SlowFast Networks for Video Recognition

Christoph Feichtenhofer Haoqi Fan Jitendra Malik Kaiming He

Facebook AI Research (FAIR)

Abstract

We present SlowFast networks for video recognition. Our model involves (i) a Slow pathway, operating at low frame rate, to capture spatial semantics, and (ii) a Fast pathway, operating at high frame rate, to capture motion at fine temporal resolution. The Fast pathway can be made very lightweight by reducing its channel capacity, yet can learn useful temporal information for video recognition. Our models achieve strong performance for both action classification and detection in video, and large improvements are pin-pointed as contributions by our SlowFast concept. We report 79.0% accuracy on the Kinetics dataset without using any pre-training, largely surpassing the previous best results of this kind. On AVA action detection we achieve a new state-of-the-art of 28.3 mAP. Code will be made publicly available.

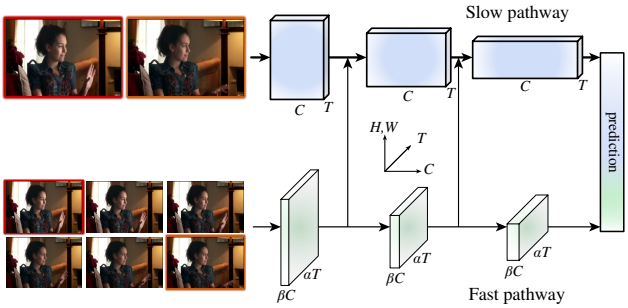

Figure 1. **A SlowFast network** has a low frame rate, low temporal resolution *Slow* pathway and a high frame rate, $\alpha\times$ higher temporal resolution *Fast* pathway. The Fast pathway is lightweight by using a fraction (β, *e.g.*, 1/8) of channels. Lateral connections fuse them. This sample is from the AVA dataset [17] (annotation: hand wave).

1. Introduction

It is customary in the recognition of images $I(x, y)$ to treat the two spatial dimensions x and y symmetrically. This is justified by the statistics of natural images, which are to a first approximation isotropic—all orientations are equally likely—and shift-invariant [38, 23]. But what about video signals $I(x, y, t)$? Motion is the spatiotemporal counterpart of orientation [1], but all spatiotemporal orientations are *not* equally likely. Slow motions are more likely than fast motions (indeed most of the world we see is at rest at a given moment) and this has been exploited in Bayesian accounts of how humans perceive motion stimuli [51]. For example, if we see a moving edge in isolation, we perceive it as moving perpendicular to itself, even though in principle it could also have an arbitrary component of movement tangential to itself (the aperture problem in optical flow). This percept is rational if the prior favors slow movements.

If all spatiotemporal orientations are not equally likely, then there is no reason for us to treat space and time symmetrically, as is implicit in approaches to video recognition based on spatiotemporal convolutions [44, 3]. We might instead "factor" the architecture to treat spatial structures and temporal events separately. For concreteness, let us study this in the context of recognition. The categorical spatial semantics of the visual content often evolve *slowly*. For example, waving hands do not change their identity as "hands" over the span of the waving action, and a person is always in the "person" category even though he/she can transit from walking to running. So the recognition of the categorical semantics (as well as their colors, textures, lighting *etc*.) can be refreshed relatively *slowly*. On the other hand, the motion being performed can evolve much *faster* than their subject identities, such as clapping, waving, shaking, walking, or jumping. It can be desired to use *fast* refreshing frames (high temporal resolution) to effectively model the potentially *fast* changing motion.

Based on this intuition, we present a two-pathway *SlowFast* model for video recognition (Fig. 1). One pathway is designed to capture semantic information that can be given by images or a few sparse frames, and it operates at *low* frame rates and *slow* refreshing speed. In contrast, the other pathway is responsible for capturing rapidly changing motion, by operating at *fast* refreshing speed and high temporal resolution. Despite its high temporal rate, this pathway is made very *lightweight*, *e.g.*, ~20% of total computation. This is because this pathway is designed to have fewer channels and weaker ability to process spatial information, while such information can be provided by the first pathway in a less redundant manner. We call the first a *Slow* pathway and the second a *Fast* pathway, driven by their different temporal speeds. The two pathways are fused by lateral connections.

Our conceptual idea leads to flexible and effective designs for video models. The Fast pathway, due to its lightweight nature, does *not* need to perform any temporal pooling—it can operate on high frame rates for all intermediate layers and maintain temporal fidelity. Meanwhile, thanks to the lower temporal rate, the Slow pathway can be more focused on the spatial domain and semantics. By treating the raw video at different temporal rates, our method allows the two pathways to have their own expertise on video modeling.

We comprehensively evaluate our method on the Kinetics [27, 2] and AVA [17] datasets. On Kinetics action classification, our method achieves 79.0% accuracy *without* any pre-training (*e.g.*, ImageNet), largely surpassing the best number in the literature of this kind by 5.1%. Ablation experiments convincingly demonstrate the improvement contributed by the SlowFast concept. On AVA action detection, our model achieves a new state-of-the-art of 28.3% mAP.

Our method is partially inspired by biological studies on the retinal ganglion cells in the primate visual system [24, 34, 6, 11, 46], though admittedly the analogy is rough and premature. These studies found that in these cells, ~80% are Parvocellular (P-cells) and ~15-20% are Magnocellular (M-cells). The M-cells operate at *high temporal frequency* and are more responsive to temporal changes, but are not sensitive to spatial details or colors. P-cells provide fine spatial details and colors, but have lower temporal resolution. Our framework is analogous in that: (i) our model has two pathways separately working at low and high temporal resolutions; (ii) our Fast pathway is designed to capture fast changing motion but fewer spatial details, analogous to M-cells; and (iii) our Fast pathway is lightweight, similar to the small ratio of M-cells. We hope these relations will inspire more computer vision models for video recognition.

2. Related Work

Spatiotemporal filtering. Actions can be formulated as spatiotemporal objects and captured by oriented filtering in spacetime, as done by HOG3D [28] and cuboids [8]. 3D ConvNets [43, 44, 3] extend 2D image models [29, 40, 42, 21] to the spatiotemporal domain, handling both spatial and temporal dimensions similarly. There are recent methods focusing on decomposing the convolutions into separate 2D spatial and 1D temporal filters [9, 45, 53, 36].

