Large-scale Video Classification with Convolutional Neural Networks

Andrej Karpathy[1,2] George Toderici[1] Sanketh Shetty[1]
karpathy@cs.stanford.edu gtoderici@google.com sanketh@google.com
Thomas Leung[1] Rahul Sukthankar[1] Li Fei-Fei[2]
leungt@google.com sukthankar@google.com feifeili@cs.stanford.edu
[1]Google Research [2]Computer Science Department, Stanford University
http://cs.stanford.edu/people/karpathy/deepvideo

Abstract

Convolutional Neural Networks (CNNs) have been established as a powerful class of models for image recognition problems. Encouraged by these results, we provide an extensive empirical evaluation of CNNs on large-scale video classification using a new dataset of 1 million YouTube videos belonging to 487 classes. We study multiple approaches for extending the connectivity of a CNN in time domain to take advantage of local spatio-temporal information and suggest a multiresolution, foveated architecture as a promising way of speeding up the training. Our best spatio-temporal networks display significant performance improvements compared to strong feature-based baselines (55.3% to 63.9%), but only a surprisingly modest improvement compared to single-frame models (59.3% to 60.9%). We further study the generalization performance of our best model by retraining the top layers on the UCF-101 Action Recognition dataset and observe significant performance improvements compared to the UCF-101 baseline model (63.3% up from 43.9%).

1. Introduction

Images and videos have become ubiquitous on the internet, which has encouraged the development of algorithms that can analyze their semantic content for various applications, including search and summarization. Recently, Convolutional Neural Networks (CNNs) [15] have been demonstrated as an effective class of models for understanding image content, giving state-of-the-art results on image recognition, segmentation, detection and retrieval [11, 3, 2, 20, 9, 18]. The key enabling factors behind these results were techniques for scaling up the networks to tens of millions of parameters and massive labeled datasets that can support the learning process. Under these conditions, CNNs have been shown to learn powerful and interpretable image features [28]. Encouraged by positive results in domain of images, we study the performance of CNNs in large-scale video classification, where the networks have access to not only the appearance information present in single, static images, but also their complex temporal evolution. There are several challenges to extending and applying CNNs in this setting.

From a practical standpoint, there are currently no video classification benchmarks that match the scale and variety of existing image datasets because videos are significantly more difficult to collect, annotate and store. To obtain sufficient amount of data needed to train our CNN architectures, we collected a new Sports-1M dataset, which consists of 1 million YouTube videos belonging to a taxonomy of 487 classes of sports. We make Sports-1M available to the research community to support future work in this area.

From a modeling perspective, we are interested in answering the following questions: what temporal connectivity pattern in a CNN architecture is best at taking advantage of local motion information present in the video? How does the additional motion information influence the predictions of a CNN and how much does it improve performance overall? We examine these questions empirically by evaluating multiple CNN architectures that each take a different approach to combining information across the time domain.

From a computational perspective, CNNs require extensively long periods of training time to effectively optimize the millions of parameters that parametrize the model. This difficulty is further compounded when extending the connectivity of the architecture in time because the network must process not just one image but several frames of video at a time. To mitigate this issue, we show that an effective approach to speeding up the runtime performance of CNNs is to modify the architecture to contain two separate streams of processing: a *context* stream that learns features on low-resolution frames and a high-resolution *fovea* stream that only operates on the middle portion of the frame. We

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.223

observe a 2-4x increase in runtime performance of the network due to the reduced dimensionality of the input, while retaining the classification accuracy.

Finally, a natural question that arises is whether features learned on the Sports-1M dataset are generic enough to generalize to a different, smaller dataset. We investigate the transfer learning problem empirically, achieving significantly better performance (65.4%, up from 41.3%) on UCF-101 by re-purposing low-level features learned on the Sports-1M dataset than by training the entire network on UCF-101 alone. Furthermore, since only some classes in UCF-101 are related to sports, we can quantify the relative improvements of the transfer learning in both settings.

