Scaling Open-Vocabulary Action Detection

Supplementary Material

In this supplementary we provide the following additional technical details which were not included in the main paper:

- Architecture
- Configuration for Training and Evaluation
- Weak-supervision Implementation

In addition, we also provide more qualitative results on the datasets used for our benchmarks, as well in-the-wild qualitative results from random videos of various sources to demonstrate the practicality of our model for downstream applications.

1. Model Architecture

We initialize our video encoder and text encoder from ViCLIP-B16 [16], pretrained on large-scale video-text pairs from InternVid-10m-FLT [16].

1.1. Video Encoder

In our proposed approach, the standard [CLS] token used in the video encoder is removed, and 100 trainable [DET] tokens are introduced into the input sequence. These [DET] tokens are processed within the encoder to perform detection tasks. To support this functionality, we incorporate two additional Multi-Layer Perceptrons (MLPs) at the end of the vision encoder: one dedicated to human classification and the other to bounding box regression. Furthermore, the final projection layer of the video encoder, originally designed for the [CLS] token, is repurposed for the [DET] tokens. This projection layer benefits from the pretraining on InternVid-10m-FLT, enabling it to generate action embeddings directly from the [DET] tokens.

To accommodate input sequences of variable lengths, online interpolation is applied to the spatial positional embeddings associated with the [PATCH] tokens. Similarly, the temporal positional embeddings from the InternVid-10m-FLT pretrained weights are interpolated to extend from 8 to 9 frames. For action classification, cosine similarity (S) is computed between L2-normalized action embeddings and text embeddings, measuring the likelihood of an action occurring. To enhance numerical stability during training, we introduce a learnable temperature parameter (T) and a bias parameter (b), resulting in the final logits calculation: $l_{final} = e^T S + b$. The temperature parameter is initialized at $\log(\frac{1}{0.07})$, and the bias parameter is initialized at 0. Importantly, both T and b are removed during inference.

For ablations using the [PATCH] token regression scheme introduced by OWL-ViT [11], we follow BMViT [13] by temporally average pooling the [PATCH] tokens at the output of the video encoder from (B, T, N_{seq}) to (B, N_{seq}), where B, T and N_{seq} represent the batch size, number of frames in the input video clip and the input sequence length per frame.

Table 1. ***Total number of parameters and number of trainable parameters*** in each model, in millions (m). After introducing 100 [DET] tokens, regression MLPs, and LoRA for the text encoder, our model shows only a 2-3% increase in total parameters compared to the original model.

Model	# params (m)		
	Text	Trainable	Total
ViCLIP-B16 [16]	-	149.604	149.604
SiA-B16	Frozen	88.721	152.126
SiA-B16	LoRA	89.212	152.617

Table 2. Training Hyperparameters

Optimizer	Adam
Learning Rate	10^{-5}
$\beta_1, \beta 2$	0.9, 0.999
Number of Frames	9
Input Video Height, Width	240, 320
Sampling Rate (AVA/AVA-K)	8
Sampling Rate (UCF-101-24)	7
Sampling Rate (MultiSports)	7
Sampling Rate (UCF-MAMA)	4

1.2. Text Encoder

LoRA modules [6] are applied on the multi-layer perceptron (MLP) in each text encoder block for all input tokens.

2. Training and Evaluation Configuration

Training hyperparameters are outlined in table 2.

Input video clips are sampled from their original videos with 9 frames × sampling rate, centered around the keyframe.

Given that the 25 fps videos from JHMDB have between 16-40 frames, we do not fix a sampling rate for them. Instead, frames are uniformly sampled from each video; atomic actions (e.g. jump) in JHMDB are guaranteed to happen at the start and stop at the end, temporally trimming these videos would remove essential temporal information, and the atomic action cannot be accurately determined (e.g. jump), especially when performing inference in a downstream manner from AVA-Kinetics pretraining.

