Stargazer: A Transformer-based Driver Action Detection System for Intelligent Transportation

Junwei Liang[1] He Zhu[2]* Enwei Zhang[1] Jun Zhang[1]
[1]Tencent Youtu Lab [2]Tsinghua University
junweiliang1114@gmail.com, zhuh20@mails.tsinghua.edu.cn
{miyozhang,bobbyjzhang}@tencent.com

Abstract

Distracted driver actions can be dangerous and cause severe accidents. Thus, it is important to detect and eliminate distracted driving behaviors on the road to save lives. To this end, we study driver action detection using videos captured inside the vehicle. We propose Stargazer, an efficient, transformer-based system exploiting rich temporal features about the human behavioral information, with a simple yet effective action temporal localization framework. The core of our system contains an improved version of the multi-scale vision transformer network, which learns a hierarchy of robust representations. We then use a sliding-window classification strategy to facilitate temporal localization of actions-of-interest. The proposed system wins the ***second place*** *in the Naturalistic Driving Action Recognition of AI City Challenge 2022 (Track 3)*[1]*. The code and models are released*[2].

1. Introduction

With the rapid growth of traffic flows on the road, traffic accidents have claimed thousands of lives each year in the US alone. Distracted driving is reported to be an important cause among those deaths. Identifying and eliminating these driver behaviors is important and can help save lives on the road. With the advancement in deep learning and computer vision, systems now are able to analyze an unprecedented amount of rich visual information from videos. An important analysis is action detection in untrimmed/extended videos, to enable applications such as distracted behavior recognition and accident avoidance. This problem has received increasing attention in the computer vision community [4, 7, 8, 11, 20, 23]. It is regarded as an essential building block in video understanding for many applications like self-driving cars [22, 24, 25], socially-aware robots, public safety monitoring [21, 28, 30], *etc*.

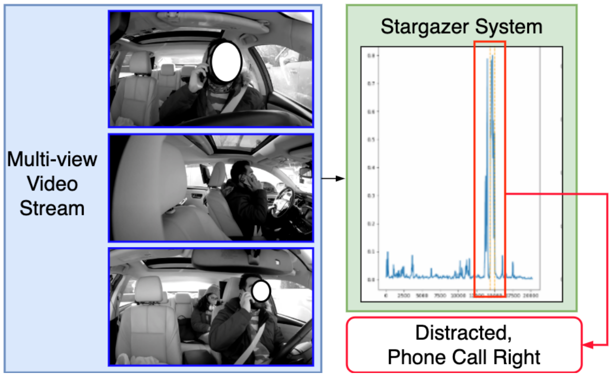

Figure 1. The Stargazer system for naturalistic driving action detection. Our system can take in multi-view or single-view video stream and report detected actions' start and end time.

Human vision can recognize video actions efficiently despite the variations of scenes and domains. Convolutional neural networks (CNNs) [5, 13, 33, 35, 36] fully utilize the power of modern computational devices and employ spatial-temporal filters to recognize actions, which outperforms traditional models such as oriented filtering in space time (HOG3D) [18]. However, due to the high variations in space-time, the state of the art of action recognition is still far from being satisfactory, compared with the success of 2D CNNs in image recognition [15]. Recently, vision transformers like ViT [10], MViT [11] that are based on the self-attention [38] mechanism are proposed to tackle the problems of image and video recognition. Instead of modeling pixels like CNNs, transformers apply attentions on top of visual tokens. The inductive bias of translation invariant in CNNs make it require less training data than pure-attention-transformers in general. However, transformer has the advantage that it could better harness the parallel processing

*Work done during a research internship at Tencent Youtu Lab.

[1] https://www.aicitychallenge.org/2022-data-and-evaluation/

[2] https://github.com/JunweiLiang/aicity_action

units of modern computing devices like GPUs and TPUs, making it more computationally efficient than CNNs. We have seen a rapid growth in image and video datasets [17] in recent years, which would make up for the shortcomings of data-hungry transformers. Meanwhile, transformers combined with low-level convolutional operations have been proposed [11] to further improve the original design.