Beyond spatiotemporal filtering or their separable versions, our work pursuits a more thorough separation of modeling expertise by using two different temporal speeds.

Optical flow for video recognition. There is a classical branch of research focusing on hand-crafted spatiotemporal features based on optical flow. These methods, including histograms of flow [30], motion boundary histograms [4], and trajectories [47], had shown competitive performance for action recognition before the prevalence of deep learning.

In the context of deep neural networks, the two-stream method [39] exploits optical flow by viewing it as another input modality. This method has been a foundation of many competitive results in the literature [9, 10, 49]. However, it is methodologically unsatisfactory given that optical flow is a hand-designed representation, and two-stream methods are often not learned end-to-end jointly with the flow.

Our work is related to the two-stream method [39], but provides conceptually different perspectives. The two-stream method [39] has not explored the potential of *different temporal speeds*, a key concept in our method. The two-stream method adopts the same backbone structure to both streams, whereas our Fast pathway is more lightweight. Our method does not compute optical flow, and therefore, our models are learned end-to-end from the raw data.

3. SlowFast Networks

Our generic architecture has a Slow pathway (Sec. 3.1) and a Fast pathway (Sec. 3.2), which are fused by lateral connections (Sec. 3.3). Fig. 1 illustrates our concept.

3.1. Slow pathway

The Slow pathway can be any convolutional model (*e.g.*, [9, 44, 3, 50]) that works on a clip of video as a spatiotemporal volume. The key concept in our Slow pathway is a *large* temporal stride τ on input frames, *i.e.*, it processes only one out of τ frames. A typical value of τ we studied is 16—this refreshing speed is roughly 2 frames sampled per second for 30-fps videos. Denoting the number of frames sampled by the Slow pathway as T, the raw clip length is $T \times \tau$ frames.

3.2. Fast pathway

In parallel to the Slow pathway, the Fast pathway is another convolutional model with the following properties.

High frame rate. Our goal here is to have a fine representation along the temporal dimension. Our Fast pathway works with a *small* temporal stride of τ/α, where $\alpha > 1$ is the frame rate ratio between the Fast and Slow pathways. Our two pathways operate on the same raw clip, so the Fast pathway samples αT frames, α times denser than the Slow pathway. A typical value is $\alpha = 8$ in our experiments.

The presence of α is in the key of the SlowFast concept (Fig. 1, time axis). It explicitly indicates that the two pathways work on *different* temporal speeds, and thus drives the expertise of the two subnets instantiating the two pathways.

High temporal resolution features. Our Fast pathway not only has a high input resolution, but also pursues high-resolution features throughout the network hierarchy. In our instantiations, we use *no* temporal downsampling layers (neither temporal pooling nor time-strided convolutions) throughout the Fast pathway, until the global pooling layer before classification. As such, our feature tensors always

have αT frames along the temporal dimension, maintaining temporal fidelity as much as possible.

Low channel capacity. Our Fast pathway also distinguishes with existing models in that it can use significantly *lower* channel capacity to achieve good accuracy for the SlowFast model. This makes it lightweight.

In a nutshell, our Fast pathway is a convolutional network analogous to the Slow pathway, but has a ratio of β ($\beta < 1$) channels of the Slow pathway. The typical value is $\beta = 1/8$ in our experiments. Notice that the computation (floating-number operations, or FLOPs) of a common layer is often *quadratic* in term of its channel scaling ratio. This is what makes the Fast pathway more computation-effective than the Slow pathway. In our instantiations, the Fast pathway typically takes ~20% of the total computation. Interestingly, as mentioned in Sec. 1, evidence suggests that ~15-20% of the retinal cells in the primate visual system are M-cells (that are sensitive to fast motion but not color or spatial detail).

The low channel capacity can also be interpreted as a *weaker* ability of representing spatial semantics. Technically, our Fast pathway has no special treatment on the spatial dimension, so its spatial modeling capacity should be lower than the Slow pathway because of fewer channels. The good results of our model suggest that it is a desired tradeoff for the Fast pathway to weaken its spatial modeling ability while strengthening its temporal modeling ability.

Motivated by this interpretation, we also explore different ways of weakening spatial capacity in the Fast pathway, including reducing input spatial resolution and removing color information. As we will show by experiments, these versions can all give good accuracy, suggesting that a lightweight Fast pathway with less spatial capacity can be made beneficial.

3.3. Lateral connections

The information of the two pathways is fused, so one pathway is not unaware of the representation learned by the other pathway. We implement this by *lateral connections*, which have been used to fuse optical flow-based, two-stream networks [9, 10]. In image object detection, lateral connections [32] are a popular technique for merging different levels of spatial resolution and semantics.

Similar to [9, 32], we attach one lateral connection between the two pathways for every "stage" (Fig. 1). Specifically for ResNets [21], these connections are right after pool_1, res_2, res_3, and res_4. The two pathways have different temporal dimensions, so the lateral connections perform a transformation to match them (detailed in Sec. 3.4). We use unidirectional connections that fuse features of the Fast pathway into the Slow one (Fig. 1). We have experimented with bidirectional fusion and found similar results.

Finally, a global average pooling is performed on each pathway's output. Then two pooled feature vectors are concatenated as the input to the fully-connected classifier layer.

