

Adaptive Intermediate Representations for Video Understanding

Juhana Kangaspunta, AJ Piergiovanni, Rico Jonschkowski, Michael Ryoo, Anelia Angelova
 Google Research

{juhana,ajpiergi,rjon,mryoo,anelia}@google.com

Abstract

A common strategy to video understanding is to incorporate spatial and motion information by fusing features derived from RGB frames and optical flow. In this work, we introduce a new way to leverage semantic segmentation as an intermediate representation for video understanding and use it in a way that requires no additional labeling. Second, we propose a general framework which learns the intermediate representations (optical flow and semantic segmentation) jointly with the final video understanding task and allows the adaptation of the representations to the end goal. Despite the use of intermediate representations within the network, during inference, no additional data beyond RGB sequences is needed, enabling efficient recognition with a single network. Finally, we present a way to find the optimal learning configuration by searching the best loss weighting via evolution. We obtain more powerful visual representations for videos which lead to performance gains over the state-of-the-art.

1. Introduction

In the task of action recognition, the model is given only a video-level action class as a supervision signal. From this, the model has to implicitly learn a wide variety of concepts from the input data: objects, people, motion and their relationships through time. Thus using explicit intermediate representations can aid in the learning process. One notable form of intermediate representation, optical flow, has been used extensively in two-stream architectures [52, 4, 15]. Optical flow is often pre-computed using an algorithm optimized for the quality of optical flow itself, rather than the downstream video understanding task.

As an addition to motion information, knowledge about the presence and location of people and objects should also be highly relevant in understanding the activity in a video sequence [24]. We propose a new way to use semantic segmentation as an intermediate representation learned by the network (Figure 1). Semantic segmentation is learned along with the main video understanding task using annotations

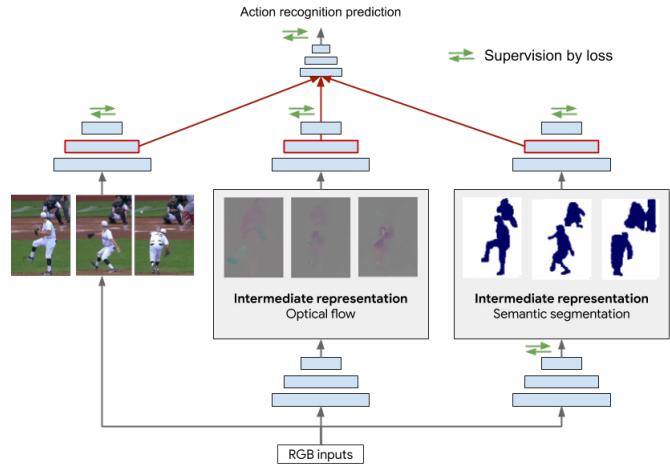


Figure 1. Intermediate representations from optical flow (middle) and semantic segmentation (right) are learned alongside the main RGB task and their losses are simultaneously optimized for improving the performance of the main video understanding task. Merging of streams is highlighted with red borders and supervision by loss is highlighted in green.

generated by an off-the-shelf teacher model that was trained on static images unrelated to the task. Therefore, the annotations are readily available and require no additional labeling effort.

How can one best take advantage of these intermediate representations? Enforcing *rigid* intermediate representations, for example by training and freezing a model for optical flow, prevents these representations from learning information that is specific to the given task, e.g., when some types of motion are more informative than others. To allow for such flexibility, we encourage intermediate representations in a “soft” way through auxiliary learning objectives that are optimized jointly with the video understanding task. Importantly, after training the network and learning the corresponding intermediate representations, the network requires RGB-only inputs at inference.

As all intermediate representations are jointly learned with the final task by optimizing a weighted sum of these

objectives, the weights of their losses determine the importance of each stream’s feature representations and intermediate losses. In addition to optimizing these representations, we also optimize the loss weights for end-to-end action recognition performance using evolution to maximize mutual usefulness of each loss (Figure 1). We call the approach ‘AIRStreams’, short from Adaptive Intermediate Representation Streams, as it provides a framework for using different streams of intermediate representations in an adaptive manner for video understanding.

Our contributions are threefold: (a) We propose to incorporate semantic segmentation as a learned intermediate representation for video understanding that is complementary to optical flow, improves performance, and does not require additional labeling. (b) We provide a novel framework that combines different intermediate representations in a way that automatically optimizes them with respect to the final task. The networks in our framework require RGB inputs only. c) We use an evolutionary strategy to further balance the learning of these representations for final action recognition task.

Our experimental evaluation shows that AIRStreams performs well across three major action recognition datasets. In an extensive analysis of the effect of adaptive intermediate representations and the proposed approach, we find that the extra modalities help, more specifically: (i) Adding the semantic segmentation stream significantly improves performance, and can do so without pre-training. (ii) Intermediate representations are more useful when optimized jointly with the task by allowing the end-to-end gradient to affect the intermediate representation. (iii) Optimizing the weights of the intermediate losses for the task provides additional improvements in performance. (iv) Combining the aforementioned concepts, AIRStreams outperforms or is competitive with the best current video understanding models in multiple action recognition datasets.

