Multi-Attention Transformer for Naturalistic Driving Action Recognition

Xiaodong Dong*, Ruijie Zhao*, Hao Sun†, Dong Wu*, Jin Wang*
Xuyang Zhou*, Jiang Liu*, Shun Cui*, Zhongjiang He*
China Telecom Corporation Ltd. Data and AI Technology Company
{dongxd1, wud21, wangj75, zhouxy26, liuj67, cuis2, hezj}@chinatelecom.cn , 19120454@bjtu.edu.cn

Abstract

To detect the start time and end time of each action in an untrimmed video in the Track 3 of AI City Challenge, this paper proposes a powerful network architecture, Multi-Attention Transformer. The previous methods extract features by setting a fixed sliding window whitch means a fixed time interval, and predict the start and end times of the action. We believe that adopting a series of fixed windows will corrupt the video feature containing contextual information. So we present a Multi-Attention transformer module which combines the local window attention and global attention to fix this problem. The method equipped with features provided by VideoMAE achieved a score of 66.34. Then use the time correction module to improve the score to 67.23 on validation set A2. Finally, we have achieved third place on Track3 A2 dataset of the AI City Challenge 2023. Our code is available at: https://github.com/wolfworld6/Aicity2023-Track3.

1. Introduction

Distracted driving can be very dangerous. Today, developments in naturalistic driving research and computer vision technology provide much needed solutions to eliminate and reduce the occurrence of distracted driving behavior. Naturalistic driving studies are essential for studying driver behavior. They can help us capture driver behavior in traffic environments and analyze driver distractions while driving, which is one of the keys to reducing distracted driving. Track 3 of AI City Challenge offers video footage of drivers in the car, which covers three different perspectives and contains 16 different types of driver actions. For this track, participants were asked to implement an algorithm to label the various actions in the video and recognize when they started and when they ended.

This task can be considered as a temporal action localization (TAL) task in the field of video understanding. The videos in this task are usually long unedited videos, but the time interval of each individual action is relatively short. In temporal action localization algorithms, an intuitive idea is to pre-define a set of sliding windows of different time lengths and slide them over the video, such as S-CNN [17], TURN [6] and CBR [5]. Then, the action categories are judged one by one for the temporal intervals within each sliding window. Inspired by the two-stage target detection algorithm, the algorithms based on candidate temporal intervals first generate some candidate temporal intervals from the video that may contain actions, and then judge the action classes within each candidate temporal interval and correct the interval boundaries, such as R-C3D [19] and TAL-Net [3]. In addition, the idea of single-stage target detection can also be applied to temporal action localization, such as SSAD [12] and GTAN [15]. Currently, transformer models have shown remarkable performance in various fields of computer vision such as object detection [2], [21], [10], [9], image classification [4], [13], and video understanding [7], [14]. However, when using transformer model for long-duration videos, the increase in the number of video frames will lead to a significant increase in computation. Gedas Bertasius et al. [1] conducted extensive experiments and found a method of separable space-time attention, which opened the door for transformer models' application to long video understanding. Secondly, as the duration of different actions can vary greatly, it is challenging to extract the appropriate features by setting a fixed window and patch size in transformer models. Kai Han et al. [8] proposed the Transformer in Transformer (TNT) model, which fused the features of the outer patches and the inner patches, enriching feature information and improving feature expression.

Motivated by the aforementioned observations, we proposed a Multi-Attention transformer module, which is used to model not only the relationship between those different clip windows, but also the relationship within global windows. Besides, we design a Time Correction module to fuse

*These authors contributed equally to this work.
†Corresponding author, China Telecom Corporation Ltd. Data and AI Technology Company, sunh10@chinatelecom.cn

Model	Rearview	Dashboard	Right side
Uniformer-L	88.28	84.47	83.07
VideoMAE-L	**89.62**	**89.23**	**84.62**

Table 1. The classification results on the A1 validation. UniformerV2-L is trained for 35 epoch, while VideoMAE-L is trained for 30 epoch.

