PERF-Net: Pose Empowered RGB-Flow Net

Yinxiao Li*, Zhichao Lu*, Xuehan Xiong, Jonathan Huang
{yinxiao, lzc, xxman, jonathanhuang}@google.com
Google Research

Abstract

In recent years, many works in the video action recognition literature have shown that two stream models (combining spatial and temporal input streams) are necessary for achieving state-of-the-art performance. In this paper we show the benefits of including yet another stream based on human pose estimated from each frame — specifically by rendering pose on input RGB frames. At first blush, this additional stream may seem redundant given that human pose is fully determined by RGB pixel values — however we show (perhaps surprisingly) that this simple and flexible addition can provide complementary gains. Using this insight, we propose a new model, which we dub PERF-Net *(short for Pose Empowered RGB-Flow Net), which combines this new pose stream with the standard RGB and flow based input streams via distillation techniques and show that our model outperforms the state-of-the-art by a large margin in a number of human action recognition datasets while not requiring flow or pose to be explicitly computed at inference time. The proposed pose stream is also part of the winner solution of the ActivityNet Kinetics Challenge 2020 [1].*

1. Introduction

Human pose is intuitively intimately linked to human centric activity recognition. For example, by localizing the two legs from a human in a collection of frames, one is often able to easily recognize actions such as jumping, walking or sitting. As such, the idea of using pose explicitly as a cue for activity recognition tasks is one that has been explored in a number of works in the computer vision literature, including [5, 6, 7, 31, 49]. In this paper we revisit this conceptually simple idea of using pose as a cue for activity recognition using modern large scale datasets and models. Specifically, we exploit pose in activity recognition using 3D CNNs, which in recent years have been a dominant architecture in the subfield due to the rise of massive scale video datasets such as Kinetics [3, 4, 18].

To achieve state-of-the-art results on Kinetics, many recent works that rely on 3D CNNs [39, 40] have found it necessary to rely on a "two-stream" approach [34] that combines spatial and temporal input streams using late fusion. Concretely, this has typically referred to models trained independently to do activity recognition on (1) a sequence of RGB images and (2) a sequence of optical flow fields (or other motion representation) and fusing the results of both models via ensembling.

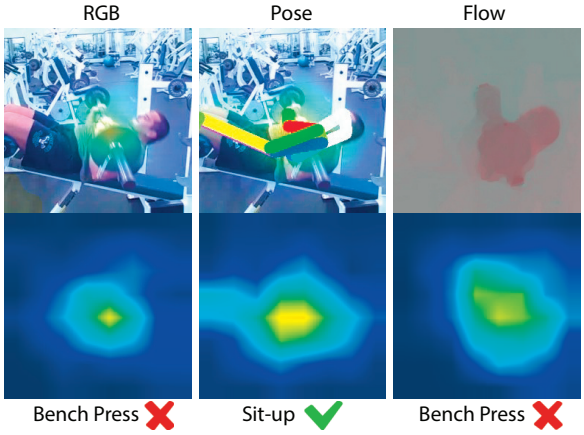

Figure 1. Visualizations of models trained on RGB, Pose, and Flow modalities. The top row shows input multi-modality data. The middle row shows the response maps from the networks using Grad-CAM [33]. Note that the response maps are overlaid on RGB and pose images for better visualization. The bottom row shows the model predictions on each of the modalities. Our proposed pose modality focuses the attention on the entire human body, providing a useful complementary cue to the standard RGB and Flow modalities, here allowing for our model to correctly predict the "sit-up" action.

In addition to this two-stream framework, we propose to add a third input stream based on human pose. Unlike the two-stream approach which is (very) loosely based on the two-stream hypothesis of the human visual system [13], our approach takes no specific inspiration from biology — instead we rely on the natural intuition that since action datasets tend to be human centric, if we had explicit pose cues, it would often be much more straightforward to infer action from pose compared to directly from raw pixels or flow. As an example, consider Figure 1 which visualizes

*Equal contribution

our model's results on a person performing a sit-up using the three possible input "modalities", RGB, pose and flow. However, this sit-up is more specifically a "barbell sit-up on a decline bench" which is easily confused for "bench press" due to strong cues from the appearance and motion of the barbell. With such, the pose modality offers a complementary signal allowing our model to infer the correct activity.

How to provide the pose cues properly requires care however — using pose alone as an input stream is intuitively not enough, as recognition often requires contextual cues (e.g. from props, objects that the human is interacting with, etc). Instead, as the "pose stream", we render pose via exaggerated colored lines on top of each corresponding RGB frame, which allows us to benefit from both a clear pose based signal as well as contextual cues from surrounding appearance. We demonstrate via ablations that this choice to superimpose pose with the corresponding RGB frame is critical for good results.

A reasonable question to ask is: why is pose not simply a redundant input stream? After all, it is fully determined by RGB values — and even more redundant given that we render poses on top of the RGB frames. So even though pose is intuitively connected to activity recognition, what additional specific benefit is pose bringing in our setting?