While incorporating extra modalities requires more computation, we opted for an orthogonal approach where we use a lightweight backbone, but utilize the compute to pack additional modalities, which bring in new valuable information and large improvements. At inference, our model is very efficient, taking 17 GFLOPs, as opposed to SlowFast [13] which is 213 GFLOPs, and I3D [4] - 216 GFLOPs.

2. Previous work

Understanding videos has had many successful approaches applying a number of deep architectures [55, 53, 52, 14, 57, 27, 64, 4, 54, 11, 13]. Optical flow, although a product of the raw video stream, has been demonstrated to be of high value to video recognition [48]. Two-stream networks [52, 4, 15] are particularly interesting constructs with RGB and flow streams, in which optical flow is used as a separate input to a deep network, instead of relying on the RGB sequence to learn flow information on its own. Early ver-

sions, e.g., Wang et al. [58], used an RGB difference stream and a warped RGB stream after applying flow. While the majority of two-stream networks apply late-fusion, by fusing features from the two streams before the final output, the work of [14] proposes fusion across layers from multiple streams.

Slow-fast networks [13] consider two individual RGB-only streams, which jointly process information at different temporal resolutions. They also advocate for fixed connections between parallel layers. We here propose the semantics stream and to optimize the performance and mutual benefit of multiple streams working jointly. The above-mentioned works can benefit from AIRStreams by incorporating extra semantic streams, by training end-to-end, and learning the importance of the loss contributions of each of the streams.

Some previous approaches have successfully used object information for video understanding, e.g. [65, 26, 25, 44, 24, 61, 35, 1, 60]. Sigurdsson et al. [51] demonstrates the potential of using objects via object labels oracles. Other works have explored multi-stream networks with various inputs. For example, PoTion [6] proposes a complementary pose representation. Chained Multi-stream Networks [69] similar to this work uses multiple different input features each optimized for recognition. This paper instead uses other features, like objects, to learn representations within the network so that during inference they are not needed. While we are not aware of prior work using semantics as a jointly optimized separate stream, Diba et al. [9] is most related to ours. They utilize a 2D stream which is built from image pretraining on an independent dataset. It is infused into a 3D stream which processes the main video information.

Multi-task learning approaches [10, 46, 67, 28, 18, 37] train several tasks in parallel (each one to a corresponding loss) and have been successfully shown to use multiple tasks for mutual benefit, e.g., for joint feature embedding learning. In another aspect of multi-task learning, various modalities (or streams) are considered as tasks but the focus is on training towards separate output goals, usually called ‘heads’, while maintaining shared representations, e.g., learning by supervising for depth, surface normals etc., while learning image segmentation. Other works also do joint optimization in the video understanding context, e.g. SegFlow [5] of optical flow and video object segmentation. Unlike the multi-task learning approaches listed above, we train the intermediate representations to help the main action recognition task and have the main task be explicitly dependent of the intermediate representations, not only indirectly dependent through feature sharing. Previous works have found that optimizing flow in conjunction with recognition improves performance [39, 12].

Architecture search [70, 71, 31, 2, 32, 45, 19] has proven to be highly successful for visual understanding. Recent work on architecture search for videos [38, 41] has demon-

