Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

By Joa˜o Carreira and Andrew Zisserman

Presenter: Zhisheng Huang

03/02/2018

Outline:

- Introduction
- Action classification architectures
- Implementation
- Experiment and results
- Discussion
- Improvements

Quo Vadis: Where are you going?

What is action recognition?

The goal of human action recognition is to automatically detect and classify ongoing activities from an input video.

Question: Is there a benefit in transfer learning from videos?

Action classification architectures

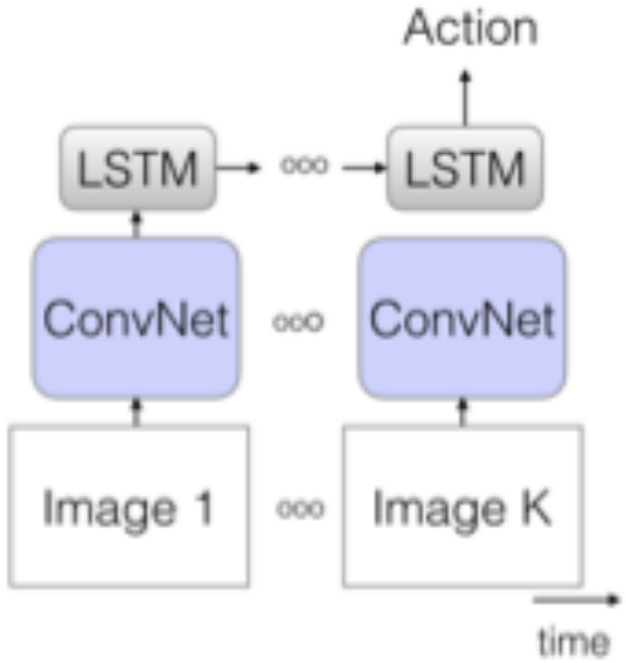
—The Old 1: ConvNet+LSTM

Structure:

- Add a LSTM layer with batch normalization after the last average pooling layer
- A FC layer is added on top to classifier
- Only consider the output on the last frame during testing

Limitations:

- May not be able to capture fine low-level motion
- Expensive to train



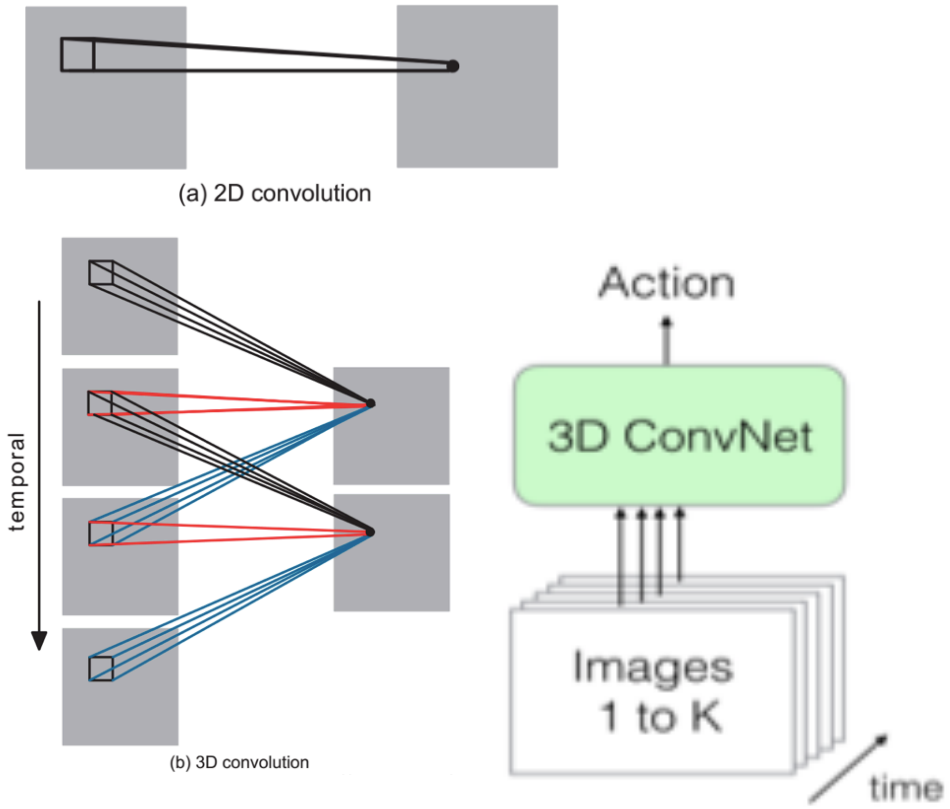
— The Old 2: 3D ConvNets

Structure:

- Having spatio-temporal filters
- 8 Conv layers, 5 pooling layers and 2 FC layers
- Using BN after all convolutional and FC layers
- Using a temporal stride of 2 in the first pooling layer
- Input: short 16-frame clips with 112x112-pixel crops

Limitations:

- Having more parameters than 2D ConvNet
- Precluding the benefits of ImageNet pre-training
- Low accuracy



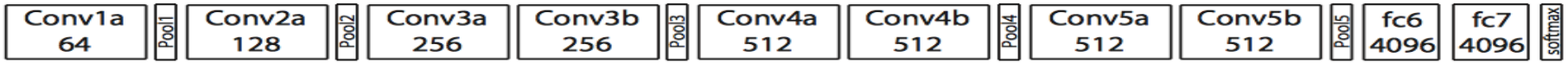


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling kernels are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

—The Old 3: Two-Stream Networks

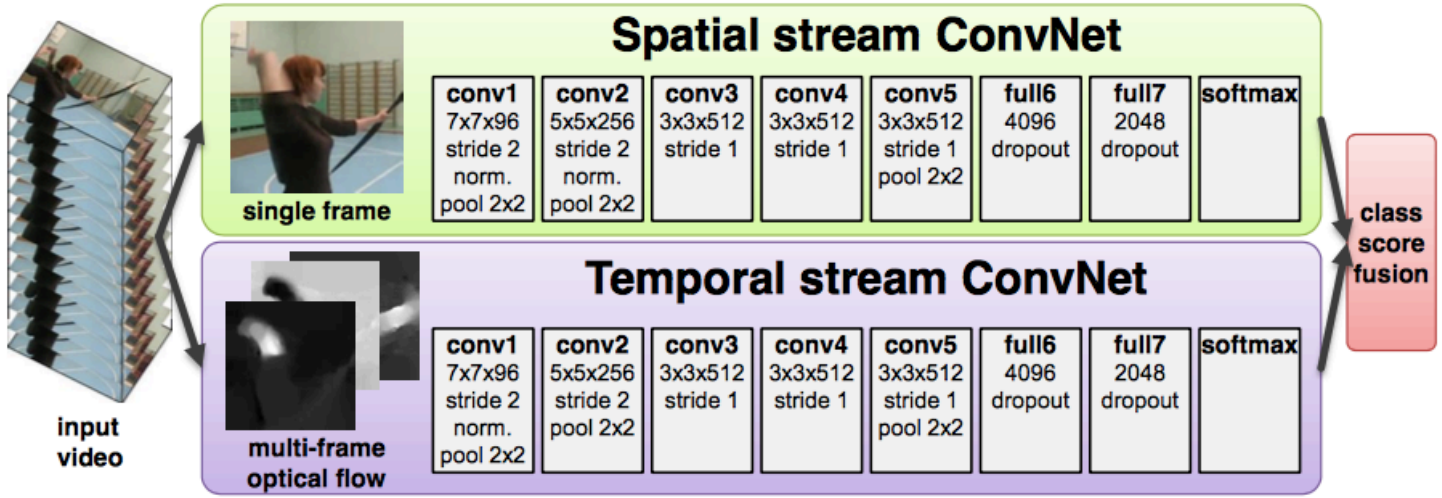
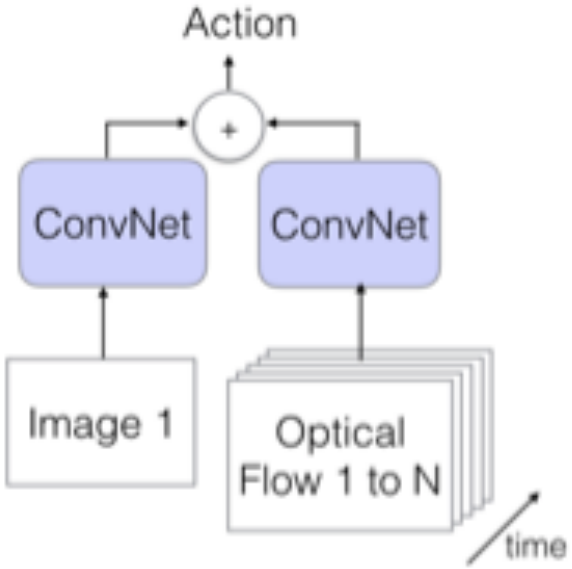