stage	*Slow* pathway	*Fast* pathway	output sizes $T\times S^2$
raw clip	-	-	64×224^2
data layer	stride 16, 1^2	stride **2**, 1^2	*Slow* : 4×224^2 *Fast* : 32×224^2
conv_1	1×7^2, 64 stride 1, 2^2	5×7^2, 8 stride 1, 2^2	*Slow* : 4×112^2 *Fast* : 32×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 4×56^2 *Fast* : 32×56^2
res_2	$\begin{bmatrix} 1\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	$\begin{bmatrix} 3\times 1^2, 8 \\ 1\times 3^2, 8 \\ 1\times 1^2, 32 \end{bmatrix}\times 3$	*Slow* : 4×56^2 *Fast* : 32×56^2
res_3	$\begin{bmatrix} 1\times 1^2, 128 \\ 1\times 3^2, 128 \\ 1\times 1^2, 512 \end{bmatrix}\times 4$	$\begin{bmatrix} 3\times 1^2, 16 \\ 1\times 3^2, 16 \\ 1\times 1^2, 64 \end{bmatrix}\times 4$	*Slow* : 4×28^2 *Fast* : 32×28^2
res_4	$\begin{bmatrix} 3\times 1^2, 256 \\ 1\times 3^2, 256 \\ 1\times 1^2, 1024 \end{bmatrix}\times 6$	$\begin{bmatrix} 3\times 1^2, 32 \\ 1\times 3^2, 32 \\ 1\times 1^2, 128 \end{bmatrix}\times 6$	*Slow* : 4×14^2 *Fast* : 32×14^2
res_5	$\begin{bmatrix} 3\times 1^2, 512 \\ 1\times 3^2, 512 \\ 1\times 1^2, 2048 \end{bmatrix}\times 3$	$\begin{bmatrix} 3\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix}\times 3$	*Slow* : 4×7^2 *Fast* : 32×7^2
global average pool, concate, fc			# classes

Table 1. **An example instantiation of the SlowFast network**. 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 $\alpha = 8$ and the channel ratio is $\beta = 1/8$. τ is 16. 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-50.

3.4. Instantiations

Our idea of SlowFast is generic, and it can be instantiated with different backbones (*e.g.*, [40, 42, 21]) and implementation specifics. In this subsection, we describe our instantiations of the network architectures.

An example SlowFast model 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. The details are described next.

Slow pathway. The Slow pathway in Table 1 is a temporally strided 3D ResNet, modified from [9]. It has $T = 4$ frames as the network input, sparsely sampled from a 64-frame raw clip with a temporal stride $\tau = 16$. We opt to not perform temporal downsampling in this instantiation, as doing so would be detrimental when the input stride is large.

Unlike typical C3D / I3D models, we use *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 (*i.e.*, in later layers).

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).

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. Also, the Fast pathway has no temporal downsampling layers 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 experiment with the following transformations in the lateral connections:

(i) *Time-to-channel*: We reshape and transpose $\{\alpha T, S^2, \beta C\}$ into $\{T, S^2, \alpha\beta C\}$, meaning that we pack all α frames into the channels of one frame.

(ii) *Time-strided sampling*: We simply sample one out of every α frames, so $\{\alpha T, S^2, \beta C\}$ becomes $\{T, S^2, \beta C\}$.

(iii) *Time-strided convolution*: We perform a 3D convolution of a 5×1^2 kernel with $2\beta C$ output channels and stride $= \alpha$.

The output of the lateral connections is fused into the Slow pathway by summation or concatenation.

4. Experiments: Kinetics Action Classification

Datasets. We investigate Kinetics-400 [27] that has ~240k training videos and 20k validation videos in 400 human action categories. We report top-1 and top-5 classification accuracy (%). We also report the computational cost (in FLOPs) of a single, spatially center-cropped clip.[1]

Training. Our models are trained *from random initialization* ("*from scratch*") on Kinetics, *without* using ImageNet [5] or any pre-training. We use the initialization method in [20].

We adopt synchronized SGD training in 128 GPUs following the recipe in [16], 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 train with Batch Normalization (BN) [25], and the BN statistics are computed within each 8 clips. We adopt a half-period cosine schedule [35] 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 the base learning rate η is set as 1.6. We also use a linear warm-up strategy [16] in the first 8k iterations. Unless specified, we train for 256 epochs (60k iterations with a total mini-batch size of 1024, in ~240k Kinetics videos) when $T \leq 4$ frames, and 196 epochs when $T > 4$ frames: it is sufficient to train shorter when a clip has more frames. We use momentum of 0.9 and weight decay of 10^{-4}. Dropout [22] of 0.5 is used before the final classifier layer.

For the temporal domain, we randomly sample a clip (of $T\times\tau$ frames) from the full-length video, 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 randomly sampled in [256, 320] pixels [40, 50].

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 [50]. We average the softmax scores for prediction.

Our implementation is based on the public code of [50].[2]

4.1. Results and Analysis

Table 2 shows a series of ablations, analyzed next:

Training from scratch. Our models are trained *from scratch*, without ImageNet training. To draw fair comparisons, it is helpful to check the potential impacts (positive or negative) of training from scratch. To this end, we train *the exact same* 3D ResNet-50 architectures specified in [50], using our large-scale SGD recipe trained from scratch.

Table 2a shows the comparisons on two architectures. In both cases, our training recipe achieves *comparably good* results as the ImageNet pre-training counterpart reported by [50]. This suggests that our training system, as the foundation of the following experiments, has no loss despite no ImageNet pre-training.

Individual pathways. Table 2b shows the results using the structure of one individual pathway *alone* (specified in Table 1). We compare them with the 3D R-50 of [50]. Our individual pathways use no temporal downsampling, leading to a total temporal reduction factor of 1 (see "t-reduce" in Table 2b). So we also train a counterpart 3D R-50 of [50] modified for total temporal reduction of 1 (Table 2b, row 2).

Table 2b shows that our individual pathways *alone* have *no unfair advantage* over the 3D R-50 design, as shown by their lower accuracy. This can be explained by the fact that our individual pathways are far more *lightweight* than the 3D R-50 ones: our Slow pathway has *fewer* frames, and our Fast pathway has *fewer* channels. Consequently, they have only 20.9 and 4.9 GFLOPs respectively, substantially cheaper than the two 3D R-50 baselines (28.1 or 44.9 GFLOPs). The two pathways are designed with their special expertise if they are used jointly, as we will show next.

[1] We use single-clip, center-crop FLOPs as a basic *unit* of computational cost. Inference-time computational cost is roughly proportional to this, *if* a fixed number of clips and crops is used, as is for our all models.