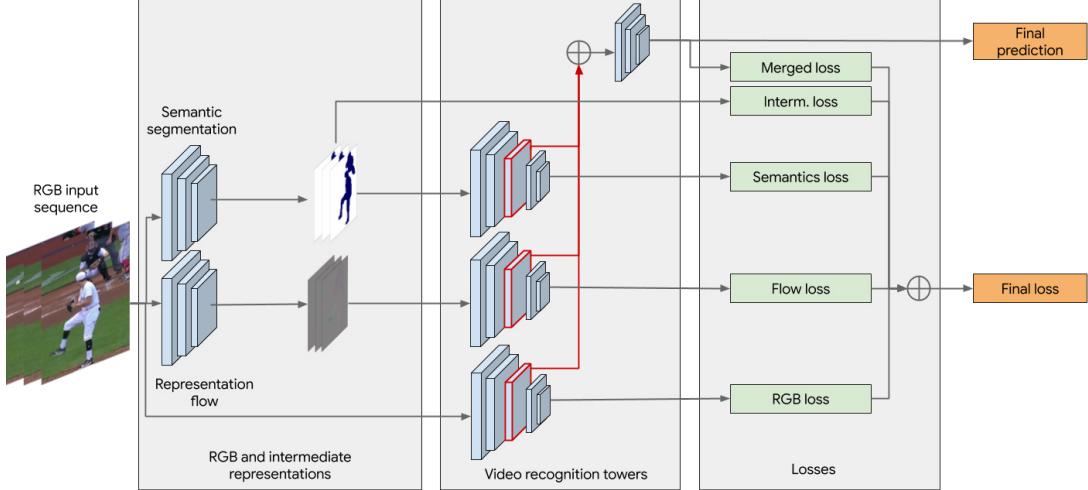


Figure 2. AIRStreams overview: The input RGB video gives rise to intermediate representations based on semantics (top stream), where common objects are automatically segmented, and optical flow (middle stream). Together with the RGB stream (bottom stream), each of the representations are passed into their own video recognition towers. Features extracted from the middle of each video recognition tower are merged (highlighted with red borders) and passed through an additional convolutional tower to produce the final action recognition prediction. The losses (highlighted in green) of each stream and the intermediate losses are simultaneously optimized for improving the performance of the main video understanding task.

strated promise. Evolutionary search is also used for selecting a subset of losses for unsupervised feature learning [42]. Our work too uses evolutionary search for loss combinations, where evolution is used as a versatile and easy-to-apply optimization technique, especially for non-differentiable tasks such as ours. While other optimization techniques could substitute evolution, we are not aware of prior work which proposed to optimize intermediate representations for video understanding and jointly optimize all representations contributions to the final task.

3. Learning with adaptive intermediate representations

In this section, we describe our video understanding model, the different modules it consists of, and the loss functions for training.

3.1. Model overview

The AIRStreams model consists of three independent streams of computation (Figure 2). An RGB stream is optimized only towards the final action recognition task without any intermediate constraints. The two other streams, in addition to being optimized towards the final task, each produce an intermediate output: optical flow and semantic segmentation. Here we first explain the overall model architecture, the individual streams in detail, how the features from the streams are merged together, what losses are used during training and how these losses are weighted.

Model architecture. The RGB video input and each individual intermediate representation are processed by a separate convolutional action recognition tower, e.g., (2+1)D ResNet [54]. These towers are shown in Figure 2 in the middle section. Different tower backbones can be used and we experiment with a lightweight alternative in this paper. Each of the individual towers attempt to predict the final task and are supervised by separate classification losses. The feature maps of each of the towers are also fused together to produce a merged stream, which has a classification loss as well. The merged stream produces the final output that is used in evaluation. More advanced connectivity between the towers is also possible, e.g., as in [14] or in RandWire Networks [63].

Since it is not immediately clear which configuration will work well, we define below a set of losses (Section 3.2) and propose to further optimize them automatically via evolution (Section 3.3).

The way in which each intermediate representation is produced is specific to the representation. The framework is extendable to a variety of different representations, which can be unsupervised or supervised. Below, we describe the computation of the intermediate representations in detail.

RGB Stream. The RGB stream takes as input a sequence of RGB images from a video and applies a convolutional tower to it. This is the standard approach for many video understanding tasks.

Segmentation Stream. One of the key aspects of AIRStreams is using semantic segmentation as an interme-

diate representation. Our rationale is that the presence and location of people and semantic objects hold key information for the activities at hand. Importantly, the interaction of the features with others (flow or RGB) as a result of the presence of the semantic stream in the AIRStreams architecture will provide opportunity for better learning of the final actions.

To produce the semantic segmentation representation, we use a Dilated ResNet-18 [66] with output stride 16 applied frame-by-frame on the input RGB sequence. We use pixelwise softmax cross-entropy as the semantic segmentation loss using model-generated annotations as ground truth as explained below. The pre-softmax activations of the semantic segmentation network are then used as the input to a subsequent representation-specific action recognition tower.

Frame-by-frame annotation of object segments is very expensive and therefore large-scale action recognition datasets do not provide such annotations. The semantic annotations used to supervise the network explained above are obtained with an off-the-shelf algorithm (trained on an independent dataset of static images, e.g., on MS-COCO dataset [30]), which are readily available, and incurs no further labeling effort or cost. Notably, this teacher model used to generate the semantic segmentation annotations is a much larger and much more complicated Mask R-CNN model [17] compared to the network generating semantic segmentation inside AIRStreams.

Many of the classes present in MS-COCO are likely not encountered in action recognition datasets, e.g., giraffe, kite, etc., but a handful of classes will be useful and prevalent in the videos such as people, furniture and objects of daily life. We conjecture that the semantic features will encode valuable information, as well as directly transfer information for some of the classes that are jointly encountered - e.g., person, chair, etc.

Optical Flow Stream. While many video understanding models use a single-stream, two-stream models [52, 4] (RGB and optical flow), have become increasingly popular. The rationale is that flow directly provides information about motion in a much more efficient way than extracting it from the raw RGB sequence. This approach has been shown to be very beneficial. Our approach includes an optical-flow based intermediate representation as well.