Our contributions can be summarized as follows:

- We provide extensive experimental evaluation of multiple approaches for extending CNNs into video classification on a large-scale dataset of 1 million videos with 487 categories (which we release as Sports-1M dataset) and report significant gains in performance over strong feature-based baselines.
- We highlight an architecture that processes input at two spatial resolutions - a low-resolution context stream and a high-resolution fovea stream - as a promising way of improving the runtime performance of CNNs at no cost in accuracy.
- We apply our networks to the UCF-101 dataset and report significant improvement over feature-based state-of-the-art results and baselines established by training networks on UCF-101 alone.

2. Related Work

The standard approach to video classification [26, 16, 21, 17] involves three major stages: First, local visual features that describe a region of the video are extracted either densely [25] or at a sparse set of interest points [12, 8]. Next, the features get combined into a fixed-sized video-level description. One popular approach is to quantize all features using a learned k-means dictionary and accumulate the visual words over the duration of the video into histograms of varying spatio-temporal positions and extents [13]. Lastly, a classifier (such as an SVM) is trained on the resulting "bag of words" representation to distinguish among the visual classes of interest.

Convolutional Neural Networks [15] are a biologically-inspired class of deep learning models that replace all three stages with a single neural network that is trained end to end from raw pixel values to classifier outputs. The spatial structure of images is explicitly taken advantage of for regularization through restricted connectivity between layers (local filters), parameter sharing (convolutions) and special local invariance-building neurons (max pooling). Thus, these architectures effectively shift the required engineering from feature design and accumulation strategies to design of the network connectivity structure and hyperparameter choices. Due to computational constraints, CNNs have until recently been applied to relatively small scale image recognition problems (on datasets such as MNIST, CIFAR-10/100, NORB, and Caltech-101/256), but improvements on GPU hardware have enabled CNNs to scale to networks of millions of parameters, which has in turn led to significant improvements in image classification[11], object detection [20, 9], scene labeling [3], indoor segmentation [4] and house number digit classification [19]. Additionally, features learned by large networks trained on ImageNet [7] have been shown to yield state-of-the-art performance across many standard image recognition datasets when classified with an SVM, even with no fine-tuning [18].

Compared to image data domains, there is relatively little work on applying CNNs to video classification. Since all successful applications of CNNs in image domains share the availability of a large training set, we speculate that this is partly attributable to lack of large-scale video classification benchmarks. In particular, commonly used datasets (KTH, Weizmann, UCF Sports, IXMAS, Hollywood 2, UCF-50) only contain up to few thousand clips and up to few dozen classes. Even the largest available datasets such as CCV (9,317 videos and 20 classes) and the recently introduced UCF-101[22] (13,320 videos and 101 classes) are still dwarfed by available image datasets in the number of instances and their variety [7]. Despite these limitations, some extensions of CNNs into the video domain have been explored. [1] and [10] extend an image CNN to video domains by treating space and time as equivalent dimensions of the input and perform convolutions in both time and space. We consider these extensions as only one of the possible generalizations in this work. Unsupervised learning schemes for training spatio-temporal features have also been developed, based on Convolutional Gated Restricted Boltzmann Machines [23] and Independent Subspace Analysis [14]. In contrast, our models are trained end to end fully supervised.

3. Models

Unlike images which can be cropped and rescaled to a fixed size, videos vary widely in temporal extent and cannot be easily processed with a fixed-sized architecture. In this work we treat every video as a bag of short, fixed-sized clips. Since each clip contains several contiguous frames in time, we can extend the connectivity of the network in time dimension to learn spatio-temporal features. There are multiple options for the precise details of the extended connectivity and we describe three broad connectivity pattern categories (Early Fusion, Late Fusion and Slow Fusion) below. Afterwards, we describe a multiresolution architecture for addressing the computational efficiency.

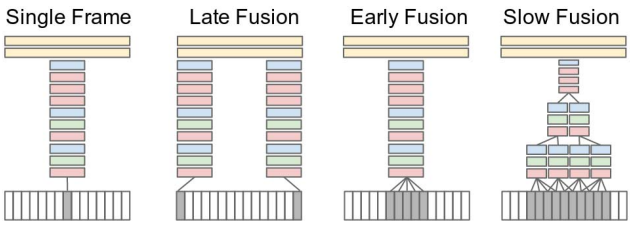

Figure 1: Explored approaches for fusing information over temporal dimension through the network. Red, green and blue boxes indicate convolutional, normalization and pooling layers respectively. In the Slow Fusion model, the depicted columns share parameters.

