D3D: Distilled 3D Networks for Video Action Recognition

Jonathan C. Stroud*† stroud@umich.edu
David A. Ross* dross@google.com
Chen Sun* chensun@google.com
Jia Deng*‡ jiadeng@cs.princeton.edu
Rahul Sukthankar* sukthankar@google.com
*Google Research † University of Michigan ‡ Princeton University

Abstract

State-of-the-art methods for action recognition commonly use two networks: the spatial stream, which takes RGB frames as input, and the temporal stream, which takes optical flow as input. In recent work, both streams are 3D Convolutional Neural Networks, which use spatiotemporal filters. These filters can respond to motion, and therefore should allow the network to learn motion representations, removing the need for optical flow. However, we still see significant benefits in performance by feeding optical flow into the temporal stream, indicating that the spatial stream is "missing" some of the signal that the temporal stream captures. In this work, we first investigate whether motion representations are indeed missing in the spatial stream, and show that there is significant room for improvement. Second, we demonstrate that these motion representations can be improved using distillation, that is, by tuning the spatial stream to mimic the temporal stream, effectively combining both models into a single stream. Finally, we show that our Distilled 3D Network (D3D) achieves performance on par with the two-stream approach, with no need to compute optical flow during inference.

1. Introduction

Motion is often a necessary cue for recognizing actions. For example, it may be difficult to tell two actions apart from a single frame, like "open a door" and "close a door", because the interpretation of the action depends on the direction of motion. To handle this, recent work treats recognition from motion as its own task, in which a "temporal stream" observes only a hand-designed motion representation as input, while another network, the "spatial stream", observes the raw RGB video frames [28]. However, when the spatial stream is a 3D Convolutional Neural Network, it has spatiotemporal filters that can respond to motion in the video [4, 41]. Conceptually, this should allow the spatial stream to learn motion features, a claim echoed in the literature [34, 20, 23]. However, we still see strong gains in accuracy by including a "temporal" 3D CNN which takes an explicit motion representation, typically optical flow, as input. For example, we see a 6.6% increase in accuracy on HMDB-51 when we ensemble a 3D CNN that takes RGB frames with a 3D CNN that takes optical flow frames [4]. It is unclear why both streams are necessary. Is the temporal stream capturing motion features which the spatial stream is missing? If so, why is the 3D CNN missing this information? In this work, we examine the spatial stream in 3D CNNs to see what motion representations they learn, and we introduce a method, depicted in Figure 1, that combines the spatial and temporal streams into a single RGB-only model that achieves comparable performance.

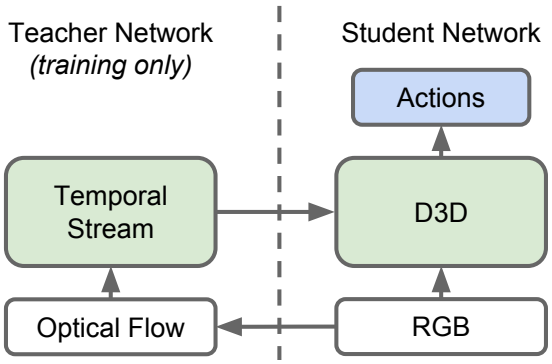

Figure 1. Distilled 3D Networks (D3D). We train a 3D CNN (the *student*) to recognize actions from RGB video while also distilling knowledge from a network (the *teacher*) that recognizes actions from optical flow sequences. The teacher network is only used during training, so optical flow is not needed for inference.

Because 3D CNNs include temporal filters, we hypothesize that they should be able to produce motion representations such as optical flow. Recent work has shown that it is possible for 3D CNNs to learn optical flow, but in these studies, the network structure is designed specifically for this purpose [22]. Instead of designing a network specif-

ically for learning motion representations, we study a network that is designed for action recognition, and we test whether it is capable of producing motion representations. To do this, we train 3D CNNs on an optical flow prediction task, described in Section 3.1, and we demonstrate experimentally that 3D CNNs are indeed capable of learning motion representations in this way.

However, while 3D CNNs are capable of learning motion representations when optimized for optical flow prediction, it is not necessarily true that these motion representations will arise naturally when 3D CNNs are trained to perform other tasks, such as action recognition. To answer whether this is the case, we evaluate the same state-of-the-art 3D CNNs on the optical flow prediction task, but we use models with fixed spatiotemporal filters that are trained on an action recognition task. We find that these models underperform those that are fully fine-tuned for optical flow prediction, suggesting that 3D CNNs have much room for improvement to learn higher-quality motion representations.

To improve these motion representations, we propose to distill knowledge from the temporal stream into the spatial stream, effectively compressing the two-stream architecture into a single model. In Section 4, we train this Distilled 3D Network (D3D) by optimizing an auxiliary loss which encourages the spatial stream to match the temporal stream's output, a technique often used for model compression [14]. During inference, we only use the distilled spatial stream, and we find that D3D achieves improved performance on the optical flow prediction task. This suggests that distillation improves motion representations in 3D CNNs.