[2] https://github.com/facebookresearch/video-nonlocal-net

model	pre-train	$T \times \tau$	t-reduce	top-1	top-5	GFLOPs
3D R-50 [50]	ImageNet	32×2	2^3	73.3	90.7	33.1
3D R-50 (our recipe)	-	32×2	2^3	73.0	90.4	33.1
3D R-50 [50]	ImageNet	8×8	2^1	73.4	90.9	28.1
3D R-50, our recipe	-	8×8	2^1	73.5	90.8	28.1

(a) **Baselines trained from scratch**: Using the same structure as [50], our training recipe achieves comparable results *without* ImageNet pre-training. "t-reduce" is the temporal downsampling factor in the network.

model	$T \times \tau$	t-reduce	top-1	top-5	GFLOPs
3D R-50	8×8	2^1	73.5	90.8	28.1
3D R-50	8×8	1	**74.6**	**91.5**	44.9
our Slow-only, R-50	4×16	1	72.6	90.3	**20.9**
our Fast-only, R-50	32×2	1	51.7	78.5	**4.9**

(b) **Individual pathways**: Training our Slow-only or Fast-only pathway alone, using the structure specified in Table 1. "t-reduce" is the total temporal downsampling factor within the network.

	lateral	top-1	top-5	GFLOPs
Slow-only	-	72.6	90.3	20.9
SlowFast	-	73.5	90.3	26.2
SlowFast	TtoC, concat	74.3	91.0	30.5
SlowFast	TtoC, sum	74.5	91.3	26.2
SlowFast	T-sample	75.4	91.8	26.7
SlowFast	T-conv	**75.6**	**92.1**	27.6

(c) **SlowFast fusion**: Fusing Slow and Fast pathways with various lateral connections is consistently better than the Slow-only baseline. Backbone: R-50.

	top-1	top-5	GFLOPs
Slow-only	72.6	90.3	20.9
$\beta = 1/4$	75.6	91.7	41.7
$1/6$	**75.8**	92.0	32.0
$1/8$	75.6	**92.1**	27.6
$1/12$	75.2	91.8	25.1
$1/16$	75.1	91.7	23.4
$1/32$	74.2	91.3	21.9

(d) **Channel capacity ratio**: Varying values of β, the channel capacity ratio of the Fast pathway. Backbone: R-50.

Fast pathway	spatial	top-1	top-5	GFLOPs
RGB	-	**75.6**	**92.1**	27.6
RGB, β=1/4	*half*	74.7	91.8	26.3
gray-scale	-	**75.5**	**91.9**	**26.1**
time diff	-	74.5	91.6	26.2
optical flow	-	73.8	91.3	26.9

(e) **Weaker spatial input to Fast pathway**: Various ways of weakening spatial inputs to the Fast pathway in SlowFast models. β=1/8 unless specified otherwise. Backbone: R-50.

	top-1	top-5	GFLOPs
Slow-only	72.6	90.3	20.9
SlowFast	**75.6**	**92.1**	**27.6**
2-Slow ens.	73.2	90.8	41.8
"SlowSlow"	70.5	88.6	75.6

(f) ***vs.* Slow+Slow**: Ensembling 2 Slow-only models (ens.), or replacing the Fast pathway with a Slow pathway ("SlowSlow") . Backbone: R-50.

	$T \times \tau$	α	top-1	top-5	GFLOPs
Slow-only	4×16	-	72.6	90.3	20.9
SlowFast	4×16	8	75.6	92.1	27.6
Slow-only	8×8	-	74.9	91.5	41.9
SlowFast	8×8	4	**77.0**	**92.6**	50.3
SlowFast	2×32	8	73.4	90.8	**13.9**
SlowFast	4×16	4	75.3	91.7	25.2
SlowFast	6×16	8	76.8	92.2	41.1
SlowFast	8×12	4	76.8	92.5	50.3

(g) **Various SlowFast instantiations**, compared to Slow-only counterparts. Here all SlowFast models use β=1/8 for the Fast pathway. Backbone: R-50.

SlowFast	$T \times \tau$	α	top-1	top-5	GFLOPs
R-50	4×16	8	75.6	92.1	27.6
R-50 + NL	4×16	8	76.3	92.2	33.8
R-50	8×8	4	77.0	92.6	50.3
R-50 + NL	8×8	4	**77.7**	**93.1**	65.5
R-101	4 ×16	8	76.9	92.7	44.5
R-101 + NL	4 ×16	8	77.4	92.7	47.4
R-101	8×8	4	77.9	93.2	81.5
R-101 + NL	8×8	4	**79.0**	**93.6**	88.0

(h) **Advanced backbones for SlowFast models**, with ResNet-101 [21] and/or non-local (NL) blocks [50]. NL blocks are added to $\text{res}_{3,4}$ for R-50 and to res_4 for R-101.

Table 2. Ablations on **Kinetics-400** action classification. We show top-1 and top-5 classification accuracy (%), as well as computational complexity measured in GFLOPs (floating-point operations, in # of multiply-adds $\times 10^9$) for a single clip input of spatial size 224^2.

SlowFast fusion. Table 2c shows various ways of fusing the Slow and Fast pathways in Table 2b.

As a naïve fusion baseline, in Table 2c we show a variant using no lateral connection: it only concatenates the final outputs of the two pathways. This variant has 73.5% accuracy, slightly better than the Slow-only counterpart by 0.8%.

Then we show SlowFast models with various lateral connections: time-to-channel (TtoC), time-strided sampling (T-sample), and time-strided convolution (T-conv). Concatenation is used for merging. For TtoC which can match channel dimensions for this model, we also show merging by element-wise summation.

Table 2c shows that these SlowFast models are *all* better than the Slow-only pathway. With the best-performing lateral connection of T-conv, our SlowFast network is **3.0% better** than the one-pathway, Slow-only baseline. We use T-conv as our default lateral connection.

Fig. 2 shows the curves of the Slow-only model (72.6%) *vs.* the SlowFast model (75.6%) during the training procedure. We plot the single-crop validation error and training error. It is clear that the SlowFast model is consistently better than its Slow-only counterpart *throughout* the entire training.

Interestingly, the Fast pathway alone has only 51.7% accuracy (Table 2b). But it brings in up to 3.0% improvement to the Slow pathway, showing that the underlying representation modeled by the Fast pathway is largely complementary. We strengthen this observation by the next set of ablations.

Channel capacity of Fast pathway. A key intuition for designing the Fast pathway is that it can employ a lower channel capacity for capturing motion *without* building a detailed spatial representation. This is controlled by the channel ratio β. Table 2d shows the effect of varying β.