The AI City Challenge has been a prominent annual competitions [31, 32, 44] that addresses key intelligent city problems. For the naturalistic driver action detection problem, we are given single-view or multi-view videos from inside the vehicle. There are a few key challenges:

- Models need to recognize different actions from different viewpoints.
- Different action's start and end time could be hard to pin-point.
- The system need to be efficient to be practical and deployable within the vehicle's on-board device.

To this end, we propose a transformer-based action detection system, termed *Stargazer*, which includes an improved version of the multi-scale vision transformers [11] for sliding-window action classification. As action temporal localization is challenging, we introduce two new techniques to address the issue. First, to address the problem of pin-pointing actions within the videos, we design an efficient vision transformer model that takes in a short video clip of 2 seconds and a sliding-window classification technique with a stride of about half a second to allow the system to accurately localize the start and end of the actions. Second, to facilitate the training of data-hungry transformers, we utilize pre-training of our models on one or multiple large-scale video action datasets and multi-crop data augmentation on the target dataset.

2. Related Work

CNNs and Vision Transformers. CNNs work as the standard backbones throughout computer vison tasks for image and video. Various effective convolutional neural architectures have been raised to improve the precision and efficiency (e.g. VGG [34], ResNet [15] and DenseNet [16]). Although CNNs are still the primary models for computer vision, the Vision Transformers gradually show their enormous potential. Vision Transformer (ViT [10]) directly applies the architecture of Transformer on image classification and get encouraging performance. ViT and its variants (e.g., MViT [11]) achieve outstanding results in recent years.

Action Recognition/Classification. The research of action recognition has advanced with both new datasets and new models. The modern benchmarks for action recognition is the Kinetics dataset [17]. The Kinetics dataset proposes a bigger benchmark with more categories and more videos (e.g., 400 categories 160,000 clips in [17]) as a harder benchmark. However, Kinetics datasets do not exhaust all the possible actions in all possible scales, for example, surveillance actions are missing in the two datasets. Many new approaches [12, 26, 37, 40, 43] have been carried on these datasets, of which the SlowFast network [12] and MViT [11] obtain good performance. We can see the trend of action recognition in the last two decades is to collect bigger datasets (e.g. Kinetics) as well as build bigger models (e.g., I3D [5] and SlowFast).

Temporal Action Localization. Temporal Action Localization (TAL) is a kind of task to locate the action instances and identify their categories. The architectures of TAL include two-stage and single-stage models. The two-stage approaches [1, 3, 6, 14] for TAL first split the video to many candidate segments as action proposals, and then classify these proposals into the corresponding action categories. Single-stage TAL [2, 27, 29, 41] methods aim to localize actions and get category in one stage without action proposals. Most of these work is to adopt the sliding anchor windows, which is called anchor-based.

3. Approach

3.1. System Architecture

Fig. 2 shows the overall network architecture of our *Stargazer* system. Our design is simple yet effective. *Stargazer* has the following key components:

Action Proposal module extracts a constant number of frames as inputs to the action recognition module. To ensure more precise action temporal localization results, we extract cube proposals with temporally overlapping frames. For the naturalistic driver action detection task, as the videos contain mostly the drivers, we directly take the whole frame as the spatial dimensions of the proposals. One can include object detection and tracking in this module to apply the system in a more complex situation where the action-of-interest region is small compare to the whole frame.

Action Recognition module takes the proposal frames (i.e., video clips) and classify the video clips into one of the distracted or normal actions. The action recognition module is based on the improved multi-scale vision transformers [11, 19].

Post Processing module aggregates the overlapping video clip scores and produces the final action outputs with start and end timestamps. As each action in this challenge is accompanied with multiple videos from different viewpoints, a simple heuristic logic (the cross-view action instance selection method in Fig. 2) is utilized to select the best action instance for each class as the final outputs.