2.1. Simulating federated training

For each training batch, instead of passing every single available action class, only a list of ground-truth actions present

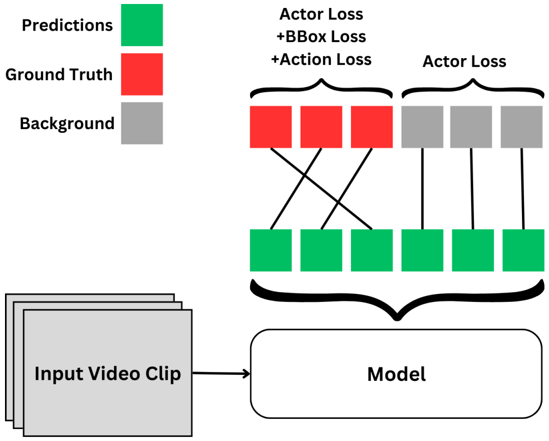

Figure 1. ***Visualization of Bipartite Matching Loss*** for our model. For predicted triplets that are matched with the BG, only the actor loss is calculated.

in that batch is gathered and each action is assigned an integer label. Action loss is calculated based on these integer label assignments. The integer label assignments may vary for each batch.

Algorithm 1 Training loop to simulate federated training

```
for Batch in Dataloader do
    Videos, Annotations = Batch
    ActionsList = collectAllActions(Annotations)
    ActionToIdx = allocateIndices(ActionsList)
    for anno in Annotations do
        anno{labels} = ActionToIdx(anno{textlabels})
    end for
    ActionDescList = Sample1Descriptor(ActionsList)
    outputs = Model(Videos, ActionDescList)
    loss = lossFunc(outputs, Annotations)
    Calculate Gradients and Update Model Weights
end for
```

2.2. Bipartite Matching Loss

Following prior works in transformer-based detection architectures [2–4, 13, 14, 17, 18], we use bipartite matching loss to train our model. The loss function has 2 stages: (1) Hungarian matching, and (2) Loss calculation.

The first stage involves matching N triplet predictions of the model to the optimal M ground truth objects. Given that the model has more predictions than the number of ground truth objects ($N \geq M$), an extra background (BG) class is introduced to match non-predictions with the background class to denote that there is no object in those predictions. The Hungarian matching step minimizes the following cost for a set of predictions $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ and ground truth objects $y = \{y_j\}_{j=1}^{M}$:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{C}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

where:

- $\mathfrak{S}_N$ is the set of all permutations of N elements.
- $\mathcal{C}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ is the matching cost between the ground truth y_i and the predicted object $\hat{y}_{\sigma(i)}$, defined as:

$$\mathcal{C}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\nVdash_{\{c_i=\hat{c}_{\sigma(i)}\}} + \cdot\text{BBox}(b_i, \hat{b}_{\sigma(i)})$$

Here:

- c_i and $\hat{c}_{\sigma(i)}$ are the class labels of the ground truth and prediction, respectively.
- b_i and $\hat{b}_{\sigma(i)}$ are the bounding box coordinates.
- BBox represents the Intersection over Union (IoU) cost summed with the L1-distance cost between predicted boxes and ground truth boxes.

The permutation $\hat{\sigma}$ minimizes the total cost. In our implementation, we only use actor scores and bounding box coordinates to perform the matching following TubeR [18], since the only object of interest is the human figure (and the background).

For the second stage, we compute 3 losses: CE_{actor}, $\mathcal{L}_{box}$ and CE_{action} where CE_{actor}, $\mathcal{L}_{box}$ and CE_{action} represent the actor classification loss, bounding box loss (IoU loss + L1 loss), and action classification loss, respectively. For CE_{actor}, we calculate the loss for every single triplet matched with the correct humans and the BG, whereas for $\mathcal{L}_{box}$ and CE_{action} we calculate the losses only for triplets that are matched with humans; triplets that are matched with the BG class are ignored as shown in Figure 1.

2.3. Multi-label training with Softmax-based Cross-Entropy

We use softmax-based cross-entropy for our action classification loss CE_{action}. The $nn.CrossEntropy()$ implementation in PyTorch allows the use of n-dimensional vectors containing class probabilites (which do not necessarily need to sum to 1) instead of one integer representing one ground-truth class. We exploit this feature to utilize $nn.CrossEntropy()$ in a multi-label manner.

For example, if the predicted probability of actions (after softmax) for a [DET] token is $[0.25, 0.25, 0.25, 0.25]$ and the multi-label ground-truth is $[2, 3]$, after one-hot encoding, the ground truth label becomes $[0, 0, 1, 1]$. After backpropagation, the prediction (after softmax) is expected to be $[0.1, 0.1, 0.4, 0.4]$.