We apply D3D to five datasets using three backbone architectures, and we find in Section 5 that D3D strongly outperforms single-stream baselines, achieving accuracy on par with the two-stream model with only a single stream. We train and evaluate D3D on Kinetics [18], and we show that the weights learned by distillation also transfer to other tasks, including HMDB-51 [19], UCF-101 [29], and AVA [13]. D3D does not require any optical flow computation during inference, making it less computationally expensive than two-stream approaches. D3D can also benefit from ensembling for better performance, still without the need for optical flow. We compare D3D to a number of strong baselines, and D3D outperforms these approaches.

In summary, we make the following contributions:

1. We investigate whether motion representations arise naturally in the spatial stream of 3D CNNs trained on action recognition.
2. We introduce a method, Distilled 3D Networks (D3D), for improving these motion representations using knowledge distillation from the temporal stream.
3. We demonstrate that D3D achieves competitive results on Kinetics, UCF-101, HMDB-51, and AVA, without the need to compute optical flow during inference.

2. Related Work

We broadly categorize video action recognition methods into two approaches. First, there are 2D CNN approaches, where single-frame models are used to process each frame individually. Second, there are 3D CNN approaches, where a model learns video-level features using 3D filters. As we will see, both categories of methods often take a two-stream approach, where one stream captures features from appearance, and another stream captures features from motion. Our work considers Two-Stream 3D CNNs.

2D CNNs. Many approaches leverage the strength of single-image (2D) CNNs by applying a CNN to each individual video frame and pooling the predictions across time [28, 6, 27]. However, naïve average pooling ignores the temporal dynamics of video. To capture temporal features, Two-Stream Networks introduce a second network called the temporal stream, which takes a sequence of consecutive optical flow frames as input [28]. The outputs of these networks are then combined by late fusion, or in other approaches by early fusion, by allowing the early layers of the spatial and temporal streams to interact [8]. Other methods have taken different approaches to incorporating motion by changing the way the features are pooled across time, for example, with an LSTM or CRF [6, 27]. These approaches have proven very effective, particularly in the case where video data is limited and therefore training a 3D CNN is challenging. However, recently released large-scale video datasets have spurred advances in 3D CNNs [18].

3D CNNs. Single-frame CNNs can be generalized to video by expanding the filters to three dimensions and applying them temporally, an approach called 3D CNNs [16]. Conceptually, 3D filters should allow CNNs to model motion, but this comes at a cost; 3D CNNs have more parameters and therefore require more data to train. Large-scale video datasets such as Sports-1M enabled the first 3D CNNs, but these were often not much more accurate than 2D CNNs applied frame-by-frame, calling into question whether 3D CNNs actually model motion [17]. To compensate, many 3D CNN approaches use additional techniques for incorporating motion. In C3D, motion is incorporated using Improved Dense Trajectory (IDT) features, which leads to a substantial improvement of 5.2% absolute accuracy on UCF-101 [34, 38]. In I3D, S3D-G, and R(2+1)D, using a two-stream approach leads to absolute improvements of 3.1%, 2.5%, and 1.1% on Kinetics, respectively [4, 41, 36]. The fact that 3D CNNs benefit from a hand designed motion representation suggests that they do not learn to model motion naturally when trained on action recognition tasks. More evidence has shed light on this, for example recent work discovered that 3D CNNs are largely unaffected in accuracy on Kinetics when their input is reversed [41]. In addition, it has been shown that using only a single frame from Kinetics videos with C3D achieves only 5% lower ac-

curacy than using all frames [15]. These results suggest that 3D CNNs do not sufficiently model motion, a hypothesis we explore further in this work.

Why Optical Flow? If 3D CNNs do not model motion when trained on action recognition, we naturally ask whether motion is even necessary for this task, and if not, what other benefits optical flow may offer. Recent work has explored several possible explanations for why optical flow is so effective for 3D CNNs [26]. One hypothesis is that optical flow is invariant to texture and color, making it difficult to overfit to small video datasets. To support this, recent work demonstrates that action recognition performance is not well correlated with optical flow accuracy, except near motion boundaries and areas of small displacement [26]. This work, as well as others, have shown that better or cheaper motion representations can be used in place of optical flow, suggesting that, while motion representations are important, optical flow itself is not crucial [7, 43, 44, 10, 26]. However, optical flow has been shown to be useful as a source of additional supervision, which is shown by ActionFlowNet [22]. This work, like ours, trains a 3D CNN to incorporate motion by using an auxiliary task. However, our work uses a different auxiliary task, distillation, which we show is more effective.