Figure 1: **Two-stream architecture for video classification.**



Structure

- 2 inputs: a RGB frame and 10 optical flow frame
- Using an ImageNet pre-trained ConvNet
- Averaging the predictions from two ConvNets
- Trained end-to-end

Limitations:

- Optical flow frames are computed externally

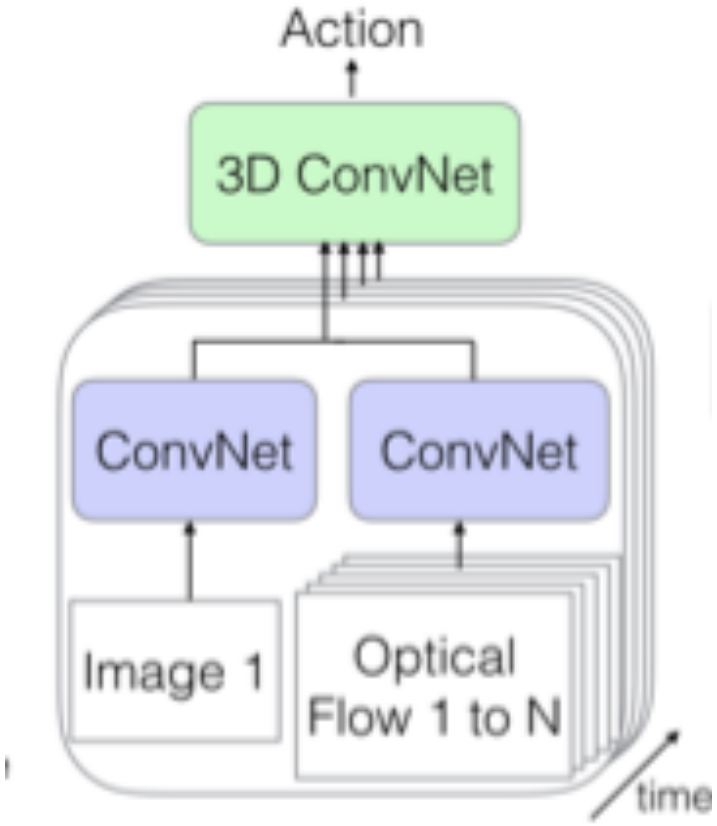
— The old 4: 3D-Fused Two-Stream Network

Structure:

- Fusing the spatial and flow streams after the last network convolutional layer
- Using Inception-V1
- 3D ConvNet with a 3x3x3 3D Conv layer with 512 output channels and a 3x3x3 3D max-pooling layer followed by a fully connected layer
- Input: 5 consecutive RGB frames and corresponding optical flow snippets
- Trained end-to-end

Limitations:

- Optical flow frames are compute externally
- Too many parameters

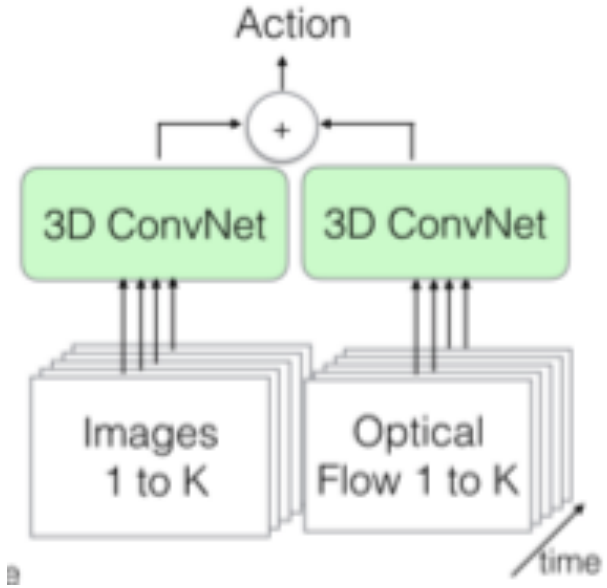


—The new: Two-Stream Inflated 3D ConvNets

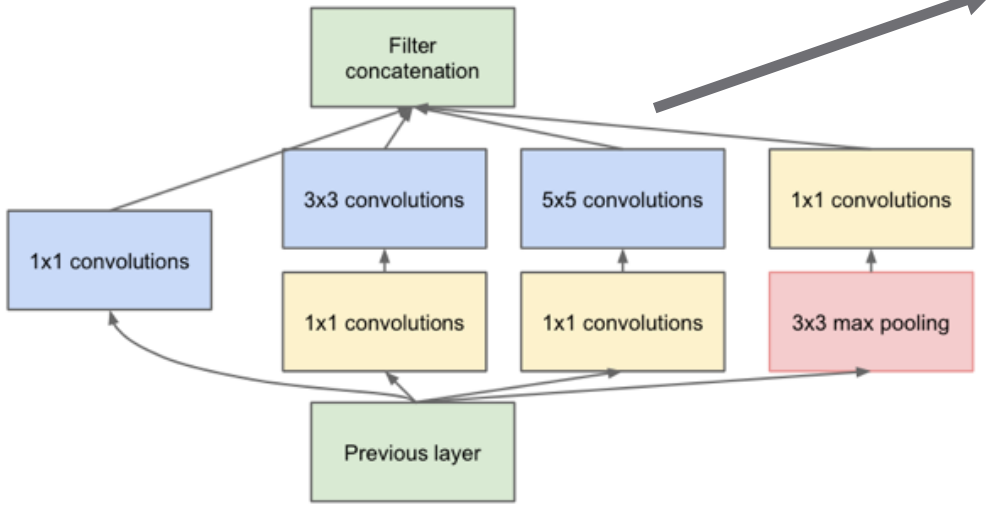
1. Inflating 2D ConvNets into 3D:

- Using a 2D architecture (ImageNet-pretrained Inception-V1) as a base and converting it into 3D ConvNets
- Inflating all the filters and pooling kernels with an additional temporal dimension

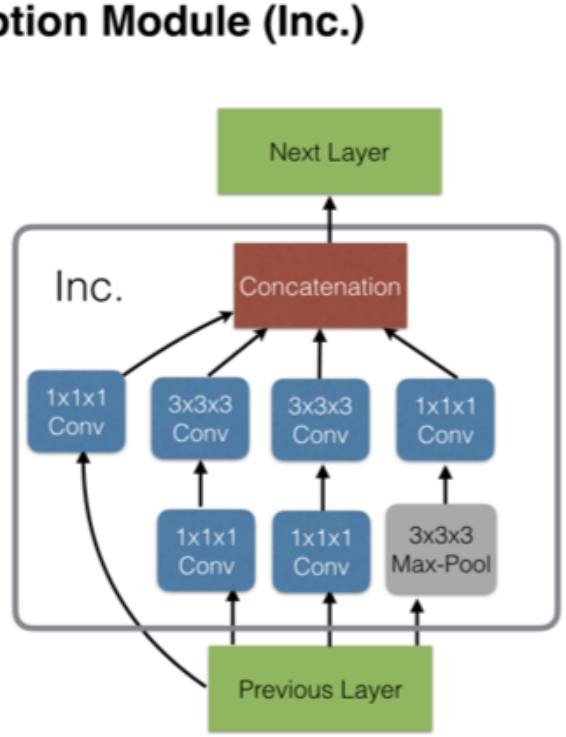
$$N \times N \rightarrow N \times N \times N$$



