The Last Lecture - COS598B

Vikash Sehwag and Julienne LaChance

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

By: Joao Carreira and Andrew Zisserman

Overview

1. Why focus on transfer learning?
2. Improved architecture for transfer learning.
3. Kinetics- a new dataset.
4. Experimental results.

Transfer Learning

How beneficial is to use pre-trained models?

Imagenet- 1000 images, 1000 categories.

Transfer imagenet pre-trained models to use in segmentation, pose estimation and action classification.

Motivation

Is there an alternate for Imagenet specifically for action classification on video dataset?

Primary characteristic:

- Scale/complexity of dataset
- Performance with transfer learning

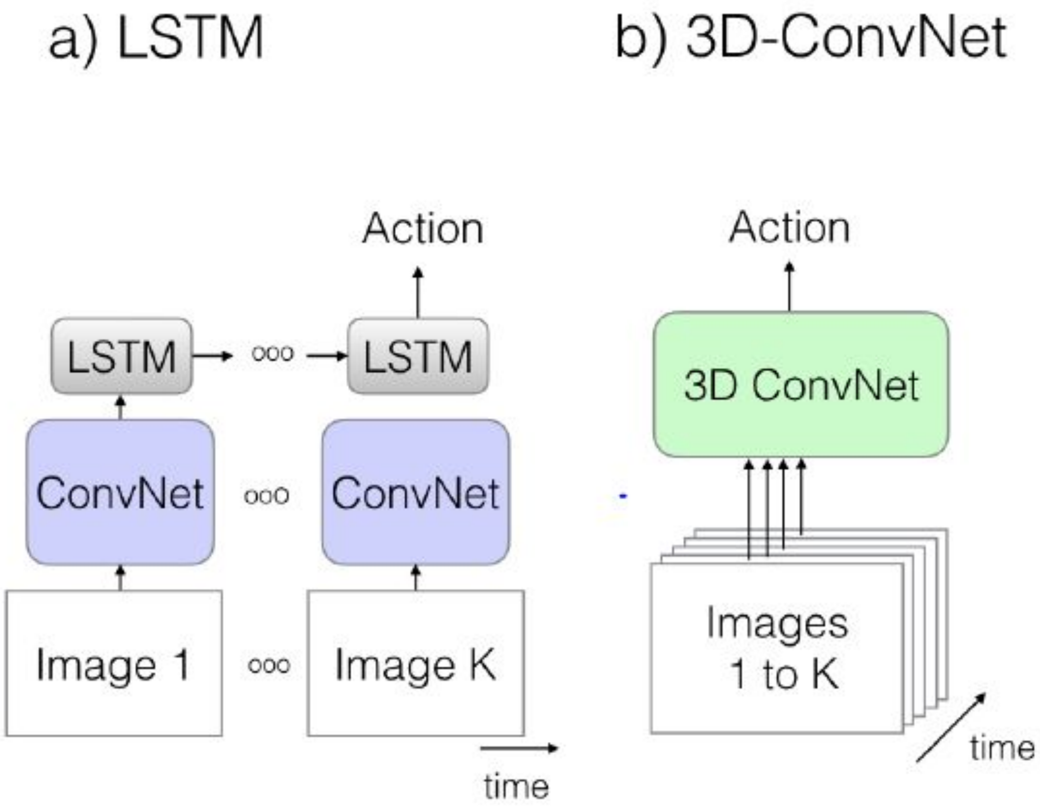
Spoiler alert: it's called Kinetics

Strategy

- Kinetics: 400 human action classes with more than 400 examples for each class
- Pretrain each prev. successful model on Kinetics and fine tune for HMDB-51 and UCF-101.
- Propose Two-Stream Inflated 3D ConvNets (I3D) to further improve the success of pre-training on Kinetics.