Model	Datasets	Feature
VideoMAE-L	Internvideo Hybraid	FL(hybrid)(1024)
VideoMAE-L	ego-4d	FL(ego)(1024)
VideoMAE-H	Kinetics-400	FH(k400)(1280)

Table 2. Public models pretrained on different datasets and feature dimensions extracted on A2 dataset.

and correct the prediction results with high confidence, and obtain a more accurate result.

2. Method

2.1. Data Preprocess

We detect the human body in the video and crop each frame. To ensure the stability of the video, we perform human body detection for each frame of the video and save the one with the largest detection area as the crop standard for the whole video, avoiding background shaking caused by different detection sizes in different frames. The crop operation retains the information related to the human body and removes the redundant information. On the one hand, it reduces the interference of other noise to the action features. On the other hand, it makes model learn human actions more easily.

2.2. Feature Extraction

We perform multiple experiments on different video representation models and three views of A1 videos. VideoMAE [18] is adopted for feature extraction on Rear and Dashboard views in this paper because of it's better performance as shown in Tab. 1. Public weights pretrained on different datasets are adopted and fine-tuned on A1 data. The weights used in this paper is shown in the Tab. 2. We finetune different models on the videos from Rear and Dashboard views respectively and extracted the features of A2 dataset.

2.3. Temporal Action Localization

Actionformer [20] combines multi-scale feature representation with local self-attention, and uses a lightweight decoder to classify every moment and estimate the corresponding action boundary. As shown in Fig. 1, on the basis of Actionformer, we propose a Multi-Attention transformer, which is applied to model not only the relationship between those different clip windows, but also the relationship within global windows.

Multi-Attention As shown in the right of Fig. 1, in the multi-scale channel transformer encoder, the feature $f1$ extracted from the video segment is input to layerNorm, multi-head attention and window attention, and then downsampled to obtain feature $\tilde{f}1$. The feature $\tilde{f}1$ is re-entered into the encoder, and the feature $\tilde{f}2$ is obtained after layerNorm, multi-head attention, window attention and downsampled. This operation is repeated N-1 times to obtain $\tilde{f}2$, $\tilde{f}3$, ...$\tilde{fN}$. After that, $\tilde{f}2$, $\tilde{f}3$, ...$\tilde{fN}$ are input into the multi-scale channel transformer decoder for decoding, and the category information and the corresponding time information of the actions are regressed through different fully connected layers.

In the multi-head attention module, feature fusion is performed between all the input features; since the input features are arranged according to the time segments of the video, the multi-head attention module will make use of the information in the time axis. In the window attention module, feature fusion is performed both for video segment features at adjacent locations and for all features, but the difference is that the feature fusion is performed only in the channel dimension.

Our model has N transformer layers with multi-scale to capture actions at different temporal scales. Each layer is composed of local multi-headed self-attention (MSA) and global multi-head self-attention(GMSA). To capture actions at different attention, the operation is formulated as:

$$\sum_{i=1}^{N} MSA_i + GMSA_i, \tag{1}$$

Where MSA_i refers to MSA in the i-th layer, $GMSA_i$ refers to GMSA in the i-th layer. A standard transformer block structure includes multi-head self-attention (MSA) and multi-layer perceptron (MLP). For efficiency, we employ a Multi-Attention module, contists of Clip Window Attention Module and Global Attention Module which can learn the information from different representation subspaces from different areas. Specifically, the embedding of each clip window channels is averaged, and the same number attention values are obtained by using a head-in-head transformer. The attentions will be multiplied or summed by the channels correspondingly. The module achieves feature enhancement through dimension-wise attention, which only increases a few parameters.

