Action recognition with spatial-temporal discriminative filter banks

Brais Martinez, Davide Modolo, Yuanjun Xiong, Joseph Tighe
Amazon
braisa,dmodolo,yuanjx,tighej@amazon.com

Abstract

Action recognition has seen a dramatic performance improvement in the last few years. Most of the current state-of-the-art literature either aims at improving performance through changes to the backbone CNN network, or they explore different trade-offs between computational efficiency and performance, again through altering the backbone network. However, almost all of these works maintain the same last layers of the network, which simply consist of a global average pooling followed by a fully connected layer. In this work we focus on how to improve the representation capacity of the network, but rather than altering the backbone, we focus on improving the last layers of the network, where changes have low impact in terms of computational cost. In particular, we show that current architectures have poor sensitivity to finer details and we exploit recent advances in the fine-grained recognition literature to improve our model in this aspect. With the proposed approach, we obtain state-of-the-art performance on Kinetics-400 and Something-Something-V1, the two major large-scale action recognition benchmarks.

1. Introduction

Action recognition has seen significant advances in terms of overall accuracy in recent years. Most existing methods [2, 11, 26, 27, 31] treat this problem as a generic classification problem, of which the only difference from ImageNet [6] classification is that the input is now a video frame sequence. Thus, numerous efforts have been devoted to leveraging the temporal information. However, unlike objects in ImageNet, which are usually centered and occupy the majority of pixels, human activities are complex concepts. They involve many factors such as body movement, temporal structure, and human-object interaction.

An example of this complexity can be found in action classes like "eating a burger" and "eating a hot-dog". They both depict eating something, and have a similar body motion pattern. To correctly distinguish them, one has to focus on the object the person is interacting with. A even more difficult situation happens when we try to distinguish "picking up a phone" versus "picking up a hot-dog", where even the shape of the objects are similar. Another example is for "dribbling basketball" and "shooting basketball", where the objects and scenes are the same. The major cue for distinguishing them is the difference in the body motion pattern.

Figure 1. *Action recognition is a fine-grained recognition problem.* ***Left****: we illustrate samples from class pairs that are easily confused by a state-of-the-art method [2]. This confusion is due to these actions being visually extremely similar and they can only be distinguished by fine-grained information. In (b) we show the classification output of the proposed approach on these classes, as well the highly activated regions with fine-grained information, such as objects (red), motion pattern (blue), object texture (green).*

This complex nature of human activities dictates that rough modeling of the visual or temporal features will lead to confusion between many classes that share similar factors. A common example of rough modeling is the widely used classifier head of one global average pooling and one linear classifier in action recognition models [2, 26], a setup typical of image object recognition models [10] on ImageNet [6]. However, as illustrated in fig. 1, even a state-of-the-art CNN architecture, when learned with this setup, fails to distinguish two classes when they share similar factors. In this case both classes can only be distinguished by capturing fine-grained patterns.

In this work, we propose to tackle the complexity of human action classification by promoting the importance of

analyzing finer details. In fact, a multitude of works in fine-grained recognition have been dedicated to solving similar problems on images, like distinguishing bird species [30], car models [33, 13] and plants [25]. Taking inspiration from a recent fine-grained recognition work [29], we propose a novel design to improve the final classification stage of action recognition models. Its main advantage is its ability to better extract and utilize the fine-grained information of human activities for classification, which are otherwise not well preserved by the global average pooling mechanism alone. In particular, we propose to use three classification branches. The *first* branch is commonly used global average pooling classification head. The *second* and the *third* branches share a set of convolutions, spatial upsampling and max-pooling layers to help surface the fine-grained information of activities, and differ in terms of the classifier used. The three branches are trained jointly in an end-to-end manner. This new design is compatible with most of current state-of-the-art action recognition models, *e.g.* [27, 38] and can be applied to both 2D and 3D CNN-based action recognition methods (sec. 6).

We evaluate the proposed approach on two major large-scale action classification benchmarks: Kinetics-400 [12] (400 classes) and Something-Something-V1 [9] (174 classes). Our results show that models built with our simple approach surpass many state-of-the-art methods on both benchmarks. Furthermore, we also provide a detailed ablation studies and visualizations (sec. 5) to demonstrate the effectiveness of our approach. Our discoveries suggest that the proposed approach does indeed help distinguishing similar classes that are otherwise confused by the baseline models, which leads to the improved overall accuracy.

2. Related Work

Action recognition in videos. Action recognition in the deep learning era has been successfully tackled with 2D [20, 26] and 3D CNNs [2, 5, 11, 22, 23, 27, 31]. Most existing works focus on modeling of motion and temporal structures. In [20] the optical flow CNN is introduced to model short-term motion patterns. TSN [26] models long-range temporal structures using a sparse segment sampling in the whole video during training. 3D CNN based models [2, 11, 22, 27] tackle the temporal modeling using the added dimension of the convolution on the temporal axis, in the hope that the models will learn the hierarchical motion patters as in the image space. Several recent works have started to decouple the spatial and temporal convolution in 3D CNNs to achieve more explicit temporal modeling [5, 23, 31]. In [38, 32] the temporal modeling is further improved by tracking feature points or body joints over time.