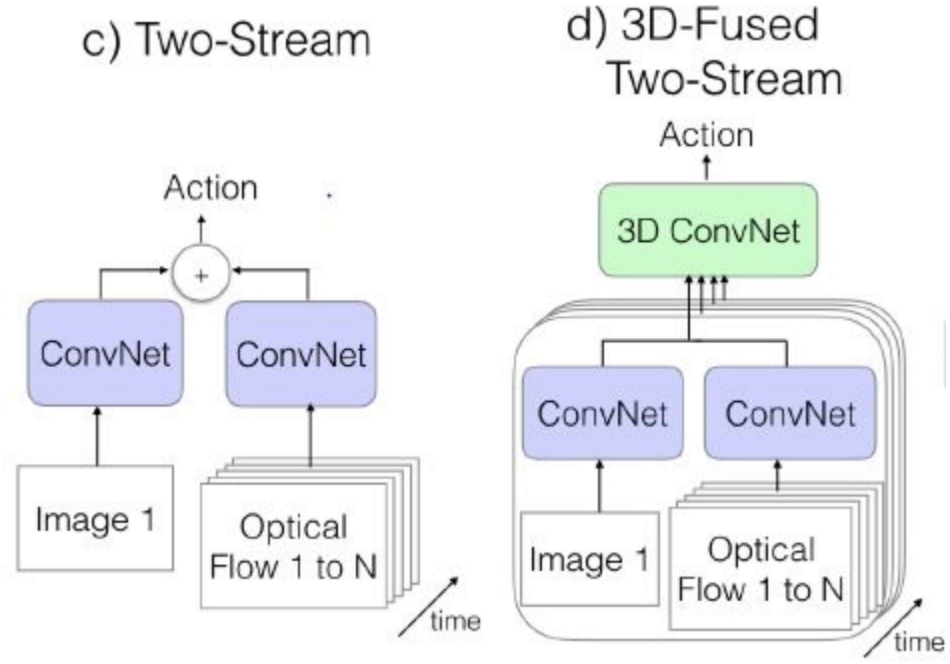
Recap

Prev. models - I



3D ConvNet is trained from scratch

Prev. models - II



Two stream networks.

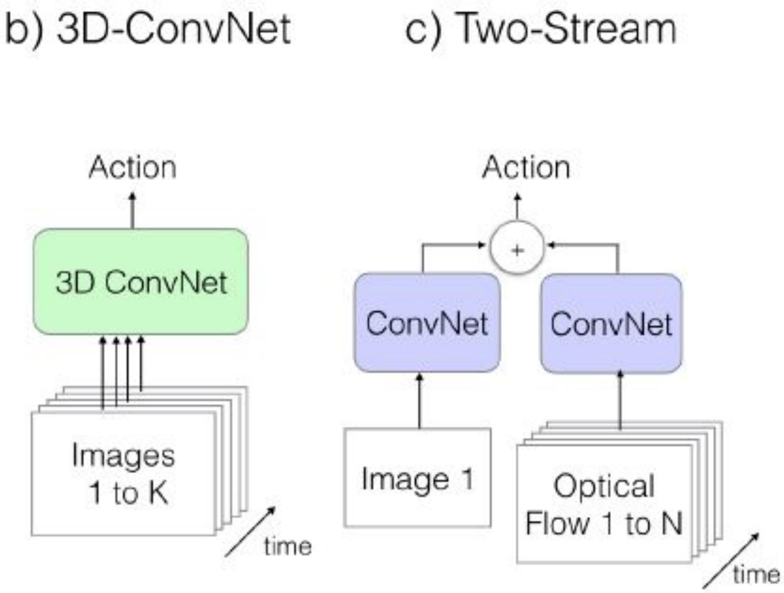
- Introduction
- Recap
- Proposed architecture
- Kinetic datasets
- Experimental results

Proposed architecture

The New: Two Stream Inflated 3D ConvNets

ConvNet+LSTM: difficult to train, only captures high level variation in motion.

3D ConvNets: Training from scratch, thus shallow networks are used.