We have a few answers. First, by using an off-the-shelf pose estimation algorithm that was trained on the COCO dataset [23], we are injecting additional semantic knowledge that the model can leverage. Second, we note that optical flow is also fully determined by the sequence of RGB inputs. And as with flow, we show that models using the pose stream are quantitatively different (better) than simply ensembling with a second RGB-only model. In very recent work, Stroud *et al*. [36] showed that the benefits of the temporal stream could be captured by an "RGB-only" model via distillation training, obviating the need for redundant input streams at inference time.

Taking inspiration from Stroud *et al*. 's flow based results [36], we similarly apply distillation techniques to our problem with both pose and flow. Combining this with a novel self-gating based architecture, we are able to obtain a state-of-the-art RGB-only model that requires us to compute neither flow nor pose. We dub this model the *Pose Empowered RGB-Flow Net (or PERF-Net)*.

To summarize, our contributions are as follows.

- We demonstrate strong evidence that pose is an important modality for video action recognition and can provide a complementary input stream to the standard RGB and Flow streams.
- We propose *PERF-Net*, an approach that leverages RGB, Flow and Pose input streams in a multi-teacher distillation setting to train an RGB-only model with state of the art performance on the challenging Kinetics dataset.
- We study the impact of using different representations of the human pose input stream. We propose a context-aware human pose rendering which can bridge the gap between pose information and RGB within a collection of frames.
- We perform detailed analysis on the response of networks from different input streams (RGB, Flow, and Pose). Our qualitative results show that when trained on our Pose stream, our model sometimes attends to different regions of a frame compared to RGB or Flow, allowing this third stream to offer complementary cues.

2. Related Work

2.1. Fusion of multiple modalities

In contrast to image data, videos are multi-modal. How to best utilize this special characteristic of video data has been a long-standing topic in the video understanding research community. One of the standard approaches, introduced by [34], captures complementary information from appearance and motion by averaging predictions from two separately trained 2D CNNs, one from RGB frames and the other from stacked optical flow frames. Following [34], Feichtenhofer *et al*. [12] investigated the optimal locations within CNNs to combine the two streams.

A more recent trend has been to train a 3D ConvNet to directly model temporal patterns without relying explicitly on optical flow. This is easier said than done, as [4] showed that performance (of their 3D convolutional architecture, *I3D*) could be greatly improved by including an optical flow stream. However there have been some promising approaches; Feichtenhofer *et al*. [11] recently proposed a two-stream architecture where both streams take RGB frames as inputs, but extracted at different frame rates. Unlike the late fusion approach taken by two-stream I3D models, the fusion in [11] is implemented as lateral connections at different layers of the network. Ryoo *et al*. [32] adapted the Evolution algorithm to search such lateral connections in a multi-stream architecture. In addition to different frame rates of RGB streams, they also include optical flow as an additional stream of input.

In addition to optical flow, human pose is another input modality that has been widely studied for understanding videos involving human activities [46, 26, 17]. Chéron *et al*. [5] showed that training RGB and flow streams on the patches centered at human joint locations can improve over the global approach. In addition to RGB and flow frames, Zolfaghari *et al*. [51] proposed a new modality using human body part segmentation results from an existing network. Another novelty from their work is that multi-stream fusion is done sequentially through a Markov chain. Choutas1 *et al*. [6] also proposed an representation to en-

code pose information and use that as an additional stream, but they used black background in the presentation, so on Kinetics, the top-1 and top-5 accuracies decreased by 2% and 1% respectively when using their representation with I3D compared to I3D alone. Our study focuses on how to best represent human pose as an input stream for a 3D CNN. Our experiments highlight the importance of this issue, and we show that a naive representation of human pose indeed degrades the final ensemble performance. More generally we run our experiments on the large scale Kinetics dataset which are properly able to leverage the expressiveness of 3D CNNs leading to stronger results and "clearer" ablation signals throughout the paper.

2.2. Distillation between modalities

While achieving state-of-the-art performance, multi-stream models are computationally more expensive. For example, the computation of optical flow could be more expensive than ConvNet inference. Distillation [2, 15] is a technique to transfer the knowledge of a complex teacher model to a smaller student model by optimizing the student model to mimic the behavior of the teacher. Recently, researchers have adapted this idea to multi-modal model training. Zhang *et al*. [50] used a teacher model trained on optical flow to guide a student CNN whose input is motion vectors, which can be directly obtained from compressed videos. Luo *et al*. [25] proposed a graph distillation approach to address the modality discrepancy between the source and target domain. Our study is most similar to recent works [36, 7] which distill the flow stream into the RGB stream (*e.g*. flow stream is the teacher while RGB stream is the student). Besides the flow stream, our experiments show the benefits of using multiple teachers, *e.g*. flow and human pose.

3. Pose Empowered RGB-Flow Nets

In this section we describe our main contribution, the Pose Empowered RGB-Flow Nets (or *PERF-Net*) approach. We begin by constructing a model that predicts actions based on pose information. Specifically we describe how we represent pose and how our pose representations can be fed to a 3D CNN. The final goal is to fuse the predictions that we can obtain via this pose stream with predictions from RGB and flow streams. The standard approach of applying "late fusion" to combine disparate input streams is accurate but very slow since it requires multiple runs through the 3d convnet architecture. Instead, in the PERF-Net setting, we propose to use multi-teacher distillation to train a final model that takes RGB inputs at test time, but can benefit all three modalities (RGB, Flow, Pose) at training time.