Most of these methods treat action recognition as a video classification problem. These works tend to focus on how motion is captured by the networks and largely ignore what makes the actions unique. In this work, we provide insights specific to the nature of the action recognition problem itself, showing how it requires an increased sensitivity to finer details. Different from the methods above, our work is explicitly designed for fine-grained action classification. In particular, the proposed approach is inspired by recent advances in the fine-grained recognition literature, such as [29]. We hope that this work will help draw the attention of the community on understanding generic action classes as a fine-grained recognition problem.

Fine-grained action understanding. Understanding human activities as a fine-grained recognition problem has been explored for some domain specific tasks [17, 21]. For example, some works have been proposed for hand-gesture recognition [8, 14, 19], daily life activity recognition [18] and sports understanding [7, 24, 3, 1]. All these works build ad hoc solutions specific to the action domain they are addressing. Instead, we present a solution for generic action recognition and show that this can also be treated as a fine-grained recognition problem and that it can benefit from learning fine-grained information.

Fine-grained object recognition. Different from common object categories such as those in ImageNet [6], this field cares for objects that look visually very similar and that can only be differentiated by learning their finer details. Some examples including distinguishing bird [30] and plant [25] species and recognizing different car models [13, 33]. We refer to Zhao et. al [36] for an interesting and complete survey on this topic. Here we just remark the importance of learning the visual details of these fine-grained classes. Some works achieve this with various pooling techniques [16, 15]; some use part-based approaches [35, 34]; and others use attention mechanisms [37, 39]. More recently, Wang et. al [29] proposed to use a set of 1×1 convolution layers as a discriminative filter bank and use a spatial max pooling to find the location of the fine-grained information. Our method takes inspiration from this method and extends it to the task of video action recognition. We emphasize the importance of finding the fine-grained details in the spatio-temporal domain with high resolutions features.

3. Methodology

In this work we enrich the last layers of a classic action recognition network with three branches (fig. 2). We preserve the original global pooling branch as it has been shown to carry important discriminative clip-level information across all activities (sec. 3.1). At the same time, we propose two more branches to respond to very localized structures (sec. 3.2). Our intuition is that in order to learn fine-grained details, the network needs per-class local discrim-

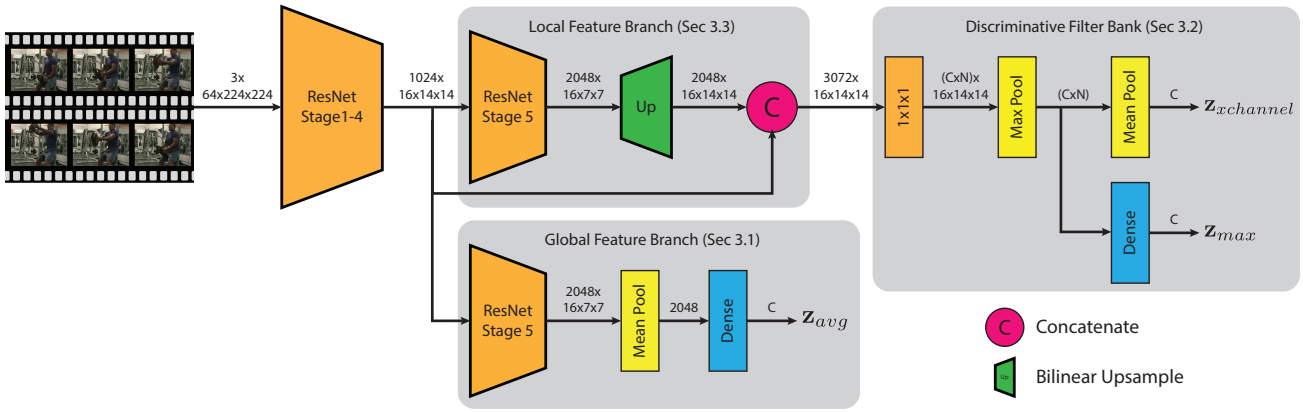

Figure 2. *Architecture diagram of our proposed approach. We illustrate the design with a 3D ResNet that takes 64 frames as input, but the overall design generalizes to both 2D and 3D architectures with an arbitrary number of input frames. The global feature branch (sec. 3.1) functions as our baseline. Our proposed approach improves upon this baseline with a bank of discriminative filters (sec. 3.2) that specialize on localized cues and a local feature extraction branch (sec. 3.3) that produces feature maps tuned to be sensitive to local patterns.*

inative classifiers from which it can pull unique signatures ($\mathbf{z}_{xchannel}$) and correlations across similar classes ($\mathbf{z}_{max}$). Finally, we propose to decouple the feature representation for the global pool branch and for the fine-grained branches to improve feature diversity (sec. 3.3).