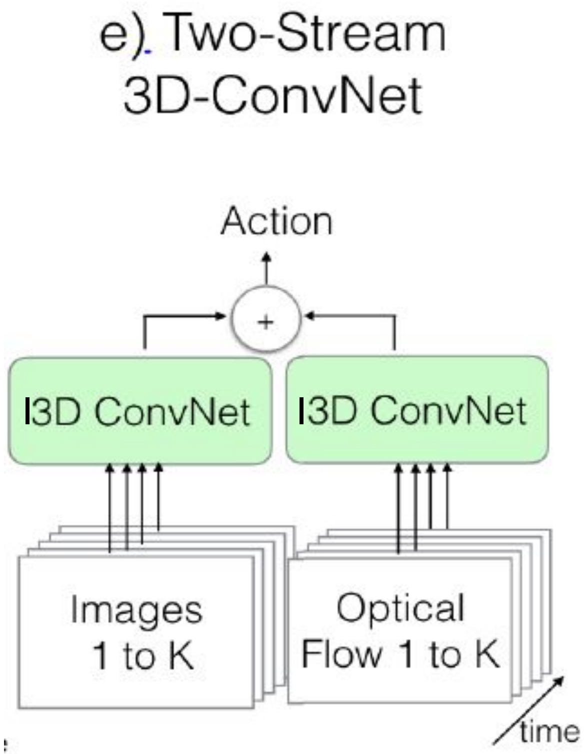
Two Stream Inflated 3D (I3D) ConvNets

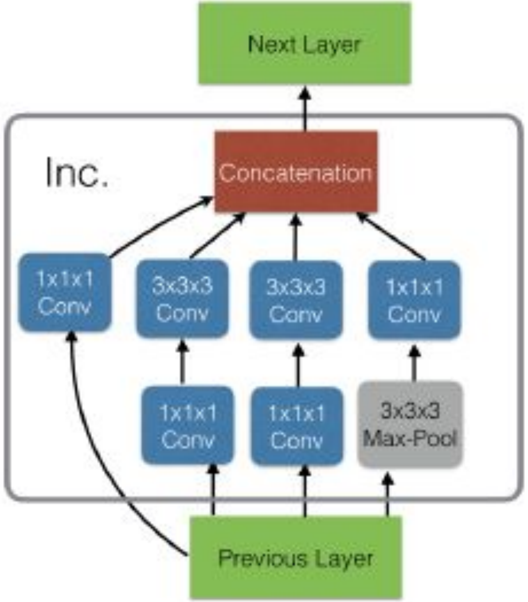
Step-I

- Inflating 2D ConvNets into 3D.
- Bootstrapping 3D filters from 2D Filters.
- Pacing receptive field growth in space, time and network depth.

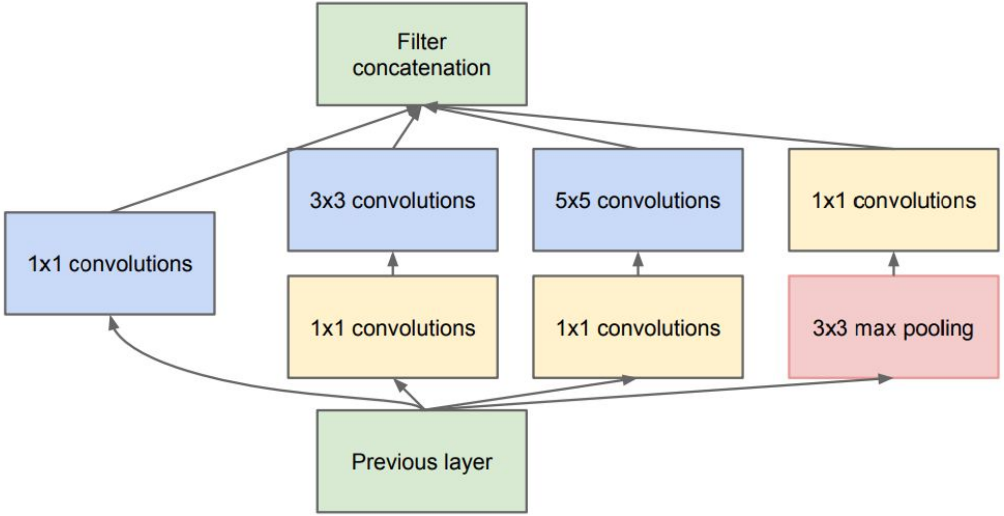
Step-II

Use two streams networks with I3D models instead of 2-D networks.

