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 state-of-the-art accuracy on major video recognition benchmarks, Kinetics, Charades and AVA. Code has been made available at: `https://github.com/facebookresearch/SlowFast`.

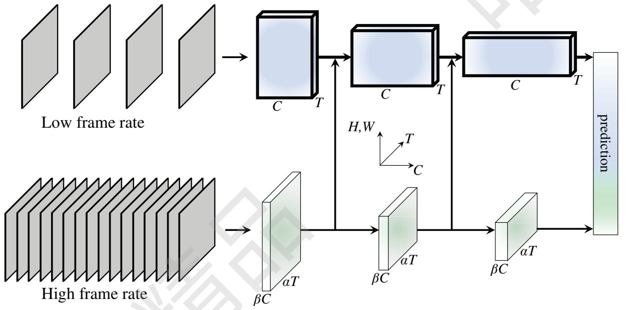

Figure 1. **A SlowFast network** has a low frame rate, low temporal resolution *Slow* pathway and a high frame rate, $\alpha\times$ higher temporal resolution *Fast* pathway. The Fast pathway is lightweight by using a fraction (β, *e.g.*, 1/8) of channels. Lateral connections fuse them.

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 [41, 26]. But what about video signals $I(x, y, t)$? Motion is the spatiotemporal counterpart of orientation [2], 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 [58]. 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 [49, 5]. 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.

There is another well known architecture for video recognition which has a two-stream design [44], but provides conceptually different perspectives. The Two-Stream method [44] 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. In our experiments we observe that the SlowFast network is empirically more effective.

Our method is partially inspired by biological studies on the retinal ganglion cells in the primate visual system [27, 37, 8, 14, 51], 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 responsive to fast temporal changes, but not sensitive to spatial detail or color. P-cells provide fine spatial detail and color, but lower temporal resolution, responding slowly to stimuli. 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.

We evaluate our method on the Kinetics-400 [30], Kinetics-600 [3], Charades [43] and AVA [20] datasets. Our comprehensive ablation experiments on Kinetics action classification demonstrate the efficacy contributed by SlowFast. SlowFast networks set a new state-of-the-art on all datasets with significant gains to previous systems in the literature.

2. Related Work

Spatiotemporal filtering. Actions can be formulated as spatiotemporal objects and captured by oriented filtering in spacetime, as done by HOG3D [31] and cuboids [10]. 3D ConvNets [48, 49, 5] extend 2D image models [32, 45, 47, 24] to the spatiotemporal domain, handling both spatial and temporal dimensions similarly. There are also related methods focusing on long-term filtering and pooling using temporal strides [52, 13, 55, 62], as well as decomposing the convolutions into separate 2D spatial and 1D temporal filters [12, 50, 61, 39].

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 [33], motion boundary histograms [6], and trajectories [53], had shown competitive performance for action recognition before the prevalence of deep learning.

In the context of deep neural networks, the two-stream method [44] exploits optical flow by viewing it as another input modality. This method has been a foundation of many competitive results in the literature [12, 13, 55]. 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.

3. SlowFast Networks

SlowFast networks can be described as a single stream architecture that operates at two different framerates, but we use the concept of pathways to reflect analogy with the biological Parvo- and Magnocellular counterparts. Our generic architecture has a Slow pathway (Sec. 3.1) and a Fast pathway (Sec. 3.2), which are fused by lateral connections to a SlowFast network (Sec. 3.3). Fig. 1 illustrates our concept.

3.1. Slow pathway

The Slow pathway can be any convolutional model (*e.g*., [12, 49, 5, 56]) 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. The 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 [12, 13]. In image object detection, lateral connections [35] are a popular technique for merging different levels of spatial resolution and semantics.

Similar to [12, 35], we attach one lateral connection between the two pathways for every "stage" (Fig. 1). Specifically for ResNets [24], 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.

