The more fine-grained, the better for transfer learning

Farzaneh Mahdisoltani
University of Toronto, Vector Institute
farzaneh@cs.toronto.edu

Guillaume Berger
Twenty Billion Neurons Inc.
guillaume.berger@twentybn.com

Waseem Gharbieh
Twenty Billion Neurons Inc.
waseem.gharbieh@twentybn.com

Roland Memisevic
Twenty Billion Neurons Inc.
roland.memisevic@twentybn.com

David Fleet
University of Toronto, Vector Institute
fleet@cs.toronto.edu

Abstract

In this paper, we investigate the correlation between the degree of detail (granularity) in the source task and the quality of the learned features for transfer learning to new tasks. For this purpose, we design a DNN for action classification and video captioning. The same video encoding architecture is trained to solve multiple tasks with different granularity levels. In our transfer learning experiments, we fine-tune a network on a target task, while freezing the video encoding learned from the source task. Experiments reveal that training with more fine-grained tasks tends to produce better features for transfer learning. We use Something-Something dataset with over $220,000$ videos, and multiple levels of granularity of the target labels. With impressive coarse-grained and fine-grained classification results, our model introduces a strong baseline on the new Something-Something captioning task.

1 Introduction

Fine-grained video understanding entails recognition of actions, objects, and spatiotemporal relations. A successful framework needs to discriminate myriad variations of actions and interactions, not unlike the emergence of fine-grained tasks in visual object recognition. To enable extracting rich features from video, right kinds of tasks are needed to train the framework. There are various levels at which we can describe actions, and these levels of granularity match naturally with compositionality of language. For example, at a coarse-grained level we have actions like *'putting a pen'*. Then we can have similar actions that differ in relatively subtle ways, for instance, *'putting a pen beside the cup'*, *'putting the pen in the cup'*, or perhaps *'pretending to put the pen in the cup'*. Adding prepositions and categories like "pretending to put" gives us fine-grained action. The complexity of the task at this level requires features that capture spatial relations. As the complexity of the task begins to match the complexity of the world that we are trying to understand, it necessitates more powerful features in order to discriminate these different scenes.

A two-channel DNN architecture is designed for video encoding. The same architecture is then used for video classification and captioning. Training is performed on Something-Something dataset [1], with 50 coarse-grained action groups, which are further broken to 174 closely related action categories, and a caption authored by the crowd actor. These captions mirror the fine-grained action category, but with placeholder *Something* replaced by the specific object(s). The main contributions of this paper include:

1. **Explore the link between label granularity and feature quality**: We exploit 3 levels of granu-

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

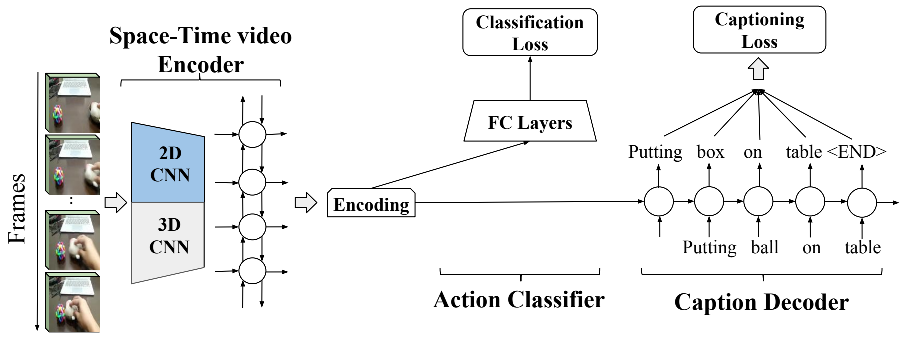

Figure 1: Our model architecture includes a video encoder, an action classifier, and an LSTM decoder for caption generation.

larity in Something-Something, namely, action groups, action categories, and captions. Experiments show that more fine-grained labels yield richer features.
2. **Baselines for captioning on Something-Something data**: We note that the captioning task is new for this dataset; the original version did not provide captions.
3. **Captioning as a source task for transfer learning**: We show that models trained for captioning learn features that transfer better to other tasks. To the best of our knowledge, captioning has, to date, been used as a target task. Our results suggest that captioning is a powerful source task.
4. **20bn-kitchenware**: We introduce a new dataset, ostensibly for video transfer learning.