3.1. Pose representation

By pose information we refer to human body joint positions (as is typical in the literature) which we first estimate from each RGB frame using an off-the-shelf pose estimation model and then feed to a 3D CNN as a sequence of frames. For pose estimation we use the PoseNet approach [19, 27] with ResNet backbones which is pre-trained on the COCO dataset [23] and produces 17 estimated pose keypoints for each detected human in a frame. We note that the success of our model does not depend on our specific choice of pose estimation approach. Additionally, we have not specifically tuned the pose model with respect to the final performance of PERF-Net. We also note that in our datasets, such as Kinetics-600, human poses are not available in many samples.

How specifically to render pose as a frame (which can then be sent as input to a convolutional network) is a more important design decision. Our approach is to render pose via colored lines (using a different color for each limb to allow the model to more easily distinguish between the limbs). The simplest approach (similar to that taken by [51]) is to simply render the estimated pose on a black background. However using pose information alone in this way is intuitively not enough, as activity recognition often requires contextual cues — for example, having a golf club in the frame is highly indicative of the action. So instead we render the pose of each human on top of each corresponding RGB frame, which as we show in experiments, can have a sizeable impact on performance. We experiment with three additional variations of the rendering scheme:

- Dots vs bars: we render joint locations with filled circles instead of limbs with line segments.
- Fine vs coarse-grained coloring: in our coarse-grained setting we use 6 colors for the joints, assigning a unique color to the left arm, right arm, body, head, left leg, and right leg. In our fine-grained setting, each limb gets its own color (e.g., left forearm vs left upper arm).
- Uniform vs ratio-aware line thickness: in the former setting, we render lines with a uniform width; whereas in the latter setting, we set line thickness proportional to the size of the corresponding person detection's bounding box.

Figure 2 shows example of these pose rendering variants. As we show in the next section, using the fine-grained coloring scheme and using ratio-aware line thicknesses can lead to improved results.

3.2. Backbone architecture

We now describe our backbone architecture which is based on a 3D version of ResNet50 where some of the convolution kernels have been "inflated" (specifically described

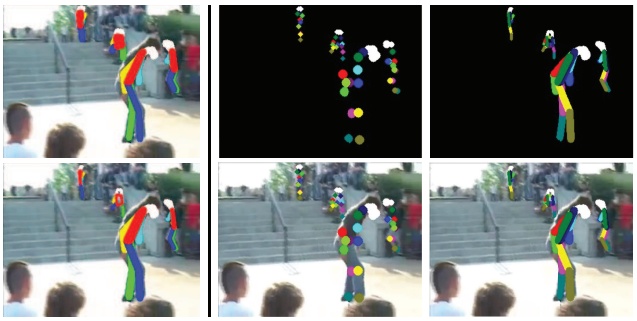

Figure 2. A few different human pose rendering effects that have been explored. Column A uses 6 different colors to represent poses, where the top row is rendered using the same thickness of the segments and bottom row uses ratio-aware thickness of the segments. Column B and C explore two different rendering markers, points and segments with 13 different colors. The top row in column B and C uses a black background. Both column B and C also add ratio-aware radius or thickness while rendering the poses.

Block		Output sizes $T \times S^2 \times C$
input		$64 \times 224^2 \times 3$
$conv_1$	5×7^2 stride 1×2^2	$64 \times 112^2 \times 64$
$pool_1$	1×3^2 stride 1×2^2	$64 \times 56^2 \times 64$
res_2	$\begin{bmatrix} 3 \times 1^2 \\ 1 \times 3^2 \\ 1 \times 1^2 \end{bmatrix} \times 3$ feature gating	$64 \times 56^2 \times 256$
res_3	$\begin{bmatrix} t_i \times 1^2 \\ 1 \times 3^2 \\ 1 \times 1^2 \end{bmatrix} \times 4$ feature gating	$64 \times 28^2 \times 512$
res_4	$\begin{bmatrix} t_i \times 1^2 \\ 1 \times 3^2 \\ 1 \times 1^2 \end{bmatrix} \times 6$ feature gating	$64 \times 14^2 \times 1024$
res_5	$\begin{bmatrix} t_i \times 1^2 \\ 1 \times 3^2 \\ 1 \times 1^2 \end{bmatrix} \times 3$ feature gating	$64 \times 7^2 \times 2048$

Table 1. R3D50-G architecture used in our experiments. The kernel dimensions are $T \times S^2$ where T is the temporal kernel size and S is the spatial size. The strides are denoted as temporal stride $\times$ spatial stride^2. For res_3, res_4, and res_5 blocks the temporal convolution only applies at every other cell. E.g., $t_i = 3$ when i is an odd number and $t_i = 1$ when i is even.

by [45] with a few key modifications). First, we remove all max pooling operations in the temporal dimension. We find that applying temporal downsampling in any layer degrades the performance. Second, we add a feature gating module [47] after each residual block. Feature gating is a self-attention mechanism that re-weights the channels based on context (i.e., the feature map averaged over time and space). We also explored adding feature gating modules after every residual cell which achieved similar results, so we decided to keep the former configuration given that it is more computationally efficient. These two modifications (no temporal downsampling, feature gating) can significantly improve the final performance and ablation studies can be found in the supplementary materials. In our experiments, we denote this modified ResNet50 as R3D50-G (see Table 1). Note that our methodology for using pose as an input stream does not depend specifically on the choice of backbone, and indeed we also demonstrate results using the recent S3D-G backbone [47].

