieval [2, 6, 13]. It requires similarity scores between all items in the test set.

4.2. Implementation Details

Architecture details. We implement video embedding network and text embedding network as a 2 layer perceptron (fully connected layers) with ReLU. Additionally, the input vectors and output vectors are L2 normalised. In all cases, we set the dimension of the video embedding and text embedding to 256, a dimension we found to be suitable across all settings.

Training details. The embedding models were implemented in Python using the pytorch library. We trained the models with an Adam solver and a learning rate of 1e-5, considering batch sizes of 256. The training in general converges after a few thousand iterations, we report all results after 4000 iterations. We adopt the Adam optimizer for all our experiments and the margin of the inter-modal ranking loss is set to 0.3 and the hyper-parameter λ of the intra-modal structure loss is set to 0.1.

4.3. Performance Comparisons

We compare our proposed method with some existing state-of-the-art methods to verify the effectiveness.

- **MLP:** The MLP methods uses a 2-layer perceptron to project both modalities into a shared action space with a triplet loss.
- **JPoSE:** The JPoSE method disentangles captions into verb, noun and action spaces learned with a triplet loss.
- **DCRL(Our):** The DCRL method simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint to preserve both the cross-modal semantic similarity and the modality-specific consistency in the embedding space. (precise details in [8])

We can see that our proposed method performs best and consistently outperforms state-of-the-art methods in both text-to-video and video-to-text retrieval, which indicates that the DCRL plays an essential role and obtains a great performance in the Multi-Instance Retrieval Challenge.

5. Conclusion

In this report, we present a solution to the EPIC-KITCHENS-100 2021 Multi-Instance Retrieval Challenge. In this report, we elaborately design a novel Dual Constraint Ranking Loss (DCRL), which simultaneously considers the inter-modal ranking constraint and the intra-modal structure constraint to preserve both the cross-modal semantic similarity and the modality-specific consistency in the embedding space. This novel method allowed us to achieve the 1st place in the CVPR 2021 workshop of EPIC KITCHENS-100 Multi-Instance Retrieval Challenge. In our future work, we will explore domain adaptive cross-modal retrieval task.

References

[1] Samuel Albanie, Yang Liu, Arsha Nagrani, Antoine Miech, Ernesto Coto, Ivan Laptev, Rahul Sukthankar, Bernard Ghanem, Andrew Zisserman, Valentin Gabeur, Chen Sun, Karteek Alahari, Cordelia Schmid, Shizhe Chen, Yida Zhao, Qin Jin, Kaixu Cui, Hui Liu, Chen Wang, Yudong Jiang, and Xiaoshuai Hao. The end-of-end-to-end: A video understanding pentathlon challenge (2020). *CoRR*, abs/2008.00744, 2020. 1

[2] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 2010. 3

[3] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 1, 3

[5] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019. 1, 2, 3

[6] FilipRadlinski and NickCraswell. Comparingthesensitivity of information retrieval metrics. *ACM SIGIR*, 2010. 3

[7] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 1

[8] Xiaoshuai Hao, Yucan Zhou, Dayan Wu, Wanqian Zhang, Bo Li, and Weiping Wang. Multi-feature graph attention network for cross-modal video-text retrieval. In *Proceedings of the ACM on International Conference on Multimedia Retrieval*, 2021. 1, 3, 4

[9] Xiaoshuai Hao, Yucan Zhou, Dayan Wu, Wanqian Zhang, Bo Li, Weiping Wang, and Dan Meng. What matters: Attentive and relational feature aggreggation network for video-text retrieval. In *IEEE International Conference on Multimedia and Expo*, 2021. 1

[10] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *British Machine Vision Conference*, 2019. 1

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L Van Gool. Temporal segment networks: Towards good practices for deep action recognition. *ECCV*, 2016. 2

[12] Michael Wray, Gabriela Csurka, Diane Larlus, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, pages 450–459, 2019. 1, 2

[13] M. Wray, H. Doughty, and D. Damen. On semantic similarity in video retrieval. *CVPR*, 2021. 3