3.1. Time Information Fusion in CNNs

We investigate several approaches to fusing information across temporal domain (Figure 1): the fusion can be done early in the network by modifying the first layer convolutional filters to extend in time, or it can be done late by placing two separate single-frame networks some distance in time apart and fusing their outputs later in the processing. We first describe a baseline single-frame CNN and then discuss its extensions in time according to different types of fusion.

Single-frame. We use a single-frame baseline architecture to understand the contribution of static appearance to the classification accuracy. This network is similar to the ImageNet challenge winning model [11], but accepts inputs of size $170 \times 170 \times 3$ pixels instead of the original $224 \times 224 \times 3$. Using shorthand notation, the full architecture is $C(96, 11, 3)$-N-P-$C(256, 5, 1)$-N-P-$C(384, 3, 1)$-$C(384, 3, 1)$-$C(256, 3, 1)$-P-$FC(4096)$-$FC(4096)$, where $C(d, f, s)$ indicates a convolutional layer with d filters of spatial size $f \times f$, applied to the input with stride s. $FC(n)$ is a fully connected layer with n nodes. All pooling layers P pool spatially in non-overlapping 2×2 regions and all normalization layers N are defined as described in Krizhevsky et al. [11] and use the same parameters: $k = 2, n = 5, \alpha = 10^{-4}, \beta = 0.5$. The final layer is connected to a softmax classifier with dense connections.

Early Fusion. The Early Fusion extension combines information across an entire time window immediately on the pixel level. This is implemented by modifying the filters on the first convolutional layer in the single-frame model by extending them to be of size $11 \times 11 \times 3 \times T$ pixels, where T is some temporal extent (we use $T = 10$, or approximately a third of a second). The early and direct connectivity to pixel data allows the network to precisely detect local motion direction and speed.

Late Fusion. The Late Fusion model places two separate single-frame networks (as described above, up to last convolutional layer $C(256, 3, 1)$ with shared parameters a distance of 15 frames apart and then merges the two streams in the first fully connected layer. Therefore, neither single-frame tower alone can detect any motion, but the first fully connected layer can compute global motion characteristics by comparing outputs of both towers.

Slow Fusion. The Slow Fusion model is a balanced mix between the two approaches that slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions. This is implemented by extending the connectivity of all convolutional layers in time and carrying out temporal convolutions in addition to spatial convolutions to compute activations, as seen in [1, 10]. In the model we use, the first convolutional layer is extended to apply every filter of temporal extent $T = 4$ on an input clip of 10 frames through valid convolution with stride 2 and produces 4 responses in time. The second and third layers above iterate this process with filters of temporal extent $T = 2$ and stride 2. Thus, the third convolutional layer has access to information across all 10 input frames.

3.2. Multiresolution CNNs

Since CNNs normally take on orders of weeks to train on large-scale datasets even on the fastest available GPUs, the runtime performance is a critical component to our ability to experiment with different architecture and hyperparameter settings. This motivates approaches for speeding up the models while still retaining their performance. There are multiple fronts to these endeavors, including improvements in hardware, weight quantization schemes, better optimization algorithms and initialization strategies, but in this work we focus on changes in the architecture that enable faster running times without sacrificing performance.

One approach to speeding up the networks is to reduce the number of layers and neurons in each layer, but similar to [28] we found that this consistently lowers the performance. Instead of reducing the size of the network, we conducted further experiments on training with images of lower resolution. However, while this improved the running time of the network, the high-frequency detail in the images proved critical to achieving good accuracy.

Fovea and context streams. The proposed multiresolution architecture aims to strike a compromise by having two separate streams of processing over two spatial resolutions (Figure 2). A 178×178 frame video clip forms an input to the network. The context stream receives the downsampled frames at half the original spatial resolution (89×89 pixels), while the fovea stream receives the center 89×89 region at the original resolution. In this way, the the total input dimensionality is halved. Notably, this design takes advantage of the camera bias present in many online videos, since the object of interest often occupies the center region.