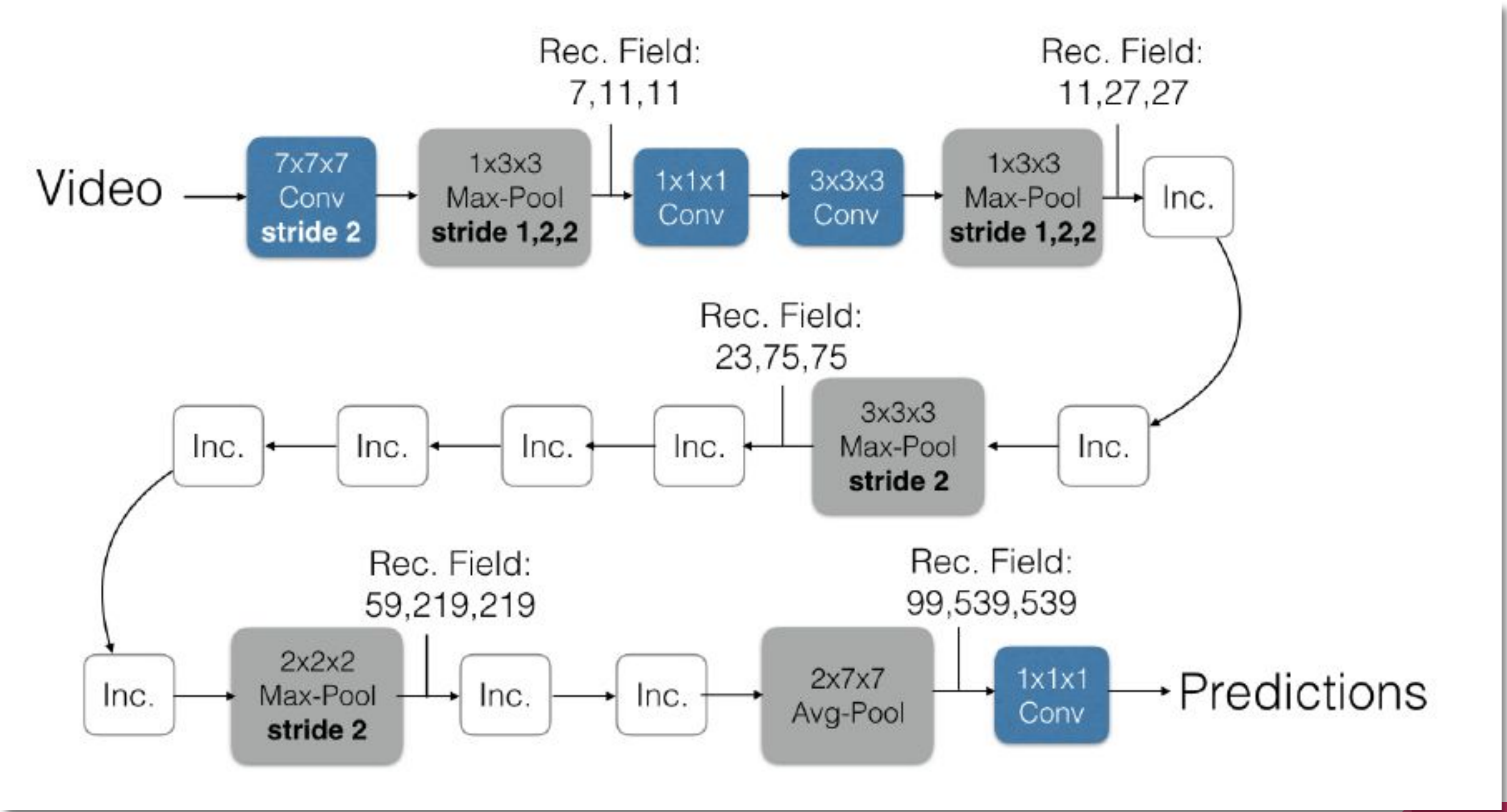




Inflated inception module



Inception module - 2D



Inflated Inception- V1

Comparison

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

Number of parameters and temporal input sizes of the models.