Incorporating Motion in 3D CNNs. Many other approaches incorporate motion information into 3D CNNs using changes to the network architecture. Motion Feature Networks, Optical Flow-Guided Features, and Representation Flow all accomplish this by introducing modules into the network which explicitly compute motion representations [20, 32, 23]. Alternatively, several approaches have proposed to replace the optical flow inputs for the temporal stream with a CNN which produces a learned motion representation. For example, Hidden Two-Stream and TVNet use a motion representation that is trained end-to-end for action recognition [7, 44]. In our work, we show that distillation is more effective at improving accuracy than these architectural changes. However, distillation is not in conflict with these changes, and can in fact be applied in combination with any network architecture. Furthermore, the approaches which introduce new modules do not answer whether "vanilla" 3D CNNs are capable of learning motion representations. In our work, we present a study which demonstrates that 3D CNNs do have this ability, and show that distillation improves these representations.

Distillation. In this work we propose to incorporate motion representations into 3D CNNs using distillation. Distillation was first introduced as a way of transferring knowledge from a teacher network to a (typically smaller) student network by optimizing the student network to reconstruct the output of the teacher network [2, 14]. Recent work on distillation has demonstrated that this technique is widely applicable and can be used to transfer knowledge between different tasks or modalities [9, 43, 25, 21, 11]. Our work is related to Motion Vector CNNs, which distill knowledge from the temporal stream into a new motion stream which uses a cheaper motion representation in place of optical flow [43]. By contrast, our work distills the temporal stream into the spatial stream, which allows us to avoid using hand-designed motion representations altogether.

The most similar work to ours is concurrent work on Motion-Augmented RGB Streams (MARS) [5]. This work proposes a similar distillation approach, but ours presents several additional analyses which shed light on the method. Specifically, in Section 3, we propose a flow prediction task to study the motion representation capacity of 3D CNNs, and we demonstrate the effect of distillation on this ability. In addition, we show that our approach can transfer to spatio-temporal action localization (Section 5.5) as well as different backbone architectures (Table 7). Finally, in our ablation studies in Section 5.6 we propose and evaluate some alternatives to distillation, and we show that distillation outperforms these alternatives.

3. Motion Representations in 3D CNNs

Two-stream methods rely on optical flow, a hand-designed motion representation, in order to learn features from motion. This begs the question: are 3D CNNs capable of learning sufficient motion representations on their own? To answer this, we train a spatial stream 3D CNN to produce optical flow. If the spatial stream is able to produce optical flow, it suggests that the temporal stream is unnecessary, since it does not have access to any information that the spatial stream cannot learn to produce on its own. On the other hand, if the 3D CNN is not able to produce optical flow, it could be due to one of two possibilities. First, it could be a fundamental limitation of 3D CNNs, that is, they are unable to learn optical flow from video. Second, it could suggest a limitation in the training procedure, that is, they are able to learn optical flow, but do not.

We will show that the second possibility is true: 3D CNNs do not learn motion representations such as optical flow naturally, and the issue lies with the training procedure. Specifically, we demonstrate that 3D CNNs do not learn sufficiently accurate optical flow when trained on action recognition, and that they can learn much more accurate optical flow when trained explicitly to do so.

3.1. Optical Flow Decoder

To predict optical flow, we use the hidden features from an intermediate layer in a 3D CNN and pass them through a decoder, as depicted in Figure 2. Since our goal is to evaluate the motion representations in the hidden features, we constrain the decoder such that it is unable to learn motion patterns beyond what is already learned by the 3D CNN. Specifically, the decoder contains no temporal convolutions,

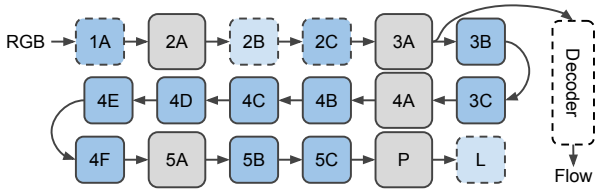

Figure 2. The network used to predict optical flow from 3D CNN features. We apply the decoder at hidden layers in the 3D CNN (depicted here at layer 3A). This diagram shows the structure of I3D/S3D-G, where blue boxes represent convolution (dashed lines) or Inception blocks (solid lines), and gray boxes represent pooling blocks [4, 41]. Layer names are the same as those used in Inception [33].

and operates on a single frame at a time.

In our experiments, the optical flow decoder is designed to mimic the optical flow prediction network from PWC-Net [31], but without the cost volume and warping layers. For more details on the architecture of this decoder, please refer to the supplementary materials.

The output of the decoder is a motion representation introduced by Im2Flow [10], which consists of three channels that encode optical flow: $(\mathrm{mag}, \sin\theta, \cos\theta)$, where mag and θ are the magnitude and angle, respectively, of the flow vector at each pixel. The decoder is trained to minimize the squared error between the predicted and target optical flow. For numerical stability, we weight the loss for the $\sin\theta, \cos\theta$ channels by mag. This encoding and training procedure have been shown in prior work to be more effective than directly regressing the optical flow vectors.