The best-performing β values are $1/6$ and $1/8$ (our default). Nevertheless, it is surprising to see that *all* values from β=1/32 to $1/4$ in our SlowFast model can improve over the Slow-only counterpart. In particular, with β=1/32, the Fast pathway only adds as small as 1.0 GFLOPs (~5% relative), but leads to 1.6% improvement.

Weaker spatial inputs to Fast pathway. Further, we experiment with using different *weaker* spatial inputs to the

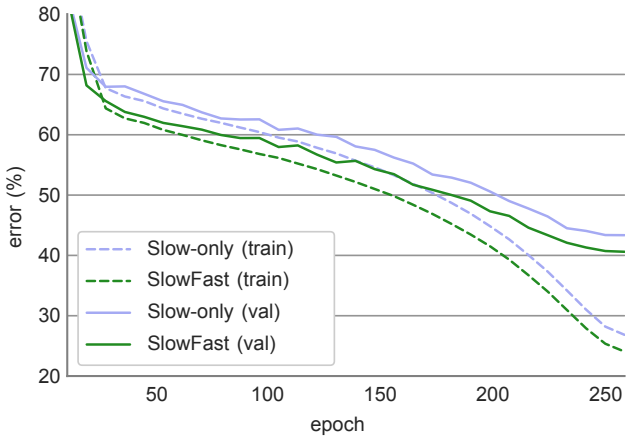

Figure 2. Training procedure on Kinetics for Slow-only (blue) *vs*. SlowFast (green) network. We show the top-1 training error (dash) and validation error (solid). The curves are single-crop *errors*; the video *accuracy* is 72.6% *vs*. 75.6% (see also Table 2c).

Fast pathway in our SlowFast model. We consider: (i) a *half spatial resolution* (112×112), with β=1/4 (*vs*. default 1/8) to roughly maintain the FLOPs; (ii) *gray-scale* input frames; (iii) "*time difference*" frames, computed by subtracting the current frame with the previous frame; and (iv) using *optical flow* as the input to the Fast pathway.

Table 2e shows that all these variants are competitive and are better than the Slow-only baseline. In particular, the *gray-scale* version of the Fast pathway is nearly as good as the RGB variant, but reduces FLOPs by ~5%. Interestingly, this is also consistent with the M-cell's behavior of being insensitive to colors [24, 34, 6, 11, 46].

We believe both Table 2d and Table 2e convincingly show that the *lightweight* but temporally *high-resolution* Fast pathway is an effective component for video recognition.

***vs*. Slow+Slow.** In Table 2f we study using *two* Slow pathways instead of SlowFast. We consider: (i) ensembling two Slow-only models that are independently trained, and (ii) replacing the Fast pathway with a Slow pathway to create a "SlowSlow" model. Table 2f shows that ensembling moderately improve accuracy by 0.5% at the cost of 2× computation, and SlowSlow severely suffers from overfitting.

Various SlowFast instantiations. In Table 2g we compare various instantiations of SlowFast models. We compare with two Slow-only baselines from Table 2b. Their SlowFast counterparts have healthy gains. In particular, even if the Slow-only model has a denser input ($T \times \tau = 8\times8$), the Fast pathway is still able to improve its accuracy by 2.1% (from 74.9% to 77.0%), with only a ~20% increase of FLOPs.

The bottom rows in Table 2g show that various SlowFast models perform competitively and offer a variety of tradeoffs between accuracy and FLOPs. In particular, the smallest SlowFast model we have tried has only 13.9 GFLOPs, which is < 50% of many Slow-only 3D convolutional models (see also Table 2b), but it still has good accuracy of 73.4%.

We believe there will be more room to optimize the tradeoffs between accuracy and FLOPs under the SlowFast framework, and the desired tradeoff is often application-dependent. But our results all suggest that the SlowFast models are more effective than the Slow-only ones.

model	flow	pretrain	top-1	top-5	inference GFLOPs×views
I3D [3]		ImageNet	72.1	90.3	108 × N/A
Two-Stream I3D [3]	✓	ImageNet	75.7	92.0	216 × N/A
S3D-G [53]	✓	ImageNet	77.2	93.0	143 × N/A
Nonlocal R-50 [50]		ImageNet	76.5	92.6	282 × 30
Nonlocal R-101 [50]		ImageNet	77.7	93.3	359 × 30
R(2+1)D Flow [45]	✓	-	67.5	87.2	152 × 115
STC [7]		-	68.7	88.5	N/A × N/A
ARTNet [48]		-	69.2	88.3	23.5 × 250
S3D [53]		-	69.4	89.1	66.4 × N/A
ECO [54]		-	70.0	89.4	N/A × N/A
I3D [3]	✓	-	71.6	90.0	216 × N/A
R(2+1)D [45]		-	72.0	90.0	152 × 115
R(2+1)D [45]	✓	-	73.9	90.9	304 × 115
SlowFast, R50 (4×16)		-	75.6	92.1	36.1 × 30
SlowFast, R50		-	77.0	92.6	65.7 × 30
SlowFast, R50 + NL		-	77.7	93.1	80.8 × 30
SlowFast, R101		-	77.9	93.2	106 × 30
SlowFast, R101 + NL		-	**79.0**	**93.6**	115 × 30

Table 3. **Comparison with the state-of-the-art on Kinetics-400**. In the column of computational cost, we report the cost of a single "view" (temporal clip with spatial crop) and the numbers of such views used. Details of the SlowFast models in this table are in Table 2h. "N/A" indicates the numbers are not available for us. The SlowFast models are the $T\times\tau = 8\times8$ versions, unless specified.

Advanced backbones. Thus far all experiments used ResNet-50 as the backbone. Next we study advanced backbones including a deeper model of ResNet-101 [21] and a non-local (NL) version [50]. For models involving R-101, to reduce overfitting, we use a scale jittering range of [256, 340] pixels and a random temporal jittering of $[-\tau/2, -\tau/2)$ when sampling the Slow pathway with a stride of τ. For all models involving NL, we initialize them with the counterparts that are trained without NL, to facilitate convergence. We only add NL to the Slow pathway. For R101+NL, we only add NL to res_4 (instead of res_3+res_4 [50]).