We do not use sigmoid-based losses given that ViCLIP is pretrained with softmax-based cross-entropy; our model does not generalize to downstream datasets when sigmoid-based losses are used to finetune the original ViCLIP weights (e.g. binary cross-entropy, sigmoid-focal cross-entropy [10]).

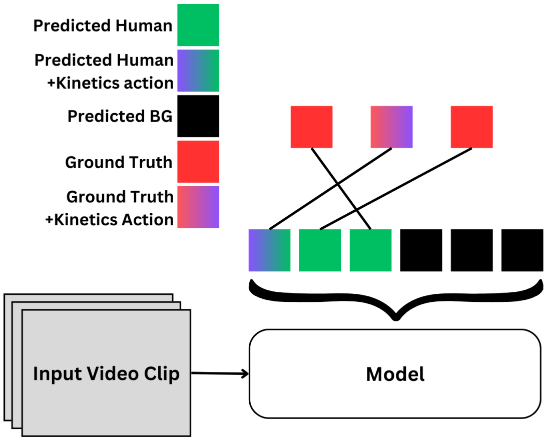
Predicted Human
Predicted Human +Kinetics action
Predicted BG
Ground Truth
Ground Truth +Kinetics Action
Input Video Clip
Model

Figure 2. ***Visualization of AWS*** for our model. We only assign the Kinetics-700 action to the Hungarian-matched ground truth annotation if the assigned prediction detects the associated Kinetics-700 action to be positive. BG denotes the background class.

Table 3. ***Number of floating point operations (in GFLOPs)*** relative to the number of [DET] tokens in the video encoder. 0* indicates that the [PATCH] token regression scheme is used.

# [DET] Tokens	GFLOPs
125	780.78
100	771.20
75	761.64
50	752.14
0*	734.70

2.4. Multi-label Evaluation

Multi-label predictions are obtained by thresholding the cosine similarity at 0.25. Although the full-range of cosine-similarity values are between -1 and 1, ViCLIP, as well as our model, would yield cosine-similarity values roughly between the range of 0 and 0.5. Based on this observation, we empirically choose 0.25 as our threshold.

3. Details on Assignment-based Weak Supervision (AWS)

For AWS, we first use our model trained on AVA-Kinetics annotations augmented with Naive Weak Supervision (NWS). Subsequently, for each Kinetics-700 video in AVA-Kinetics, we predict the associated Kinetics-700 action for all humans in that video and use Hungarian matching to assign the predicted human box to the ground truth box; the Kinetics-700 action is only appended to the ground truth box if the assigned predictions detect the said Kinetics-700 action to be positive as shown in Figure 2. We use the same cost function from the bipartite matching loss used to perform assignment.

4. Computational Complexity

In Table 3, we observe that adding more [DET] tokens increases the computational complexity in the video encoder. However, the relative increase compared to the model using the [PATCH] token regression scheme (0 [DET] tokens) is minimal, as the main bottleneck is the inefficient nature of the Type-1 ViViT used by ViCLIP [16], which is the least efficient ViViT out of the 4 versions introduced in [1]. Nevertheless, we elect to adapt this version as pretraining another video-language model on InternVid [16] from scratch with a different video backbone is too resource-intensive and time-consuming.

5. Additional Qualitative Results

We show additional visualizations from AVA [5], UCF101-24 [15], JHMDB [7], MultiSports [9] and UCF-MAMA [12] from our model in Figures 3, 4, 5, 6 and 7 trained on AVA-Kinetics [8] and using the assignment-based weak supervision scheme (AWS) to cover more than 700 action classes. Red boxes/labels denote the ground truth and green boxes/labels denote out the output of our model.

6. Qualitative Results in the Wild

We show a few visualizations in Figure 8 using videos from various sources other than AVA, UCF101-24, JHMDB, MultiSports or UCF-MAMA, showcasing the practicality of our model in the wild without the need for collecting videos, annotating and training with desired actions.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 3

[2] 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. 2

[3] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *arXiv preprint arXiv:2106.00666*, 2021.