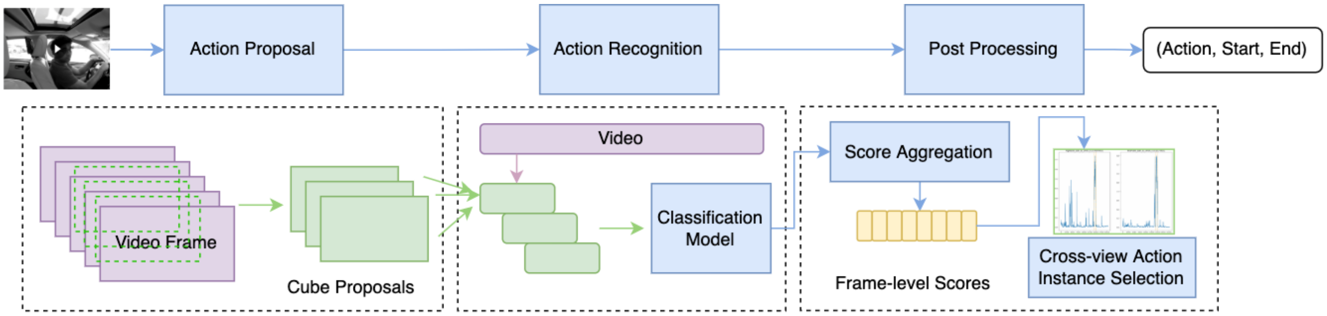

Figure 2. The inference pipeline of our Stargazer system for the AI City Challenge. Given a sequence of frames, our action proposal module produces cube proposals that are input to the action recognition module to get classification scores. The post processing module aggregates the scores from multiple-view videos and produces the final submission results.

3.2. Action Proposal Module

Given an extended or untrimmed videos, we first extract action proposals where the actions might appear. We utilize the *spatial-temporal cube* as the action proposal, which is defined as:

$$p_i = (x_0^i, x_1^i, y_0^i, y_1^i, t_0^i, t_1^i) \tag{1}$$

This six-tuple design has been proven to be effective compared to tube proposals in spatial-temporal action detection task [28]. Since in this challenge the drivers take up the main content of the frame, we set the spatial region of the proposals to be the size of the frame. In other more complex data where cameras are away from the action-of-interest, a object detection and tracking module can be added to produce more precise spatial regions.

The temporal length of the proposal is decided by the input length of the action recognition model. In this challenge we have tried both 16x4 and 32x3 settings (number of frame x frame stride). The 16x4 setting means the total proposal length is of 64 frames, which corresponds to about 2 seconds of videos of 30 FPS. In order to get a finer temporal resolution of the prediction scores, we use a overlapping-sliding window technique to generate the proposals, with a stride of 16 frames, for example, which corresponds to a temporal resolution of about 0.5 seconds. In terms of implementation, we utilize PyTorch's multi-process Dataloader[3] and the Decord package[4] for efficient video frame decoding with CPUs (in parallel with the action recognition module, which mostly using GPUs).

3.3. Action Recognition Module

Our key component is the action recognition module, which takes the action proposal frames as input and produces per-proposal action classification scores. Our model is based on the improved multi-scale vision transformers (MViT v2) [11, 19], which learn a hierarchy from dense (in space) and simple (in channels) to coarse and complex features. Fig. 3 shows the detailed architecture of the model (using 16x4 and 224x224 spatial-temporal proposal input as an example). MViT v2 first utilizes 3D convolution as the Patch Embedding to produce visual tokens, and then they are added with separate spatial and temporal positional embedding before input to the self-attention block computation. Each self-attention block consists of a multi-head pooling attention layer (MHPA) and a multi-layer perceptron (MLP), and the residual connections are built in each layer. The feature of each self-attention block is computed by:

$$\begin{aligned} X_1 &= \text{MHPA}(LN(X)) + X \\ Block(X) &= \text{MLP}(LN(X_1)) + X_1 \end{aligned} \tag{2}$$

where X is the input to each block. Multiple self-attention blocks are group into stages. The channel dimensions are expanded before the multi-head attention computation as in [19] at the start of each resolution stage, while the spatial dimensions are reduced through 3D convolutions and max pooling, as shown in Fig. 3. As the features go through each stage of the model, the spatial and temporal dimensions of the features are reduced while the channel dimension is increased. Finally, the spatial and temporal features are averaged before inputting to the classification layer. For more details of the backbone architecture, please refer to the released code and original papers [11, 19].