Table 2h shows the results. As expected, using advanced backbones is orthogonal to our SlowFast concept, and they give additional improvement over our SlowFast baselines.

Comparison with state-of-the-art results. Table 3 shows the comparisons with state-of-the-art results in Kinetics-400. We also show the actual *inference-time* computation. As *existing papers differ in their inference strategy for cropping/clipping in space and in time*, in Table 3, 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

model	pretrain	top-1	top-5	inference GFLOPs×views
I3D [2]	-	71.9	90.1	108 × N/A
StNet-IRv2 RGB [18]	ImgNet+Kinetics400†	79.0	N/A	N/A
SlowFast, R50	-	79.9	94.5	65.7 ×30
SlowFast, R101	-	80.4	94.8	106 × 30
SlowFast, R101 + NL	-	**81.1**	**94.9**	115 × 30

Table 4. **Kinetics-600 results**. SlowFast models are with $T\times\tau = 8\times8$. †: The Kinetics-400 training set partially overlaps with the Kinetics-600 validation set, and "*it is therefore not ideal to evaluate models on Kinetics-600 that were pre-trained on Kinetics-400*" [2].

of 224^2), and 10 temporal clips each with 3 spatial crops are used (so 30 views in total).

Table 3 shows that our results are substantially better than existing results that are also *without ImageNet pre-training*. In particular, our model (79.0%) is **5.1%** absolutely better than the previous best result of this kind (73.9%). Our results are also better than those using ImageNet pre-training.

Our results are achieved at low inference-time cost. We notice that many existing works (if reported) use *extremely dense* sampling of clips along the temporal axis, which can lead to >100 views at inference time. This cost has been largely overlooked. In contrast, our method does not require many temporal clips, thanks to our high temporal resolution yet lightweight Fast pathway. Our cost per spacetime view can be low (*e.g.*, 36.3 GFLOPs), while still being more accurate than existing methods.

Kinetics-600. Kinetics-600 [2] has ~392k training videos and 30k validation videos in 600 classes. As this dataset is larger, we extend the training epochs (and the learning rate schedule) by 2×. We set the base learning rate η as 0.8.

Kinetics-600 is relatively new, and existing results are limited. So our goal is mainly to provide results for future reference. See Table 4. Note that the Kinetics-600 validation set overlaps with the Kinetics-400 training set [2], so we do *not* pre-train on Kinetics-400. The winning entry [18] of the latest ActivityNet Challenge 2018 [12] reports a best single-model, single-modality accuracy of 79.0%. Our method achieves 81.1%.

5. Experiments: AVA Action Detection

Dataset. The AVA dataset [17] focuses on spatiotemporal localization of human actions. The data is taken from 437 movies. Spatiotemporal labels are provided for 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 [17] of evaluating on 60 classes (see Fig. 3). The performance metric is mean Average Precision (mAP) over 60 classes, using a frame-level IoU threshold of 0.5.

Detection architecture. Our detector is similar to Faster R-CNN [37] with minimal modifications adapted for video.

model	$T\times\tau$	α	mAP
Slow-only, R-50	4×16	-	19.0
SlowFast, R-50	4×16	8	**24.2**

Table 5. **AVA action detection baselines**: Slow-only *vs*. SlowFast.

model	$T\times\tau$	α	mAP
SlowFast, R-50	4×16	8	24.2
SlowFast, R-50	8×8	4	24.8
SlowFast, R-101	8×8	4	**26.1**

Table 6. **More instantiations of SlowFast models on AVA**.

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 [14] 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 [17]. We compute RoI features by RoIAlign [19] 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.

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* [15]. It is a Faster R-CNN with a ResNeXt-101-FPN [52, 32] backbone. It is pre-trained on ImageNet and the COCO human keypoint images [33]. We fine-tune this detector on AVA for person (actor) detection. The person detector produces 93.9 AP@50 on the AVA validation set. Then, the region proposals for action detection are detected person boxes with a confidence of > 0.9, which has a recall of 91.1% and a precision of 90.7% for the person class.

Training. We initialize the network weights from the Kinetics-400 classification models. We use step-wise learning rate, reducing the learning rate 10× when validation error saturates. We train for 14k iterations (68 epochs for ~211k data), with linear warm-up [16] 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. Ground-truth boxes, and proposals overlapping with ground-truth boxes by IoU > 0.75, are used as the samples for training. The input is instantiation-specific $T\times\tau$ frames of size 224×224.

Inference. We perform inference on a single clip with $T\times\tau$ 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 convolutional, as in standard Faster R-CNN [37].

5.1. Results and Analysis

Table 5 compares a Slow-only baseline with its SlowFast counterpart, with the *per-category* AP shown in Fig. 3. Our method improves massively by **5.2** mAP (relative 28%) from 19.0 to 24.2. This is *solely* contributed by our SlowFast idea.