To match prior work, we use TV-L1 optical flow [42] as the motion representation [10, 37, 24]. TV-L1 optical flow is commonly used as the input to the temporal stream in many two-stream approaches [4, 26]. Therefore, it is known to be a useful motion representation for action recognition, and reconstructing it with a 3D CNN demonstrates how well the 3D CNN can capture useful motion representations.

3.2. Evaluation Metrics

After training the optical flow decoder, we evaluate the learned optical flow using endpoint error (EPE), a common metric that is adopted in prior work [10, 37, 24].

We evaluate in two settings. In the first setting, we freeze the 3D CNN and train the decoder. This setting tests what motion representations are learned by the 3D CNN naturally by training on action recognition. In the second setting, we fine-tune decoder and 3D CNN end-to-end. This setting tests what motion representations *can* be learned by a 3D CNN when optimized specifically for this purpose.

In Section 5.2, we demonstrate much better results in the second setting than in the first, suggesting there is room for improvement in the training procedure for spatial stream 3D CNNs. We also demonstrate that our proposed distilled method achieves improvements in this direction.

4. Distilled 3D Networks

Our goal is to incorporate motion representations from the temporal stream into the spatial stream. We approach this using distillation, that is, by optimizing the spatial stream to behave similarly to the temporal stream. Our approach uses the learned temporal stream from the typical two-stream pipeline as a teacher network, and the spatial stream as a student network. During training, we distill the knowledge from the teacher network into the student network, as depicted in Figure 1. This is accomplished by introducing a new loss function, which penalizes the outputs of the spatial stream if they are dissimilar to those of the temporal stream. More concretely, we train the network parameters θ to minimize the sum of two losses L_a and L_d,

$$L(\theta) = L_a(\theta) + \lambda L_d(\theta) \tag{1}$$

where the action classification loss L_a is the cross-entropy and the distillation loss L_d is the mean squared error between the pre-softmax outputs of the spatial stream $f_s(x;\theta)$ and that of the fixed temporal stream $f_t(x)$, i.e.

$$L_d(\theta) = \frac{1}{N}\sum_{i=0}^{N-1}(f_s(x^{(i)};\theta) - f_t(x^{(i)}))^2, \tag{2}$$

where $\{x^{(0)}, ..., x^{(N-1)}\}$ are the video clips. The hyperparameter λ allows us to flexibly rescale the contribution of the distillation loss. In our experiments, we find that $\lambda = 1$ conveniently serves as a good setting in many cases. Note that we use a mean squared error loss, as opposed to the cross-entropy loss proposed in prior work [14]. We find that this approach achieves similar results, and can be more flexibly applied to intermediate layers in the network.

We refer to a spatial stream f_s trained using distillation as a Distilled 3D Network (D3D). For inference, we discard the temporal stream f_t, skipping the optical flow step and relying only on RGB input. As we show in Section 5, D3D is able to achieve accuracy on par with two-stream methods without the need for two separate spatial and temporal streams. In addition, unlike other approaches for incorporating motion representations, we add no additional computational overhead to the spatial stream [23, 40, 32, 20]. We use S3D-G as the backbone architecture for both the spatial and temporal stream, since it achieves comparable accuracy at lower computational cost than competing architectures such as I3D and Non-local I3D [4, 40].

4.1. Implementation Details

We train D3D in two steps. First, we train the temporal stream using TV-L1 optical flow inputs. Second, we train the spatial stream using the distillation procedure described in Section 4. For inference, we discard the temporal stream.

When training the temporal stream, we use the same hyperparameters as those described in prior work [41]. When training the spatial stream, we also use the same hyperparameters as prior work, with the only change being the addition of our distillation loss. We use scaling parameter $\lambda = 1$ unless otherwise specified. We train the model for 140k steps on 56 GPUs with a batch size of 6 clips per GPU. For more details, please refer to prior work on S3D-G [41].

5. Experiments

We train and evaluate D3D on several datasets, and we demonstrate that D3D outperforms single-stream models and achieves accuracy on par with that of two-stream models that require explicit optical flow computation.

5.1. Datasets

Kinetics. Kinetics is a large-scale video classification dataset with approximately 500K 10-second clips annotated with one of 600 action categories [18, 3]. Kinetics has two variants: Kinetics-600 is the full dataset, and Kinetics-400 is an approximate subset containing 400 categories.

Kinetics consists of publicly available YouTube videos, which can be deleted by their owners at any time. Thus, Kinetics, like similar large-scale Internet datasets, gradually decays over time. Our experiments were conducted on a snapshot of the Kinetics dataset captured in October 2018, when Kinetics-400 contained 226K of the original 247K training examples (-8.4%) and Kinetics-600 contained 369K of the original 393K training examples (-6.1%). The change in both training and validation sets generates a small discrepancy between experiments conducted at different times. We explicitly denote results on the original Kinetics dataset with an asterisk (*) in all tables and provide the list of videos available at the time of our experiments to enable others to reproduce our results.