3.4. Post Processing Module

As shown in Fig. 2, the action recognition module produces a classification score for each action proposal, which is temporally overlapped with each other. We aggregate the scores by averaging all the scores of each frame position. The finest temporal resolution of the frame-level scores are decided by the stride of the action proposals. In most of our experiments, we use 64-frame proposals with a 16-frame stride, which means the frame-level scores would be changed about every 0.5 seconds. In the multi-view action detection scenario, multiple videos are provided for

[3] https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader
[4] https://github.com/dmlc/decord

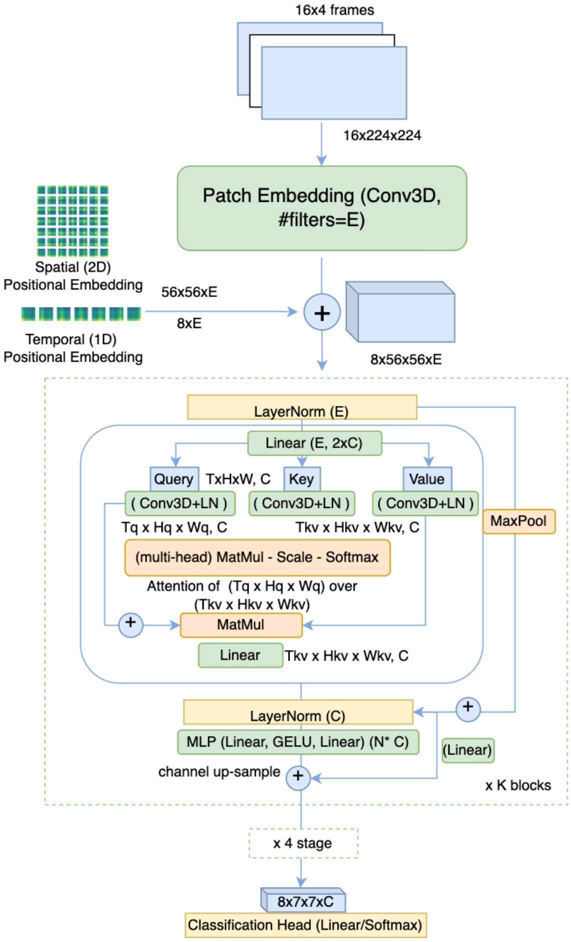

Figure 3. Improved multi-scale vision transformers.

the same testing instances, i.e., we have multiple lists of frame-level scores to produce the final action instance output. We adopt the following cross-view action instance selection strategy: For each action class, given the frame-level scores from multiple videos of different views, we first extract continuous frames as action candidates with scores larger than a threshold within each video. For each video, the action candidate with the highest averaged scores is selected. For cross-view selection, we select the action candidate with most number of frames for each action class. Finally, the selected action candidate is output by the system after a rounding operation on the start and end times.

3.5. Multi-crop Data Augmentation Training

As mentioned in the action recognition module, our input to the model is a fixed-length video clip while our action targets are of variable length. To better facilitate training, we first convert the original variable length annotations of the training set into a list of video segments with action class IDs. The empty segments (the video clip without any annotations) are considered as the class 0 (normal driving). During training, for each action sample, a random 16x4 (or 32x3) clip is sampled temporally within the positive segments, as well as randomly spatially cropped, to combine into a mini-batch. Note that we do not use random flipping and Rand Aug [9] as in the original MViT [11] paper since this challenge distinguishes actions of left and right. The process is shown in Fig. 4. Through this training process, although the original annotations are of variable length and our backbone model takes in fixed-length video clips, the model can be trained with all possible clips within the training set.