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		$\underline{5\times7^2}$, 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$_2$	1×1^2, 64 1×3^2, 64 1×1^2, 256	$\times3$	$\underline{3\times1^2}$, 8 1×3^2, 8 1×1^2, 32	$\times3$	*Slow* : 4×56^2 *Fast* : 32×56^2
res$_3$	1×1^2, 128 1×3^2, 128 1×1^2, 512	$\times4$	$\underline{3\times1^2}$, 16 1×3^2, 16 1×1^2, 64	$\times4$	*Slow* : 4×28^2 *Fast* : 32×28^2
res$_4$	$\underline{3\times1^2}$, 256 1×3^2, 256 1×1^2, 1024	$\times6$	$\underline{3\times1^2}$, 32 1×3^2, 32 1×1^2, 128	$\times6$	*Slow* : 4×14^2 *Fast* : 32×14^2
res$_5$	$\underline{3\times1^2}$, 512 1×3^2, 512 1×1^2, 2048	$\times3$	$\underline{3\times1^2}$, 64 1×3^2, 64 1×1^2, 256	$\times3$	*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.

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.

3.4. Instantiations

Our idea of SlowFast is generic, and it can be instantiated with different backbones (*e.g.*, [45, 47, 24]) 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 [12]. 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. Further, 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: Action Classification

We evaluate our approach on four video recognition datasets using standard evaluation protocols. For the action classification experiments, presented in this section we consider the widely used Kinetics-400 [30], the recent Kinetics-600 [3], and Charades [43]. For action detection experiments in Sec. 5, we use the challenging AVA dataset [20].

Training. Our models on Kinetics are trained *from random initialization* ("*from scratch*"), *without* using ImageNet [7] or any pre-training. We use synchronized SGD training following the recipe in [19]. See details in Appendix.

For the temporal domain, we randomly sample a clip (of $\alpha T\times\tau$ 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 [45, 56].

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 [56]. 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).

Datasets. Kinetics-400 [30] consists of ~240k training videos and 20k validation videos in 400 human action categories. Kinetics-600 [3] has ~392k training videos and 30k validation videos in 600 classes. We report top-1 and top-5 classification accuracy (%). We report the computational cost (in FLOPs) of a single, spatially center-cropped clip.

Charades [43] has ~9.8k training videos and 1.8k validation videos in 157 classes in a multi-label classification setting of longer activities spanning ~30 seconds on average. Performance is measured in mean Average Precision (mAP).

4.1. Main Results

Kinetics-400. Table 2 shows the comparison with state-of-the-art results for our SlowFast instantiations using various input samplings ($T\times\tau$) and backbones: ResNet-50/101 (R50/101) [24] and Nonlocal (NL) [56].

In comparison to the previous state-of-the-art [56] our best model provides 2.1% higher top-1 accuracy. Notably, all our results are substantially better than existing results that are also *without ImageNet pre-training*. In particular, our model (79.8%) is **5.9%** absolutely better than the previous best result of this kind (73.9%). We have experimented with ImageNet pretraining for SlowFast networks and found that they perform similar (±0.3%) for both the pre-trained and the train from scratch (random initialization) variants.

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, due to the high temporal resolution yet lightweight Fast pathway. Our cost per spacetime view can be low (*e.g.*, 36.1 GFLOPs), while still being accurate.

The SlowFast variants from Table 2 (with different backbones and sample rates) are compared in Fig. 2 the with their corresponding Slow-only pathway to assess the improvement brought by the Fast pathway. The horizontal axis measures model capacity for a single input clip of 256^2 spatial size, which is proportional to 1/30 of the overall inference cost.