[4] Alexey A Gritsenko, Xuehan Xiong, Josip Djolonga, Mostafa Dehghani, Chen Sun, Mario Lucic, Cordelia Schmid, and Anurag Arnab. End-to-end spatio-temporal action localisation with video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18373–18383, 2024. 2

[5] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018. 3

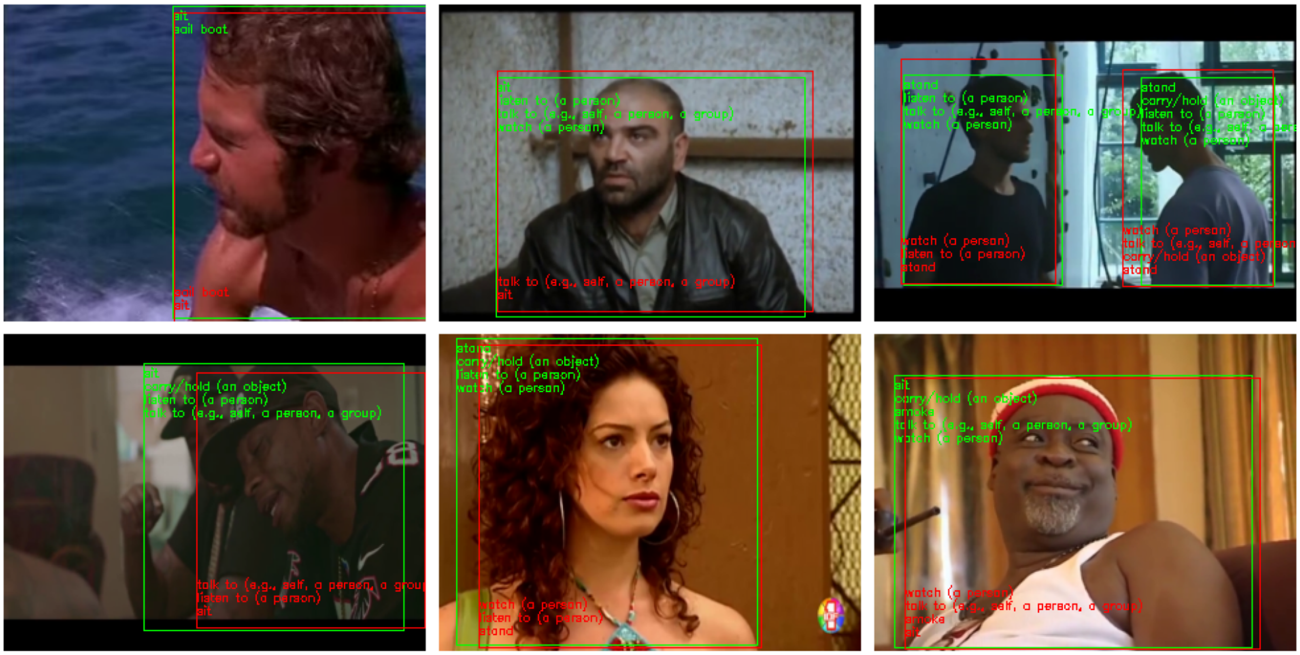

Figure 3. ***Additional Visualizations from AVA***

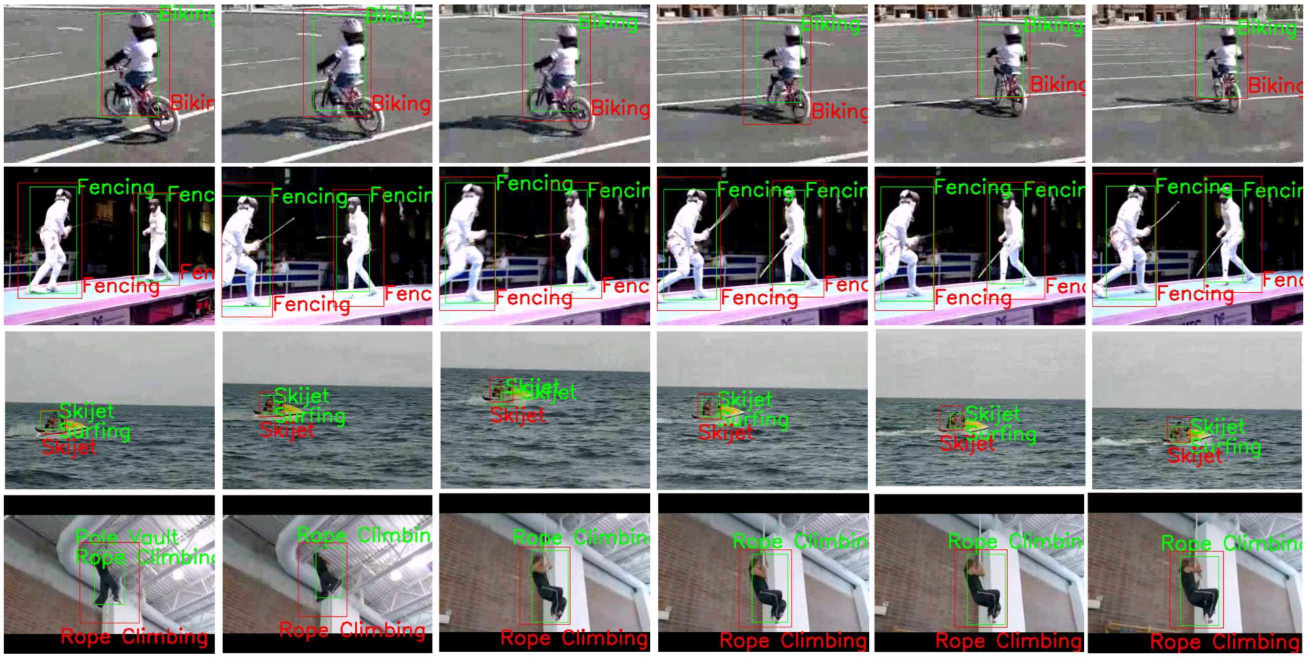

Figure 4. ***Additional Visualizations from UCF101-24:*** Class confusion persists for certain actions as shown in row 3 (Skijet) and row 4 (Rope Climbing).

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1