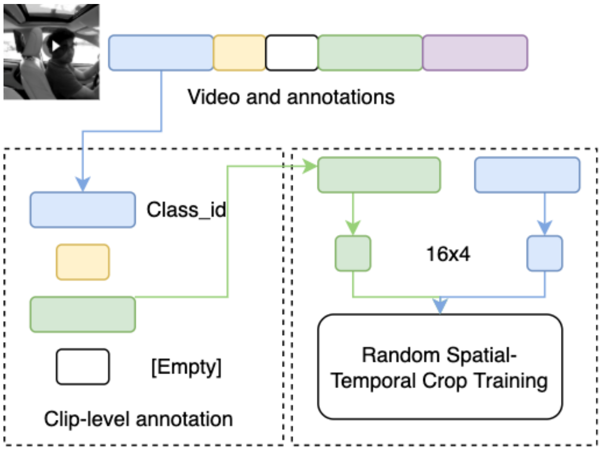

Figure 4. Multi-crop data augmentation training.

4. Experiments

We evaluate the proposed *Stargazer* system on the Naturalistic Driver Action Recognition of AI City Challenge dataset [32]. We demonstrate that our model performs favorably against other systems on this challenging task.

Dataset. The whole dataset contains 90 video clips captured from 15 drivers. The length of each video is about 10 minutes, and it is about 14 hours in total. These drivers perform every one of the 18 different distracted actions once in random order in each video. There are three synchronous cameras recording from different angles mounted in the car. Each driver is recorded twice because of performing two kinds of tasks: one is performing without appearance block, and another is performing with some appearance blocks (e.g., sunglasses, hat). Therefore, record each driver twice to collect 6 videos, 3 videos in sync without appearance block and 3 videos in sync with some appearance block.

Evaluation metrics. The evaluation metrics are measured by the F1-score. An activity is correctly identified when its starting time and ending time both are within one second of the ground truth. In order to compute the F1-score, an identified activity is true-positive (TP) when it is cor-

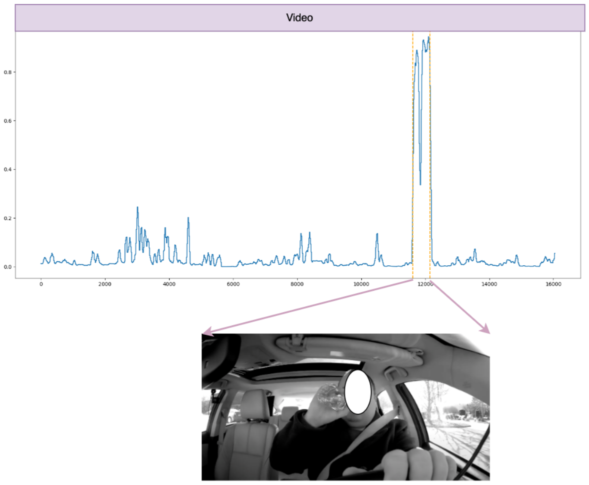

Figure 5. Example of the single video prediction scores and threshold selection.

rectly identified. And correspondingly, an identified activity is false-positive (FP) when it is not a TP activity. When a ground-truth activity is not correctly identified, it is considered a false-negative (FN).

Implementation Details. We initialize our backbone MViT v2 model with pre-trained model weights from Kinetics-700 dataset [17]. The pre-trained MViT v2 models has a Top-1/5 accuracy of 71.91/90.52 and 74.08/91.87, respectively for 16x4 and 32x3 models, both with a 224x224 spatial crop inputs. We first down-sample all the videos to 540p resolution for faster training and testing. In terms of annotation conversion as mentioned in Section 3.5, we have experimented with or without the empty video segments as the normal driving class (see also Fig. 4), and found that training without the empty video segments is slightly better, but the grid-searched thresholds with models trained using the empty video segments are better across all other models (even used for models that are not trained in this way). We train all our models on the AI City training set using a learning rate of 0.0001 for 200 epochs, with a warm-up period of 30 epochs and a cosine decay. We utilize the adamw optimizer with a batch size of 128 or 32 (for 32x3 models). For the initialization of the larger 448x448 spatial crop models, we convert the positional embedding of the pre-trained model weights to the proper resolution, as done in previous works [11]. For the proposal generation, we use a stride of 16 frames for the 64-frame long model. To select the final post-processing threshold for each action class, we either use the grid-searched weights from split 1 or manually select them by looking at the prediction visualization with the videos on the validation set, as shown in Fig. 5.