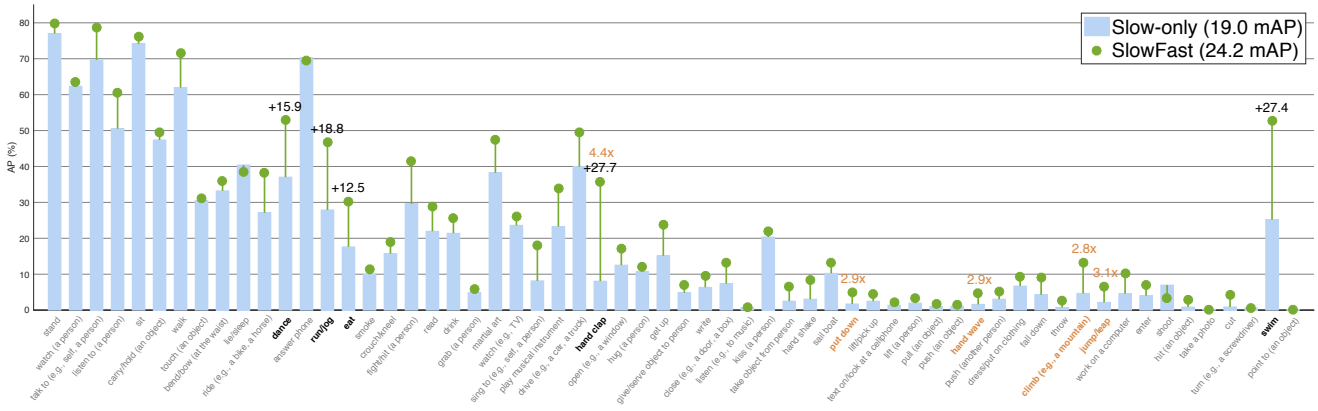

Figure 3. **Per-category AP on AVA**: a Slow-only baseline (19.0 mAP) *vs*. its SlowFast counterpart (24.2 mAP). The highlighted categories are the 5 highest absolute increase (**black**) or 5 highest relative increase with Slow-only AP > 1.0 (**orange**). Categories are sorted by number of examples. Note that the SlowFast instantiation in this ablation is not our best-performing model.

model	flow	video pretrain	val mAP	test mAP
I3D [17]		Kinetics-400	14.5	-
I3D [17]	✓	Kinetics-400	15.6	-
ACRN, S3D [41]	✓	Kinetics-400	17.4	-
ATR, R50 + NL [26]		Kinetics-400	20.0	-
ATR, R50 + NL [26]	✓	Kinetics-400	21.7	-
9-model ensemble [26]	✓	Kinetics-400	25.6	21.1
I3D [13]		Kinetics-**600**	21.9	21.0
SlowFast, R101		Kinetics-400	26.1	-
SlowFast, R101		Kinetics-**600**	26.8	**26.6**
SlowFast, R101 + NL		Kinetics-**600**	27.3	-
SlowFast++, R101 + NL		Kinetics-**600**	**28.3**	-

Table 7. **Comparison with the state-of-the-art on AVA**. Here "++" indicates a version of our method that is tested with multi-scale and horizontal flipping augmentation (testing augmentation strategies for existing methods are not always reported).

Category-wise (Fig. 3), our SlowFast model improves in **57 out of 60** categories, *vs*. its Slow-only counterpart. The largest absolute gains are observed for "*hand clap*" (+27.7 AP), "*swim*" (+27.4 AP), "*run/jog*" (+18.8 AP), "*dance*" (+15.9 AP), and "*eat*" (+12.5 AP). We also observe large relative increase in "*jump/leap*", "*hand wave*", "*put down*", "*throw*", "*hit*" or "*cut*". These are categories where modeling dynamics are of vital importance. The SlowFast model is worse in only 3 categories: "*answer phone*" (-0.1 AP), "*lie/sleep*" (-0.2 AP), "*shoot*" (-0.4 AP), and their decrease is relatively small *vs*. others' increase.

Table 6 shows more SlowFast instantiations. Their performance is consistent with the Kinetics action classification accuracy (see Table 3), suggesting that our models are robust.

Comparison with state-of-the-art results. Finally, we compare with previous results on AVA in Table 7. An interesting observation is on the potential benefit of using optical flow (see column 'flow' in Table 7). Existing works have observed mild improvements: +1.1 mAP for I3D in [17], and +1.7 mAP for ATR in [26]. In contrast, our improvement from the Fast pathway is +5.2 mAP (Table 5). Moreover, two-stream methods using optical flow can *double* the computational cost, whereas our Fast pathway is lightweight.

As system-level comparisons, our SlowFast model has 26.1 mAP using only Kinetics-400 pre-training. This is **4.4** mAP higher than the previous best number under similar settings (21.7 of ATR [26], single-model), and **6.1** mAP higher than that using no optical flow (Table 7).

The work in [13] pre-trains on the larger Kinetics-600 and achieves 21.9 mAP. With our Kinetics-600 pre-trained model, we approach 26.8 mAP. Augmenting with NL blocks [50] increases to this 27.3 mAP. By using 3 spatial scales and horizontal flip for testing (SlowFast++, Table 7), we achieve **28.3 mAP**, a new state-of-the-art on AVA.

We further provide an oracle experiment on ground-truth region proposals resulting in **35.1 mAP**, suggesting future work could also explore to improve the proposal generation.

For our baseline with 26.8 validation mAP, we train it on train+val (and by $1.5\times$ longer) and submit it to the official test server [31]. It achieves 26.6 mAP on the test set; therefore signalizing consistency with our other results (Table 7).

Visualization. We show qualitative results of the SlowFast model in Fig. 4. Overall, even given some discrepancy between our predictions and ground-truth, our results are visually reasonable. Predictions and ground-truth labels are in several cases extending each other, illustrating the difficulty of this dataset. For example, consider the sequence in the last row of Fig. 4. The SlowFast detector is able to predict the right person's reading action with a score of 0.54, but is penalized as this label is not present in the annotations.

6. Conclusion

The time axis is a special dimension. This paper has investigated an architecture design that contrasts the speed along this axis. It achieves state-of-the-art accuracy for video action classification and detection. We hope that this SlowFast concept will foster further research in video recognition.