Architecture changes. Both streams are processed by identical network as the full frame models, but starting at

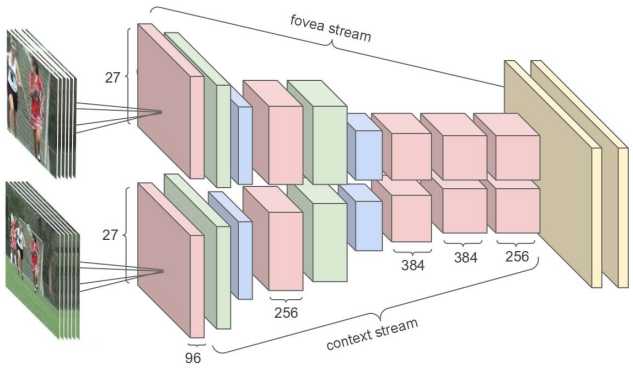

Figure 2: Multiresolution CNN architecture. Input frames are fed into two separate streams of processing: a *context stream* that models low-resolution image and a *fovea stream* that processes high-resolution center crop. Both streams consist of alternating convolution (red), normalization (green) and pooling (blue) layers. Both streams converge to two fully connected layers (yellow).

89×89 clips of video. Since the input is only of half the spatial size as the full-frame models, we take out the last pooling layer to ensure that both streams still terminate in a layer of size $7 \times 7 \times 256$. The activations from both streams are concatenated and fed into the first fully connected layer with dense connections.

3.3. Learning

Optimization. We use Downpour Stochastic Gradient Descent [6] to optimize our models across a computing cluster. The number of replicas for each model varies between 10 and 50 and every model is further split across 4 to 32 partitions. We use mini-batches of 32 examples, momentum of 0.9 and weight decay of 0.0005. All models are initialized with learning rates of $1e^{-3}$ and this value is further reduced by hand whenever the validation error stops improving.

Data augmentation and preprocessing. Following [11], we take advantage of data augmentation to reduce the effects of overfitting. Before presenting an example to a network, we preprocess all images by first cropping to center region, resizing them to 200×200 pixels, randomly sampling a 170×170 region, and finally randomly flipping the images horizontally with 50% probability. These preprocessing steps are applied consistently to all frames that are part of the same clip. As a last step of preprocessing we subtract a constant value of 117 from raw pixel values, which is the approximate value of the mean of all pixels in our images.

4. Results

We first present results on our Sports-1M dataset and qualitatively analyze the learned features and network predictions. We then describe our transfer learning experiments on UCF-101.

4.1. Experiments on Sports-1M

Dataset. The Sports-1M dataset consists of 1 million YouTube videos annotated with 487 classes. The classes are arranged in a manually-curated taxonomy that contains internal nodes such as *Aquatic Sports, Team Sports, Winter Sports, Ball Sports, Combat Sports, Sports with Animals,* and generally becomes fine-grained by the leaf level. For example, our dataset contains 6 different types of bowling, 7 different types of American football and 23 types of billiards.

There are 1000-3000 videos per class and approximately 5% of the videos are annotated with more than one class. The annotations are produced automatically by analyzing the text metadata surrounding the videos. Thus, our data is weakly annotated on two levels: first, the label of a video may be wrong if the tag prediction algorithm fails or if the provided description does not match the video content, and second, even when a video is correctly annotated it may still exhibit significant variation on the frame level. For example, a video tagged as soccer may contain several shots of the scoreboard, interviews, news anchors, the crowd, etc.

We split the dataset by assigning 70% of the videos to the training set, 10% to a validation set and 20% to a test set. As YouTube may contain duplicate videos, it is possible that the same video could appear in both the training and test set. To get an idea about the extent of this problem we processed all videos with a near-duplicate finding algorithm on the frame level and determined that only 1755 videos (out of 1 million) contain a significant fraction of near-duplicate frames. Furthermore, since we only use a random collection of up to 100 half-second clips from every video and our videos are 5 minutes and 36 seconds in length on average, it is unlikely that the same frames occur across data splits.

Training. We trained our models over a period of one month, with models processing approximately 5 clips per second for full-frame networks and up to 20 clips per second for multiresolution networks on a single model replica. The rate of 5 clips per second is roughly 20 times slower than what one could expect from a high-end GPU, but we expect to reach comparable speeds overall given that we use 10-50 model replicas. We further estimate the size of our dataset of sampled frames to be on the order of 50 million examples and that our networks have each seen approximately 500 million examples throughout the training period in total.