3.1. Global Branch: Baseline Network

Our baseline method follows the recent approaches in action recognition [2, 26] and it consists of a classification backbone encoder followed by the standard global average pooling and a linear classifier. We denote this classification output as $\mathbf{z}_{avg}$. This pooling method is highly effective in capturing large contextual cues from across the video sequence, as it aggregates information from the entire spatial-temporal volume of the video via average pooling. It is designed to capture the gist of the full video sequence and, as such, it is quite effective at separating actions that take place in different scenes, like dancing vs playing golf. However, this pooling methodology limits the networks ability to focus on a particular local, spatial-temporal region as the global average forces the classifier to consider all parts of the video at once.

3.2. Discriminative Filter Bank

To allow the network to focus on key local regions of the video we introduce a set of local discriminative classifiers. Taking inspiration from [29], we model these classifiers as filters. Each of them specializes on a particular local cue that is important for a specific action class. The discriminative filters are implemented as $N \cdot C$, 1×1 (for 2D) or $1 \times 1 \times 1$ (for 3D) convolutions followed by global max pooling over the feature volume to compute the highest activation value of each discriminative classifier. Here C is the number of classes (400 for Kinetics-400 and 174 for Something-Something-V1), while N is a hyper-parameter indicating how many discriminative classifiers are associated to each class (in this work we set $N = 5$, as we did not observe any substantial improvement with a larger value). The max response from each filter for a given class is averaged to produce a final classification prediction. This produces a classification prediction that is highly localized, as the response comes from just N locations in the feature volume. Wang *et al*. [29] coined this classification method cross-channel pooling, as responses are pooled in blocks of N across the channel feature dimension. We adopt this nomenclature and denote this classifier output as $\mathbf{z}_{xchannel}$. We apply the standard softmax cross-entropy loss on this output and denote this loss as $\mathcal{L}_{xchannel}$. This loss directly encourages each filter to specialize on its class as their outputs are directly aggregated to produce the classification prediction.

This classifier is an aggregate of the N classifiers specialized for that particular class but does not include any of the $N \cdot (C-1)$ other discriminative filters. To allow each class to benefit from all filters, we add a dense layer to the discriminative filter output after the maxpool. We denote this classifier output as $\mathbf{z}_{max}$ and apply a softmax cross-entropy loss denoted as $\mathcal{L}_{max}$. This classifier draws from local activation across weak classifiers from all classes, thus it can be thought of as a middle ground between the highly localized $\mathbf{z}_{xchannel}$ and the global $\mathbf{z}_{avg}$

Finally, we combine the three classifier outputs ($\mathbf{z}_{avg}$, $\mathbf{z}_{xchannel}$ and $\mathbf{z}_{max}$) into a single prediction ($\mathbf{z}_{comb}$) through simple summation, pass it through another softmax layer and compute a combined loss $\mathcal{L}_{comb}$. The final loss used in training is just a summation of all factors:

$$\mathcal{L} = \mathcal{L}_{comb} + \mathcal{L}_{avg} + \mathcal{L}_{max} + \mathcal{L}_{xchannel} \quad (1)$$

We apply an auxiliary loss to each individual classification output ($\mathbf{z}_{avg}$, $\mathbf{z}_{xchannel}$ and $\mathbf{z}_{max}$) to force each to learn in isolation. All results reported in the experimental section use the aggregate classifier ($\mathbf{z}_{comb}$).

3.3. Local Detail Preserving Feature Branch

While our discriminative filters provide fine-grain local cues, they still operate on the same feature volume as the average pooled classifier ($\mathbf{z}_{avg}$). This has two issues: one, the feature volume has been down-sampled to such a high degree that the filters cannot learn the finer details; and two, this feature volume is shared between the average pooled ($\mathbf{z}_{avg}$) and the discriminative filters ($\mathbf{z}_{xchannel}$ and $\mathbf{z}_{max}$) classifiers, meaning that it cannot specialize for either task.

In order to overcome these issues and improve feature diversity, we propose to branch the last stage of our backbone and use one branch for our average pooling classifier (global branch) and the other (local branch) for our discriminative filters. This allows the global branch to specialize on context while the local branch can specialize on finer details.

Next, as we seek sensitivity to discriminative finer details, we add a bilinear upsampling operation in the local branch in charge of computing the discriminative classifiers (fig. 2) and add a skip connection from the features from stage 4. These modules provide a specialized and detailed feature volume for the discriminative filters, further enriching the information that the fine-grain classifiers can learn.

4. Experimental Setting

4.1. Datasets

We experiment on the two largest datasets for action recognition: Kinetics-400 [12] and Something-Something-V1 [9]. **Kinetics-400** consists of 400 actions. Its videos are from Youtube and most of them are 10 seconds long. The training set consists of around 240,000 videos and the validation set of around 19,800 videos, well balanced across all 400 classes. The test set labels are withheld for challenges, so it is a standard practice to report performance on the validation set. Kinetics-400 is an excellent dataset for evaluation thanks to its very large scale nature, its large intra-class variability and the extensive set of action classes. **Something-Something-V1** consists of 174 actions and it contains around 110,000 videos. These videos are shorter on average than those of Kinetics-400 and their duration typically spans from 2 to 6 seconds.