2.4. Time Correction

The output of Temporal Action Localization model contains a large number of predictions with low scores, and these results have a large range of overlapping temporal regions. The scoring criteria requires that each correct result is matched to only one prediction as far as possible and that the time range of that prediction is less different from the

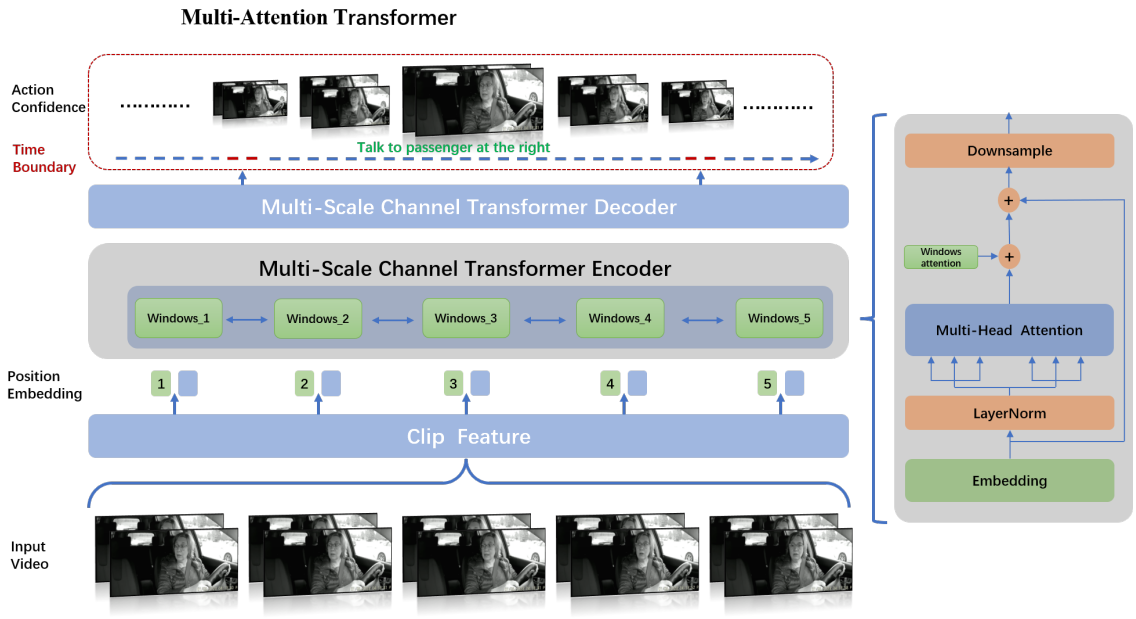

Figure 1. Overview of our model architecture. Our approach builds a Transformer-based model, using the action classification and to estimate action boundaries for each moment. In feature extraction stage, we extract a sequence of video clip features by VideoMAE, then embed each of these features. The embedded features will be encoded via window attention and global attention module. A candidate action is generated at each time step through using the classification head to predict the action category and the regression head to predict the boundaries of the action time boundaries.

correct one. This means that the large number of results need to be filtered and only the results with high confidence could be retained. Therefore, we design the time correction module to fuse and correct the prediction results with high confidence, and obtain the final results with more accurate time.

The time correction operation consists of 3 main steps, which are:

1. For all prediction results of each video-id, keep only the one with the highest score among the results of the same label, and discard the other items.

2. Perform step 1 separately for several different models and stitch the obtained results;

3. For the results obtained in step 2, fuse the results of same labels and same video-id according to the time Intersection over Union (tIoU); The specific fusion operations are:

(1) Remove all results whose time length is less than 1 second; remove all results whose time length is greater than 30 seconds;

(2) For all results with the same video-id and the same label, divide the results into different sets such that in each set, the tIoU of all time regions is greater than the set threshold;

(3) Remove the set whose length is equal to 1;

(4) For all the sets obtained in step (3), calculate the mean value of all the time points in each set; since all the results in each set have the same video-id and the same label, calculate the mean of all start times in the set as the start time for that video-id and that label; calculate the mean of all end times in the set as the end time for that video-id and that label. The start time and end time of the action in the i-th video-id and the j-th label can be calculated by the following formula:

$$
\begin{aligned}
{ts_i}^j &= \frac{1}{N}\sum_{p=1}^{N} start_p, \\
{te_i}^j &= \frac{1}{N}\sum_{p=1}^{N} end_p, \\
(start_p, end_p) &\in {S_i}^j.
\end{aligned}
\tag{2}
$$