Our use of optical flow has two key aspects. First, in order to learn and fine-tune the flow representation for the final task, we need a fully differentiable version of flow. We show later in experiments that backpropagating from the final classification through the intermediate representation computations is important. For a fully differentiable flow we use Representation Flow [43], which is trained on-the-fly and can be backpropagated fully. Representation flow mimics the implementation of the variational flow algorithms [3]

via fully differentiable learnable layers, minimizing the total variational energy. We chose this over approaches like FlowNet/ActionFlowNet [22, 39] as this requires no pre-training to obtain good flow representations.

Second, similar to the semantic segmentation representation, the optical flow is passed into its own action recognition tower. Unlike regular two-stream towers, which all apply late fusion, features from this video recognition tower are fused early with the other video recognition towers as shown in Figure 2 and supplementary materials, as well as given its own video classification loss.

While we use representation flow for our experiments, other versions of flow can be incorporated.

Merging streams. The features from each of the three action recognition towers (flow, semantic segmentation and RGB) are combined via averaging at a pre-defined level. We merge after the 3rd block and our net has a total of 6 blocks. After the features have been merged, the rest of the video recognition tower for the merged stream is identical in structure for blocks 4 to 6 of the individual feature towers. Similarly as for the feature action recognition towers, the merged tower is supervised by the final action recognition task. Merging at more than one levels (including the common late-fusion) is also possible and can easily be added to the AIRStreams framework as separate losses.

3.2. Losses

In AIRStreams, we propose to use multiple independently-weighted losses, described below.

Activity recognition losses. Each action recognition tower, one consuming flow, one consuming semantic segmentation, one consuming RGB and one consuming the merged features, has an individual softmax cross-entropy (SCE) loss L_{rgb} , L_{flow} , $L_{semantic}$, L_{merged} with respect to the final action recognition target: $L_* = SCE(Y_*, Y_{target})$, where Y_{target} are the target labels for the final activity recognition task, and the $*$ denotes each of the output streams' logits, i.e. rgb , $flow$, $semantic$, $merged$.

Intermediate representation losses. Additionally, each intermediate representation can have its own loss that is optimized for the specific task. In this work, we have semantic segmentation and optical flow as intermediate representations, from which only semantic segmentation requires its own loss. We use standard pixelwise softmax cross-entropy (SCE) between the logits $Y_{interm_semantic}$ and the 'target' semantic labels $Y_{labels_semantic}$ as the loss for semantic segmentation L_{interm} . It is applied per frame. Note that the 'ground truth' labels used for this loss are generated by an off-the-shelf segmentation model trained on the MS-COCO

dataset [30] with 90 classes. Therefore, we do not require pixelwise labeling of the action recognition datasets.

$$L_{interm} = SCE(Y_{interm_semantic}, Y_{labels_semantic}). \quad (1)$$

Final loss. Putting it all together, we have a number of losses from each of the action recognition streams (flow, semantic segmentation, RGB and merged) and an intermediate representation loss for semantic segmentation. The losses are naturally weighted by parameters $\lambda_{rgb}, \lambda_{flow}, \lambda_{semantic}, \lambda_{merged}, \lambda_{interm}$ which determine the relative importance of each sub-task.

$$L = \lambda_{rgb}L_{rgb} + \lambda_{flow}L_{flow} + \lambda_{semantic}L_{semantic} + \lambda_{merged}L_{merged} + \lambda_{interm}L_{interm} \quad (2)$$

3.3. Evolution of loss weighting

AIRStreams can be viewed as a collection of losses which train jointly for the final action recognition task. With that, its de-facto topology can depend on which losses are switched on or off (via their weights). For example, presence or absence of certain losses can make a big difference, as Section 4 shows accuracy of only 42.82 is achieved on HMDB for uniform weights (all weights are 1s) vs. 50.02 when the flow loss and the semantic losses are both 0s (all others 1s).

We propose to learn the combination of losses via *evolution* of weights for the purposes of achieving a higher performance on the main task [42]. Evolution [16, 45] is a well established technique in which a random set of individuals in a population (here combination of weights) are generated, after which random mutations are applied. Their performance is tested and the best performing combination of weights are preserved for the next iterations.

For AIRStreams, we use a tournament selection evolutionary strategy [16]. For this experiment, we use a population size 10 and tournament size 3. At each evolutionary step, we mutate an individual weighting by randomly selecting one weight and assigning it a random value between 0 and 1. The evolution evaluates 100 different loss weightings in total.

This approach can be viewed as an automatic optimization of weights, in lieu of parameter tuning. We note that the number of losses is likely to increase with more advanced tasks and representations, thus tuning them by hand or exhaustive search will be a prohibitively expensive task in such scenarios.

3.4. Inference

We measure end-to-end performance for the final action recognition task by using the output of the merged stream. We note that RGB is the only input and no pre-computed flow or semantics information is needed. Despite computing

multiple towers corresponding to ‘modalities’, our model is very fast with 17 GFLOPs per frame, much fewer than Slow-Fast [13] with 213 GFLOPs, and I3D [4] with 216 GFLOPs.