2 Related Work

Video-based action classification dates back to seminal work by Laptev et al [2] with hand-tuned features, while most recent approaches have focused on DNN features. Existing methods differ in the way they aggregate information through time. Many approaches rely primarily on spatial features with CNNs applied to individual frames [3]. Other approaches make use of spatiotemporal information [4, 5]. Video captioning have received significant attention since the release of large-scale captioning corpora, notably, Microsoft COCO [6] and MSR-VTT [7]. Captioning tasks, if designed appropriately, could represent extremely detailed scene properties. Most existing captioning architectures are based on an encoder-decoder framework [8, 9, 10]. The encoder is typically a convolutional or recurrent convolutional network. Despite the significant attention to Video tasks, progress has lagged compared to static images, in part because of the lack of large-scale corpora. Using web sources and human annotators, larger datasets have been collected in recent years [11, 12]. More recently, crowd-sourced data have emerged, where crowd actors are asked to generate videos depicting template actions [1, 13]. One of the most astonishing properties of neural networks is their ability to learn representations that can be successfully transferred to other tasks[14, 15]. One motivation for studying fine-grained video tasks is to understand and improve the potential for transfer learning on video domain.

3 Architecture

The *video encoder*, inspired in part by magno- and parvo-cellular pathways in visual cortex, first processes the video through a spatial 2D-CNN and a spatio-temporal 3D-CNN in parallel (Fig. 2). Our video encoder is most closely related to approaches that perform temporal reasoning via a recurrent convolutional architecture [16, 17, 18]. It is also related to TwoStream architectures [19]; but our model does not explicitly use optical flow, opting instead for generic 3D CNN features. The basic building block of each channel is a $3 \times 3 \times 3$ (3×3 in 2D-CNN channel) convolution filter with batchnorm [20] and ReLU activation. Feature vectors from two channels are concatenated and then fed to a 2-layer bidirectional LSTM. We average these features to get an encoding of the entire video, h. This encoding is used by both the classifier and the captioning decoder(See Fig. 1). The *action classifier* applies an FC layer to the encoder output h, followed by a softmax layer. For training we use a cross-entropy loss over the action categories.

$$\text{loss}_{\textit{classification}} = -\log p(c|h;\theta). \tag{1}$$

The *caption decoder* is a two-layer LSTM which generates captions using a softmax over the vocabulary words, conditioned on previously generated words. The loss used for a caption is the

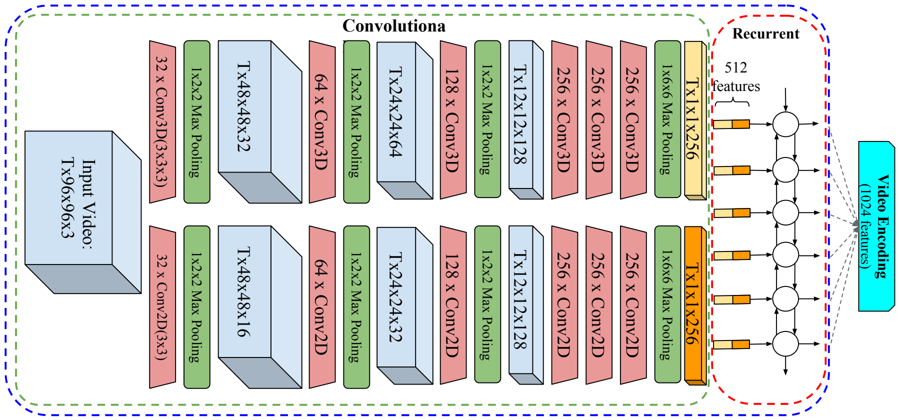

Figure 2: Our encoder includes a two-channel CNN followed by an LSTM for aggregating features.

Figure 3: 20bn-kitchenware samples: Using a knife to cut something (left), Trying but failing to pick something up with tongs(right).

usual negative log-probability of the word sequence:

$$\text{loss}_{captioning} = -\sum_{i=0}^{N-1} \log p(w^{i+1}|w^{\leq i}, h; \theta). \tag{2}$$

where w^i denotes the i^{th} word of the caption, h is the video encoding, and θ denotes model parameters. In order to optimize speed and memory usage during training, the length of captions generated by the decoder is fixed at 14 words. We train using teacher-forcing [21], however at test time, the input to the decoder at each time-step is the token generated at the previous time-step.

4 Tasks

We have trained our model end-to-end on 4 different tasks: Coarse-grained classification (on 50 action groups), fine-grained classification (on 174 action categories), captioning with simplified object placeholders and fine-grained captioning with full object placeholders. Labels with more subtle and fine-grained distinctions expose the ability (or inability) of a network to correctly infer the scene properties encoded in the captions.