a) New Model



b) 2D Inception Module



c) Inflated 3D Inception Module

—The new: Two-Stream Inflated 3D ConvNets

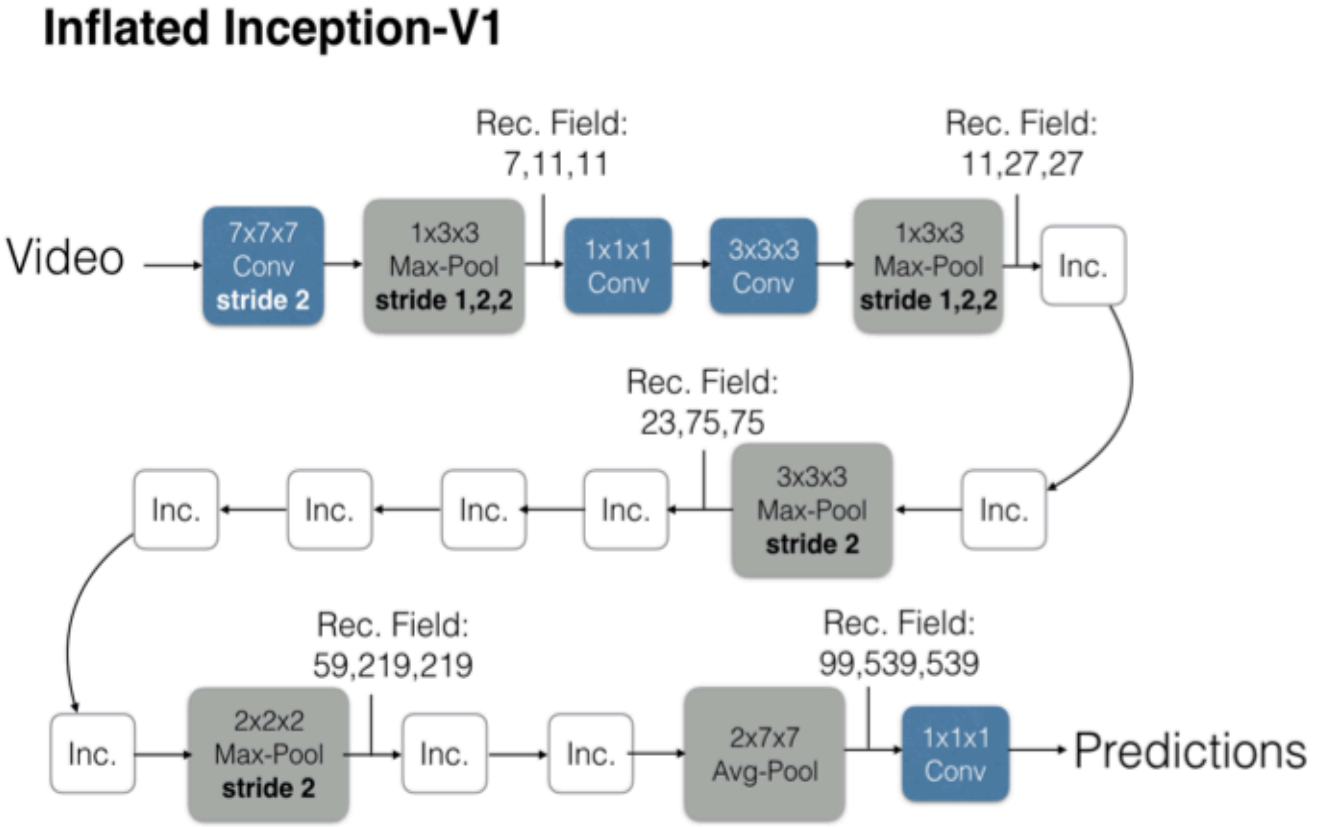
2. Bootstrapping 3D filters from 2D filters:

- Bootstrap **parameters** from the pre-trained ImageNet models
- Converting an image into a (boring)video by copying it repeatedly into a video sequence
- Satisfying the boring-video fixed point: the pooled activations on a boring video should be the same as on the original single-image input
- The outputs of pointwise non-linearity layers and average and max-pooling layers are the same as for the 2D case

—The new: Two-Stream Inflated 3D ConvNets

3. Pacing receptive field growth in space, time and network depth:

- A symmetric receptive field is not necessarily optimal
- Receptive field is dependent on frame rate and image dimensions
- Input videos are processed at 25 fps
- The first two max-pooling layers use 1x3x3 kernels and stride 1 in time
- The final average pooling layer uses a 2x7x7 kernel