model	flow	pretrain	top-1	top-5	GFLOPs×views
I3D [5]		ImageNet	72.1	90.3	108 × N/A
Two-Stream I3D [5]	✓	ImageNet	75.7	92.0	216 × N/A
S3D-G [61]	✓	ImageNet	77.2	93.0	143 × N/A
Nonlocal R50 [56]		ImageNet	76.5	92.6	282 × 30
Nonlocal R101 [56]		ImageNet	77.7	93.3	359 × 30
R(2+1)D Flow [50]	✓	-	67.5	87.2	152 × 115
STC [9]		-	68.7	88.5	N/A × N/A
ARTNet [54]		-	69.2	88.3	23.5 × 250
S3D [61]		-	69.4	89.1	66.4 × N/A
ECO [63]		-	70.0	89.4	N/A × N/A
I3D [5]	✓	-	71.6	90.0	216 × N/A
R(2+1)D [50]		-	72.0	90.0	152 × 115
R(2+1)D [50]	✓	-	73.9	90.9	304 × 115
SlowFast 4×16, R50		-	75.6	92.1	36.1 × 30
SlowFast 8×8, R50		-	77.0	92.6	65.7 × 30
SlowFast 8×8, R101		-	77.9	93.2	106 × 30
SlowFast 16×8, R101		-	78.9	93.5	213 × 30
SlowFast 16×8, R101+NL		-	**79.8**	**93.9**	234 × 30

Table 2. **Comparison with the state-of-the-art on Kinetics-400**. In the last column, we report the inference cost with a single "view" (temporal clip with spatial crop) × the numbers of such views used. The SlowFast models are with different input sampling ($T \times \tau$) and backbones (R-50, R-101, NL). "N/A" indicates the numbers are not available for us.

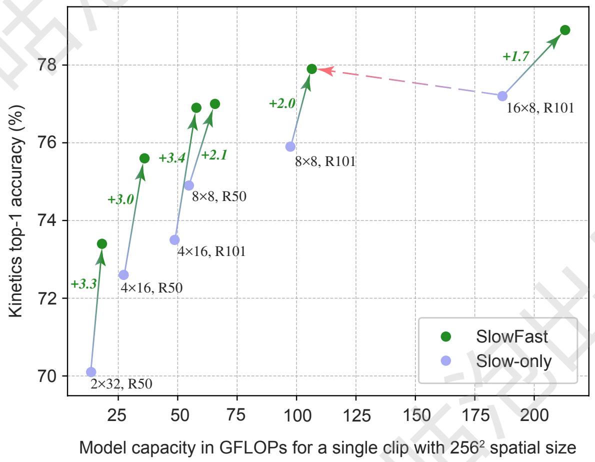

Figure 2. **Accuracy/complexity tradeoff** on Kinetics-400 for the SlowFast (green) *vs*. Slow-only (blue) architectures. SlowFast is consistently better than its Slow-only counterpart in all cases (green arrows). SlowFast provides higher accuracy *and* lower cost than temporally heavy Slow-only (*e.g*. red arrow). The complexity is for a single 256^2 view, and accuracy are obtained by 30-view testing.

Fig. 2 shows that for all variants the Fast pathway is able to consistently improve the performance of the Slow counterpart at comparatively low cost. The next subsection provides a more detailed analysis on Kinetics-400.

Kinetics-600 is relatively new, and existing results are limited. So our goal is mainly to provide results for future reference in Table 3. Note that the Kinetics-600 validation set overlaps with the Kinetics-400 training set [3], and therefore we do *not* pre-train on Kinetics-400. The winning entry [21] of the latest ActivityNet Challenge 2018 [15] reports a best single-model, single-modality accuracy of 79.0%. Our variants show good performance with the best model at 81.8%. SlowFast results on the recent Kinetics-700 [4] are in [11].

model	pretrain	top-1	top-5	GFLOPs×views
I3D [3]	-	71.9	90.1	108 × N/A
StNet-IRv2 RGB [21]	ImgNet+Kin400	79.0	N/A	N/A
SlowFast 4×16, R50	-	78.8	94.0	36.1 × 30
SlowFast 8×8, R50	-	79.9	94.5	65.7 ×30
SlowFast 8×8, R101	-	80.4	94.8	106 × 30
SlowFast 16×8, R101	-	81.1	95.1	213 × 30
SlowFast 16×8, R101+NL	-	**81.8**	**95.1**	234 × 30

Table 3. **Comparison with the state-of-the-art on Kinetics-600**. SlowFast models the same as in Table 2.