HMDB-51 and UCF-101. HMDB-51 and UCF-101 are action classification datasets composed of brief video clips, each containing one action [19, 29]. HMDB-51 contains 7,000 videos from 51 classes, and UCF-101 contains 13,320 videos from 101 classes. For both datasets, we report classification accuracy on the first test split.

AVA. AVA is a large-scale spatiotemporal action localization dataset that consists of 430 15-minute movie clips [13]. Each clip contains bounding box annotations at 1-second intervals for all actors in frame, and each actor is annotated with one or more action labels. In our experiments, we train on AVA v2.1, and report results on the validation set.

5.2. Predicting Optical Flow

In this experiment, we decode optical flow from the intermediate layers of a 3D CNN as described in Section 3.1. For the 3D CNN, we use the spatial stream of S3D-G, which is pretrained on Kinetics-400 and takes RGB videos as inputs. We train the decoder on 2 GPUs with a batchsize of 6 clips per GPU for 100K iterations, and otherwise use the same hyperparameters as S3D-G [41]. We measure performance using endpoint error (EPE) between the predicted and ground truth optical flow.

Features	Modality	EPE
All zeros	-	2.92
S3D-G	RGB	2.08
D3D	RGB	1.76
S3D-G+FT	RGB	1.34
S3D-G	Flow	0.63

Table 1. Effect of feature extractor on optical flow prediction. "All zeros" is a trivial decoder. "S3D-G" and "S3D-G+FT" refer to the 3D CNN with and without end-to-end fine-tuning. We add the optical flow decoder to the "3A" layer of S3D-G and train it to predict optical flow. Fine-tuning vastly improves performance, showing that motion representations can be improved during training.

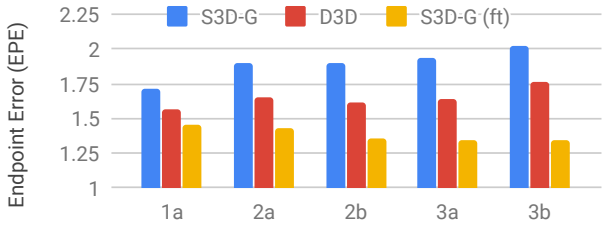

Figure 3. Predicting optical flow from multiple layers in S3D-G and D3D. The horizontal axis indicates which layer (see Figure 2) is used as input to the decoder. D3D features are able to more accurately reproduce optical flow across the board. Fine-tuning S3D-G end-to-end for flow prediction (indicated "ft") serves as a lower bound.

Fixed vs. Finetuning In Table 1, we demonstrate that the decoder can reproduce optical flow, but also that there is significant room for improvement. To bracket performance, we evaluate three baselines: (1) a trivial flow model that predicts "All zeros", (2) a decoder that is trained end-to-end with the 3D CNN, and (3) a decoder trained on the activations of a temporal stream model, which is provided TV-L1 flow as input. Compared to the baselines, the decoder trained on spatial stream S3D-G is able to approximately estimate optical flow. However, we find that the decoded flow is improved by finetuning the model end to end, meaning that motion representations could be improved by changing the training procedure of the 3D CNN.

Distillation and Flow Prediction In Figure 3, we compare the flow prediction performance of S3D-G and D3D when the decoder is applied at earlier layers. We observe lower error across the board when attempting to predict optical flow from D3D activations versus S3D-G activations.

While distillation improves optical flow prediction, it does not improve it to the same extent as full end-to-end fine-tuning. This shows that the two objectives, flow predic-

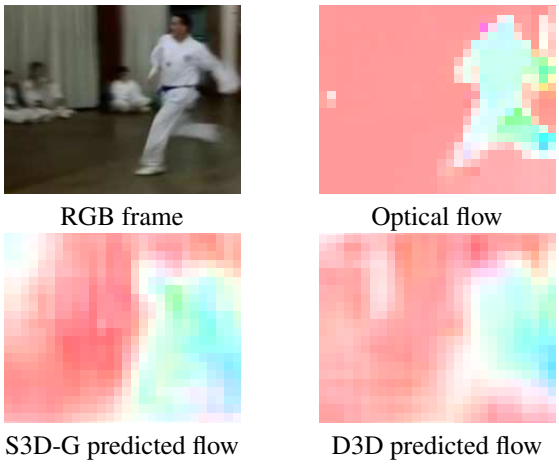

Figure 4. Examples of optical flow produced by S3DG and D3D (without fine-tuning) with the decoder applied at layer 3A. The color and saturation of each pixel corresponds to the angle and magnitude of motion, respectively. Optical flow is displayed at 28×28px, the output resolution of the decoder. Both S3D-G and D3D miss fine details, but D3D makes fewer mistakes.

tion and distillation, are complimentary but not completely overlapping. As we will show in Section 5.6, distillation improves action recognition accuracy while fine-tuning does not. This result leads to an important finding: improving motion representations directly does not improve action recognition performance, but improving action recognition performance does improve motion representations. Therefore, in order to improve action recognition performance, it is not sufficient to optimize directly for better optical flow prediction. Distillation takes an alternative approach. By imitating the behavior of the temporal stream, we are able to capture the motion features that are used by the temporal stream while ignoring those that are not.