- Introduction
- Recap
- Proposed architecture
- Kinetic datasets
- Experimental results

Kinetics Dataset

Kinetics Dataset

- A large-scale, diverse dataset designed specifically for *human action recognition*. Focus on classification, rather than temporal localization
  - So, short clips: around 10s each
  - Source: YouTube video; each clip taken from a different video (unlike in UCF-101)
- Contains 400 human action classes, with at least 400 video clips per class
  - Designed to cover a broad range of action categories, including human-object interaction and human-human interaction
- Clips labeled by Amazon Mechanical Turk (AMT) workers...

Evaluating Actions in Videos

Can you see a human performing the action

riding mule?

Instructions

We would like to find videos that contain real humans performing actions e.g. scrubbing their face, jumping, kissing someone etc.

Please click on the most appropriate button after watching each video:

- Yes, this contains a true example of the action
- No, this does not contain an example of the action
- You are unsure if there is an example of the action
- Replay the video
- Video does not play, does not contain a human, is an image, cartoon or a computer game.
- We have turned off the audio, you need to judge the clip using the visuals only.

Image courtesy of Kay et al., “The Kinetics Human Action Video Dataset”, 2017

Kinetics Dataset

https://storage.googleapis.com/deepmind-media/_widgets/kinetics/app.html#/labels-all

(a) headbanging

(b) stretching leg

Images courtesy of Kay et al., “The Kinetics Human Action Video Dataset”, 2017

Kinetics Dataset

- Content
  - Person Actions (singular) - drawing, drinking, laughing, pumping fist.
  - Person-Person Actions - hugging, kissing, shaking hands.
  - Person-Object Actions - opening, present, mowing lawn, washing dishes
- Scale

Dataset	Year	Actions	Clips	Total	Videos
HMDB-51 [15]	2011	51	min 102	6,766	3,312
UCF-101 [20]	2012	101	min 101	13,320	2,500
ActivityNet-200 [3]	2015	200	avg 141	28,108	19,994
Kinetics	2017	400	min 400	306,245	306,245

- Introduction
- Recap
- Proposed architecture
- Kinetic datasets
- Experimental results

Experiments

Baseline evaluation

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**

Test set accuracy on UCF-101, HMDB-51 and Kinetics.

Note that the models (except 3D CNN) are pre-trained on Imagenet.

Is Imagenet pre-training helpful?

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)

Performance training and testing on Kinetics with and without ImageNet pretraining.