[7] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In *2013 IEEE International Conference on Computer Vision*, pages 3192–3199, 2013. 3

[8] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020. 3

[9] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video

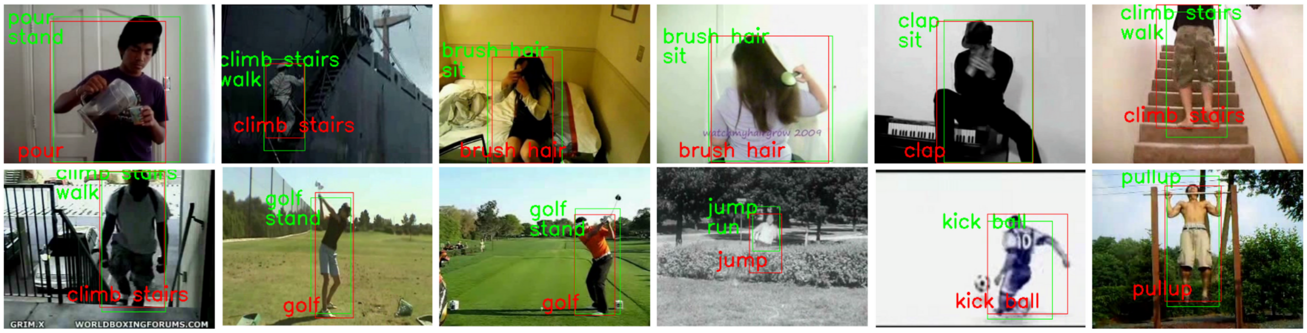

Figure 5. ***Additional Visualizations from JHMDB:*** Our model is able to detect passive actions (e.g. stand, sit, walk) that are happening concurrently with the annotated ground truth actions. This observation signifies the need for better annotations, or a better evaluation scheme for open-vocabulary action detection.