3.5. Network backbone

Instead of using a standard ResNet-50 or ResNet-101 architectures, we here experimented with a more lightweight model, so that we can take advantage of multiple towers. The network architecture backbone we are using is a Tiny Video Network [40], and open source code [21, 7], which is comparable to a ResNet-18. The network architecture itself contains 6 blocks with 1d convolution and spatial convolutional layers, and efficient blocks, such as context gating [64] and squeeze-and-excite [20]; it is described in more detail in the supplementary materials. This model is of lesser capacity to standard architectures for video understanding e.g., ResNet-50, but it is more efficient to train so we prefer it for faster convergence. Importantly, it allows us to perform well without pre-training on small datasets. Of note is that it is less powerful than a ResNet-50, as evidenced by the performance without pre-training (see baseline models in Table 3). Despite that, within the AIRStreams framework, it outperforms the best state-of-the-art approaches on very challenging datasets (as seen later, e.g., in Table 4).

4. Experiments and results

First we introduce the datasets we used. In the subsequent sections, we investigate multiple ways intermediate representations impact video classification tasks and how different components contribute to the performance of AIRStreams.

4.1. Datasets

Moments in Time (MiT). The MiT dataset [36] is a large scale dataset which consists of about 1M videos, each one approximately 3 seconds long. It has 339 classes, 802,264 training, 33,900 test videos.

Charades dataset. The Charades [50] dataset is a multi-class multi-label dataset of 157 activities. It has 7,985 training and 1,863 test videos. We follow the standard protocols for evaluation for multi-class multi-label setting and thus use a multi-class sigmoid cross-entropy classification loss instead of a softmax cross-entropy loss for this dataset.

Toyota Smarthomes dataset The Toyota Smarthomes dataset [8] consists of daily lives activities, e.g., making coffee, reading, watching TV. It contains 16,115 videos and 31 action classes.

HMDB Video dataset. HMDB [29] is a smaller video recognition dataset with about 7,000 videos and 51 activities. We use this dataset for our ablation experiments.

4.2. Analysis of AIRStreams

We start by investigating various aspects of learning adaptive intermediate representations for AIRStreams.

4.2.1 Intermediate representations help the final task

We first examine how additional intermediate representations help in the action recognition task. We evaluate the use of flow, semantics and both, in addition to an RGB stream.

Table 1 shows the results. We can see that each of the intermediate representations help, and that the use of each of the streams alone is highly beneficial. When combined, they increase the overall performance even further for an overall improvement of above 8.3% in absolute value, which constitutes about 20% relative improvement. These results are obtained when respective intermediate representation are optimized for the downstream task at hand. They are also conducted without pre-training from another dataset. Interestingly, we can see that the effects of the semantic stream are very positive, despite the fact that the semantic labels are trained on an independent dataset from an off-the-shelf algorithm, and not related to the data on hand.

4.2.2 Intermediate representations should be optimized for the final task

One of the key aspects of the AIRStreams architecture is learning adaptive intermediate representations jointly with the task at hand. This is in contrast to many prior works, which use pre-computed versions of flow. In this section we investigate whether optimizing intermediate representations for the final task is of importance. In order to evaluate this, we test two configurations. In both configurations, we train the intermediate representations jointly with the main task. However, in one configuration, we don't let gradients propagate from the final action recognition classification losses through the intermediate representation computations. This means that the intermediate representations are independent from final task and, for example, the semantic segmentation network is only optimizing towards the semantic segmentation pixelwise cross-entropy loss. In the other configuration, we let gradients propagate freely. This means that the intermediate representations are allowed to adjust for the final task, balancing the rigidity of the intermediate form with the needs of the downstream task.

Table 2 shows the corresponding ablation study. We see that full optimization of the intermediate representations (or combination of them) for the task at hand is better than using pre-computed intermediate representations. Each task is able to gain from aligning its intermediate features to the final task at hand. While this result is not surprising, it solidifies the insight that one needs to optimize jointly, rather than pre-compute. Importantly, backpropagating with the semantics labels brings more improvements, even though the semantic information is transferred from another task. Figure 4 shows why – some objects which were missed by the segmentation model, can be recovered during training.

Method	Acc. (%)	Improvement
RGB	41.72	-
RGB + Semantics	46.32	+4.6
RGB + Flow	47.66	+5.94
RGB + Semantics + Flow	50.02	+8.3

Table 1. Ablation study. Intermediate representations improve performance, for both semantics and flow intermediate representations. Combining them helps most, with an 8.3% improvement. HMDB dataset, training is done from raw data (no pretraining), to isolate other effects. The middle column shows classification accuracy on the dataset. The right column shows the corresponding improvements from RGB only.

Learning mode	Stop gradient at interm. repr.	Propagate gradient
RGB (no interm. repr.)	41.72	-
RGB + Semantics	43.57	46.32
RGB + Flow	46.84	47.66
RGB + Semantics + Flow	47.76	50.02

Table 2. Ablation study for intermediate representations for HMDB. All intermediate representations are beneficial (compare column-wise to RGB only). For every intermediate representation, allowing it to be learned end-to-end is more beneficial than using a pre-specified representation (compare row-wise). Accuracy in % shown for both columns.