Video-level predictions. To produce predictions for an entire video we randomly sample 20 clips and present each clip individually to the network. Every clip is propagated through the network 4 times (with different crops and flips)

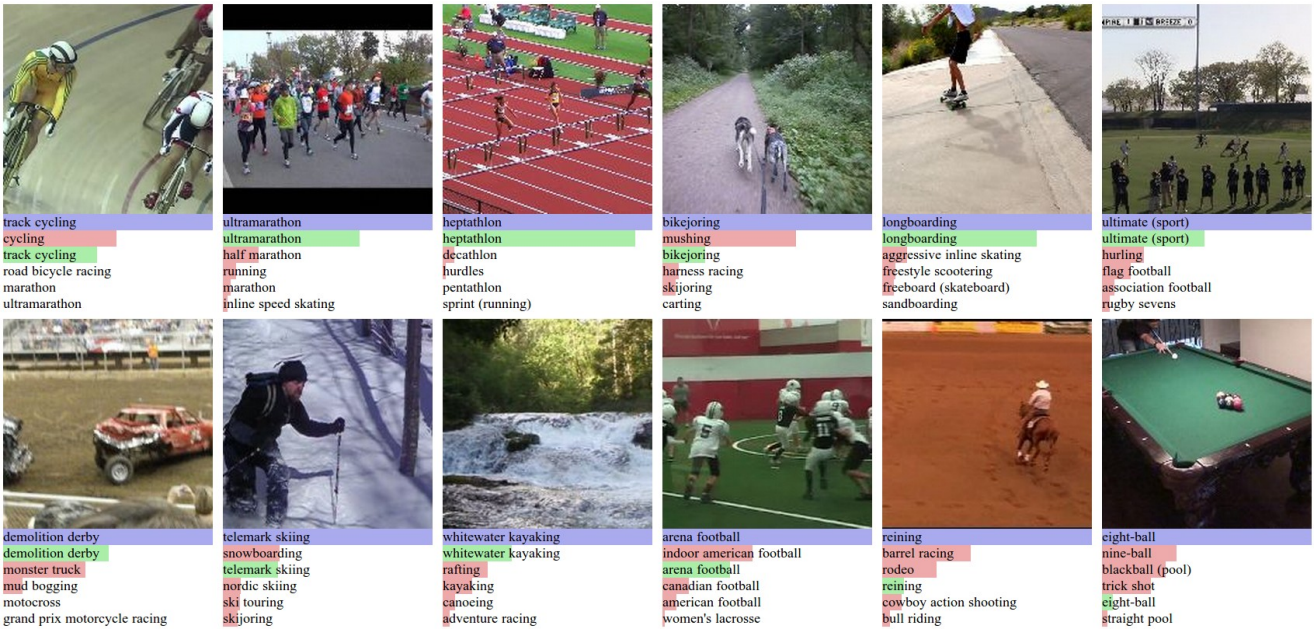

Figure 4: Predictions on Sports-1M test data. Blue (first row) indicates ground truth label and the bars below show model predictions sorted in decreasing confidence. Green and red distinguish correct and incorrect predictions, respectively.

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	**60.9**	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

Table 1: Results on the 200,000 videos of the Sports-1M test set. Hit@k values indicate the fraction of test samples that contained at least one of the ground truth labels in the top k predictions.

and the network class predictions are averaged to produce a more robust estimate of the class probabilities. To produce video-level predictions we opted for the simplest approach of averaging individual clip predictions over the durations of each video. We expect more elaborate techniques to further improve performance but consider these to be outside of the scope of the paper.