Coarse- and fine-grained classification Something-Something provides coarse-grained categories called action groups, which comprise disjoint sets of fine-grained actions. Classification accuracy of our model is at 57.60% on action groups. We use the same architecture and train it on fine-grained action categories, and achieve 51.62%.

Captioning with simplified object placeholders We consider a captioning task in which we modify the ground truth captions to only contain one word per placeholder. Table 1 shows an example of the process. In the spectrum of granularity, captioning with simplified objects can be considered as a middle ground between fine-grained action classification and captioning with full labels.

Fine-grained captioning with full object placeholders We also train networks on the full object placeholders. This constitutes the finest level of action granularity. Table 2 summarizes the captioning results. We evaluate the models using standard captioning metrics: BLEU [22], ROUGE-L [23] and METEOR [24]. The captioning models produce impressive qualitative results with a high degree of approximate action and object accuracy. For qualitative examples of captioning and classification, please refer to the supplementary material.

5 Transfer Learning to 20bn-kitchenware:

We introduce *20bn-kitchenware*, a few-shot video classification dataset that contains 390 videos of 13 action categories. This dataset contains video clips of manipulating a kitchen utensil for roughly 4

Video ID	81955
Action Group	Holding [something]
Action Category	Holding [something] in front of [something]
Somethings	"a blue plastic cap", "a men's short sleeve shirt"
Simplified somethings's	"cap", "shirt"
Simplified-object Caption	Holding cap in front of shirt
Full Caption	Holding a blue plastic cap in front of a men short sleeve shirt

Table 1: An example of labels with different granularity levels for a Something-Something video

Task	BLEU@4	ROUGE-L	METEOR	Exact-Match Accuracy	Classification Accuracy
SO captions	23.04	44.89	22.60	8.63	51.38
Full captions	17.61	41.28	19.69	3.76	50.56

Table 2: Performance of our two-channel models for captioning with simplified and full object placeholders.

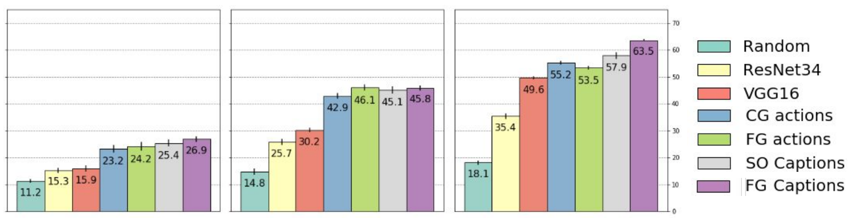

Figure 4: 20bn-kitchenware transfer learning results: averaged scores obtained using a VGG16, an Inflated ResNet34, as well as two-channel models trained on coarse-grained classification(CG), fine-grained classification(FG), simplified-object captions(SO), and full captions(FG). We report results using 1, 5, or 10 training samples per class.

seconds(see Fig. 8). For each utensil $X \in \{fork, spoon, knife, tongs\}$, the target label belongs to 1 of 3 actions, namely, *"Using X"*, *"Pretending to use X"* or *"Trying but failing to use X"*. We also include a fall-back class of *"Doing other things"*. We encourage the model to pay attention to visual details by including unused 'negative' objects in the scene.

5.1 Experiments

We explore transfer learning performance on 20bn-kitchenware as a function of source task granularity. We consider two-channel models that are pre-trained on the four aforementioned tasks. We also include a VGG16 network pre-trained on ImageNet, and an Inflated-ResNet34 pre-trained on Kinetics[1]. For each pre-trained model, we fine-tune an MLP with 512 units on top of the penultimate features from the frozen encoder, using only 10 samples per class. We evaluate 1-shot, 5-shot and 10-shot performance, averaging scores obtained over 10 runs. Figure 4 shows the average scores as well as 95% confidence intervals. Our results support the contention that training on fine-grained tasks leads to better features. The best model on this benchmark is our model trained on full captions. In all our experiments we use frame rate of $12fps$. During training we randomly pick 48 consecutive frames. For videos with less than 48 frames, we replicate the first and last frames to achieve the intended length. We resize the frames to 128×128, and then use random cropping of size 96×96. For validation and testing, we use 96×96 center cropping. We optimize all models using Adam, with an initial learning rate of 0.001.

6 Conclusion

Ever since ImageNet became popular as a generic feature extractor, a hypothesis has been that the dataset size, the amount of detail and the variety of labels, drive a network's capability to learn useful features. This paper provides further evidence for that hypothesis, showing that task granularity has a strong influence on the quality of the learned features for transfer learning. For the new task, given the limited amount of training data, the action granularity and the presence of negative objects, we hypothesize that only models that have some understanding of physical world properties will perform well on this dataset. Our experiments support that fine-grained tasks generally leads to better features.