Dataset	No aux. losses	Fixed weights	Evolved weights
HMDB (no pretr.)	41.72	50.02	50.53
Charades (no pretr.)	28.56	35.05	35.45

Table 3. Ablation study: Performance when evolving weights for balancing all losses from all intermediate representations, compared to manually tuned loss combinations. Accuracy(%) for HMDB and mAP(%) for Charades are shown.

4.2.3 Learning how to combine intermediate feature representations.

From the above-mentioned ablation results, we learned that (i) intermediate representation are beneficial and their combination is of highest value and (ii) they should be optimized jointly with the final task. The above experiments are run with fixed weights of 1 on ‘merged’, ‘interm’ and ‘RGB’ and 0s for flow and semantic tower, which is already a very successful model (close to the evolved coefficients). In this subsection, we run ablation experiments to evaluate the evolution of the weights of auxiliary losses via the evolution procedure. More specifically we compare: the baseline model without intermediate representations, the model with all in-



Figure 3. Samples from the learned intermediate representations for optical flow. The optical flow is learned as a part of the overall training process and gradients are backpropagated from video classification losses allowing the flow to be tuned for the end task.

termediate representations with fixed loss weights, and the full AIRStreams evolved model, where the weights are additionally evolved for the task at hand.

Table 3 compares the performance from weights obtained by evolution vs fixed weights vs a baseline with no intermediate weights. Here the gradients are allowed to flow so that full learning and optimization is done. We observe that the evolved weights give further boost, obtaining the best performance. The improvement by evolution is small but consistent, as also observed across our experiments. This is due to the initial good choice of fixed coefficients; other trivial combinations are not as good, for example using **uniform weights**, achieves 42.82 on HMDB, which is above the baseline, but much worse than 50.02 and 50.53, with evolution. We note that the evolved weights applied to the Charades dataset are evolved on a different dataset (MiT). In summary, for both datasets, AIRStreams improves performance over the baseline by large margins. The best evolved loss weights are displayed in Figure 5.

4.2.4 Model visualizations

Figures 3 and 4 visualize the learned intermediate representations. Figure 4 shows intermediate representations for semantics. The visualization shows the output of the teacher model as well as the intermediate representations learned as a part of the AIRStreams training. In order to show the visualization, we use an argmax over the logits, whereas during training, actual logits are passed to the video recognition network. We observe that the learned representations are often less sharp, but in some cases the training is able to correct errors in the segmentation algorithm (mostly by proposing a missing segment, as in the bottom left example). We conjecture that letting gradients flow through from the final action recognition task, the network is able to learn additional elements not present in the frame-by-frame data from the teacher model.

Figure 5 visualizes the loss coefficients learned from the evolutionary search. The figure shows the learned top-5 co-



Figure 4. Samples from the learned intermediate representations for semantics. These are the class labels perceived at the input point of the semantic tower. Each panel contains image, learned segmentation, and ‘ground truth’, the latter being trained on another dataset and may miss objects or be incorrect. The learned segments are less sharp, but end-to-end training is able to correct errors in the ‘ground truth’ by recovering full segments, e.g., bottom left. Interestingly, the majority of segments are persons (dark blue), other classes may appear too. Best viewed in color.

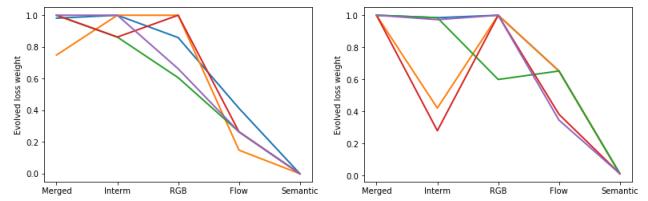


Figure 5. Learned best coefficients after evolution. Evolved on MiT (left), HMDB (right).

efficients on HMDB and MiT. As seen, the evolution tends to assign larger weights to merged and the main RGB towers, and to the semantics intermediate representation loss, whereas smaller weights to the flow and semantics action recognition towers. While multiple evolution runs will produce different sets of coefficients, the final performance is similar for top performing evolved coefficients. For example, accuracy on MiT with top coefficients evolved on MiT is 34.51 vs 34.44 when evolved on HMDB.

4.3 Comparison to previous methods

Here we compare AIRStreams to the state-of-the-art on various action recognition datasets.