Feature histogram baselines. In addition to comparing CNN architectures among each other, we also report the accuracy of a feature-based approach. Following a standard bag-of-words pipeline we extract several types of features at all frames of our videos, discretize them using k-means vector quantization and accumulate words into histograms with spatial pyramid encoding and soft quantization. Every histogram is normalized to sum to 1 and all histograms are concatenated into a 25,000 dimensional video-level feature vector. Our features are similar to Yang & Toderici [27] and consist of local features (HOG [5], Texton [24], Cuboids [8], etc.) extracted both densely and at sparse interest points, as well as global features (such as Hue-Saturation, Color moments, number of faces detected). As a classifier we use a multilayer neural network with Rectified Linear Units followed by a Softmax classifier. We found that a multilayer network performs consistently and significantly better than linear models on separate validation experiments. Furthermore, we performed extensive cross-validations across many of the network's hyperparameters by training multiple models and choosing the one with best performance on a validation set. The tuned hyper parameters include the learning rate, weight decay, the number of hidden layers (between 1-2), dropout probabilities and the

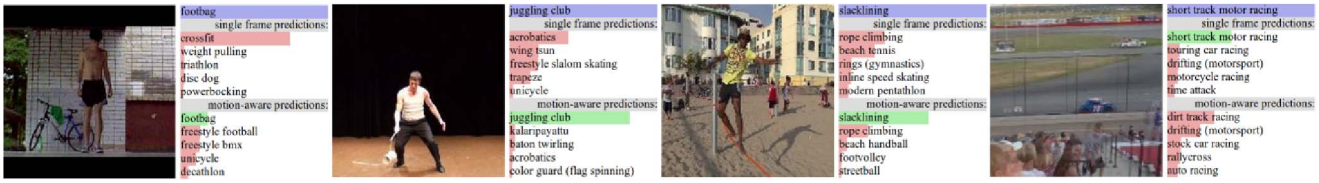

Figure 5: Examples that illustrate qualitative differences between single-frame network and Slow Fusion (motion-aware) network in the same color scheme as Figure 4. A few classes are easier to disambiguate with motion information (left three).

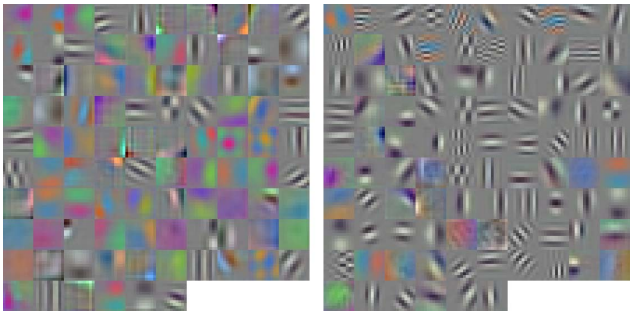

Figure 3: Filters learned on first layer of a multiresolution network. Left: context stream, Right: fovea stream. Notably, the fovea stream learns grayscale, high-frequency features while the context stream models lower frequencies and colors. GIFs of moving video features can be found on our website (linked on first page).

Sports class	Δ AP	Δ AP	Sports class
Juggling Club	0.12	-0.07	Short Track Motor Racing
Pole Climbing	0.10	-0.07	Road Racing
Mountain Unicycling	0.08	-0.07	Jeet Kune Do
Tricking	0.07	-0.06	Paintball
Footbag	0.07	-0.06	Freeride
Skipping Rope	0.06	-0.06	Cricket
Rope Climbing	0.06	-0.06	Wrestling
Slacklining	0.05	-0.06	Modern Pentathlon
Tee Ball	0.05	-0.06	Krav Maga
Sheepdog Trial	0.05	-0.05	Rally Cross

Table 2: Classes for which a (motion-aware) Slow Fusion CNN performs better than the single-frame CNN (left) and vice versa (right), as measured by difference in per-class average precision.

number of nodes in all layers.

Quantitative results. The results for the Sports-1M dataset test set, which consists of 200,000 videos and 4,000,000 clips, are summarized in Table 1. As can be seen from the table, our networks consistently and significantly outperform the feature-based baseline. We emphasize that the feature-based approach computes visual words densely over the duration of the video and produces predictions based on the entire video-level feature vector, while our networks only see 20 randomly sampled clips individually. Moreover, our networks seem to learn well despite significant label noise: the training videos are subject to incorrect annotations and even the correctly-labeled videos often contain a large amount of artifacts such as text, effects, cuts, and logos, none of which we attempted to filter out explicitly.

