SlowFast Networks for Video Recognition

Technical report: AVA action detection in ActivityNet challenge 2019

Christoph Feichtenhofer Haoqi Fan Jitendra Malik Kaiming He

Facebook AI Research (FAIR)

Abstract

This technical report documents our entry to the AVA action detection track of the ActivityNet challenge 2019. Our entry is based on our paper on SlowFast networks [4]. Extended technical details are elaborated for the AVA action detection challenge, together with the pre-training specifics on the Kinetics dataset. We report 34.3 mAP on the test set for AVA action detection. This result is achieved by using only visual input (e.g. audio information is not exploited).

1. Approach

Our approach closely follows the methodology outlined in the original SlowFast publication [4], for details on the architecture, please refer to [4]. In the next section we summarize the SlowFast instantiations used for the challenge.

1.1. SlowFast instantiations

An example SlowFast model used in this challenge is specified in Table 1. We denote spatiotemporal size by $T\times S^2$ where T is the temporal length and S is the height and width of a square spatial crop.

Slow pathway. The Slow pathway shown in Table 1 is a temporally strided 3D ResNet, modified from [5]. It has $T = 8$ frames as the network input, sparsely sampled from a 64-frame raw clip with a temporal stride $\tau = 8$.

We employ *non-degenerate* temporal convolutions (temporal kernel size > 1, underlined in Table 1) only in res_4 and res_5; all filters from conv_1 to res_3 are essentially 2D convolution kernels in this pathway. This is motivated by our experimental observation that using temporal convolutions in earlier layers degrades accuracy. We argue that this is because when objects move fast and the temporal stride is large, there is little correlation within a temporal receptive field unless the spatial receptive field is large enough.

Fast pathway. Table 1 shows an example of the Fast pathway with $\alpha = 8$ and $\beta = 1/8$. It has a much higher temporal resolution (green) and lower channel capacity (orange).

stage	*Slow* pathway	*Fast* pathway	output sizes $T\times S^2$
raw clip	-	-	64×224^2
data layer	stride 8, 1^2	stride 2, 1^2	*Slow* : 8×224^2 *Fast* : 32×224^2
conv_1	1×7^2, 64 stride 1, 2^2	$\underline{5\times 7^2}$, 8 stride 1, 2^2	*Slow* : 8×112^2 *Fast* : 32×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 8×56^2 *Fast* : 32×56^2
res_2	$\begin{bmatrix} 1\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	$\begin{bmatrix} \underline{3\times 1^2}, 8 \\ \underline{1\times 3^2}, 8 \\ 1\times 1^2, 32 \end{bmatrix}\times 3$	*Slow* : 8×56^2 *Fast* : 32×56^2
res_3	$\begin{bmatrix} 1\times 1^2, 128 \\ 1\times 3^2, 128 \\ 1\times 1^2, 512 \end{bmatrix}\times 4$	$\begin{bmatrix} \underline{3\times 1^2}, 16 \\ \underline{1\times 3^2}, 16 \\ 1\times 1^2, 64 \end{bmatrix}\times 4$	*Slow* : 8×28^2 *Fast* : 32×28^2
res_4	$\begin{bmatrix} \underline{3\times 1^2}, 256 \\ \underline{1\times 3^2}, 256 \\ 1\times 1^2, 1024 \end{bmatrix}\times 23$	$\begin{bmatrix} \underline{3\times 1^2}, 32 \\ \underline{1\times 3^2}, 32 \\ 1\times 1^2, 128 \end{bmatrix}\times 23$	*Slow* : 8×14^2 *Fast* : 32×14^2
res_5	$\begin{bmatrix} \underline{3\times 1^2}, 512 \\ \underline{1\times 3^2}, 512 \\ 1\times 1^2, 2048 \end{bmatrix}\times 3$	$\begin{bmatrix} \underline{3\times 1^2}, 64 \\ \underline{1\times 3^2}, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	*Slow* : 8×7^2 *Fast* : 32×7^2
global average pool, concate, fc			# classes

Table 1. **SlowFast** $T\times\tau = 8\times 8$, instantiation. The dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes. Strides are denoted as {temporal stride, spatial stride^2}. Here the speed ratio is $\alpha = 4$ and the channel ratio is $\beta = 1/8$. τ is 8. The green colors mark *higher* temporal resolution, and orange colors mark *fewer* channels, for the Fast pathway. Non-degenerate temporal filters are underlined. Residual blocks are shown by brackets. The backbone is ResNet-101.