One interesting difference between these two datasets is that, on Kinetics-400, temporal information and temporal ordering of frames are not very important and a single 2D RGB CNN already achieves competitive results compared to a much more complex 3D architecture; on the other hand, temporal information is essential for Something-Something-V1 and a simple 2D RGB CNN achieves much lower performance than its 3D counterpart. This is due to the different nature of the actions in these datasets: while they are very specific on Kinetics-400 (e.g., 'building cabinet'), they are relatively generic on Something-Something-V1 (e.g., 'plugging something into something'). By experimenting on these diverse datasets, we show that our approach is generic and suitable to different action domains.

4.2. Implementation Details

Approaches. We experiment with two of the best performing backbones for our system: 2D TSN [26] and inflated 3D [2] networks. Briefly, **2D TSN networks** treat each frame as a separate image. The TSN sampling method divides an input video evenly into segments (we used 3 segments in this work) and samples 1 snippet (a single RGB frame for this work) per segment randomly during training. This ensures that a diverse set of frames are observed during training. Finally, a consensus function aggregates predictions from multiple frames to produce a single prediction per video clip, on which the standard soft-max cross entropy loss is applied. In this work we use the average consensus function, as it has been shown to be effective. As these networks operate on each frame independently, we use the models provided by Gluon [4], which is pre-trainined on ImageNet [6]. Rather than processing each frame independently, **inflated 3D networks** operate on a set of frames as a unit, convolving in both the spatial and temporal dimensions. These 3D networks are typically modeled after 2D architectures, converting 2D convolutions into 3D ones (i.e. a 3×3 convolution becomes a $3 \times 3 \times 3$ convolution). We follow the common practice to initialize the network weights by "inflating" [2] weights from a 2D network trained on ImageNet. We follow the 3D implementation of Wang et. al [27], which down-samples the temporal resolution by 4 in the initial convolutions at the beginning of stage 1, via strided convolutions. From that point on the temporal dimension remains fixed. 3D networks produce a spatial temporal feature tensor as illustrated in Fig. 2.

Network architecture. We use the ResNet backbone [10] in all our experiments. For all network configurations, instead of using stride 2 to down-sample the spatial resolution in the initial 1×1 convolution ($1 \times 1 \times 1$ for 3D) of the first Bottleneck block, we use it in the 3×3 convolution ($3 \times 3 \times 3$) of the block instead.

Network Initialization. We initialize our networks with the weights of a model pre-trained on the ImageNet dataset [6] as described above. The local stage 5 branch of the network does not have a corresponding pre-trained set of weights as it is not part of the standard ResNet architecture. Two obvious choices to initialize the local branch are random initialization or use the stage 5 weights from ImageNet pre-training. We conducted preliminary experiments

	2D		3D	
Method	Top-1	Top-5	Top-1	Top-5
GB (Baseline) sec. 3.1	69.6	88.3	66.8	86.0
GB + DF sec. 3.2	70.9	89.5	68.0	86.9
GB + DF + LB sec. 3.3	71.2	89.3	68.8	87.3

Table 1. *Results of the different components of our model on the Kinetics-400 dataset. Our baseline, which consists of a single global branch (GB) is consistently outperformed by our discriminative filters (DF) and add our specialized local branch (LB) compliments these filters, pushing performance even higher. Performance is computed using the training-time setting. Thus, 2D models use 3 segments while 3D models use one 16-frame segment.*

and found that using a random initialization for the local stage 5 branch weights gave consistently better results. We use this initialization for all results presented in this work. This is intuitive as we want the local branch to specialize for local cues, while the pre-trained ImageNet weights are already specialized for global context and are prone to remain close to that local minimum.

Optimization parameters. We use an SGD optimizer and a starting learning rate of 0.01, which we reduce three times by factor 10 during training. We set weight decay to $10e-5$, momentum to 0.9 and dropout to 0.5. Our batch size is as large as permitted by the hardware and it changes with the depth of the model, the approach used (i.e., 2D or 3D) and with the number of frames of the video. For example, when using 16 frames, ResNet152 and a 3D CNN, we can only use a batch of 8. Moreover, when using 64 frames, we use mixed precision (FP16). This allows us to use larger batch sizes than with FP32.

Input size and data augmentation. We resize all videos so the short edge is 256 pixels. We keep the temporal resolution (FPS) the same as the source files. During training, we augment the training examples with horizontal flipping, random resizing and random cropping of 224×224 pixels.

Test-time settings. We follow the standard procedures employed in the literature when comparing to other state-of-the-art methods. For the 2D models, we use 20 regularly-sampled segments at test time (a segment consists of only a frame in our case) and perform oversampling, which consists on extracting 5 crops and their flips for each segment. This results in 200 forward passes for each video. All these outputs are then averaged to obtain the final prediction. For the 3D networks, we use 10 segments. Instead of resizing each frame to a pre-defined input size, we run fully convolutional inference and average the predictions. Finally, we flip horizontally every other segment. While this is the standard testing protocol for Kinetics-400, on Something-Something-V1 we follow the standard practice to use a single segment for testing.