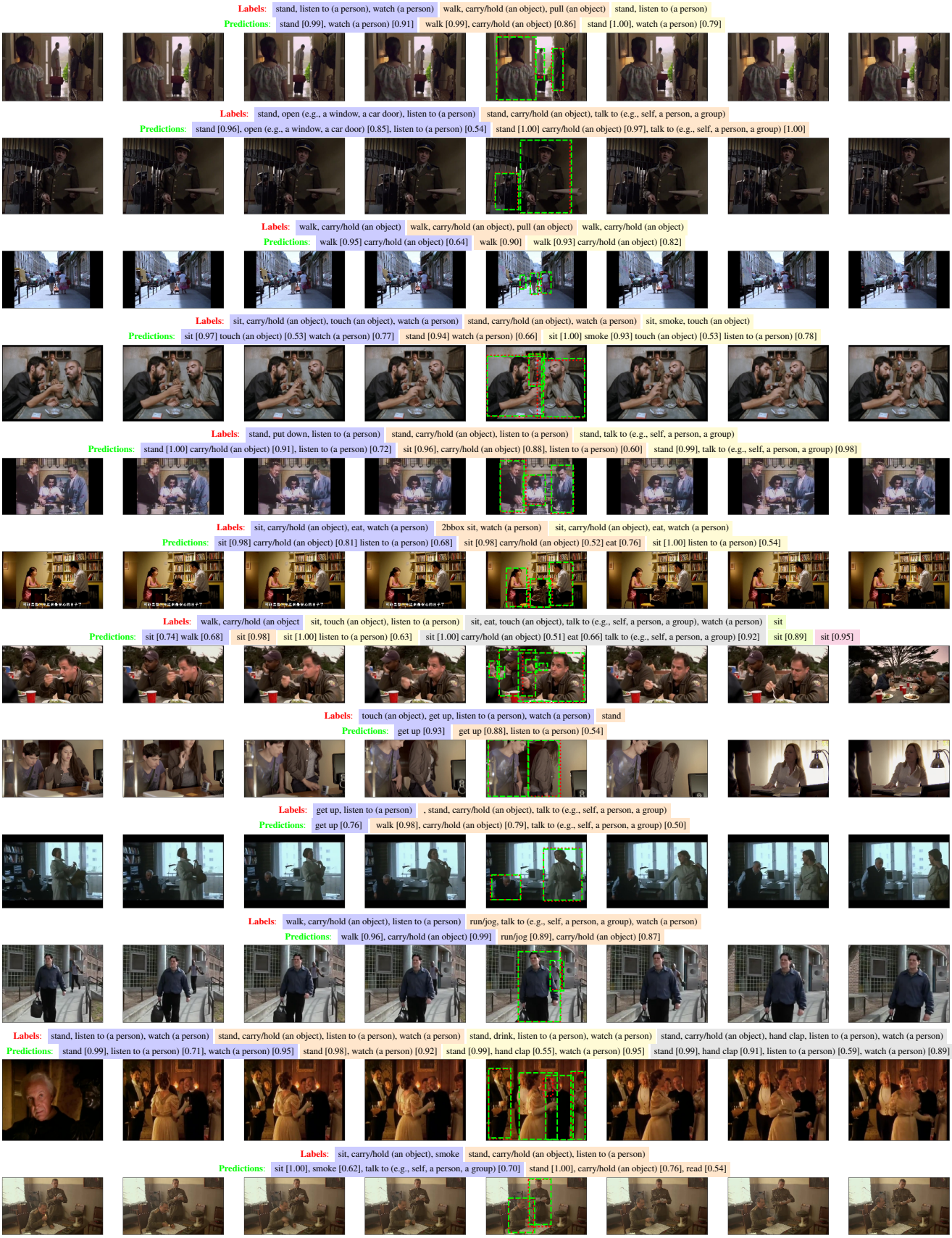

Figure 4. **Visualization on AVA**. Our model's predictions (green, with confidence > 0.5) *vs.* ground-truth labels (red), on the AVA validation set. We only show the predictions/labels in the center frame that is annotated. Multiple tags of one instance are marked with one background color (instances are tagged from left to right). This is a SlowFast model of $T\times\tau = 8\times8$, with 26.8 mAP. Best viewed with zoom.

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, 1985. 1

[2] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 2, 7

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 1, 2, 6

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. ECCV*, 2006. 2

[5] 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. 4

[6] A. Derrington and P. Lennie. Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque. *The Journal of physiology*, 357(1):219–240, 1984. 2, 6

[7] A. Diba, M. Fayyaz, V. Sharma, M. M. Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proc. ECCV*, 2018. 6

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS Workshop, ICCV*, 2005. 2

[9] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 2, 3

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016. 2, 3

[11] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991. 2, 6

[12] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, V. Escorcia, R. Khrisna, S. Buch, and C. D. Dao. The ActivityNet large-scale activity recognition challenge 2018 summary. *arXiv:1808.03766*, 2018. 7

[13] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for AVA. *arXiv:1807.10066*, 2018. 8

[14] R. Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 7

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

[16] 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. 4, 7

[17] 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. 1, 2, 7, 8

[18] D. He, F. Li, Q. Zhao, X. Long, Y. Fu, and S. Wen. Exploiting spatial-temporal modelling and multi-modal fusion for human action recognition. *arXiv:1806.10319*, 2018. 7

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 7

[20] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. CVPR*, 2015. 4

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2, 3, 5, 6

[22] 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. 4

[23] J. Huang and D. Mumford. Statistics of natural images and models. In *Proc. CVPR*, 1999. 1

[24] D. H. Hubel and T. N. Wrisel. Receptive fields and functional architecture in two non-striate visual areas of the cat. *J. Neurophysiol*, 28:229–289, 1965. 2, 6

[25] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 4

[26] 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. 8

[27] 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. 2, 4

[28] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. BMVC.*, 2008. 2

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[30] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008. 2

[31] Leaderboard:ActivityNet-AVA. http://activity-net.org/challenges/2018/evaluation.html. 8

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 3, 7

[33] 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. 7

[34] M. Livingstone and D. Hubel. Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, 240(4853):740–749, 1988. 2, 6

[35] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 4

[36] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. ICCV*, 2017. 2

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 7

[38] D. L. Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517–548, 1994. 1

[39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 2, 3, 4

[41] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *ECCV*, 2018. 8

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 2, 3

[43] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proc. ECCV*, 2010. 2

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proc. ICCV*, 2015. 1, 2

[45] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proc. CVPR*, 2018. 2, 6

[46] D. C. Van Essen and J. L. Gallant. Neural mechanisms of form and motion processing in the primate visual system. *Neuron*, 13(1):1–10, 1994. 2, 6

[47] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. ICCV*, 2013. 2

[48] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. In *Proc. CVPR*, 2018. 6

[49] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. ECCV*, 2016. 2

[50] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proc. CVPR*, 2018. 2, 4, 5, 6, 8

[51] Y. Weiss, E. P. Simoncelli, and E. H. Adelson. Motion illusions as optimal percepts. *Nature neuroscience*, 5(6):598, 2002. 1

[52] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017. 7

[53] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv:1712.04851*, 2017. 2, 6

[54] M. Zolfaghari, K. Singh, and T. Brox. ECO: efficient convolutional network for online video understanding. In *Proc. ECCV*, 2018. 6