Compared to the wide gap relative to the feature-based baseline, the variation among different CNN architectures turns out to be surprisingly insignificant. Notably, the single-frame model already displays strong performance. Furthermore, we observe that the foveated architectures are between 2-4$\times$ faster in practice due to reduced input dimensionality. The precise speedups are in part a function of the details of model partitioning and our implementation, but in our experiments we observe a speedup during training of 6 to 21 clips per second (3.5x) for the single-frame model and 5 to 10 clips per second (2x) for the Slow Fusion model.

Contributions of motion. We conduct further experiments to understand the differences between the single-frame network and networks that have access to motion information. We choose the Slow Fusion network as a representative motion-aware network because it performs best. We compute and compare the per-class average precision for all Sports classes and highlight the ones that exhibit largest differences (Table 2). Manually inspecting some of the associated clips (Figure 5), we qualitatively observe that the motion-aware network clearly benefits from motion information in some cases, but these seem to be relatively uncommon. On the other hand, balancing the improvements from access to motion information, we observe that motion-aware networks are more likely to underperform when there is camera motion present. We hypothesize that the CNNs struggle to learn complete invariance across all possible angles and speeds of camera translation and zoom.

Qualitative analysis. Our learned features for the first convolutional layer can be inspected on Figure 3. Interestingly, the context stream learns more color features while the high-resolution fovea stream learns high frequency grayscale filters.

As can be seen on Figure 4, our networks produce interpretable predictions and generally make reasonable mistakes. Further analysis of the confusion matrix (attached in the supplementary material) reveals that most errors are among the fine-grained classes of our dataset. For example, the top 5 most commonly confused pairs of classes are deer hunting vs. hunting, hiking vs. backpacking, powered paragliding vs. paragliding, sledding vs. toboggan, and bujinkan vs. ninjutsu.

Model	3-fold Accuracy
Soomro et al [22]	43.9%
Feature Histograms + Neural Net	59.0%
Train from scratch	41.3%
Fine-tune top layer	64.1%
Fine-tune top 3 layers	**65.4**%
Fine-tune all layers	62.2%

Table 3: Results on UCF-101 for various Transfer Learning approaches using the Slow Fusion network.

Group	mAP from scratch	mAP fine-tune top 3	mAP fine-tune top
Human-Object Interaction	0.26	0.55	0.52
Body-Motion Only	0.32	0.57	0.52
Human-Human Interaction	0.40	0.68	0.65
Playing Musical Instruments	0.42	0.65	0.46
Sports	**0.57**	**0.79**	**0.80**
All groups	0.44	0.68	0.66

Table 4: Mean Average Precision of the Slow Fusion network on UCF-101 classes broken down by category groups.

4.2. Transfer Learning Experiments on UCF-101

The results of our analysis on the Sports-1M dataset indicate that the networks learn powerful motion features. A natural question that arises is whether these features also generalize to other datasets and class categories. We examine this question in detail by performing transfer learning experiments on the UCF-101 [22] Activity Recognition dataset. The dataset consists of 13,320 videos belonging to 101 categories that are separated into 5 broad groups: Human-Object interaction (Applying eye makeup, brushing teeth, hammering, etc.), Body-Motion (Baby crawling, push ups, blowing candles, etc.), Human-Human interaction (Head massage, salsa spin, haircut, etc.), Playing Instruments (flute, guitar, piano, etc.) and Sports. This grouping allows us to separately study the performance improvements on Sports classes relative to classes from unrelated videos that are less numerous in our training data.

Transfer learning. Since we expect that CNNs learn more generic features on the bottom of the network (such as edges, local shapes) and more intricate, dataset-specific features near the top of the network, we consider the following scenarios for our transfer learning experiments:

Fine-tune top layer. We treat the CNN as a fixed feature extractor and train a classifier on the last 4096-dimensional layer, with dropout regularization. We found that as little as 10% chance of keeping each unit active to be effective.

Fine-tune top 3 layers. Instead of only retraining the final classifier layer, we consider also retraining both fully connected layers. We initialize with a fully trained Sports CNN and then begin training the top 3 layers. We introduce dropout before all trained layers, with as little as 10% chance of keeping units active.

Fine-tune all layers. In this scenario we retrain all network parameters, including all convolutional layers on the bottom of the network.