4.3.1 Moments in Time dataset

We evaluate on the challenging large-scale Moments-in-time dataset [36], comparing to prior methods including 2-stream methods (RGB and flow), most of which use pre-training (Table 4). Our approach, without using pre-training, outperforms all state-of-the-art methods. We further note that AIRStreams outperforms the powerful AssembleNet [47] which additionally pre-trained from the large Kinetics video dataset, whereas our semantics stream is obtained from a non-related image dataset. For direct comparison we also show (RGB+Flow)-only AIRStreams,

Method	Mode	Acc. (%)
TSN [58]	RGB	24.11
ResNet50-ImageNet [4] (RN50)	RGB	27.16
TRN-Multiscale [68]	RGB	27.20
Spatio-temporal attention[34]	RGB	27.86
TSN-2Stream [58]	RGB+Flow	25.32
TRN-Multiscale [68]	RGB+Flow	28.27
I3D [4]	RGB+Flow	29.51
EvaNet [41]	RGB+Flow	30.5
EvaNet [41] (Ensemble)	RGB+Flow	31.8
AssembleNet [47] RN50	RGB+Flow	31.41
AssembleNet [47] RN50	RGB+Flow (+Kin.)	33.91
AssembleNet [47] RN101	RGB+Flow (+Kin.)	34.27
AIRStreams (Ours)	RGB+Flow	33.53
AIRStreams (Ours)	RGB+Flow+Sem.	34.51

Table 4. Performance on the Moments-in-Time (MiT) dataset. AIRStreams is able to outperform the SOTA (state-of-the-art) on this challenging dataset, both with flow-only and with semantics+flow, and **without using any pre-training**.

which outperforms 4-stream AssembleNet [47] when both are trained from scratch (33.53 for AIRStreams vs 31.41 for AssembleNet); (RGB+Flow)-only AIRStreams without pre-training also outperforms the other 2-stream methods.

4.3.2 Charades dataset

Charades dataset is also a challenging dataset of longer video sequences in which multiple labels are provided per video clip, i.e., a multi-class multi-label setting. We use the standard evaluation protocols, and since the data is small compared to the number of classes available, we use MiT pre-training. AIRStreams works very well (Table 5) compared to the prior state-of-the-art on this dataset which also used pre-training. It outperforms all methods, including SlowFast [13] and video architecture search EvaNet [41], with the exception of the powerful 4-stream AssembleNet with connection learning between streams [47], which on this dataset performs best. This may be due to the longer duration of videos in Charades for which the architecture searched connectivity of AssembleNet is more beneficial.

4.3.3 Toyota Smarthome dataset

We further evaluate on the Toyota Smarthome dataset [8] which contains activities within peoples’ homes. We use the standard protocol in prior work [8] and report both the activity classification accuracy (%) and the ‘mean per-class’ accuracy (%). Table 6 shows the results for the more challenging ‘Cross-Subject’ (CS) evaluation setting. As seen, AIRStreams, without pretraining, outperforms notably previous work which used pre-training (some from Kinetics) and additional 3D human joints information [8]. More specifically, it outperforms the best prior work by 3% and 8%

Method	mAP(%)
BatchNorm Inception [23]	11.6
TSN-Flow [58]	15.7
Two-stream [52] (from [49])	18.6
Asyn-TF [49]	22.4
CoViAR [62]	21.9
TRN [68]	25.2
I3D [4]	32.9
I3D [4] (from [59])	35.5
EvaNet [41]	38.1
SlowFast [13]	45.2
Two-stream (2+1)D ResNet50 (Baseline)	46.5
AssembleNet [47] (MiT pretraining)	53.0
AIRStreams (Ours, MiT pretraining)	50.1

Table 5. Performance on Charades. AIRStreams is able to outperform other methods with the exception of powerful AssembleNet, which uses architecture search for learning connectivity.

Method	Uses pose	Uses pretr.	Classif. %	Mean per-cls. %
LSTM [33]	Yes	No	-	42.5
Dense Trajectories [56]	No	No	-	41.9
I3D [4]	No	Yes	72.0	53.4
NonLocal [59]	No	Yes	-	53.6
Separable STA [8]	Yes	Yes	75.3	54.2
AIRStreams (Ours)	No	No	78.28	62.11

Table 6. Classification and mean per-class accuracies on Toyota Smarthome. The pose column indicates whether the precomputed pose was used and the pretraining column indicates whether pre-training on another video dataset was used. AIRStreams **does not use pre-training**. Some prior methods are reported by [8].

for the classification and mean per-class accuracy, respectively, which indicates it is performing very well on under-represented classes.

5. Conclusion

We present the ‘AIRStreams’ approach for video understanding which introduces semantic segmentation as an intermediate representation, alongside with flow, and jointly optimizes both the representations themselves and their importance for the final task. AIRStreams extends the current 2-stream models. By incorporating semantics and training jointly, it outperforms or is competitive to the best SOTA models SlowFast and AssembleNet. Our work may encourage researchers to incorporate other intermediate representations, which will help tackle even more challenging and more fine-grained activity recognition tasks.