model	pretrain	mAP	GFLOPs×views
CoViAR, R-50 [59]	ImageNet	21.9	N/A
Asyn-TF, VGG16 [42]	ImageNet	22.4	N/A
MultiScale TRN [62]	ImageNet	25.2	N/A
Nonlocal, R101 [56]	ImageNet+Kinetics400	37.5	544 × 30
STRG, R101+NL [57]	ImageNet+Kinetics400	39.7	630 × 30
our baseline (Slow-only)	Kinetics-400	39.0	187 × 30
SlowFast	Kinetics-400	42.1	213 × 30
SlowFast, +NL	Kinetics-400	42.5	234 × 30
SlowFast, +NL	Kinetics-600	**45.2**	234 × 30

Table 4. **Comparison with the state-of-the-art on Charades**. All our variants are based on $T \times \tau = 16 \times 8$, R-101.

Charades [43] is a dataset with longer range activities. Table 4 shows our SlowFast results on it. For fair comparison, our baseline is the Slow-only counterpart that has 39.0 mAP. SlowFast increases over this baseline by 3.1 mAP (to 42.1), while the extra NL leads to an additional 0.4 mAP. We also achieve 45.2 mAP when pre-trained on Kinetics-600. Overall, our SlowFast models in Table 4 outperform the previous best number (STRG [57]) by solid margins, at lower cost.

4.2. Ablation Experiments

This section provides ablation studies on Kinetics-400 comparing accuracy and computational complexity.

Slow *vs*. SlowFast. We first aim to explore the SlowFast complementarity by changing the sample rate ($T \times \tau$) of the Slow pathway. Therefore, this ablation studies α, the frame rate ratio between the Fast and Slow paths. Fig. 2 shows the accuracy *vs*. complexity tradeoff for various instantiations of Slow and SlowFast models. It is seen that doubling the number of frames in the Slow pathway increases performance (vertical axis) at double computational cost (horizontal axis), while SlowFast significantly extends the performance of all variants at small increase of computational cost, even if the Slow pathways operates on higher frame rate. Green arrows illustrate the gain of adding the Fast pathway to the corresponding Slow-only architecture. The red arrow illustrates that SlowFast provides higher accuracy *and* reduced cost.

Next, Table 5 shows a series of ablations on the Fast pathway design, using the default SlowFast, $T \times \tau = 4 \times 16$, R-50 instantiation (specified in Table 1), analyzed in turn.

	lateral	top-1	top-5	GFLOPs
Slow-only	-	72.6	90.3	27.3
Fast-only	-	51.7	78.5	**6.4**
SlowFast	-	73.5	90.3	34.2
SlowFast	TtoC, sum	74.5	91.3	34.2
SlowFast	TtoC, concat	74.3	91.0	39.8
SlowFast	T-sample	75.4	91.8	34.9
SlowFast	T-conv	**75.6**	**92.1**	36.1

(a) **SlowFast fusion**: Fusing Slow and Fast pathways with various types of lateral connections throughout the network hierarchy is consistently better than the Slow and Fast only baselines.

	top-1	top-5	GFLOPs
Slow-only	72.6	90.3	27.3
$\beta = 1/4$	75.6	91.7	54.5
$1/6$	**75.8**	92.0	41.8
$1/8$	75.6	**92.1**	36.1
$1/12$	75.2	91.8	32.8
$1/16$	75.1	91.7	30.6
$1/32$	74.2	91.3	28.6

(b) **Channel capacity ratio**: Varying values of β, the channel capacity ratio of the Fast pathway to make SlowFast lightweight.

Fast pathway	spatial	top-1	top-5	GFLOPs
RGB	-	**75.6**	**92.1**	36.1
RGB, β=1/4	*half*	74.7	91.8	34.4
gray-scale	-	**75.5**	**91.9**	**34.1**
time diff	-	74.5	91.6	34.2
optical flow	-	73.8	91.3	35.1

(c) **Weaker spatial input to Fast pathway**: Alternative ways of weakening spatial inputs to the Fast pathway in SlowFast models. β=1/8 unless specified otherwise.

Table 5. Ablations on the Fast pathway design on **Kinetics-400**. 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 256^2. Inference-time computational cost is proportional to this, as a fixed number of 30 of views is used. Backbone: 4×16, R-50.