3.3. Multi-stream fusion via distillation

Much as flow is used as a complementary signal to RGB input streams in typical action recognition papers, the intention of our pose model is to be used as a complementary signal to both RGB and flow. We now turn to how to combine these multiple streams (RGB, flow, pose) into a single model that takes RGB as its only input. Specifically we assume now that we have trained 3 models based on RGB, flow and pose respectively. The goal of our distillation approach will be to train an RGB-only model that requires much less computation compared to running all three models separately while capturing their complementary strengths.

Our approach is inspired by the D3D model [36], an RGB-only model which captures the benefits of having a temporal stream by using distillation techniques. Specifically, Stroud *et al*. [36] trained a student model which takes a spatial (RGB-only) stream as input to do action recognition, adding an additional distillation loss which compares against the output of a teacher model that was trained on temporal stream inputs.

We apply a natural extension of the D3D approach to allow it to handle multiple distillation losses (corresponding to multiple non-spatial input streams). The total loss that we jointly minimize encourages our PERF-Net RGB-only student model to mimic logits from each teacher network while simultaneously minimizing the loss from groundtruth labels via backpropagation, and can be written as follows:

$$\mathbf{L} = L^c(S^\ell) + \sum_i^N MSE(T_i^\ell, S^\ell) \quad (1)$$

where S^ℓ denotes logits from student network and T_i^ℓ denotes the logits from the ith teacher network. We use mean squared loss (applied to logits of student and teacher models) as the distillation loss. Figure 3 shows the structure of our multi-teacher distillation framework.

Note that our loss function is distinct from the natural alternative of training the student to directly mimic the standard late fusion model (by regression towards the sum of all teacher-produced logits, referred as unified loss). In experiments we show that our approach achieves significantly better performance (See Table 4).

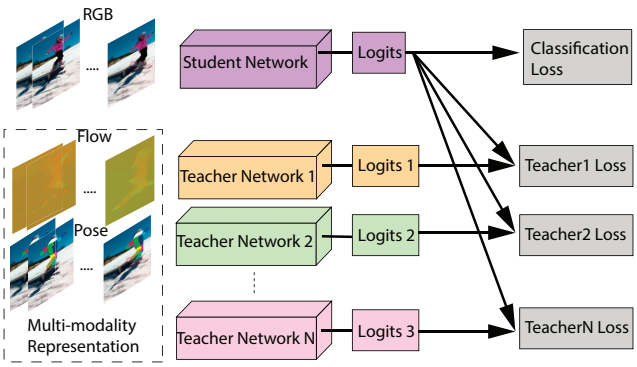

Figure 3. The distillation framework is composed of two pieces: student network and teacher network(s). The input modality can be any representations, such as RGB, flow, or pose. The losses are computed on each of the logits from the corresponding teacher networks (separate loss). Additionally, we experimented with the loss computed on the summation of logits (1, 2, ... N) from all teacher networks and added to the regression loss (unified loss). We show separate loss outperforms unified loss in the experimental section.

4. Experiments

Training details Our R3D50-G models are trained on Google TPUs (v3) [21] using Momentum SGD with weight decay 0.0001 and momentum 0.9. We construct each batch using 2048 clips on 256 TPU cores, yielding a per-core batch size of 8. In order to fit 8 clips in TPU memory, we use mixed precision training with bfloat16 type in all our TPU training runs [44]. We train our R3D50-G models on Kinetics-600 with random initialization ("from scratch"). We also experimented with initializing from an inflated [4] ImageNet [8] pre-trained model but this turns out to be unnecessary in our setup. We train using a linear learning rate warm-up for the first 2k steps increasing from 0 to a base learning rate of 1.6, then use a cosine annealed learning rate [24] for 20K steps.

Our S3D-G models are trained on 51 GPUs with a per-core batch size of 6 clips (so the total mini-batch size is 306). All S3D-G models are initialized using *inflation* [4] with a pre-trained Inception [37] model on ImageNet [8].

All models are trained on 64 consecutive frames (at 25 FPS) from the original videos and those clips are randomly cropped from the original sequence. For each frame in the clip, we first resize the video to have a shorter side equaling to 256, and randomly crop a 224×224 region as the input to the networks. For UCF-101 and HMDB-51, we use random crops of 224×298 as inputs. Random mirroring, contrast, and brightness are also applied as data augmentation. Finally, to extract flow, we use the TV-L1 approach [38].

Inference. Unlike previous work [45, 11], we use a single central crop of the video to evaluate our models' performance. The crop size is set to $250 \times 256 \times 256 \times 3$ for Kinetics-600, $128 \times 224 \times 298 \times 3$ for UCF-101, and $64 \times 224 \times 298 \times 3$ for HMDB-51, (input shapes follow the frames$\times$ height$\times$ width$\times$channels convention). For sequences that do not have sufficiently many frames, we pad by duplicating the first or the last frame.