The Fast pathway has non-degenerate temporal convolutions in *every* block. This is motivated by the observation that this pathway holds fine temporal resolution for the temporal convolutions to capture detailed motion. The Slow and Fast pathway perform no temporal downsampling by design.

Lateral connections. Our lateral connections fuse from the Fast to the Slow pathway. It requires to match the sizes of features before fusing. Denoting the feature shape of the Slow pathway as $\{T, S^2, C\}$, the feature shape of the Fast pathway is $\{\alpha T, S^2, \beta C\}$. We employ time-strided convolution (default in [4]) as lateral connection, which performs a 3D convolution with a 5×1^2 kernel with $2\beta C$ output channels and stride $= \alpha$ before concatenation to fuse information into the Slow pathway.

Instantiations used for challenge. For our entry to the ActivityNet Challenge 2019 [1] we train three model variants, all using a single-modality (rgb). First, **SlowFast** 8×8, is a SlowFast model as described above. Second, **Fast** (β=1), 32×2, is an extremely heavy model that consists only of the Fast pathway of the SlowFast architecture above, but without reduced number of channels, *i.e.*, the channel reduction ratio is set to $\beta = 1$. It has 4.8×higher computational cost than the SlowFast variant. Third, the **SlowFast** 16×8, model as presented in [4] that uses twice the input duration, spanning a total window of 16×8 = 128 frames on the input.

2. Experiments: AVA Action Detection

Dataset. The AVA dataset [9] focuses on spatiotemporal localization of human actions. The data is taken from 437 movies and labels are provided for 80 different categories. Spatiotemporal ground-truth is available at one frame per second, with every person annotated with a bounding box and (possibly multiple) actions. There are 211k training and 57k validation video segments. We follow the standard protocol [9] of training on all 80 categories and evaluating on 60 classes. The performance metric is mean Average Precision (mAP), using a frame-level IoU threshold of 0.5.

Detection architecture. Our detector is similar to Faster R-CNN [21] with minimal modifications adapted for video. We use the SlowFast network or its variants as the backbone. We set the spatial stride of res_5 to 1 (instead of 2), and use a dilation of 2 for its filters. This increases the spatial resolution of res_5 by 2×. We extract region-of-interest (RoI) features [6] at the last feature map of res_5. We extend each 2D RoI at a frame into a 3D RoI by replicating it along the temporal axis, following [9]. We compute RoI features by RoIAlign [10] spatially, and global average pooling temporally. The RoI features are then max-pooled and fed to a per-class, sigmoid-based classifier for multi-label prediction.

We follow previous works that use pre-computed proposals [9, 22, 15]. Our region proposals are computed by an off-the-shelf person detector, *i.e.*, that is not jointly trained with the action detection models. We adopt a person-detection model trained with *Detectron* [7]. It is a Faster R-CNN with a ResNeXt-101-FPN [26, 18] backbone. It is pre-trained on ImageNet and the COCO human keypoint images [19]. We fine-tune this detector on AVA for person (actor) detection. Then, the region proposals for action detection are detected person boxes with a confidence of > 0.8.

Training. We initialize the network weights from the Kinetics-600 classification models presented in [4]. We use step-wise learning rate, reducing the learning rate 10× when validation error saturates. We train for 12k iterations (58 epochs for ∼211k data), with linear warm-up [8] for the first 1k iterations. We use a weight decay of 10^{-7}. All other hyper-parameters are the same as in the Kinetics experiments used for pre-training and are described in the next section. Ground-truth boxes, and proposals overlapping with ground-truth boxes by IoU > 0.9, are used as the samples for training.The input is instantiation-specific $\alpha T \times \tau$ frames of size 224×224. For the test submission we train the models on train+val (and by 1.5× longer than when training on train only) and submit it to the official test server [17].

model	video pretrain	val mAP	test mAP
SlowFast, 8×8, AVA v2.1	Kinetics-600	28.2	-
SlowFast, 8×8	Kinetics-600	29.0	-
Fast (β=1) 32×2,	Kinetics-600	27.4	-
SlowFast, 16×8	Kinetics-600	29.8	-
SlowFast++, 16×8	Kinetics-600	**30.7**	-
SlowFast++, ensemble	Kinetics-600	-	**34.3**

Table 2. **SlowFast models on AVA**. All our variants are on the AVA v2.2 dataset, except the first row which is on v2.1. Here "++" indicates a version of our method that is tested with multi-scale and horizontal flipping augmentation. The backbone is R-101+NL.