Meta-categories			
Significant improvement		*Marginal improvement*	
Waxing	+5.5	Interact w/ animals	+1.0
Swimming	+5.1	Makeup	+0.9
Cooking	+3.0	Watersports	+0.5
Hair	+3.0	Gymnastic	+0.4
Using Tools	+2.9	Athletics-Jumping	+0.4
Eating & Drinking	+2.6	Raquet-Batsports	+0.3

Table 2. *Top-1 accuracy improvement brought by our full 3D approach over the baseline 3D network.*

5. Analysis of our system

In this section we experiment with the proposed approach on the Kinetics-400 dataset. This dataset provides a challenging benchmark for our analysis as it consists of a large set of 400 classes. First, we present an ablation study on the components of our approach (sec. 5.1). Then, we look at what actions our model is helping and hurting the most (sec. 5.2). Finally, we examine qualitative results from our discriminative filter banks by visualizing their max response (sec. 5.3). We follow the standard convention and report results in terms of Top-1 and Top-5 accuracy.

5.1. Ablation study

We investigate how the various components of our approach contribute to its final performance. We conduct this ablation study with a simpler inference setting than that described in sec. 4.2. We sample 3 segments with our 2D network and only 1 16-frames segment with our 3D network, as it is computationally prohibitive to train and test for each model variation. Results are reported in table 1. The top row reports the results of our global branch (**GB**) baseline, which is the standard action recognition approach (sec. 3.1). Adding our discriminative filter bank (**DF**) to it (sec. 3.2) consistently improves performance across all models. Finally, further enriching our model with a specialized local branch (**LB** sec. 3.3) achieves the best performance. This shows that each component of our approach is important to improve the baseline performance. Moreover, the results show that this improvement is consistent on both 2D and 3D based networks and it shows that our approach is also generic to different action recognition baselines.

5.2. How does performance change across actions?

In this section we investigate how our model performs on different action classes and compare the results of the baseline 3D CNN network with those of our full approach.

First, we look at the improvement of our model on high level meta-categories. These are defined by the Kinetics-400 dataset and were originally generated by manually clustering the 400 classes into 38 parent classes, each one containing semantically similar actions [12]. We compute the Top-1 accuracy of a meta-category as the average Top-1 ac-

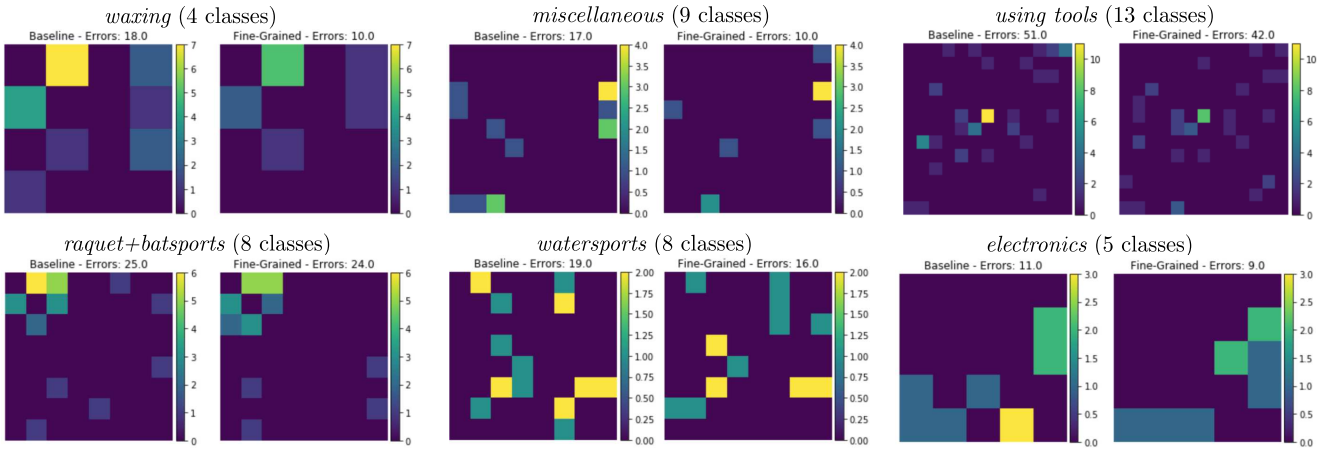

Figure 3. *Confusion matrices of 6 meta-categories. Our model (Fine-Grained) significantly improves Top-1 accuracy over the Baseline for the meta-categories in the top row, but only marginally for those in the bottom. Nevertheless, these confusion matrices show that our approach is much better at separating the actions within a meta-category, especially when these are visually similar.*

curacy of all its child actions. In table 2, we show some of the meta-categories for which we observed the largest (left) and the smallest (right) improvement. These results highlight some interesting facts. First, we observe a substantial improvement on meta-categories that contain very fine-grained actions. For example, *waxing* consists of classes like "waxing back", "waxing chest", "waxing legs", etc., which picture the same exact action, but performed on different body parts. Our approach is able to focus its attention on the relevant and discriminative regions of the videos (fig. 4, 5) and improve the performance on these fine-grained actions considerably (+5.5 on average for *waxing*). Similar trends can be observed for *swimming* (backstroke, breast stroke, and butterfly stroke) and several other meta-categories (table 2, left).