Train from scratch. As a baseline we train the full network from scratch on UCF-101 alone.

Results. To prepare UCF-101 data for classification we sampled 50 clips from every video and followed the same evaluation protocol as for Sports across the 3 suggested folds. We reached out to the authors of [22] to obtain the YouTube video IDs of UCF-101 videos, but unfortunately these were not available and hence we cannot guarantee that the Sports-1M dataset has no overlap with UCF-101. However, these concerns are somewhat mitigated as we only use a few sampled clips from every video.

We use the Slow Fusion network in our UCF-101 experiments as it provides the best performance on Sports-1M. The results of the experiments can be seen on Table 3. Interestingly, retraining the softmax layer alone does not perform best (possibly because the high-level features are too specific to sports) and the other extreme of fine-tuning all layers is also not adequate (likely due to overfitting). Instead, the best performance is obtained by taking a balanced approach and retraining the top few layers of the network. Lastly, training the entire network from scratch consistently leads to massive overfitting and dismal performance.

Performance by group. We further break down our performance by 5 broad groups of classes present in the UCF-101 dataset. We compute the average precision of every class and then compute the mean average precision over classes in each group. As can be seen from Table 4, large fractions of our performance can be attributed to the Sports categories in UCF-101, but the other groups still display impressive performance considering that the only way to observe these types of frames in the training data is due to label noise. Moreover, the gain in performance when retraining only the top to retraining the top 3 layers is almost entirely due to improvements on non-Sports categories: Sports performance only decreases from 0.80 to 0.79, while mAP improves on all other categories.

5. Conclusions

We studied the performance of convolutional neural networks in large-scale video classification. We found that CNN architectures are capable of learning powerful features from weakly-labeled data that far surpass feature-based methods in performance and that these benefits are surprisingly robust to details of the connectivity of the architectures in time. Qualitative examination of network outputs and confusion matrices reveals interpretable errors.

Our results indicate that while the performance is not particularly sensitive to the architectural details of the connectivity in time, a Slow Fusion model consistently performs better than the early and late fusion alternatives. Sur-

prisingly, we find that a single-frame model already displays very strong performance, suggesting that local motion cues may not be critically important, even for a dynamic dataset such as Sports. An alternative theory is that more careful treatment of camera motion may be necessary (for example by extracting features in the local coordinate system of a tracked point, as seen in [25]), but this requires significant changes to a CNN architecture that we leave for future work. We also identified mixed-resolution architectures that consist of a low-resolution context and a high-resolution fovea stream as an effective way of speeding up CNNs without sacrificing accuracy.

Our transfer learning experiments on UCF-101 suggest that the learned features are generic and generalize other video classification tasks. In particular, we achieved the highest transfer learning performance by retraining the top 3 layers of the network.

In future work we hope to incorporate broader categories in the dataset to obtain more powerful and generic features, investigate approaches that explicitly reason about camera motion, and explore recurrent neural networks as a more powerful technique for combining clip-level predictions into global video-level predictions.

Acknowledgments: We thank Saurabh Singh, Abhinav Shrivastava, Jay Yagnik, Alex Krizhevsky, Quoc Le, Jeff Dean and Rajat Monga for helpful discussions.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer, 2011. 2, 3

[2] D. Ciresan, A. Giusti, J. Schmidhuber, et al. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, 2012. 1

[3] L. N. Clement Farabet, Camille Couprie and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8), 2013. 1, 2

[4] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *Internatinal Conference on Learning Representation*, 2013. 2

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, 2005. 5

[6] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In *NIPS*, 2012. 4

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005. 2, 5

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013. 2, 3

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 3, 4

[12] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 2

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[14] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 2

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1, 2

[16] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. 2

[17] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, pages 392–405. Springer, 2010. 2

[18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014. 1, 2

[19] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *ICPR*, 2012. 2

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1, 2

[21] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2

[22] 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. 2, 7

[23] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*. Springer, 2010. 2

[24] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-2):61–81, 2005. 5

[25] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*. IEEE, 2011. 2, 8

[26] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009. 2

[27] W. Yang and G. Toderici. Discriminative tag learning on youtube videos with latent sub-tags. In *CVPR*, 2011. 5

[28] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013. 1, 3