Where ${ts_i}^j$ refers to the start time of the action in the i-th video-id and the j-th label, ${te_i}^j$ refers to the end time of the action in the i-th video-id and the j-th label. ${S_i}^j$ denotes the set of predictions where video-id is i and label is j. N is the length of ${S_i}^j$. $start_p$ refers to the start time of the p-th predictions in ${S_i}^j$, and end_p refers to the end time of

the p-th predictions in ${S_i}^j$.

Besides, when fusing the results of the same video-id and the same label, in addition to the method of taking the mean of all time points as described in the fourth step, we also try another method: weighting the fusion of time nodes according to their scores, which is formulated as:

$$
\begin{aligned}
{ts_i}^j &= \frac{\sum_{p=1}^{N} start_p * score_p}{\sum_{p=1}^{N} score_p}, \\
{te_i}^j &= \frac{\sum_{p=1}^{N} end_p * score_p}{\sum_{p=1}^{N} end_p}, \\
(start_p, end_p, score_p) &\in {S_i}^j.
\end{aligned} \tag{3}
$$

where $score_p$ refers to the score of the p-th predictions in ${S_i}^j$.

3. Experiment

3.1. Training

3.1.1 Feature Extraction Models

Both VideoMAE-L and VideoMAE-H are fine-tuned on A1 dataset with training crop size 224. The initial earning rate is 2e-3. The number of frames is 16, sampling rate is 4. The experiment is executed with batch size 2 on 8 Nvidia V100 GPU. For VideoMAE-L is trained for 35 epoch, while VideoMAE-H is trained for 40 epoch.

3.1.2 Temporal Action Localization Models

We conduct experiments on A1 dataset, dividing the data into training set and test set with the ratio of 7:3. After the experiment, we use all A1 as the training set. In the first place, we feed 32 consecutive frames as the input to pre-trained model UniFormerV2 [11], use a sliding window with stride 16 and extracted 3072-D features which cover three parts (left, center and right) from the original video to optimize the network structure. mAP@[0.1:0.5:5] is used to evaluate our model. In the second place, we also employ some ablation experiments of the features extracted 1024-D features. Our TAL model is trained for 40 epochs with a linear warmup of 5 epochs. The initial learning rate is 1e-3 and a cosine learning rate decay is used, and a weight decay of 5e-2 is used. In the end, we use the pre-trained VideoMAE to extract 1028-D features and 1024-D features. The method can be adapted to different features, and the performance of the model on multiple views, single view and multiple features can be tested.

3.2. Results

3.2.1 Results on Data Preprocess

Regarding whether the crop operation enhances the effect of the model, ablation experiments are applied to verify it.

Data Preprocess	**mAP@tIOU**
no crop	83.67
crop	**87.79**

Table 3. The results on the crop operation.

Task	Method	**mAP@tIOU**					
		0.1	0.2	0.3	0.4	0.5	Avg
Action	Actionformer	63.00	61.15	58.33	52.83	47.18	56.50
	Ours	73.03	71.13	64.77	68.53	59.02	67.33
	Ours(fpn+8h+w13+r2.5)	76.07	73.71	71.95	68.22	63.48	70.69

Table 4. The results on the A1 validation.

Task	Method	**mAP@tIOU**					
		0.1	0.2	0.3	0.4	0.5	Avg
Action	Center	63.88	61.99	58.77	51.08	41.93	55.53
	Right	62.89	61.20	57.45	53.69	44.26	55.90
	L+R+C	67.52	65.29	62.03	58.62	53.69	61.43
	Resize	74.34	71.54	67.66	62.98	58.51	67.01

Table 5. The results on the different screen.

