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-

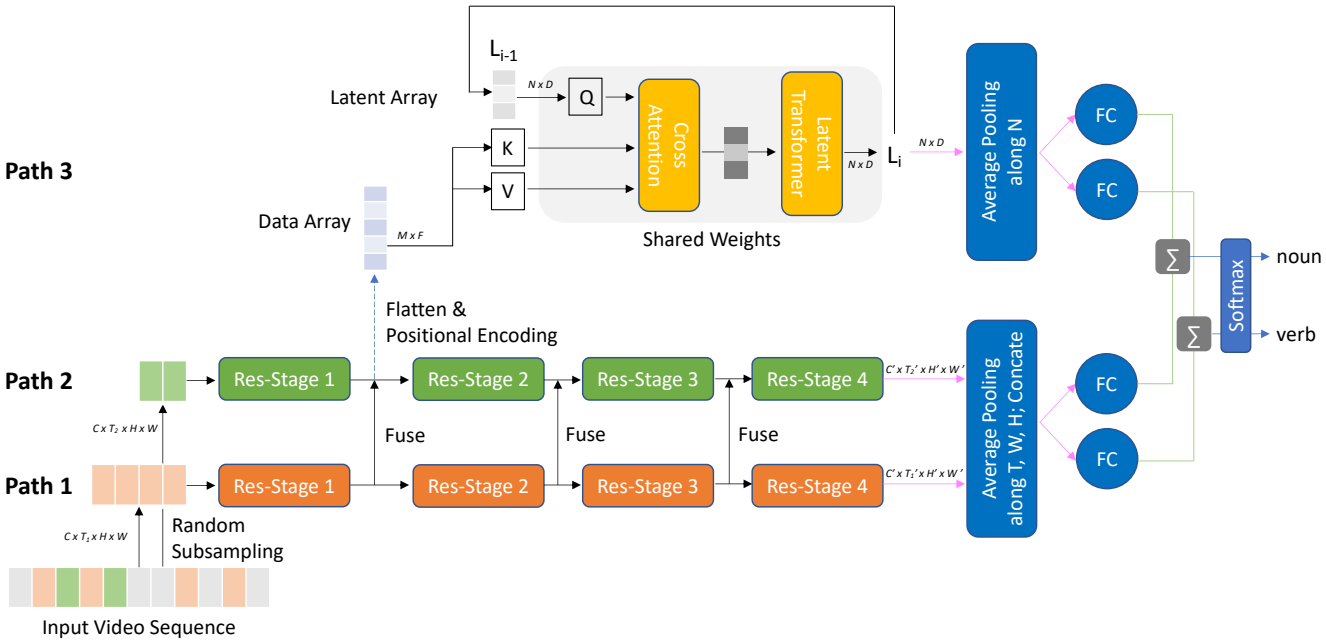

Figure 1: Overview of EgoAugment: given an input video sequence, we will randomly sample T_1 and T_2 frames as the input to the Path 1 and Path 2 respectively, where $T_1 = 2T_2$. Main Branch (Path 1 + Path 2) takes in two sets of video frames and calculates the logits for noun and verb via a data layer, 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