References

- [1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [2] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning (ICML)*, 2018. [2](#)
- [3] T. Brox, N A. Bruhn, Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2004. [4](#)
- [4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [1, 2, 4, 5, 8](#)
- [5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017. [2](#)
- [6] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7024–7033, 2018. [2](#)
- [7] Tensorflow code. Tiny video networks. In https://github.com/google-research/google-research/tree/master/tiny_video_nets, 2020. [5](#)
- [8] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. [5, 8](#)
- [9] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jurgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Holistic large scale video understanding. In [arxiv.org/pdf/1904.11451](https://arxiv.org/pdf/1904.11451.pdf), 2019. [2](#)
- [10] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. [2](#)
- [11] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [2](#)
- [12] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6016–6025, 2018. [2](#)
- [13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018. [2, 5, 8](#)
- [14] C. Feichtenhofer, A. Pinz, and R.P. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. [2, 3](#)
- [15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [1, 2](#)
- [16] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991. [5](#)
- [17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. [4](#)
- [18] Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*, 2019. [2](#)
- [19] Hanzhang Hu, John Langford, Rich Caruana, Saurajit Mukherjee, Eric Horvitz, and Debadeepta Dey. Efficient forward architecture search. 2019. [2](#)
- [20] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [5](#)
- [21] Tensorflow TF Hub. Tiny video networks models. In <https://tfhub.dev/s?module-type=video-classification>, 2020. [5](#)
- [22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margaret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017. [4](#)
- [23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)*, 2018. [8](#)
- [24] Mihir Jain, Jan C. van Gemert, and Cees G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [1, 2](#)
- [25] Jingwei Ji, Shyamal Buch, Alvaro Soto, and Juan Carlos Niebles. End-to-end joint semantic segmentation of actors and actions in video. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [26] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Joint learning of object and action detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. [2](#)
- [28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. [2](#)
- [29] Hildegarde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of the*

IEEE International Conference on Computer Vision (ICCV). IEEE, 2011. 5

[30] TsungYi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll’ar, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *abs/1405.0312*, 2014. 4, 5

[31] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2

[32] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2019. 2

[33] Behrooz Mahasseni and Sinisa Todorovic. Regularizing long short term memory with 3d human-skeleton sequences for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[34] Lili Meng, Bo Zhao, Bo Chang, Wei Sun, Frederich Tung, and Leonid Sigal. Interpretable spatio-temporal attention for video action recognition. In *ICCV Workshop on Holistic Video Understanding*, 2019. 8

[35] Pascal Mettes and Cees G. M. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[36] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *arXiv preprint arXiv:1801.03150*, 2018. 5, 7

[37] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CoRR:2001.09691*, 2020. 2

[38] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Architecture search of dynamic cells for semantic video segmentation. In *CoRR:1904.02371*, 2019. 2

[39] Joe Yue-Hei Ng, Jonghyun Choi, Jan Neumann, and Larry S Davis. Actionflownet: Learning motion representation for action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1616–1624. IEEE, 2018. 2, 4

[40] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Tiny video networks. In *CoRR:1910.06961*, 2019. 5

[41] AJ Piergiovanni, Anelia Angelova, Alexander Toshev, and Michael S Ryoo. Evolving space-time neural architectures for videos. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 8

[42] AJ. Piergiovanni and Ryoo-Michael Angelova, Anelia. Evolving losses for unsupervised video representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 5

[43] AJ Piergiovanni and Michael S Ryoo. Representation flow for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[44] J. Ray, H. Wang, D. Tran, Y. Wang, M. Feiszli, L. Torresani, and M. Paluri. Scenes-objects-actions: A multi-task, multi-label video dataset. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2

[45] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 2, 5

[46] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[47] Michael Ryoo, AJ. Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: searching for multi-stream neural connectivity in video architectures. In *International Conference on Learning Representations*, 2020. 7, 8

[48] Laura Sevilla-Lara, Yiyi Liao, Fatma Guney, Varun Jampani, Andreas Geiger, and Michael Black. On the integration of optical flow and action recognition. In *German Conference on Pattern Recognition*, 2018. 2

[49] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

[50] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018. 5

[51] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[52] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2014. 1, 2, 4, 8

[53] Du Tran, Lubomir D Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3d: generic features for video analysis. *CoRR, abs/1412.0767*, 2(7):8, 2014. 2

[54] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018. 2, 3

[55] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2011. 2

[56] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 8

[57] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[58] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016. 2, 8

[59] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 8

[60] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 399–417, 2018. 2

[61] Yifan Wang, Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Two-stream sr-cnns for actionrecognition in videos. In *British Machine Vision Conference*, 2016. 2

[62] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6026–6035, 2018. 8

[63] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[64] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 2, 5

[65] Chenliang Xu, Shao-Hang Hsieh, Caiming Xion, and Jason J. Corso. Can humans fly? action understanding with multiple classes of actors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[66] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017. 4

[67] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[68] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 8

[69] Mohammadreza Zolfaghari, Gabriel L Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2904–2913, 2017. 2

[70] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017. 2

[71] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2