[1]https://github.com/kenshohara/3D-ResNets-PyTorch

References

[1] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, 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 *Proceedings of the IEEE International Conference on Computer Vision*, ICCV17, 2017.

[2] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR08, 2008.

[3] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.

[4] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. pages 4489–4497, 2015.

[5] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, 2015.

[7] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 5288–5296. IEEE, 2016.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[9] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[10] Dotan Kaufman, Gil Levi, Tal Hassner, and Lior Wolf. Temporal tessellation for video annotation and summarization. *arXiv preprint arXiv:1612.06950*, 2016.

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. preprint arXiv:1705.06950.

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

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

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[15] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[16] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. pages 4207–4215, 2016.

[17] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. pages 29–39, 2011.

[18] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 2017.

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. pages 568–576, 2014.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, 2015.

[21] R.J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, 2002.

[23] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[24] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Supplementary Material

20bn-kitchenware:

Table 3 provides the full list of 20bn-kitchenware action categories.

Action categories
Using a fork to pick something up
Pretending to use a fork to pick something up
Trying but failing to pick something up with a fork
Using a spoon to pick something up
Pretending to use a spoon to pick something up
Trying but failing to pick something up with a spoon
Using a knife to cut something
Pretending to use a knife to cut something
Trying but failing to cut something with a knife
Using tongs to pick something up
Pretending to use tongs to pick something up
Trying but failing to pick something up with tongs
Doing other things

Table 3: The 13 action categories represented in 20bn-kitchenware.

The action categories in this dataset are somewhat ambiguous by design, we further encourage the model to pay attention to visual details by including unused 'negative' objects in the scene. The last row of Figure 8 shows one such example; while the target label indicates a manipulation of tongs, the clip also contains a spoon with an egg in it that could fool a model which simply recognizes objects.

Figure 5: Using a knife to cut something

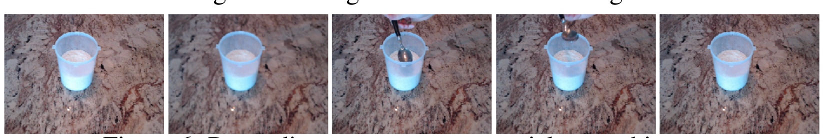

Figure 6: Pretending to use a spoon to pick something up

Figure 7: Trying but failing to pick something up with tongs

Figure 8: 20bn-kitchenware samples.

6.1 Baseline models for classification and captioning

As a classification baseline, we use ImageNet-pretrained models on individual frames, to which we then add additional layers. For the first baseline, we use just the middle frame of the video, with a classifier comprising a 2-layer MLP with 1024 hidden units. We also consider a baseline in which we apply this approach to all 48 frames, after which we average the frame by frame predictions. Lastly, we aggregate temporal information an LSTM layer with 1024 units. We report results in Table 5. There is a marked improvement with the LSTM, confirming that this task requires some form of temporal analysis. The number of features for VGG16 and Resnet152 are 4096 and 2048 respectively.
To the best of our knowledge there are no baselines for the Something-Something captioning task. To

quantify the performance of our captioning models, we count the percentage of generated captions that match ground truth word by word. We refer to this as "Exact-Match Accuracy". This is a challenging metric as the model is deemed correct only if it generates the entire caption correctly. If we use the action category predicted by our model trained for classification, and replace all occurrences of [something] with the most likely object string conditioned on that action class, the Exact-Match accuracy is 3.15%. The same baseline for simplified object placeholders is 5.69%. We also implemented a conventional encoder-decoder model for captioning 4.

Models	**BLEU@4**	**ROUGE-L**	**METEOR**	**Exact-Match Accuracy**	**Classification Accuracy**
VGG16+LSTM	31.83	52.22	24.79	3.13	31.69
Resnet152+LSTM	31.93	51.76	24.89	3.25	28.82

Table 4: Captioning baselines using a conventional encoder-decoder architecture.

Models	**Test Accuracy**
VGG16 + MLP 1024 (averaged over 48 frames)	17.57
VGG16 + LSTM 1024(48 steps)	**31.69**
ResNet152 + MLP 1024 (averaged over 48 frames)	16.79
ResNet152 + LSTM 1024 (48 steps)	**28.82**

Table 5: Classification results on 174 action categories using VGG16 and ResNet152 as frame encoders. For both MLP and LSTM we use 1024 hidden units before producing predictions.