Individual pathways. The first two rows in Table 5a show the results for using the structure of one individual pathway *alone*. The default instantiations of the Slow and Fast pathway are very lightweight with only 27.3 and 6.4 GFLOPs, 32.4M and 0.53M parameters, producing 72.6% and 51.7% top-1 accuracy, respectively. The pathways are designed with their special expertise if they are used jointly, as is ablated next.

SlowFast fusion. Table 5a shows various ways of fusing the Slow and Fast pathways. As a naïve fusion baseline, 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 counterpart by 0.9%.

Next, we ablate SlowFast models with various lateral connections: time-to-channel (TtoC), time-strided sampling (T-sample), and time-strided convolution (T-conv). For TtoC, which can match channel dimensions, we also report fusing by element-wise summation (TtoC, sum). For all other variants concatenation is employed for fusion.

Table 5a shows that these SlowFast models are *all* better than the Slow-only pathway. With the best-performing lateral connection of T-conv, the SlowFast network is **3.0% better** than Slow-only. We employ T-conv as our default.

Interestingly, the Fast pathway alone has only 51.7% accuracy (Table 5a). 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 5b 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 1.3 GFLOPs (~5% relative), but leads to 1.6% improvement.

model	pre-train	top-1	top-5	GFLOPs
3D R-50 [56]	ImageNet	73.4	90.9	36.7
3D R-50, recipe in [56]	-	69.4	88.6	36.7
3D R-50, our recipe	-	73.5	90.8	36.7

Table 6. **Baselines trained from scratch**: Using the same network structure as [56], our training recipe achieves comparable results *without* ImageNet pre-training.

Weaker spatial inputs to Fast pathway. Further, we experiment with using different *weaker* spatial inputs to the 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 5c 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 [27, 37, 8, 14, 51].

We believe both Table 5b and Table 5c convincingly show that the *lightweight* but temporally *high-resolution* Fast pathway is an effective component for video recognition.

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 [56], using our large-scale SGD recipe trained from scratch.

Table 6 shows the comparisons using this 3D R-50 baseline architecture. We observe, that our training recipe achieves *comparably good* results as the ImageNet pre-training counterpart reported by [56], while the recipe in [56] is not well tuned for directly training from scratch. This suggests that our training system, as the foundation of our experiments, has no loss for this baseline model, despite not using ImageNet for pre-training.

5. Experiments: AVA Action Detection

Dataset. The AVA dataset [20] 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. Note the difficulty in AVA lies in action detection, while actor localization is less challenging [20]. There are 211k training and 57k validation video segments in AVA v2.1 which we use. We follow the standard protocol [20] 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 [40] 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 [17] at the last feature map of res$_5$. We first extend each 2D RoI at a frame into a 3D RoI by replicating it along the temporal axis, similar to the method presented in [20]. Subsequently, we compute RoI features by RoIAlign [22] 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 [20, 46, 29]. 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* [18]. It is a Faster R-CNN with a ResNeXt-101-FPN [60, 35] backbone. It is pre-trained on ImageNet and the COCO human keypoint images [36]. 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 > 0.8, 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 [19] 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 are used as the samples for training. The input is instantiation-specific $\alpha T \times \tau$ frames of size 224×224.

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 [40].

model	flow	video pretrain	val mAP	test mAP
I3D [20]		Kinetics-400	14.5	-
I3D [20]	✓	Kinetics-400	15.6	-
ACRN, S3D [46]	✓	Kinetics-400	17.4	-
ATR, R50+NL [29]		Kinetics-400	20.0	-
ATR, R50+NL [29]	✓	Kinetics-400	21.7	-
9-model ensemble [29]	✓	Kinetics-400	25.6	21.1
I3D [16]		Kinetics-600	21.9	21.0
SlowFast		Kinetics-400	26.3	-
SlowFast		Kinetics-600	26.8	-
SlowFast, +NL		Kinetics-600	27.3	**27.1**
SlowFast*, +NL		Kinetics-600	**28.2**	-