The experimental results show that the crop operation does play a role. As shown in Tab. 3, using the Multi-Attention model with FH(k400)(1280) feature, the crop operation can raise the map from 83.67 to 87.79 on Rear and Dashboard view.

3.2.2 Results on Multi-views Model

We start with our experiments and get results on A1, extracting 3072-D features of all the views videos with pre-trained model UniFormerV2 [11], we use the validation set for training. The hyper-parameters of window size is 9, the mini-batch size is 4, the max segments number is set 2304. Tab. 4 summarizes the results, our method achieves mAP of 67.33% on average, and mAP of 59.02% at tIoU=0.5. The model performance is improved with the combination of a simple design and a strong multi-attention transformer model. In addition, we experiment with various hyper-parameters, including utilizing an FPN architecture with 8 heads, increasing the window size to 13, and setting the center sample radius to 2.5. As a result, the mAP improves from 67.33% to 70.69%.

Considering the different areas contain different information on the target of the screen, we conduct as much experiments as possible for a fair comparison to the features with different crop parts and different resize mode of the videos. The results are shown in Tab. 5. Compared with the results(mAP 67.33%) in Tab. 4, there is not much difference, so we remove the crop pattern in the follow-up experiments.

3.2.3 Results on Rear View Model

It is found through experiments that accuracy of classification model is higher using Rear views, as shown in

Method	view	pre-trained	mAP@tIOU					
			0.1	0.2	0.3	0.4	0.5	mean
VideoMAE	Rear	Kinetics-400	97.16	96.95	96.67	96.26	95.26	96.46
		Ego-4D	93.74	90.72	88.09	84.14	79.10	87.16
		Hybraid	93.73	92.27	90.11	88.12	80.70	88.99
	Rear + Dashboard	Kinetics-400	91.81	90.03	87.62	84.31	79.98	86.75
		Ego-4D	92.61	91.46	89.05	84.09	78.63	87.17
		Hybraid	93.10	91.30	89.33	85.56	81.43	88.14

Table 6. The results on the different features of Rear or Dashboard view.

Model combinations	Average overlap score
M1	0.6382
M1+M2+M3(No Removing)	0.6325
M1+M2+M3	0.6482
M4+M5+M6(Weighting)	0.6514
M4+M5+M6	0.6593
M7+M8+M9	0.6634
M7+M8+M9+M10	**0.6723**

Table 7. The results on Time Correction Module. The model combination $M7 + M8 + M9 + M10$ is the final prediction obtained by our team.

Tab. 1. We test the performance of the model in Rear and Dashboard view features extracted with different backbone model from VideoMAE. As shown in Tab. 6, different features have variable performance on TAL.

3.2.4 Results on Time Correction Module

The experimental results on Time Correction Module are shown in Tab. 7. And the corresponding specific information for The $M1$ to $M10$ models is shown in Tab. 8. Due to the limited number of evaluations provided by the system, we cannot traverse all the methods for each model combination to obtain the optimal evaluation results. Therefore, we use the optimal processing method directly for the better model after summarizing the pattern in each comparison experiment.

We introduce the Time Correction Module in Sec. 2.4. In the second step, the results of several different models are stitched together, which resulting in higher results compared with using a single model. The average overlap score of the single model $M1$ is 0.6382, but after stitching the results of the three models and adopting time correction, the performance of evaluation result improves into 0.6482. Besides, the stitching result of the four models $M7 + M8 + M9 + M10$ improves the evaluation score from 0.6634 to 0.6723 compared to the stitching results of the three models $M7 + M8 + M9$.

In the third step, the set of length 1 is removed. This operation also boosts the results. Compared to the results of the time correction module without this operation, the evaluation score improves from 0.6325 to 0.6482 after adding this operation. The statistics of action durations are conducted for the A1 dataset, and it is found that the durations are all in the range of 1-30 seconds, so we constrain the duration range of the prediction results. Besides, when fusing the results of the same video-id and the same label, we try two methods: taking the mean of all time points and weighting the fusion of time nodes according to their scores. However, the experimental results show that using the mean value is more satisfactory. Using the same combination of models $M4 + M5 + M6$, the weighted and mean methods yields 0.6514 and 0.6593, respectively.