Interestingly, our models do not bring a similar improvement on some sport-related meta-categories, like *water-sports*, *gymnastic*, *athletics-jumping* and *raquet-batsports*. This is because these meta-categories contain actions that are not necessarily similar with each other and for which the baseline model is already accurate. For example, "playing tennis", "playing cricket" and "playing badminton" can be easily differentiated by looking at their background scenes (i.e., the courts). Nevertheless, it is worth noticing that even on these meta-categories that are by construction unlikely to benefit from our fine-grained design, our approach still slightly improves over the baseline. This shows that our approach is robust and, overall preferable over the baseline.

Next, we go beyond the accuracy of a meta-category and investigate its individual actions. We analyze the confusion matrices between the actions of each meta-category (fig. 3). These visualizations clearly show that our approach is able to produce sparser confusion matrices compared to the baseline, further verifying that our approach is more appropriate to distinguish visually similar classes. For example, let's consider the meta-category *watersports*. Table 2 shows that our approach outperforms the baseline by a tiny Top-1 accuracy improvement of 0.5. Nevertheless, their confusion matrices (fig. 3, mid-bottom) show that our fine-grained approach is capable of separating the classes within this meta-category much better than the baseline (e.g., the baseline confuses "surfing water" with "waterskiing" - cell (7,5) - but our approach differentiates them well).

Action	is most confused w/	# confused videos	
		baseline	ours
Bending metal	Welding	4	1
Mopping floor	Cleaning floor	8	4
Peeling potatoes	Baking cookies	6	2
Plastering	Laying bricks	4	0
Swing dancing	Salsa dancing	8	4
Brushing hair	Fixing hair	7	2
Waxing legs	Shaving legs	4	1
Getting a haircut	Shaving head	1	7
Stretching legs	Yoga	4	8
Strumming guitar	Playing guitar	6	13

Table 3. *Pairs of actions most confused by the baseline and our approach, along with the number of videos each model confuses.*

Finally, in table 3-top we list some of the actions that are most confused by the baseline and for which our approach is more accurate. Again, we can appreciate how our approach can better separate fine-grained actions, like "mopping floor" and "cleaning floor": while the baseline wrongly predicts "cleaning floor" on 8 out of the 40 validation videos of "mopping floor", our fine-grained approach only makes 4 mistakes. Moreover, in table 3-bottom we list actions for which our approach is more confused. While these results may seem surprising at first, they are easily explainable. These actions are connected to each other and they are in a parent-child relationship (e.g., "shaving head" is a type of "haircut") or they co-occur (e.g., "stretching

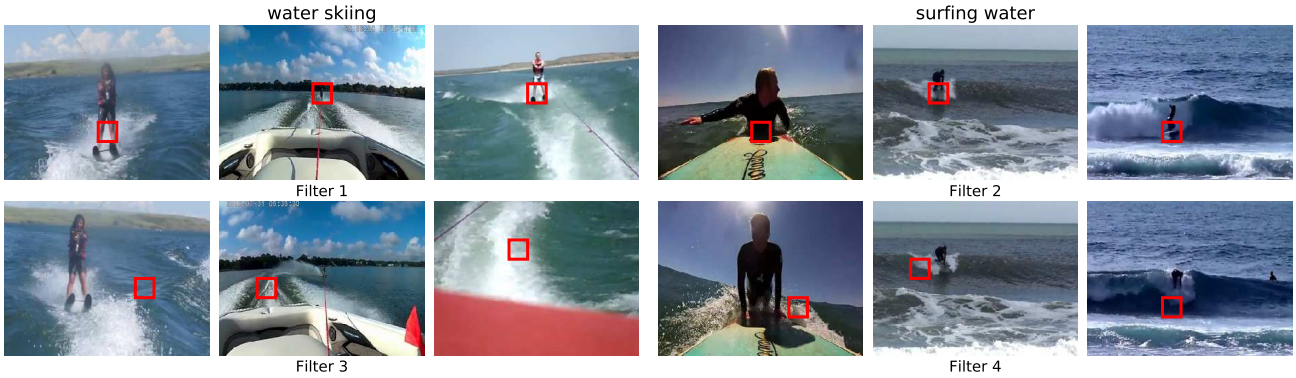

Figure 4. *Examples of 2 discriminative filters from two classes that roughly capture the same concepts. The top row shows three examples of the max response from filters that capture the person and object being ridden for the water skiing and surfing water classes. The second row shows filters that capture the texture of the water on the wake or wave in the video. Both of these cues combine to help the final classifier differentiate between these challenging classes.*