In Figure 4, we give examples of optical flow estimates given using our method. Both S3D-G and D3D can capture coarse motion, but miss fine details. Results using D3D appear to have slightly more accurate motion boundaries, a quality which is known to be useful for temporal stream action recognition [7, 26], explaining the quantitative improvements in Table 1 and Figure 3. We provide more qualitative examples in the supplementary materials.

These results confirm our original hypothesis: 3D CNNs provided with RGB input have a limited natural tendency to capture the motion signal present in optical flow when trained on action classification. The ability to capture motion signal can be significantly enhanced with modified training objectives, such as distillation loss or by fine-tuning for optical flow prediction.

5.3. Distillation on Kinetics

Kinetics-400. In Table 2, we compare D3D with several competitive baselines. We report accuracy for I3D and S3D-G trained and evaluated on the reduced Kinetics-400 dataset described in Section 5.1. These replications were run with code provided by the original authors and use identical settings to the published papers. Direct comparison with S3D-G shows that the distillation procedure leads to a 1.9% improvement in top-1 accuracy, without any additional computational cost during inference. Per-class accuracy is provided in the supplementary materials. Furthermore, we ensemble D3D with S3D-G ("D3D+S3D-G") by averaging their softmax scores, and achieve a small boost in performance over the two-stream S3D-G approach which uses optical flow. Our ensemble achieves better performance than the two-stream equivalent, without the need to compute optical flow.

Method	Modality	Kinetics-400
ARTNet [39]	RGB+Flow	72.4*
TSN [35]	RGB+Flow	73.9*
R(2+1)D [36]	RGB+Flow	75.4*
NL I3D [40]	RGB	77.7*
SAN [1]	RGB+Flow+Audio	77.7*
I3D [4]	RGB	70.6 / 71.1*
I3D [4]	Flow	62.1 / 63.9*
I3D [4]	RGB+Flow	72.6 / 74.1*
S3D-G [41]	RGB	74.0 / 74.7*
S3D-G [41]	Flow	67.3 / 68.0*
S3D-G [41]	RGB+Flow	76.2 / 77.2*
D3D	RGB	75.9
D3D+S3D-G	RGB	76.5

Table 2. D3D on Kinetics-400. All numbers given are top-1 accuracy on the validation set. "D3D+S3D-G" refers to an ensemble of D3D and S3D-G. Numbers marked with an asterisk (*) are reported on the full Kinetics-400 set, those without are reported on the subset available as of October 2018 as described in Section 5.1.

Method	Modality	Kinetics-600
I3D [3]	RGB	73.6 / 71.9*
S3D-G [41]	RGB	76.6
S3D-G [41]	Flow	69.7
S3D-G [41]	RGB+Flow	78.6
D3D	RGB	77.9
D3D+S3D-G	RGB	79.1

Table 3. D3D on Kinetics-600. All numbers given are top-1 accuracy on the validation set. "D3D+S3D-G" refers to an ensemble of D3D and S3D-G. Numbers marked with an asterisk (*) are reported on the full Kinetics-600 set, those without are reported on the subset available as of October 2018 as described in Section 5.1. Results on I3D use different settings than in Table 2 [3].

Kinetics-600. In Table 3, we compare D3D with baseline methods on Kinetics-600. Both the teacher and student network are trained using Kinetics-600 in these experiments. We achieve a 1.3% improvement in single-model perfor-

Method	UCF-101	HMDB-51
P3D [25]	88.6	-
C3D [34]	82.3	51.6
Res3D [35]	85.8	54.9
ARTNet [39]	94.3	70.9
I3D [4]	95.6	74.8
R(2+1)D [36]	96.8	74.5
S3D-G [41]	96.8	75.9
I3D Two-Stream [4]	98.0	80.7
ActionFlowNet [22]	83.9	56.4
MFNet [20, 23]	-	56.8
Rep. Flow [23]	-	65.4
MV-CNN [43]	86.4	-
TVNet+IDT [7]	95.4	72.6
Hidden Two-Stream [44]	97.1	78.7
D3D (Kinetics-400 pretrain)	97.0	78.7
D3D (Kinetics-600 pretrain)	97.1	79.3
D3D Ensemble	97.6	80.5

Table 4. Fine-tuning D3D on UCF-101 and HMDB-51. Our numbers are top-1 accuracy on test split 1 for both datasets. "D3D Ensemble" refers to an ensemble of the two D3D models with different pretraining. No distillation is performed during fine-tuning.

mance using D3D, and further improvements by ensembling D3D and S3D-G together, outperforming two-stream S3D-G without the need for optical flow.