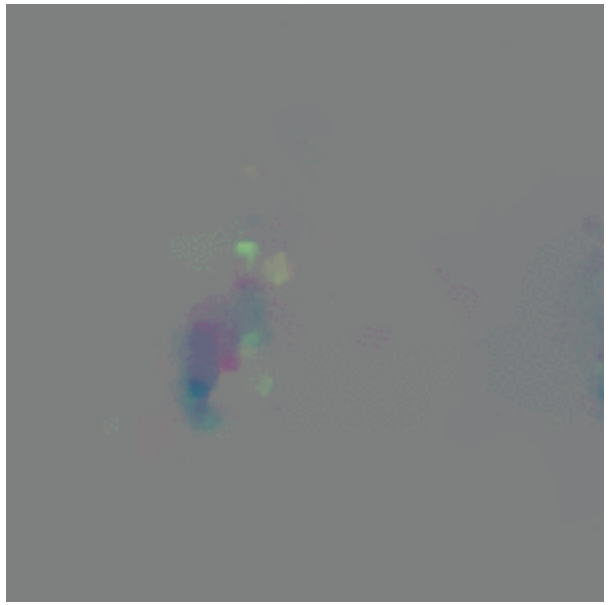


—The new: Two-Stream Inflated 3D ConvNets

4. Two 3D Streams:

- A 3D ConvNet performs pure feedforward computation
- Optical flow algorithms provide recurrence
- One I3D network trained on RGB inputs
- One I3D network trained on flow inputs
- Training separately and averaging their predictions at test time

a) RGB input



b) Optical Flow Input

—The number of parameters and temporal input sizes

- 3D-ConvNet and 3D-Fused models have much more parameters
- Two-Steam I3D use more RGB and flow frames as inputs

Method	#Params	Training		Testing	
		# Input Frames	Temporal Footprint	# Input Frames	Temporal Footprint
ConvNet+LSTM	9M	25 rgb	5s	50 rgb	10s
3D-ConvNet	79M	16 rgb	0.64s	240 rgb	9.6s
Two-Stream	12M	1 rgb, 10 flow	0.4s	25 rgb, 250 flow	10s
3D-Fused	39M	5 rgb, 50 flow	2s	25 rgb, 250 flow	10s
Two-Stream I3D	25M	64 rgb, 64 flow	2.56s	250 rgb, 250 flow	10s

Table 1. Number of parameters and temporal input sizes of the models.

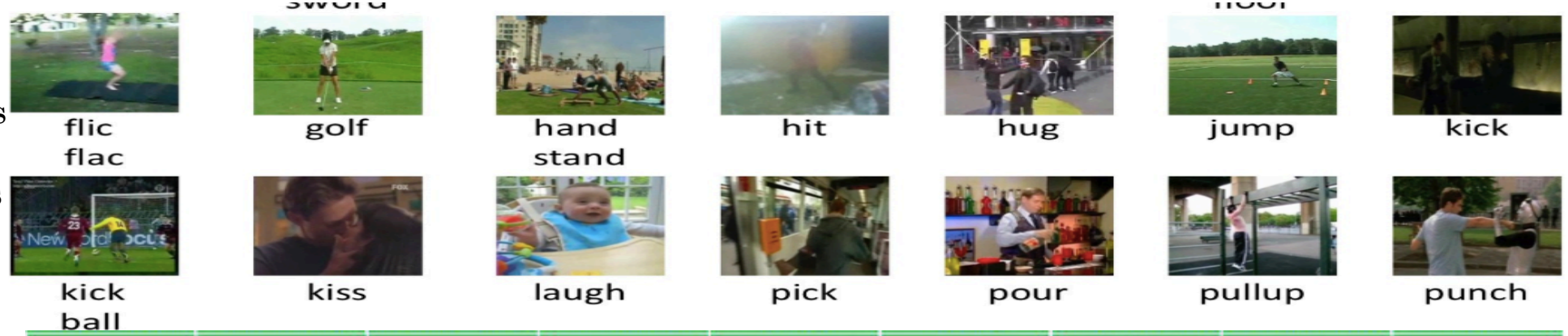
Implementation

—Training

- Base Network(except C3D like model): ImageNet-pretrained Inception-V1
- add a batch normalization and a ReLU activation function after each convolutional layer (except for the last convolutional layer)
- Standard SGD with momentum set to 0.9
- 10x reduction of learning rate when validation loss saturated
- Train on Kinetics for 110K steps
- Train on UCF-101 and HMDB-51 up to 5k steps