Table 7. **Comparison with the state-of-the-art on AVA v2.1**. All our variants are based on $T \times \tau = 8 \times 8$, R101. Here "*" indicates a version of our method that uses our region proposals for training.

model	flow	video pretrain	val mAP	test mAP
SlowFast, 8×8		Kinetics-600	29.0	-
SlowFast, 16×8		Kinetics-600	29.8	-
SlowFast++, 16×8		Kinetics-600	**30.7**	-
SlowFast++, ensemble		Kinetics-600	-	**34.3**

Table 8. **SlowFast models on AVA v2.2**. Here "++" indicates a version of our method that is tested with multi-scale and horizontal flipping augmentation. The backbone is R-101+NL and region proposals are used for training.

5.1. Main Results

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 [20], and +1.7 mAP for ATR in [29]. In contrast, our baseline improves by the Fast pathway by +5.2 mAP (see Table 9 in our ablation experiments in the next section). 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 26.3 mAP using only Kinetics-400 pre-training. This is **5.6** mAP higher than the previous best number under similar settings (21.7 of ATR [29], single-model), and **7.3** mAP higher than that using no optical flow (Table 7).

The work in [16] pre-trains on the larger Kinetics-600 and achieves 21.9 mAP. For fair comparison, we observe an improvement from 26.3 mAP to 26.8 mAP for using Kinetics-600. Augmenting SlowFast with NL blocks [56] increases this to 27.3 mAP. We train this model on train+val (and by 1.5× longer) and submit it to the AVA v2.1 test server [34]. It achieves **27.1 mAP** single crop test set accuracy.

By using predicted proposals overlapping with ground-truth boxes by IoU > 0.9, in addition to the ground truth boxes, for training we achieve **28.2 mAP** single crop validation accuracy, a new state-of-the-art on AVA.

Using the AVA v2.2 dataset (which provides more consistent annotations) improves this number to 29.0 mAP (Table 8). The longer-term **SlowFast**, 16×8 model produces 29.8 mAP and using multiple spatial scales and horizontal flip for testing, this number is increased to **30.7 mAP**.

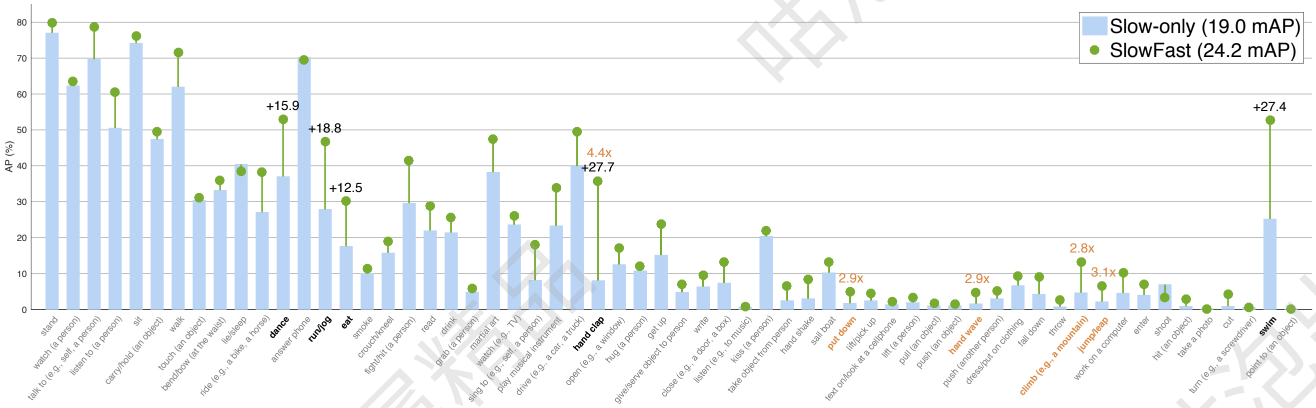

Figure 3. **Per-category AP on AVA**: a Slow-only baseline (19.0 mAP) *vs*. its SlowFast counterpart (24.2 mAP). The highlighted categories are the 5 highest absolute increase (**black**) or 5 highest relative increase with Slow-only AP $>$ 1.0 (**orange**). Categories are sorted by number of examples.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

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

Table 9. **AVA action detection baselines**: Slow-only *vs*. SlowFast.