5.4. Transfer to UCF101, HMDB51

We demonstrate that D3D transfers to other action recognition datasets by fine-tuning D3D on UCF-101 and HMDB-51. For these experiments, we initialize the model using D3D pretrained on Kinetics. However, during fine-tuning, we use only the action classification loss, and not distillation. This avoids the temporal stream altogether, during both training and inference. While we could potentially benefit from applying distillation during fine-tuning as well, these experiments demonstrate that it is not necessary to do so. Each model is fine-tuned for 10k steps on 10 GPUs with a batch size of 6 per GPU, as described in [41].

In Table 4, we demonstrate that fine-tuning D3D outperforms many competitive baselines. The models in the top section of the table are strong baselines based on 3D CNNs, including S3D-G, which serves as a direct comparison to show that the benefit of distillation during pretraining persists after fine-tuning. The models in the middle section of the table all specifically address the problem of learning motion features without the use of optical flow. D3D outperforms all baselines and achieves essentially equal performance to Hidden Two-Stream when pretrained on Kinetics-400. Hidden Two-Stream uses two I3D models plus an optical flow prediction network, so for fair comparison we also ensemble two D3D models together, and show that this ensemble outperforms Hidden Two-Stream [44].

Method	Pretraining	AVA
I3D w/ RPN [12]	Kinetics-600	21.9
I3D w/ RPN + JFT [12]	Kinetics-400	22.8
S3D-G w/ ResNet RPN [13]	Kinetics-400	22.0
D3D w/ ResNet RPN	Kinetics-400	23.0

Table 5. Performance on AVA using different backbone networks. All numbers are frame-mAP on the validation set. Models with "+ ResNet RPN" use a separate pretrained RPN stream based on ResNet, while the others use the 3D features directly for the RPN. The S3D-G baseline includes changes over the previously published numbers, described in Section 5.5.

5.5. Transfer to AVA

We fine-tune D3D on the spatiotemporal localization dataset AVA, and demonstrate that D3D transfers to this new task. We use a similar approach to the baseline described in the original AVA paper [13], but adopt some changes introduced by a top entry in the 2018 AVA competition [12]. Like the AVA baseline, we use a Faster RCNN-style approach, with a pretrained region proposal network (RPN) based on ResNet, and video feature extractor backbone network based on 3D CNNs. Unlike this work, we use D3D in place of I3D as the backbone network. We also adopt the three key changes introduced in the competition entry [12]. First, we regress only one set of bounding box offsets per region proposal, rather than a different set of offsets per action class. Second, we train for 500k steps using synchronous training on 11 GPUs using a higher learning rate. Third, we add cropping and flipping augmentation during training. Unlike [12], we do not remove the ResNet RPN in either D3D or the S3D-G baseline.

In Table 5, we compare the use of D3D as a backbone network with S3D-G and I3D. Our approaches use 50 RGB frames and no optical flow. Direct comparison between S3D-G and D3D shows that using D3D leads to a 1% improvement in Frame-mAP over S3D-G. We also see comparable gains over I3D, and we still outperform the I3D-based approach when it includes additional ResNet features pretrained on JFT, an internal Google dataset [30].

5.6. Ablation study

In the top section of Table 6, we experiment with two alternative approaches to distillation, and demonstrate that D3D outperforms both alternatives. In both cases, we make slight modifications to prior work, described below, to allow for fair comparison with distillation.

S3D-G with 3D CNN flow. Recent approaches, such as TVNet and Hidden Two-Stream networks, improve the temporal stream by learning their motion representations end-to-end [44, 7]. To compare, we use the first few layers of S3D-G as an optical flow prediction network, and use this learned flow as input to the temporal stream. We use the optical flow prediction network as described in Section 3.1,

Method	Kinetics-400
S3D-G spatial stream	74.0
S3D-G temporal stream	67.3
S3D-G with 3D CNN flow	69.7
S3D-G with flow loss	74.3
D3D distilled at layer 2C	74.4
D3D distilled at layer 4C	74.5
D3D distilled from spatial stream	74.3
D3D	75.9

Table 6. Ablation studies. All numbers given are top-1 accuracy on the reduced Kinetics-400 validation set described in Section 5.1. D3D using our proposed approach outperforms all other approaches listed. See Section 5.6 for details.

and train this end-to-end with an S3D-G temporal stream. we use S3D-G pretrained to predict actions from optical flow. In our experiments, we find that this approach outperforms the temporal stream applied to TV-L1 optical flow, but still underperforms the spatial stream and D3D.

S3D-G with flow loss. Similar to ActionFlowNet [22], we use optical flow prediction as an auxiliary task to improve the spatial stream. We use the flow prediction network described in Section 3.1, but we optimize the model to jointly minimize the flow prediction loss and action classification loss. This is a more direct way of encouraging the network to learn motion representations. However, we find that this does not generally lead to better results on action classification, and distillation gives significantly better results. This is possibly due to the fact that the flow loss is dominated by background pixels, which take up most of the field of view but are not typically important cues for action recognition.

Distillation at other layers. The middle section of Table 6 demonstrates applying the distillation loss at intermediate layers. We find that applying the distillation loss at intermediate layers is not as effective as at the network outputs.