Rank	Team ID	F1 Score
1	VTCC - UTVM	0.3492
2	**Ours**	**0.3295**
3	CybercoreAI	0.3248
4	OPPilot	0.3154
5	SIS Lab	0.2921

Table 1. Comparison to other submissions methods on the A2 validation set. Results are from the official leaderboard of the competition. Numbers denote F1 scores thus higher are better. Our system has won the runner-up place.

4.1. Main Results

Table 1 lists the top-5 leaderboard results of our system and other submissions. Our system has won the second place. The leaderboard numbers are from the validation set (A2 videos). Our best submission uses the 16x4 MViT v2 model with a spatial crop of 448x448. The annotations are converted without the empty segments and the models are trained for 200 epochs, which takes 2-3 hours with 2x8 A100 GPUs. In terms of data augmentation, as mentioned in Section 3.5, only random spatial and temporal jittering are used. The final action thresholds are selected empirically through looking the videos and the prediction scores as described in Section 3.4.

4.2. Ablation Studies

Training / Val Data	Top-1 Err	Top-5 Err
Split - 1	67.86	35.71
Split - 2	45.83	16.67
Split - 3	63.39	29.46
Split - 4	76.79	41.07
Split - 5	54.17	17.50

Table 2. Comparison of different data splits. See text for details.

We conduct most of our ablation experiments locally with the official training set. As we mention before, the official training set consists of performances from 5 different drivers. Hence, we split the official training set into training and validation based on the identity of the drivers. We answer the following questions:

Which local splits are better? We train and validate our baseline model on the 5 splits of the official training set as shown in Table 2. We train on 4 drivers' video data and test on the other one. The reported numbers are top-1/top-5 error rates on the validation set. For validation, a single center clip of each annotated segment is used for fast evaluation. This baseline model is the 16x4 model with 224x224 crop trained with adamw optimizer and a learning rate of 0.001. As we see, the results vary a lot across different splits as the data within a split is too small. By comparing to the results on the leaderboard, we conclude that split 1 is better correlated with the A2 dataset hence for the rest experiments we only use split 1 data.

Run Name	Top-1 Err	Top-5 Err
Baseline (lr = 0.001; 224)	67.86	35.71
+ MixUp	66.96	38.39
+ lr = 0.0001	61.61	32.14
+ lr = 0.005	81.25	58.04
+ 100 epochs	67.86	37.50
+ 448 crop	52.68	23.21
+ 32x3 Model	53.57	32.14
+ 32x3 Model, 448 crop	47.66	17.97

Table 3. Comparison of different hyper-parameters. See text for details.

Does stronger data augmentation method help? In Table 3, we utilize Mix Up [42] and find that it does not significantly improve the results.
Does larger model help? We experiment with models with larger spatial crop (from 224 to 448) and longer video clip input (32x3), and the results show that larger models improve the accuracy significantly.
Other hyper-parameters. We also compare the baseline model with using other hyper-parameters. We find that shorter training schedule (100 epochs vs. 200 epochs) hurts performance and a learning rate of 0.0001 is optimal.

4.3. Post Processing Experiments

Run Name	F1 Score	Precision	Recall
Ours - Leaderboard	0.3295	0.3184	0.3413
+ A2 grid search	0.3333	0.3240	0.3432
+ A1 val threshold	0.2767	0.2682	0.2857
+ 32x3 Model	0.2407	0.2626	0.2814
- rounding function	0.2139	0.2087	0.2216
DanTAD [39]	0.0446	0.0447	0.0444

Table 4. Comparison of different post-processing strategies on the leaderboard. We also compare our method with a strong method in temporal action localization. See text for details.