Finally, we create an ensemble of 7 models and submit it to the official test server for the ActivityNet challenge 2019 [1]. As shown in Table 8 this entry (**SlowFast++**, ensemble) achieved **34.3 mAP** accuracy on the test set, ranking first in the AVA action detection challenge 2019. Further details on our winning solution are provided in the corresponding technical report [11].

5.2. Ablation Experiments

Table 9 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.

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.

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.

A. Appendix

Implementation details. We study backbones including ResNet-50 and the deeper ResNet-101 [24], optionally augmented with non-local (NL) blocks [56]. For models involving R-101, we use a scale jittering range of [256, 340]. The $T{\times}\tau = 16{\times}8$ models are initilaized from the 8×8 counterparts and trained for half the training epochs to reduce training time. For all models involving NL, we initialize them with the counterparts that are trained without NL, to facilitate convergence. We only use NL on the (fused) Slow features of res$_4$ (instead of res$_3$+res$_4$ [56]).

On *Kinetics*, we adopt synchronized SGD training in 128 GPUs following the recipe in [19], 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 [23]. We train with Batch Normalization (BN) [28] with BN statistics computed within each 8 clips. We adopt a half-period cosine schedule [38] of learning rate decaying: the learning rate at the n-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\max}}\pi)+1]$, 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 [19] in the first 8k iterations. For Kinetic-400, 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 [25] of 0.5 is used before the final classifier layer.

For *Kinetics-600*, we extend the training epochs (and schedule) by 2× and set the base learning rate η to 0.8.

For *Charades*, we fine-tune the Kinetics models. A per-class sigmoid output is used to account for the mutli-class nature. We train on a single machine for 24k iterations using a batch size of 16 and a base learning rate of 0.0375 (Kinetics-400 pre-trained) and 0.02 (Kinetics-600 pre-trained) with 10× step-wise decay if the validation error saturates. For inference, we temporally max-pool scores [56].

References

[1] ActivityNet-Challenge. http://activity-net.org/challenges/2019/evaluation.html. 8

[2] 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

[3] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 2, 4, 5

[4] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 5

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

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

[7] 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

[8] 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

[9] 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. 5

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

[11] C. Feichtenhofer, H. Fan, J. Malik, and K. He. SlowFast networks for video recognition in ActivityNet challenge 2019. http://static.googleusercontent.com/media/research.google.com/en//ava/2019/fair_slowfast.pdf, 2019. 5, 8

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

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

[14] 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

[15] 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. 5

[16] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for AVA. *arXiv:1807.10066*, 2018. 7

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

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

[19] 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, 8

[20] 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, 4, 7

[21] 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. 5

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

[23] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. CVPR*, 2015. 8

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2, 3, 4, 8

[25] 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. 8

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

[27] 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

[28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 8

[29] 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. 7

[30] 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

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

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

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

[34] Leaderboard:ActivityNet-AVA. http://activity-net.org/challenges/2018/evaluation.html. 7

[35] 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

[36] 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

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

[38] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 8

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

[40] 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

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

[42] G. A. Sigurdsson, S. K. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017. 5

[43] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 4, 5

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

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

[46] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *ECCV*, 2018. 7

[47] 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

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

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

[50] 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, 5

[51] 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

[52] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE PAMI*, 2018. 2

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

[54] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. In *Proc. CVPR*, 2018. 5

[55] 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

[56] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proc. CVPR*, 2018. 2, 4, 5, 6, 7, 8

[57] X. Wang and A. Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 5

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

[59] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Compressed video action recognition. In *CVPR*, 2018. 5

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

[61] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv:1712.04851*, 2017. 2, 5

[62] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 2, 5

[63] M. Zolfaghari, K. Singh, and T. Brox. ECO: efficient convolutional network for online video understanding. In *Proc. ECCV*, 2018. 5