Inference. We perform inference on a single clip with $\alpha T \times \tau$ frames around the frame that is to be evaluated. We resize the spatial dimension such that its shorter side is 256 pixels. The backbone feature extractor is computed fully convolutionally, as in standard Faster R-CNN [21]. For ensembling we average prediction scores across models.

2.1. Main Results

We show our results on AVA in Table 2. As a baseline, our **SlowFast**, 8×8 model produces 28.2 mAP on AVA v2.1. This number is slightly higher than in the original publication [4] as we use more proposal boxes for training and testing (boxes with confidence > 0.8). Using the AVA v2.2 dataset (which provides more consistent annotations) improves this number to 29.0 mAP. Next, we also train a **Fast** (β=1) 32×2 model, corresponding to only of the Fast pathway of the SlowFast architecture above, but without channel reduction, *i.e.*, $\beta = 1$. It has 4.8×higher computational cost than the SlowFast variant and therefore is a very heavy model. The performance of this model is 27.4 mAP. Next, we investigate the longer-term **SlowFast**, 16×8 model. For this model we also employ a non-local context branch [25, 24], that computes self-attention between global-pooled features and the region features, which is concatenated to the region features before classification. The performance of this model is 29.8 mAP. By using multiple spatial scales and horizontal flip for testing (SlowFast++, Table 2), this number is increased to **30.7 mAP** on the validation set.

Finally, we create an ensemble of 7 models, **SlowFast**, 8×8, **Fast** (β=1) 32×and five **SlowFast++**, 16×8 models, which differ in the IoU (0.75 *vs.* 0.9) threshold used for training, the non-local context branch (on *vs.* off), the training duration (18k *vs.* 16k iterations), and scale jitter used for training ([256, 340] *vs.* [256, 360]). This diverse set of models achieves **34.3 mAP** accuracy on the test set.

3. Pretraining: Kinetics Action Classification

Datasets. We pre-train our approach on the Kinetics action classification dataset as it can have a large impact on AVA performance [4]. Kinetics-400 [16] consists of ~240k training videos and 20k validation videos in 400 human action categories, and Kinetics-600 [2] has ~392k training videos and 30k validation videos in 600 classes. Kinetics-700 is an approximate superset of Kinetics-400 [16] and Kinetics-600 [2], holding 650,000 video clips, and covering 700 human action categories with at least 600 video clips for each action class. The validation and test sets of Kinetics-700 contain around 50 and 100 videos per class, holding a total of around 35k and 70k videos, respectively. We report top-1 and top-5 classification accuracy (%). Performance is shown on the validation sets, unless stated otherwise. The performance criterion used in the challenge is the average of top-1 and top-5 error; therefore we also report these numbers. Finally, we show the computational cost (in FLOPs) of a single, spatially center-cropped test clip as a measure of model complexity.

Training. Model training follows our recipe outlined in [4] and is summarized here for consistency. Our models on Kinetics are trained *from random initialization* ("*from scratch*"), *without* using ImageNet [3] or any pre-training. We adopt synchronized SGD training with 128 GPUs following the recipe in [8], and we found its accuracy is as good as typical training in one 8-GPU machine but it scales out well. The mini-batch size is 8 clips per GPU (so the total mini-batch size is 1024). We use the initialization method in [11], and train with Batch Normalization (BN) [14], and the BN statistics are computed within each 8 clips. We adopt a half-period cosine schedule [20] of learning rate decaying: the learning rate at the n-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\text{max}}}\pi) + 1]$, where n_{max} is the maximum training iterations and η is the base learning rate. We also use a linear warm-up strategy [8] for the initial phase of training. We use momentum of 0.9 and weight decay of 10^{-4}. Dropout [13] of 0.5 is used before the final classifier layer. For Kinetic-600, we use a base learning rate of $\eta = 0.8$ and train for 240 epochs (92k iterations with a total mini-batch size of 1024, in ~392k Kinetics videos). For the models trained on Kinetics-700, we use the same schedule as for Kinetics-600, and initialize from the models trained on this dataset to allow faster convergence.

We study a ResNet-101 [12] backbone that is equipped with non-local (NL) blocks [23]. We only use NL blocks on the (fused) Slow features of res_4 (instead of res_3+res_4 [23]).