Distilling from the spatial stream. In the bottom section, "D3D distilled from spatial stream" uses the S3D-G spatial stream as the teacher network in place of the temporal stream. This shows that distillation alone does not explain the improvement of D3D over S3D-G. Crucially, we only see benefits when distilling from the temporal stream.

Different backbones. Distillation is agnostic to the 3D CNN architecture, and therefore can be used in combination with any architecture. In Table 7, we show that D3D improves I3D, S3D-G, and a modified version of S3D-G which includes 2 non-local blocks [40]. More details about non-local S3D-G are given in the supplementary. In all cases, we use S3D-G as the teacher network, showing that distillation can still work with cross-model transfer.

Ensembling D3D with Spatial and Temporal Streams. In Tables 2, 3, and 4, we demonstrate that it is beneficial to ensemble D3D with an additional spatial stream model.

Method	Modality	Kinetics-400
I3D [4]	RGB	70.6
S3D-G [41]	RGB	74.0
NL S3D-G	RGB	74.7
D3D (I3D)	RGB	72.3
D3D (S3D-G)	RGB	75.9
D3D (NL S3D-G)	RGB	76.0

Table 7. Backbone architectures. All numbers given are top-1 accuracy on the validation set. "D3D (I3D)" and "D3D (NL S3D-G)" refer to D3D with I3D and Non-Local S3D-G as the backbone architectures, respectively. Distillation gives a boost in performance in all architectures.

Method	Modality	Kinetics-600
S3D-G	RGB	76.6
D3D	RGB	77.9
D3D+S3D-G	RGB+Flow	77.6
D3D+S3D-G	RGB+RGB	79.1

Table 8. Ensembling. D3D benefits from ensembling with an additional spatial stream, but not a temporal stream.

However, in Table 8, we find that there is no similar benefit when ensembling D3D with a temporal stream model. This suggests that D3D already captures the signal present in S3D-G Flow, otherwise we would expect to see benefits by performing this ensemble.

6. Conclusions

We introduce D3D, a distilled 3D CNN which does not require optical flow during inference and still outperforms two-stream approaches. D3D does not require any changes to the network architecture, and therefore can be used in combination with any backbone network. Furthermore, we show that D3D transfers to other action recognition datasets without the need for further distillation. Finally, we study the ability to predict optical flow with 3D CNNs, and we show that while 3D CNNs have some limited capacity to learn motion representations, D3D improves these representation, and distillation is a more effective objective than directly optimizing for optical flow prediction. Our work shows that the optical flow stream can be discarded during inference for no penalty, calling into question whether optical flow is really necessary for action recognition. However, further work in this area needs to be done to see whether optical flow can be avoided during training as well.

Acknowledgements. Work was completed while JCS was an intern at Google Research. We thank our colleagues for their helpful feedback, including Cordelia Schmid, George Toderici, and Carl Vondrick.

References

[1] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017.

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

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

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[5] N. Crasto, P. Weinzaepfel, K. Alahari, and C. 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.

[6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[7] L. Fan, W. Huang, S. E. Chuang Gan, B. Gong, and J. Huang. End-to-end learning of motion representation for video understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[8] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[9] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born again neural networks. In *International Conference on Machine Learning (ICML)*, 2018.

[10] R. Gao, B. Xiong, and K. Grauman. Im2flow: Motion hallucination from static images for action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[11] N. C. Garcia, P. Morerio, and V. Murino. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[12] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018.

[13] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

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

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*. IEEE, 2011.

[20] S. Lee, M. Lee, S. Son, G. Park, and N. Kwak. Motion feature network: Fixed motion filter for action recognition. In *European Conference on Computer Vision (ECCV)*. IEEE, 2018.

[21] Z. Luo, J.-T. Hsieh, L. Jiang, J. C. Niebles, and L. Fei-Fei. Graph distillation for action detection with privileged modalities. In *European Conference on Computer Vision (ECCV)*. IEEE, 2018.

[22] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis. Actionflownet: Learning motion representation for action recognition. In *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.

[23] A. Piergiovanni and M. S. Ryoo. Representation flow for action recognition. *arXiv preprint arXiv:1810.01455*, 2018.

[24] S. L. Pintea, J. C. van Gemert, and A. W. Smeulders. Déja vu. In *European Conference on Computer Vision (ECCV)*. IEEE, 2014.

[25] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[26] L. Sevilla-Lara, Y. Liao, F. Guney, V. Jampani, A. Geiger, and M. J. Black. On the integration of optical flow and action recognition. *arXiv preprint arXiv:1712.08416*, 2017.

[27] G. A. Sigurdsson, S. K. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[29] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[30] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[31] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[32] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[35] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017.

[36] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[37] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[38] H. Wang and C. Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[39] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[40] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[41] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision (ECCV)*. IEEE, 2018.

[42] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*. Springer, 2007.

[43] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[44] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.