In this section, we present our experience with the post-processing techniques and compare them on the leaderboard, as shown in Table 4. If the round function on the start and end time is removed (mentioned in Section 3.4), the performance drop significantly. We utilize the thresholds from a different model trained on split 1 of the official training set. The action score thresholds for each class are grid-searched on the split 1 validation set and applied on the full-dataset trained model. We obtain a 0.2767 F1 for this run. Under the same setting, we also submit a larger model with 32x3 and 448 crop inputs. It is surprising that the performance does not reflect what we observe in the ablation experiments (Table 3). Finally, given a sub-set of the A2 videos, we manually label the videos, and apply grid-search on the annotations to get the best action score thresholds and submit to the **general** leaderboard, which returns better results than our official scores. The large variance of performance given the same model with different post-processing techniques may suggest that the evaluation metric of the temporal localization part is too strict. A temporal Intersection-over-Union method might be more suitable. We also compare our method with a recent strong model in temporal action localization, namely DaoTAD [39], on this dataset. DaoTAD performs well on THUMOS'14. As shown in Table. 4, DaoTAD performs much worse than our method, which suggests that more research is needed to make those models work well on this dataset.

5. Conclusion

In this paper, we have presented a new transformer-based system for action detection. We refer to the resulting system as *Stargazer*. We showed the efficacy of our model on the Naturalistic Driver Action Recognition of AI City Challenge 2022.

References

[1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 2

[2] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *Procedings of the British Machine Vision Conference 2017*. British Machine Vision Association, 2019. 2

[3] 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

[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. 1

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017. 1, 2

[6] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Jiashi Feng. Augmented transformer with adaptive

graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024*, 2021. 2

[7] Xiaojun Chang, Wenhe Liu, Po-Yao Huang, Changlin Li, Fengda Zhu, Mingfei Han, Mingjie Li, Mengyuan Ma, Siyi Hu, Guoliang Kang, Junwei Liang, et al. Mmvg-inf-etrol@ trecvid 2019: Activities in extended video. 2019. 1

[8] Jia Chen, Jiang Liu, Junwei Liang, Ting-Yao Hu, Wei Ke, Wayner Barrios, Dong Huang, and Alexander G Hauptmann. Minding the gaps in a video action analysis pipeline. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 41–46. IEEE, 2019. 1

[9] 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. 4

[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. In *International Conference on Learning Representations*, 2020. 1, 2

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 1, 2, 3, 4, 5

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2

[13] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NeurIPS*, 2016. 1

[14] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[17] 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, 5

[18] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In Mark Everingham, Chris J. Needham, and Roberto Fraile, editors, *BMVC*, 2008. 1

[19] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021. 2, 3

[20] Junwei Liang, Liangliang Cao, Xuehan Xiong, Ting Yu, and Alexander Hauptmann. Spatial-temporal alignment network for action recognition and detection. *arXiv preprint arXiv:2012.02426*, 2020. 1

[21] Junwei Liang, Desai Fan, Han Lu, Poyao Huang, Jia Chen, Lu Jiang, and Alexander Hauptmann. An event reconstruction tool for conflict monitoring using social media. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1

[22] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from simulation for trajectory prediction. 2020. 1

[23] Junwei Liang, Lu Jiang, Deyu Meng, and Alexander G Hauptmann. Learning to detect concepts from webly-labeled video data. In *IJCAI*, 2016. 1

[24] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *CVPR*, 2020. 1

[25] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019. 1

[26] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[27] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017. 2

[28] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Yijun Qian, Junwei Liang, Liangke Gui, Jing

Wen, and Peng Chen. Argus: Efficient activity detection system for extended video analysis. In *WACVW*, 2020. 1, 3

[29] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019. 2

[30] Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *ICRA*, 2010. 1

[31] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E Lopez, et al. The 5th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4263–4273, 2021. 2

[32] Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Jiyang Wang, Senem Velipasalar Gursoy, David Anastasiu, and Shuo Wang. Synthetic distracted driving (syndd1) dataset for analyzing distracted behaviors and various gaze zones of a driver. *arXiv preprint arXiv:2204.08096*, 2022. 2, 4

[33] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017. 1

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[35] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV*, 2010. 1

[36] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1

[39] Chenhao Wang, Hongxiang Cai, Yuxin Zou, and Yichao Xiong. Rgb stream is enough for temporal action detection. *arXiv preprint arXiv:2107.04362*, 2021. 6

[40] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, 2020. 2

[41] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020. 2

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 6

[43] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *NeurIPS*, 2018. 2

[44] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. Vehiclenet: Learning robust visual representation for vehicle re-identification. *IEEE Transactions on Multimedia*, 23:2683–2693, 2020. 2