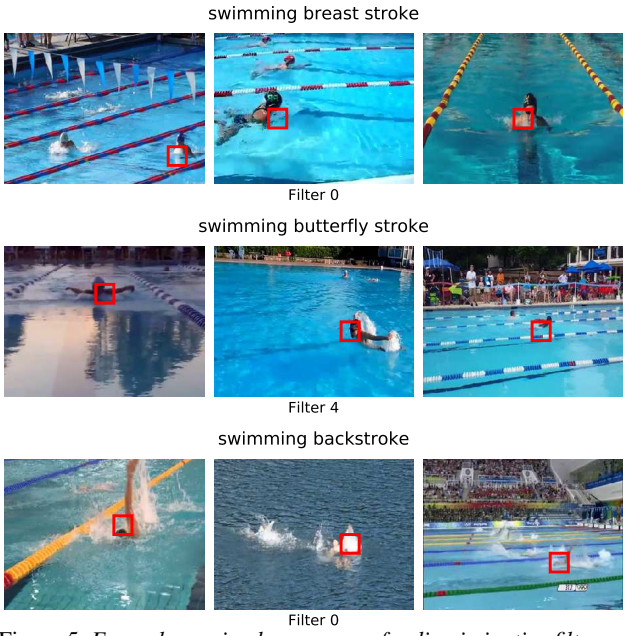

Figure 5. *Example maximal responses of a discriminative filter on three challenging swimming classes. Notice that each filter consistently fires on the frame and location of the swimmer in a canonical pose for that stroke. These filters are robust to variations in view-point and scenes.*

legs" is a common pose in "yoga"). While our model tries to learn fine-grained details of these actions, it gets confused by the ambiguity in their definition and the missing annotations, which lead to larger mistakes.

5.3. Qualitative analysis

Next, we examine qualitative examples of how our discriminate filters behave. To this end, we take activations from across an entire video of a single discriminative filter, which is trained to have a high response for a specific class. We then find the spatial-temporal location where this filter has its maximal response and draw a bounding box around the area in the frame that corresponds to that response. First, in fig. 5, we show a single filter maximal response on three test videos for each of three swimming classes (breast stroke, backstroke, and butterfly). Notice how each filter learns to fire on the precise frame and location of the corresponding stroke's canonical pose and is robust to view-point and scene variations. We further explore this phenomenon in fig. 4, where we visualize two filters for two often confused classes (water skiing and surfing water). Here we find two filters for each class that roughly fire on the same types of local cues. The top row of fig. 4 shows responses for filters that focus on the person and object they are riding (water skis or surf board), while the bottom row fires on the specific wave characteristics of the filter's corresponding class. The dense classifier ($\mathbf{z}_{max}$) is then able to take advantage of these cues when making the final prediction. Finally, in our teaser figure (fig. 1) we show more examples of how our filters are able to localize and disambiguate objects, motions, and textures.

6. Comparison with the state-of-the-art

In this section, we compare our method against the state-of-the-art in the literature. We use the train and test settings reported in sec. 4.2 and report Top-1 and Top-5 accuracy.

Kinetics-400 (table 4). We compare against 2D and 3D approaches. Our models achieve state-of-the-art performance on both scenarios. For 2D, enriching the TSN RGB stream with our fine-grained components improves both Top-1 and Top-5 accuracy by 1 point. Interestingly, our model also achieves slightly higher performance than the TSN two stream model, while remaining computationally much more efficient (Flow is extremely slow to compute and requires a forward pass through an entire second network). For 3D CNNS, we improve state-of-the-art Top-1 accuracy (1 point) and runtime. The previous best performing approach (Non-local Neural Networks) employs a more computationally intensive version of ResNet that has addi-

	Method	Modality	Top-1	Top-5	Pre-trained
2D	Two Stream (NIPS14) [20]	RGB + Flow	65.6	–	ImageNet
	TSN Two Stream (TPAMI18) [26]	RGB + Flow	73.9	91.1	ImageNet
	TSN One Stream (our impl.)	RGB	73.4	90.4	ImageNet
	Our approach	RGB	**74.3**	**91.4**	ImageNet
3D	I3D (CVPR'17) [2]	RGB	71.1	89.3	ImageNet
	I3D (CVPR'17) [2]	RGB+Flow	74.2	91.3	ImageNet
	R(2+1)D (CVPR'18) [23]	RGB	74.3	91.4	Sports-1M
	R(2+1)D (CVPR'18) [23]	RGB+Flow	75.4	91.9	Sports-1M
	Multi-Fiber (ECCV'18) [5]	RGB	72.8	90.4	None
	S3D (ECCV'18) [31]	RGB	72.2	90.6	ImageNet
	S3D-G (ECCV'18) [31]	RGB	74.7	93.4	ImageNet
	Non-local NN (CVPR'18) [27]	RGB	77.7	93.3	ImageNet
	Our approach	RGB	**78.8**	**93.6**	ImageNet

Table 4. *Comparison with state-of-the-art methods in the literature on the Kinetics-400 dataset.*