—Datasets

HMDB-51:
- YouTube videos
- 51 classes
- about 7k videos



UCF101:
- YouTube Videos
- 101 classes
- About 13k videos

Kinetics Human Action Video Dataset:
- YouTube Videos
- 400 classes
- Each class>400 videos.
- 240k training videos
- 100 testing clips for each class

Human actions:
- Person actions
- Person-person actions
- Person-object actions

(g) riding a bike

—Data Augmentation

During training:

- Randomly cropping both **spatially**

(resizing the smaller vides side to 256 pixels and randomly cropping a 224x224 patch) and **temporally**(picking the starting frame among early frames)

- **Looping** the short video as many times as necessary
- Applying random **left-right flipping** for each video

During test time:

- Taking 224x224 center crops
- The whole video as inputs
- Averaging the predictions

Experiment and Results —Architecture comparison

- Showing the classification accuracy of five architectures
- New I3D models do best with large number of parameters and small datasets
- The performance on Kinetics is far lower than that on UCF-101, but better than that on HMDB-51
- The ranking of the different architectures is consistent
- Two-stream architectures perform well on all datasets

Architecture	UCF-101			HMDB-51			Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	63.3	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	56.1	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	62.2	52.4	65.6
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	–	–	67.2
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**71.1**	**63.4**	**74.2**

—Architecture comparison

They also evaluate the performance training and testing on **Kinetics** with and without **ImageNet** pre-training.

- The imageNet pre-training helps in the performance for all cases
- Two-stream architectures have better performance
- The new model has highest accuracy.

Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	–	–	63.3	–	–
(b) 3D-ConvNet	56.1	–	–	–	–	–
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	–	–	62.7	–	–	67.2
(e) Two-Stream I3D	**68.4** (88.0)	**61.5** (83.4)	**71.6** (90.0)	**71.1** (89.3)	**63.4** (84.9)	**74.2** (91.3)

*Numbers in brackets () are Top-5 accuracy, others are Top-1 accuracy

—Evaluation of Features

They investigate the generalizability of networks trained on Kinetics using two measures:

1. **Freezing** the network **weights** and using it to produce features for the videos of UCF101/HMDB51 dataset, only training on the last layer(Fixed)
2. **Fine-tuning** each network for the UCF101/HMDB51 classes and evaluating on the UCF101/HMDB51 datasets(Full-FT)

The results show:

- Pre-training on Kinetics improves the performance
- For 3D-ConvNet and I3D models, the performance of only training the last layer is better than training directly
- The performance of the two-stream models is good

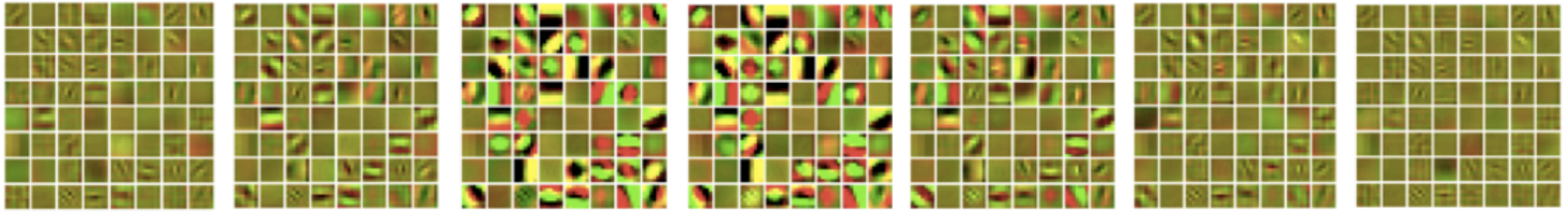
Architecture	UCF-101			HMDB-51		
	Original	Fixed	Full-FT	Original	Fixed	Full-FT
(a) LSTM	81.0 / 54.2	88.1 / 82.6	91.0 / 86.8	36.0 / 18.3	50.8 / 47.1	53.4 / 49.7
(b) 3D-ConvNet	– / 51.6	– / 76.0	– / 79.9	– / 24.3	– / 47.0	– / 49.4
(c) Two-Stream	91.2 / 83.6	93.9 / 93.3	94.2 / 93.8	58.3 / 47.1	66.6 / 65.9	66.6 / 64.3
(d) 3D-Fused	89.3 / 69.5	94.3 / 89.8	94.2 / 91.5	56.8 / 37.3	69.9 / 64.6	71.0 / 66.5
(e) Two-Stream I3D	93.4 / 88.8	97.7 / 97.4	98.0 / 97.6	66.4 / 62.2	79.7 / 78.6	81.2 / 81.3