For augmentation, we randomly sample a clip (of αT frames) from the videos, and the input to the Slow and Fast pathways are respectively T and αT frames; for the spatial domain, we randomly crop 224×224 pixels from a video, or its horizontal flip, with a shorter side sampled in [256, 340] pixels. For models that span longer temporal duration on the input. $T \times \tau > 64$, we increase the scale jitter to [256, 360].

Inference. Following common practice, we uniformly sample 10 clips from a video along its temporal axis. For each clip, we scale the shorter spatial side to 256 pixels and take 3 crops of 256×256 to cover the spatial dimensions, as an approximation of fully-convolutional testing, following the code of [23]. We average the softmax scores for prediction.

We report the actual *inference-time* computation. As existing papers differ in their inference strategy for cropping/clipping in space and in time. When comparing to previous work, we report the FLOPs per spacetime "view" (temporal clip with spatial crop) at inference *and* the number of views used. Recall that in our case, the inference-time spatial size is 256^2 (instead of 224^2 for training) and 10 temporal clips each with 3 spatial crops are used (30 views).

3.1. Baseline Results

Kinetics-700 is a new dataset and existing results are not existant, so our goal is to report results for future reference in Table 3. We employ pre-trained models from Kinetics-600.

In terms of performance shown in Table 3, we see that the SlowFast 8×8 model performs similar to the Fast model, that has much higher model complexity (552 *vs*. 115 GFLOPs per view), producing 70.6% top-1 accuracy. The SlowFast 16×8 performs slightly better with 71.0% top-1 accuracy.

One interesting observation we make is that overall accuracy on Kinetics-700 is considerably lower than for the Kinetics-400 and Kinetics-600 datasets. This might be because our models did not train long enough before the submission deadline. Our SlowFast 16×8 model produces 79.8% / 93.9% top-1 / top-5 accuracy on Kinetics-400, 81.1% / 95.1% on Kinetics-600, and 71.0% and 89.6% on Kinetics-700, more results on Kinetics-400 and 600 are in [4].

We also test an ensemble consisting of SlowFast 8×8, Fast (β=1), 32×2, and SlowFast 16×8 to the test server. The final average of top-1 and top-5 test error is 18.95%. This baseline result is achieved using only visual information (rgb) and we expect better performance using multi-modal input (*e.g.* by employing audio information, as is commonly done for challenges) and longer training schedules (full model convergence could not be completed in time before submission deadline).

4. Acknowledgments

We are grateful for discussions with Ross Girshick and Chao-Yuan Wu. The region proposals used in our submission are originating from the LFB project [25].

References

[1] ActivityNet-Challenge. http://activity-net.org/challenges/2019/index.html, 2019. 2

[2] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 3

model	modality	pretrain	top-1 accuracy	top-5 accuracy	avg val† error	avg test† error	GFLOPs×views
SlowFast 8×8	rgb	Kinetics-600	70.6	89.7	19.9		115 × 30
Fast (β=1), 32×2	rgb	Kinetics-600	70.6	89.7	19.9		552 × 30
SlowFast 16×8	rgb	Kinetics-600	71.0	89.6	19.7		234 × 30
SlowFast ensemble	rgb	Kinetics-600	-	-	-	18.95	

Table 3. SlowFast baselines on Kinetics-700. Our three instantiations are differing in model type (Fast/SlowFast), input sampling ($T\times\tau$) and complexity (GFLOPS) which is shown in the last column where we report the inference cost with a single "view" (temporal clip with spatial crop) × the numbers of such views used. †: "avg" is the average of top-1 and top-5 error. The backbones is R-101 with NL blocks.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 3

[4] C. Feichtenhofer, H. Fan, J. Malik, and K. He. SlowFast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018. 1, 2, 3

[5] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 1

[6] R. Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 2

[7] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 2

[8] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv:1706.02677*, 2017. 2, 3

[9] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. CVPR*, 2018. 2

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. CVPR*, 2015. 3

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 3

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012. 3

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 3

[15] J. Jiang, Y. Cao, L. Song, S. Z. Y. Li, Z. Xu, Q. Wu, C. Gan, C. Zhang, and G. Yu. Human centric spatio-temporal action localization. In *ActivityNet workshop, CVPR*, 2018. 2

[16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 3

[17] Leaderboard:ActivityNet-AVA. http://activity-net.org/challenges/2019/evaluation.html. 2

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 2

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 2

[20] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 3

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[22] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *ECCV*, 2018. 2

[23] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proc. CVPR*, 2018. 3

[24] X. Wang and A. Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 2

[25] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krähenbühl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 2, 3

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017. 2