Method	Backbone	Top-1	Top-5	Pre-trained
3D-CNN [9] (ICCV'17)	BN-Inception	11.5	29.7	Sports-1M
MultiScale TRN [40] (ECCV'18)	BN-Inception	34.4	–	ImageNet
ECO lite [41] (ECCV'18)	BN-Inception+ResNet18	46.4	–	Kinetics-400
I3D + GCN [28] (ECCV'18)	ResNet50	43.3	75.1	Kinetics-400
Non-local I3D + GCN [28] (ECCV'18)	ResNet50	46.1	76.8	Kinetics-400
TrajectoryNet [38] (NIPS'18)	ResNet18	44.0	–	ImageNet
TrajectoryNet [38] (NIPS'18)	ResNet18	47.8	–	Kinetics-400
Our baseline (GB, sec. 3.1)	ResNet18	42.3	72.3	ImageNet
Our approach	ResNet18	45.0	74.8	ImageNet
Our approach	ResNet50	50.1	79.5	ImageNet
Our approach	ResNet152	**53.4**	**81.8**	ImageNet

Table 5. *Comparison with state-of-the-art methods in the literature on the Something-Something-V1 dataset.*

tional convolutions after most of its blocks. Instead, our approach uses a standard ResNet model that doesn't incur into this overhead.

Something-Something-V1 (table 5). We compare against 3D CNNs approaches which train on RBG only. Unfortunately, the literature reports results on Something-Something-V1 using different backbone architectures, which makes a direct comparison difficult, as some architectures are trivially better than others. In order to present a fair comparison, we state the backbones used by the different approaches in our table. We make the following observations. First, our approach outperforms our baseline model considerably, as it improves its Top-1 accuracy from 42.3 to 45.0 using Resnet18. Second, our approach with a ResNet18 backbone pre-trained on ImageNet improves over the previous state-of-the art (with the same settings) by 1% in Top-1 accuracy (45.0 vs 44.0, TrajectoryNet [38]). Third, we further improve our performance by training on deeper backbones and substantially increase Top-1 accuracy by 8.4% with ResNet152 backbone instead of ResNet18. Also note that TrajectoryNet improves its Top-1 accuracy by 3.8% by pre-training on the Kinetics-400 dataset. While we have not tried it, we expect a similar improvement, which can further boost our performance.

Our state-of-the-art results on these two datasets validate the strength of our technique and highlight the importance of modelling action recognition as a fine-grained problem.

7. Conclusions

We showed that the performance of action recognition can be pushed to a new state-of-the-art by improving the sensitivity of the network to finer details. Our approach only changes the later stages of the network, thus not adding significant computational cost. It is also compatible with other methods that focus on either adding representational capacity to the backbone, or improving its computational efficiency. We achieved state-of-the-art performance on two major large-scale action recognition benchmark datasets. Moreover, this improvement is shown to generalize across different backbones, and for both 2D and 3D networks. Given these results, we hope that this work will bring attention to a previously neglected aspect of action recognition, i.e., how to enable the network to represent and learn the finer details in human activities.

References

[1] Jürgen Assfalg, Marco Bertini, Alberto Del Bimbo, Walter Nunziati, and Pietro Pala. Soccer highlights detection and recognition using HMMs. In *IEEE International Conference on Multimedia and Expo*, 2002. 2

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 3, 4, 8

[3] Hua-Tsung Chen, Chien-Li Chou, Wen-Jiin Tsai, Suh-Yin Lee, and Jen-Yu Yu. Extraction and representation of human body for pitching style recognition in broadcast baseball video. In *IEEE International Conference on Multimedia and Expo*, 2011. 2

[4] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MxNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015. 4

[5] Yunpeng Chen, Yannis Kalantidis, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. In *European Conference on Computer Vision*, 2018. 2, 8

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1, 2, 4

[7] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, 2003. 2

[8] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[9] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision*, 2017. 2, 4, 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 4

[11] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. 1, 2

[12] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics human action video dataset. *arXiv:1705.06950*, 2017. 2, 4, 5

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 2

[14] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European Conference on Computer Vision*, 2016. 2

[15] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. *IEEE International Conference on Computer Vision*, 2017. 2

[16] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNNs for fine-grained visual recognition. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2017. 2

[17] Donald J Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *IEEE International Symposium on Wearable Computers*, 2005. 2

[18] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[19] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2016. 2

[20] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances on Neural Information Processing Systems*, 2014. 2, 8

[21] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[23] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 8

[24] Takamasa Tsunoda, Yasuhiro Komori, Masakazu Matsugu, and Tatsuya Harada. Football action recognition using hierarchical LSTM. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2017. 2

[25] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 2

[26] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2018. 1, 2, 3, 4, 8

[27] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 4, 8

[28] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *European Conference on Computer Vision*, 2018. 8

[29] Yaming Wang, Vlad I. Morariu, and Larry S. Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3

[30] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

[31] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2018. 1, 2, 8

[32] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[33] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[34] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[35] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision*, 2014. 2

[36] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017. 2

[37] Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 2017. 2

[38] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *Advances on Neural Information Processing Systems*. 2018. 2, 8

[39] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[40] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision*, 2018. 8

[41] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: Efficient convolutional network for online video understanding. In *European Conference on Computer Vision*, 2018. 8