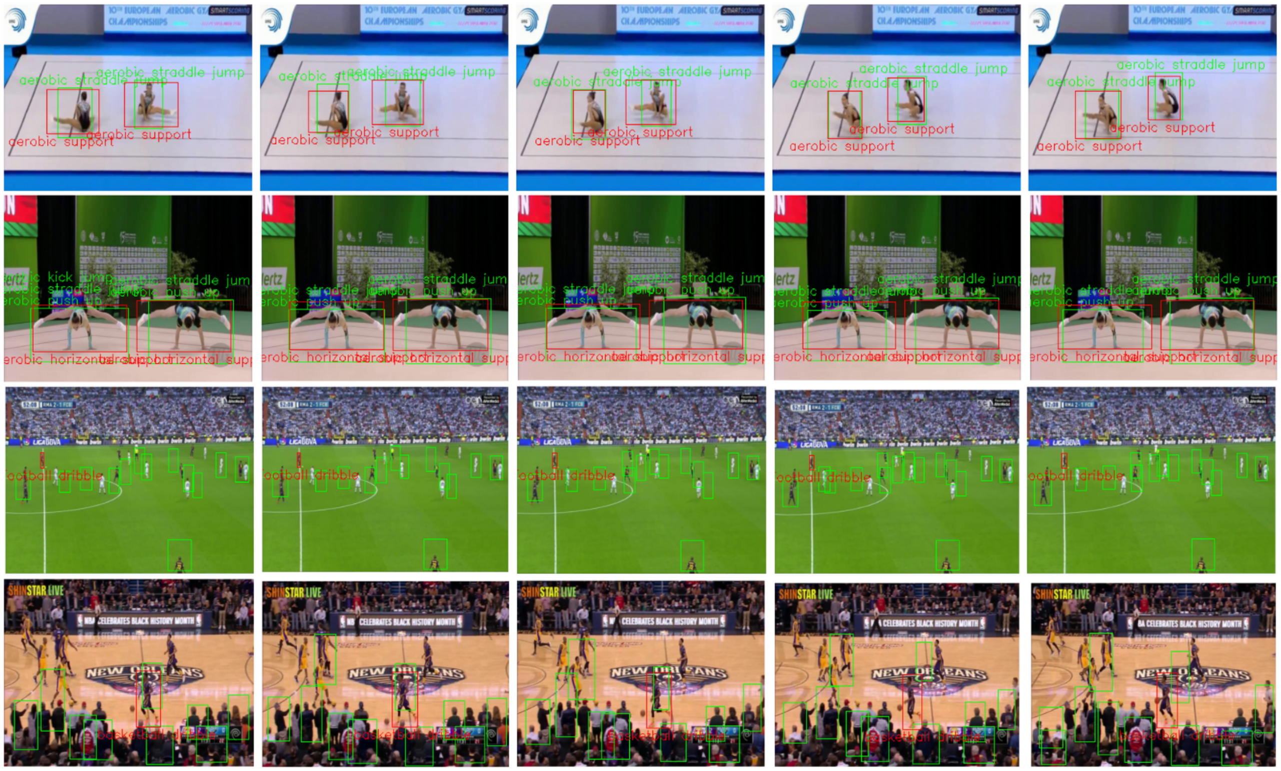

Figure 6. ***Additional Visualizations from MultiSports:*** Our model is able to detect the sport, but not the associated fine-grained action in each sport. In addition, our model fails at detecting long-range humans. The predicted actions in row 3 (football) and row 4 (basketball) are removed for cleaner visualizations since all the fine-grained actions in both sports are detected as positive by our model.

dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13536–13545, 2021. 3

[10] T Lin. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 2

[11] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 1

[12] Rajat Modi, Aayush Jung Rana, Akash Kumar, Praveen Tirupattur, Shruti Vyas, Yogesh Singh Rawat, and Mubarak Shah. Video action detection: Analysing limitations and challenges. *arXiv preprint arXiv:2204.07892*, 2022. 3

[13] Ioanna Ntinou, Enrique Sanchez, and Georgios Tzimiropoulos. Multiscale vision transformers meet bipartite matching for efficient single-stage action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18827–18836, 2024. 1, 2