The overall ranking and score of the track is shown in Tab. 9. The results of model combination $M7 + M8 + M9 + M10$ shown in Tab. 7 is the final prediction obtained by our team, with an average overlap score of 0.6723 and a ranking of third on public leaderboard. In this combination, $M7$, $M8$ and $M9$ represent the temporal action localization models trained with features $FL(hybrid)(1024)$, $FL(ego)(1024)$, and $FH(k400)(1280)$ in Tab. 2, respectively. And $M10$ represents the Tridet [16] model. It is no doubt that the results of our method on Track 3 of AI City Challenge can be further improved by combining a more powerful backbone of video features with object detection results.

4. Conclusion

In this paper, we present a Multi-Attention transformer based method for temporal action localization. The power of method lies in our design choices, especially combining features with the method of Multi-Attention module to model longer-range temporal context in videos. In addition, we conduct extensive experiments to compare with different video views, different feature extraction networks, different pre-trained datasets, so as to find the views and networks with better feature representation capability. Besides, we also propose a Time Correction module to improve temporal accuracy.

Name	Feature	View	Model
M1	FL(hybrid)(1024)+FL(ego)(1024)+FH(k400)(1280)	Rear+Dashboard	actionformer
M2	FL(hybrid)(1024)+FL(ego)(1024)+FH(k400)(1280)	Rear+Dashboard	actionformer
M3	FH(k400)(1280)	Rear	actionformer
M4	FH(k400)(1280)	Rear	actionformer
M5	FL(ego)(1024)	Rear	actionformer
M6	FL(ego)(1024)	Rear	actionformer
M7	FH(k400)(1280)	Rear+Dashboard	Multi-Attention
M8	FL(ego)(1024)	Rear+Dashboard	actionformer
M9	FL(hybrid)(1024)	Rear+Dashboard	actionformer
M10	FH(k400)(1280)	Rear	Tridet

Table 8. The specific information of different models mentioned in Tab. 7.

Rank	Team name	Average overlap score
1	Meituan-IoTCV	0.7416
2	JNU_boat	0.7041
3	**ctc-AI**	**0.6723**
4	RW	0.6245
5	Purdue Digital Twin Lab	0.5921
6	BUPT-MCPRL	0.5907
7	DiveDeeper	0.5881
8	INTELLI_LAB	0.5426
9	AILAB	0.5424
10	AIMIZ	0.5409

Table 9. Public ranking and score on Track3.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020. 1

[3] Y. W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 1

[4] 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. *ArXiv*, abs/2010.11929, 2021. 1

[5] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. 2017. 1

[6] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. *IEEE*, 2017. 1

[7] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2019. 1

[8] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 1

[9] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, Wenxuan Liu, Yunhai Tong, Lizhuang Ma, and Liqing Zhang. End-to-end video object detection with spatial-temporal transformers. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 1

[10] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim. Hotr: End-to-end human-object interaction detection with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 74–83, 2021. 1

[11] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 4

[12] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. *ACM*, pages 988–996, 2017. 1

[13] 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. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 1

[14] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *ArXiv*, abs/2106.13230, 2021. 1

[15] F. Long, T. Yao, Z. Qiu, X. Tian, and T. Mei. Gaussian temporal awareness networks for action localization. *IEEE*, 2019. 1

[16] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. *arXiv preprint arXiv:2303.07347*, 2023. 5

[17] Z. Shou, D. Wang, and S. F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[18] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 2

[19] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. *IEEE Computer Society*, 2017. 1

[20] Chen-Lin Zhang, Jian Zhai Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *ArXiv*, abs/2202.07925, 2022. 2

[21] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ArXiv*, abs/2010.04159, 2021. 1