4.1. What is the best representation for pose?

Our first question is which pose rendering methods achieve the best performance (Figure 2)? We first take the approach of rendering pose on a black background, which as shown in Table 2 yields an accuracy much lower than the other approaches. We argue that the reason is because there are quite a few action training examples that are missing more than 50% of the human body; thus pose cannot be determined in such frames. Instead, pose rendered on top of the RGB frames not only provides rich context beyond the pose itself, but also learns useful signals on the frames without pose.

We also experiment with dot and bar rendering markers and notice that bars yield slightly better results. We believe that this is because bars provides more geometric information about joint connections.

We also see that the fine-grained coloring scheme with ratio-aware rendering achieves the highest accuracy. This outcome is intuitive for the following reasons. First, fine-grained pose rendering can provide detailed body joint relations such as fore-arm vs. upper-arm. Actions like pull-ups, hug, and throw can benefit from such joint relations. Second, with the ratio-aware line thickness, the pose itself provides information about relative distances which can serve as useful hints about group actions, e.g. playing games. Figure 4 shows a few such rendered examples used in the pose stream for the training.

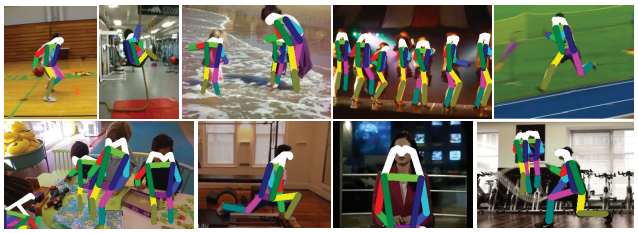

Figure 4. Samples of fine-grained, ratio-aware rendering of PoseNet detection results.

4.2. Is pose complementary to RGB?

We demonstrate that pose offers a complementary signal to the RGB (and Flow) streams. In order to demonstrate the value-add of Pose, we use the standard late-fusion approach to combining multiple streams (so as to not have potential confounding effects from distillation, which requires a more complex training setup).

Background	Marker	Color	Ratio	Top1	Top5
RGB Frame	bar	6	×	77.8	93.9
RGB Frame	bar	6	✓	78.1	93.9
RGB Frame	bar	13	×	77.9	93.8
Black	dot	13	✓	33.7	52.5
Black	bar	13	✓	34.0	52.9
RGB Frame	dot	13	✓	78.0	93.6
RGB Frame	bar	13	✓	**79.3**	**94.3**

Table 2. Pose stream results using R3D50-G on Kinetics-600 dataset with markers: dot or bar, and ratio-aware marker size. The pose model is trained to validate performance. We also evaluate the approach of rendering on a black background, but since many training frames have no detected pose the performance of this naïve approach tends to be very low.

Backbone	Modalities/Net	Top-1	Top-5	pretrain
S3D-G	RGB	77.8	93.9	Imagenet
	Flow	68.3	88.4	Imagenet
	Pose	76.8	93.4	Imagenet
	RGB+Flow	79.2	94.5	-
	RGB+Pose	79.2	94.6	-
	RGB+Flow+Pose	**80.3**	**95.4**	-
R3D50-G	RGB	80.4	95.2	-
	Flow	69.5	89.2	-
	Pose	79.3	94.5	-
	RGB+RGB	80.4	95.6	-
	RGB+Flow	81.4	95.6	-
	RGB+Pose(BB)	79.9	94.2	-
	RGB+Pose	81.1	95.9	-
	RGB+Flow+Pose	**82.0**	**96.5**	-

Table 3. Late Multi-Stream Fusion Results on Kinetics-600. To test our multi-fusion framework, we employ S3D-G and R3D50-G backbones. Here, the "G" refers to the usage of self-gating. The first block shows results using S3D-G (pretrained with Imagenet) as the backbone. The second block shows results on R3D50-G as the backbone. Pose(BB) refers to the model trained with pose rendered on black background in Table 2. Among all settings, combination of all three modalities outperform other combinations.

4.2.1 Kinetics datasets

In this section we focus on the the Kinetics-600 dataset [4], a large-scale, high-quality dataset containing YouTube video URLs with a diverse range of human focused actions. The dataset consists of approximately 500k video clips, and covers 600 human action classes with at least 600 video clips for each type of action. Each clip is at least 10 seconds and is labeled with one single class. The actions cover a broad range of classes including human-object interactions such as playing instruments, working out, as well as human-human interactions such as sword fighting and hugging.

4.2.2 Late multi-stream fusion

In the standard "late-fusion" approach, we run models independently on multiple streams, combining their predicted logits at the end through simple addition (see [12] for details). Table 3 shows a comparison of standard late-fusion (across different combinations of the three streams, RGB, Flow and Pose) among our two backbone models (R3D50-G and S3D-G).

For both S3D-G and R3D50-G backbones, we can see that by incorporating additional modalities, we can always achieve performance gains. Adding flow or pose to the existing RGB stream yields similar improvements. Since flow and pose are somewhat independent modalities, by adding both of them to the RGB stream, we also observe "stacking" of the performance gains. Most importantly, we see that adding the pose stream always yields benefits (independent of backbone network and independent of whether we are already using a flow stream).