[14] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-

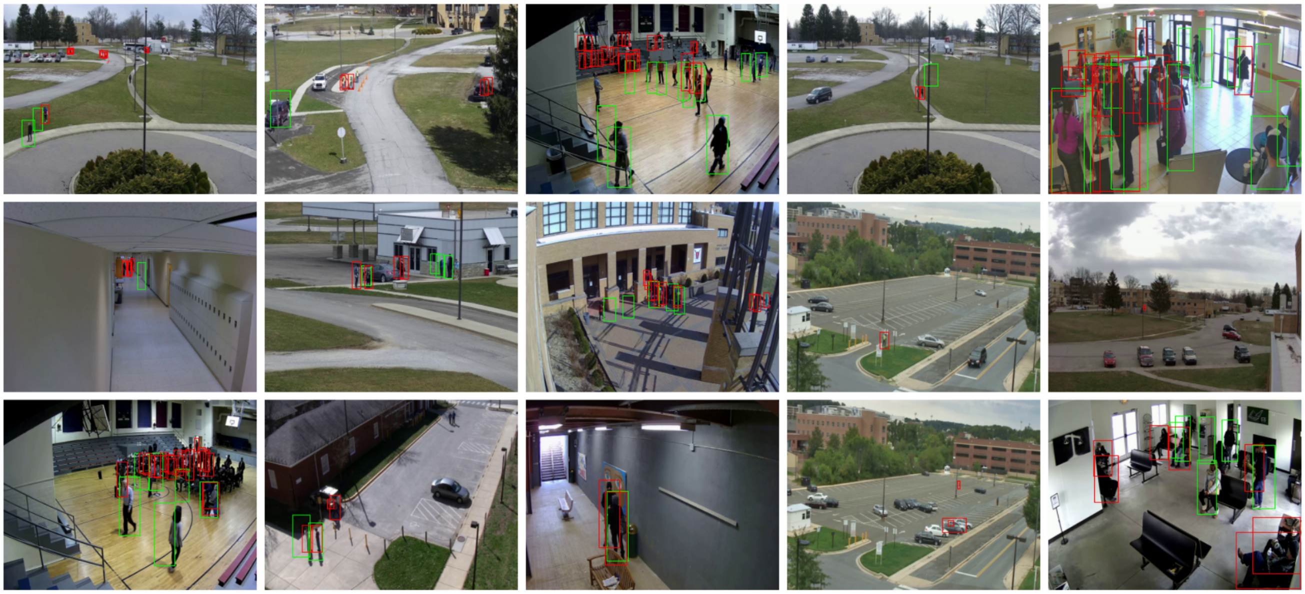

Figure 7. ***Additional Visualizations from UCF-MAMA:*** Our model fails at detecting long-range humans as well as people in crowded scenes. Actions are removed for cleaner visualization.

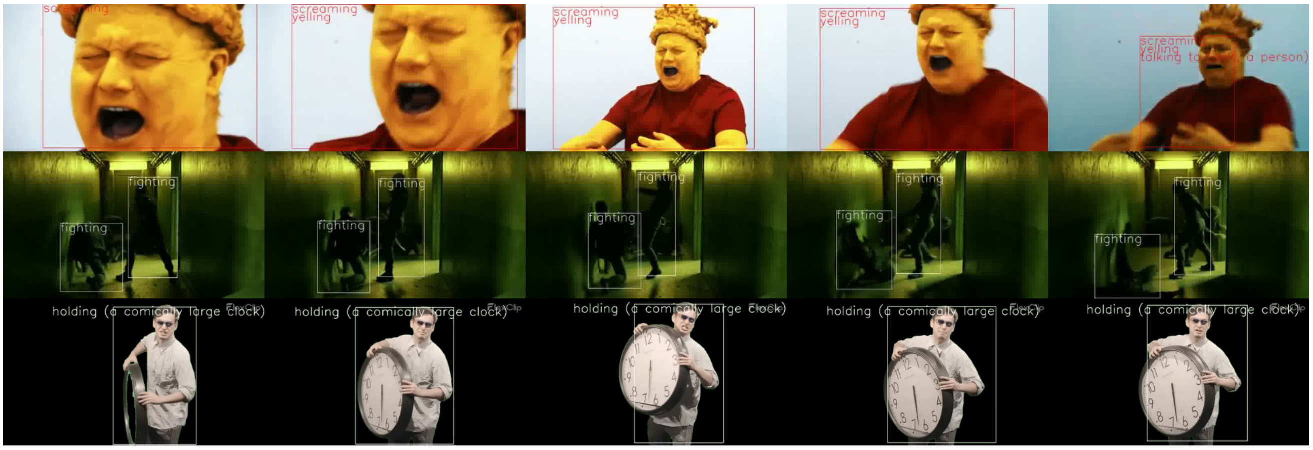

Figure 8. ***Additional Visualizations from select Youtube Videos:*** Our model, having seen more than 700 action classes during training, can be used in the wild with any actions as textual inputs without training, a significant departure from prior works in action detection that are predominantly closed-set, limiting their practicality.

Hsuan Yang. Vidt: An efficient and effective fully transformer-based object detector. *arXiv preprint arXiv:2110.03921*, 2021. 2

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3

[16] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 1, 3

[17] Tao Wu, Mengqi Cao, Ziteng Gao, Gangshan Wu, and Limin Wang. Stmixer: A one-stage sparse action detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14720–14729, 2023. 2

[18] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Shuai Bing, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. *arXiv preprint arXiv:2104.00969*, 2021. 2