Original: no pre-training on Kinetics; Fixed: features from kinetics; Full-FT: Kinetics pre-training with end-to-end fine-tuning; with/without ImageNet pre-trained weights

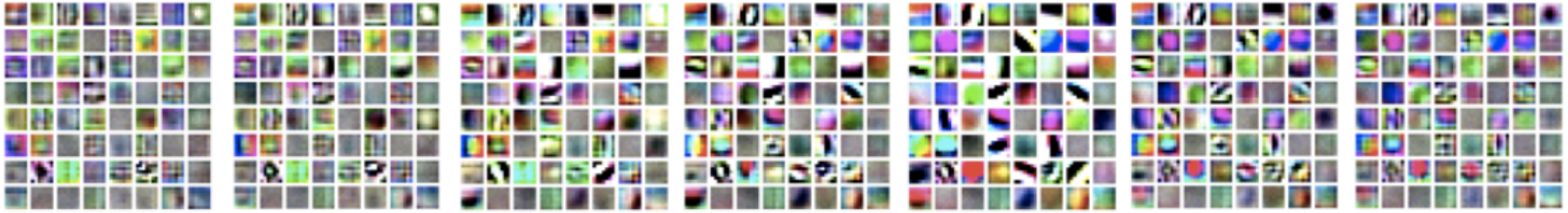
—Conv1 filters of the pre-trained models

- Showing 64 conv1 filters for each I3D ConvNet after training on Kinetics
- I3D filters have rich temporal structure
- The filters of the flow network are closer to the original ImageNet-trained Inception-v1 filters

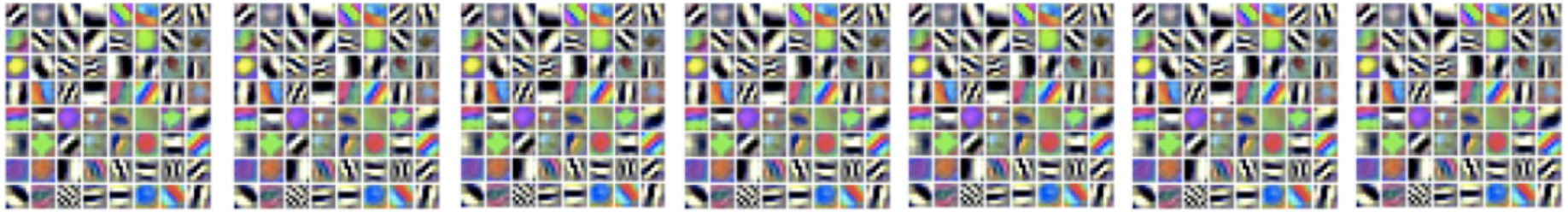
Flow network filter



RGB I3D network filer



Original incepton-v1 filter



—Comparison with the State-of-the-Art

The results show:

- RGB-I3D or RGB-Flow models outperform the current best performance
- Combined two-stream architectures do best
- Difference between Kinetics pre-trained I3D model and prior C3D is larger

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

Discussion

- There is a considerable benefit in pre-training on the large video datasets such as Kinetics.
- Their proposed the new two-stream inflated 3D ConvNets outperform the current methods.
- They did not perform a comprehensive exploration of architectures.

Improvements

- They can apply their Kinetics pre-training to other video tasks, such as semantic video segmentation, video object detection or optical flow computation.
- They can replace the optical flow with the advanced motion vector to increase the efficiency.
- They can add the anchor boxes as many image classification networks do to increase the accuracy.
- They can try to capture long-range temporal structure using sparse temporal sampling strategy on the whole long video.
- They can add a warped optical flow as another input to see the performance.

Reference

- Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset
- Batch normalization: Accelerating deep network training by reducing internal covariate shift
- https://github.com/deepmind/kinetics-i3d
- Two-Stream Convolutional Networks for Action Recognition in Videos
- Learning Spatiotemporal Features with 3D Convolutional Networks
- The Kinetics Human Action Video Dataset