One might wonder if the benefits of adding a pose stream come simply from the ensembling effect of two models — to show that this is not the case, we show that ensembling two RGB-only models (RGB+RGB in Table 3) does not lead to measurable improvements. Additionally, we show that adding pose stream always introduces complementary gain to RGB or RGB+Flow modalities.

Backbone	Student	Teacher(s)	Top-1	Top-5	pretrain
S3D-G	RGB	-	77.8	93.9	-
	RGB	Flow	78.3	94.3	-
	RGB	Pose	78.4	94.2	-
	RGB	Flow+Pose (SL)	**78.9**	**94.6**	-
R3D50-G	RGB	-	80.4	95.2	-
	RGB	Flow	80.6	94.6	-
	RGB	Pose	80.4	94.7	-
	RGB	Flow+Pose (UL)	80.7	95.3	-
	RGB	Flow+Pose (SL)	**82.0**	**95.7**	-

Table 4. Results on Kinetics-600 distillation. SL stands for separate loss, and UL stands for unified loss. The last row (SL) is the PERF-Net results.

4.2.3 Visualization and explanation

Figure 5 shows 9 examples of RGB, pose, and flow, as well as the corresponding response map from a layer from block5 in R3D50-G. The main purpose of this figure is to show the performance of the individual models trained on each modality.

The first row shows three sets of examples where the pose model is correct, and the RGB and Flow models are incorrect. For example, the leftmost example depicts an arm wrestling action. The pose response map responds most on the hands region of the frame where the wrestling happens. The response heatmap can be treated as an attention area in a tube of action sequences. For such actions, flow is not informative as there is little motion. Moveover, the RGB response could be distracted by elements in the background. However, pose can provide clear signal to the hand-to-hand

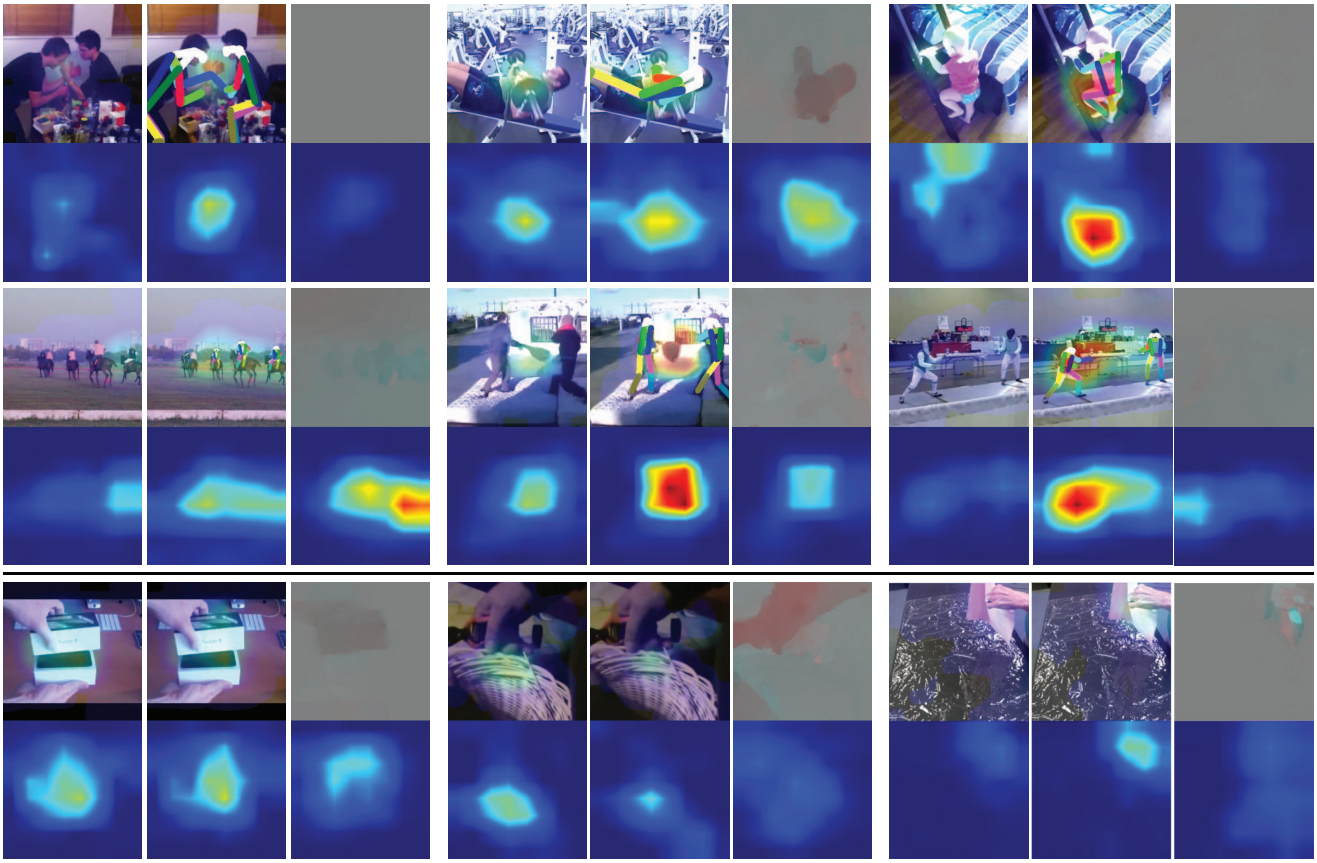

Figure 5. Nine Grad-CAM visualizations [33] of our R3D50-G model. Each row contains three examples. For each example, the top row contains the original RGB, pose overlay, and Flow frames and the bottom row are the normalized response maps from RGB, pose, and flow streams, respectively. ROW1: arm wrestling, situp action, ladder climbing. ROW2: playing polo, pillow fight, swording. ROW3: unboxing, weaving basket, napkin folding. The top two rows show examples with pose detected. The bottom row shows three actions without any pose.

interaction. The middle example shows a person performing a situp at a gym. It is difficult to classify this action correctly by focusing on the barbell regions of the image, as the RGB and flow model do. Instead, pose drives the model to "look at" the entire body configuration which allows the model to decide that it is a situp and not bench press, etc. The rightmost example shows a baby climbing a ladder. The pose stream focuses the attention on the legs where the climbing action happens, providing a useful complementary cue to the standard RGB and Flow modalities.

The second row comprises three examples where all modalities make the prediction correctly. From the response map, we can tell the three modalities mostly focus on similar locations among the video frames. For the leftmost example (playing polo), pose helps to focus more on the entire group of players, where the other two modalities put more weight on the right-most player. By looking at the original video clip, the motion of the right-most player is the largest, which is likely why RGB and flow give more weight to this player. The middle example shows a pillow fight where the pose modality response is greater on the pillow region. The pose model may learn additional information from the interaction of the two persons by looking at the pose and arm orientation, etc. The rightmost example shows swording where the pose stream focus more on the left-side acting player.

The third row shows three examples without any pose detected. There are quite a few frames in Kinetics-600 and other datasets where no pose is available. In such cases, since the RGB is still available via the pose stream, our pose based model can still learn reasonably good responses.

4.3. Distilling down to PERF-Net

As discussed in Section 3.3, distillation can effectively incorporate multiple modalities with no additional cost to the complexity of the final model. In Table 4, we show the results of multi-teacher distillation using Kinetics-600 dataset, which can jointly optimize over multiple input modalities. The advantage of the distillation is that our model size can remain the same while leveraging knowledge distilled from other modalities. Taking RGB as an example, after distilling on flow and pose using separate

Model	Backbone	Top-1	Top-5	GFLOPs
I3D [4]	Inception	71.9	90.1	544
StNet-IRv2 RGB [14]	InceptionResNet-V2	79.0	-	440
P3D two-stream [30]	ResNet152	80.9	94.9	-
SlowFast R101+NL [11]	ResNet101	81.8	95.1	7020
LGD-3D RGB [31]	ResNet101	81.5	95.6	-
X3D-XL [10]	Custom	81.9	95.5	1452
PERF-Net (ours)	ResNet50-G	**82.0**	**95.7**	3666

Table 5. Comparison with the state-of-the-art on Kinetics-600.

Backbone	Student	Teacher(s)	Top-1	Top-5	pretrain
S3D-G	RGB	-	63.5	85.1	-
	Flow	-	51.0	75.8	-
	Pose	-	61.3	83.5	-
	RGB	Flow+Pose	**67.9**	**87.9**	-

Table 6. Results on Kinetics-700 distillation. The first three rows are single stream results. The last row is the PERF-Net results.

losses, the performance can be improved beyond single modality training — thus our final RGB-only model (*a.k.a. PERF-Net*) achieves 82.0 top-1 accuracy on Kinetics-600, which outperforms the state-of-the-art work.

Table 5 shows a comparison between PERF-Nets and other state-of-the-art single-stream works. Note that PERF-Net can easily achieve state-of-the-art performance by using a shallower ResNet50-G network. One can apply PERF-Net on stronger backbones to further boost the performance.

Table 6 shows the three single stream results, along with the distillation on RGB stream with flow and pose streams as the teacher models on the Kinects-700 dataset [3]. With 700 classes, the training tasks become considerably more challenging. In this setting, PERF-Net results show even more gain from distillation compared to the model trained on Kinectics-600, as shown in Table 4.

4.4. Will distilled checkpoint transfer well?

We select two human action datasets for transfer learning experiments initialized using checkpoints on Kinetics-600 or Kinetics-700 with distillation. The Kinetics-700 dataset has 100 more classes with more video clips, which is harder to learn. During fine-tuning, we use only the classification loss, but not distillation. For both of the datasets, we show that PERF-Net achieves the state-of-the-art performance among single stream models. The results also indicate that PERF-Net generalizes well given a harder dataset for pre-training.

4.4.1 HMDB-51

HMDB-51 [20] contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips for each category. We apply the same pose detection and rendering method to the HMDB-51 dataset. We finetune S3D-G model pre-trained on Kinetics-600 or Kinetics-700 for 30 epochs and report the accuracy by averaging the results from 3 splits. Table 7 shows the averaged performance of our PERF-Net models. Our PERF-Net with backbone S3D-G, outperforms the current best on the leaderboard using single stream model [16]. Note that it also outperforms two ensemble models.

Model	UCF-101	HMDB-51
P3D [30]	88.6	-
C3D [40]	82.3	51.6
Res3D [41]	85.8	54.9
TSM [22]	95.9	73.5
I3D [4]	95.6	74.8
R(2+1)D [42]	96.8	74.5
S3D-G [47]	96.8	75.9
HATNet [9]	97.7	76.2
MARS+RGB+Flow [7]	97.8	80.9
Two-stream I3D [4]	98.0	80.9
RepFlow-50 [29]	-	81.1
EvaNet-top individual [28]	-	81.3
PA3D+I3D [48]	-	82.1
EvaNet-ensemble [28]	-	82.3
PERF-Net (ours, Kinetics-600 pretrain)	98.2	82.0
PERF-Net (ours, Kinetics-700 pretrain)	**98.6**	**83.2**

Table 7. Comparison with state-of-the-art on UCF-101 and HMDB-51. The backbone of the PERF-Net here is S3D-G.

4.4.2 UCF-101

UCF-101 [35] is an action recognition data set of 13,320 realistic action videos, collected from YouTube, with 101 action categories. Similar to HMDB51, in Table 7, we also report the accuracy by averaging over the 3 dataset splits. Similarly, for both Kinetics-600 and Kinetics-700 pretrainings, our PERF-Net model achieves the state-of-the-art at time of submission on the leaderboard [43].

5. Conclusions

We have presented an empirical study of the effects of different pose rendering methods and how to effectively incorporate it into a video recognition model to benefit human action recognition. We have shown strong evidence that, with the human pose modality and the proposed rendering method, by using distillation, the model can outperform the state-of-the-art performance. We hope such pose modality can be further studied to extend to other domains.

References

[1] ActivityNet Kinetics Challenge 2020. http://activity-net.org/challenges/2020/tasks/guest_kinetics.html. 0

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 0, 7

[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*, pages 6299–6308, 2017. 0, 1, 4, 5, 7

[5] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3218–3226, 2015. 0, 1

[6] Vasileios Choutas, Philippe Weinzaepfel, Jerome Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. pages 7024–7033, 06 2018. 0, 1

[7] Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. Mars: Motion-augmented rgb stream for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7882–7891, 2019. 0, 2, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[9] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jurgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large scale holistic video understanding. 2019. 7

[10] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, 2020. 7

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 1, 4, 7

[12] 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*, pages 1933–1941, 2016. 1, 5

[13] Melvyn A Goodale, A David Milner, et al. Separate visual pathways for perception and action. 1992. 0

[14] Dongliang He, Fu Li, Qijie Zhao, Xiang Long, Yi Fu, and Shilei Wen. Exploiting spatial-temporal modelling and multi-modal fusion for human action recognition. 2018. 7

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[16] HMDB-51-Leaderboard. Action recognition in videos on hmdb-51. In *https://paperswithcode.com/sota/action-recognition-in-videos-on-hmdb-51*, 2020. 7

[17] Umar Iqbal, Martin Garbade, and Juergen Gall. Pose for action-action for pose. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 438–445. IEEE, 2017. 1

[18] 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. 0

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. pages 2938–2946, 12 2015. 2

[20] Hilde Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb51: A large video database for human motion recognition. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2556–2563, 11 2011. 7

[21] Sameer Kumar, Victor Bitorff, Dehao Chen, Chiachen Chou, Blake Hechtman, HyoukJoong Lee, Naveen Kumar, Peter Mattson, Shibo Wang, Tao Wang, et al. Scale mlperf-0.6 models on google tpu-v3 pods. *arXiv preprint arXiv:1909.09756*, 2019. 4

[22] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 7

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 2

[24] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[25] Zelun Luo, Jun-Ting Hsieh, Lu Jiang, Juan Carlos Niebles, and Li Fei-Fei. Graph distillation for action detection with privileged modalities. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 166–183, 2018. 2

[26] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018. 1

[27] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3711–3719, July 2017. 2

[28] AJ Piergiovanni, Anelia Angelova, Alexander Toshev, and Michael S. Ryoo. Evolving space-time neural architectures for videos. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7

[29] AJ Piergiovanni and Michael S. Ryoo. Representation flow for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7

[30] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. pages 5534–5542, 10 2017. 7

[31] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. 06 2019. 0, 7

[32] Michael S Ryoo, AJ Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. *arXiv preprint arXiv:1905.13209*, 2019. 1

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 0, 6

[34] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 0, 1

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 7

[36] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 625–634, 2020. 1, 2, 3

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4

[38] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. TV-L1 Optical Flow Estimation. *Image Processing On Line*, 3:137–150, 2013. 4

[39] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pages 140–153. Springer, 2010. 0

[40] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 0, 7

[41] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 7

[42] 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*, pages 6450–6459, 2018. 7

[43] UCF101-Leaderboard. Leaderboard: Action recognition in videos on ucf101. 2020. 7

[44] Shibo Wang and Pankaj Kanwar. Bfloat16: the secret to high performance on cloud tpus. *Google Cloud Blog*, 2019. 4

[45] 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*, pages 7794–7803, 2018. 3, 4

[46] Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1293–1301, 2015. 1

[47] 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 the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 3, 7

[48] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[49] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation? In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.67. 0

[50] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2718–2726, 2016. 2

[51] 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*, pages 2904–2913, 2017. 1, 2