Imagenet+Kinetics pre-training

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

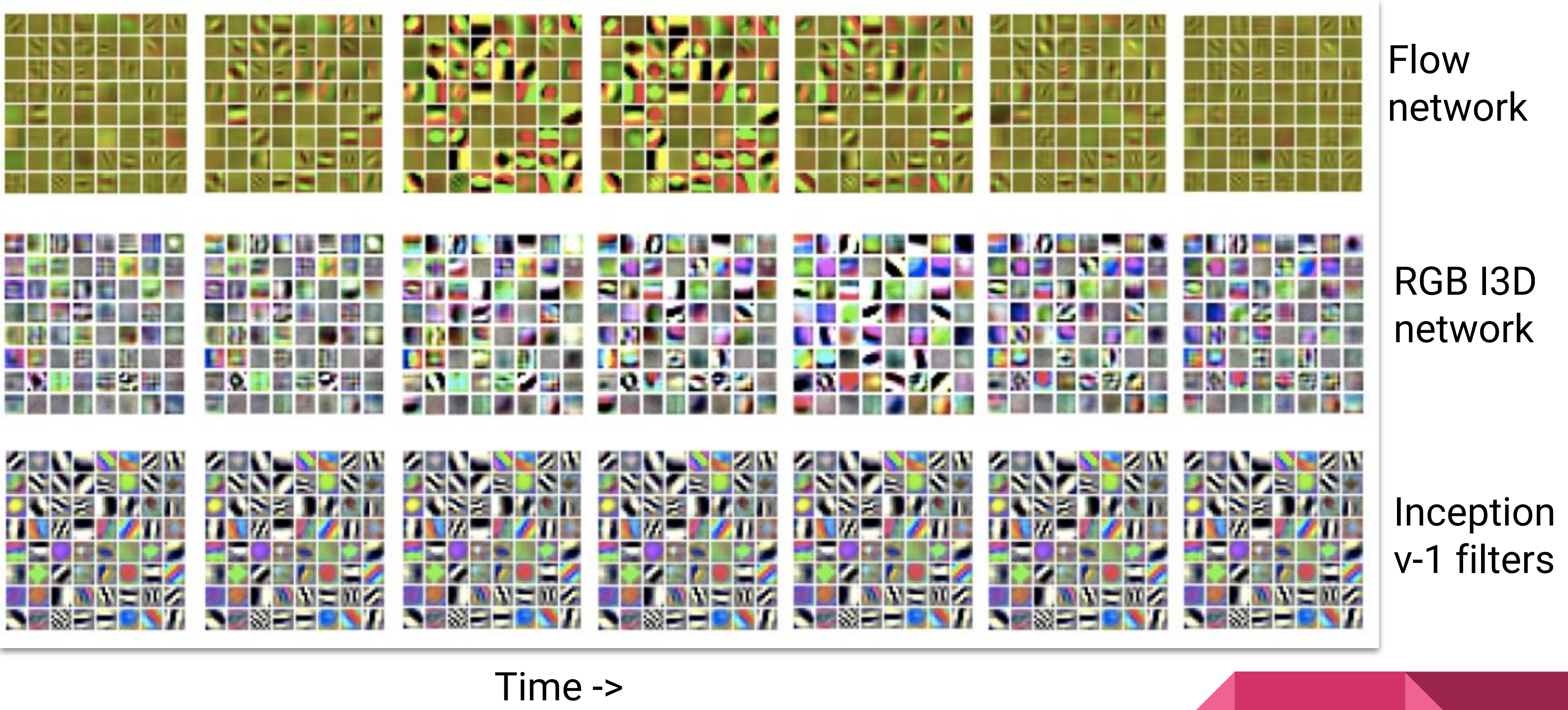
Performance on the UCF-101 and HMDB-51 for architectures starting with / without ImageNet pretrained weights.

The performance gains for two stream I3D networks are significant.

Comparison -IV

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**

Comparison with state-of-the-art on the UCF-101 and HMDB-51 datasets, averaged over three splits.



All 64 conv1 filters of each Inflated 3D ConvNet after training on Kinetics

Conclusion

- Inclusion of innovation in 2-D Convnets architectures.
- Better baseline due to pre-training on Kinetics.

Strategy:

Pre-trained model on Imagenet + inflation of 2-d filters to 3-d + Training on Kinetics (Just keep an eye on space-time relationship).

Still unclear whether these improvement will last across other tasks such as semantic video segmentation, video object detection.

Non-local neural networks

By: Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He.
April 2018.

Non-local NNs: Overview

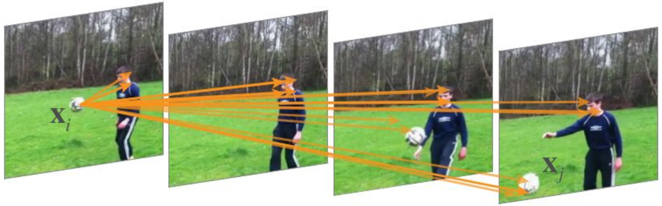
- What are non-local neural networks?
- Intuition and background
- Video classification: brief overview of datasets
- Non-local operations
- Neural network implementation
- Experiments
- Conclusions

What are non-local neural networks?

Capturing *long-range dependencies* is essential when it comes to neural networks:

- *Recurrent* operations for sequential data (e.g. speech)
- *Convolutional* operations for image data

But in each of these, regions are processed locally (either in space or in time), and the operations must be applied repeatedly to capture long-range dependencies.



Non-local operations: a simpler, more efficient way of capturing these dependencies.

- Competes with or outperforms current SOTA methods on video classification tasks (Kinetics, Charades) + COCO

Image courtesy of Wang et al.

Intuition and background

Non-local means (Buades and Morel, 2005)

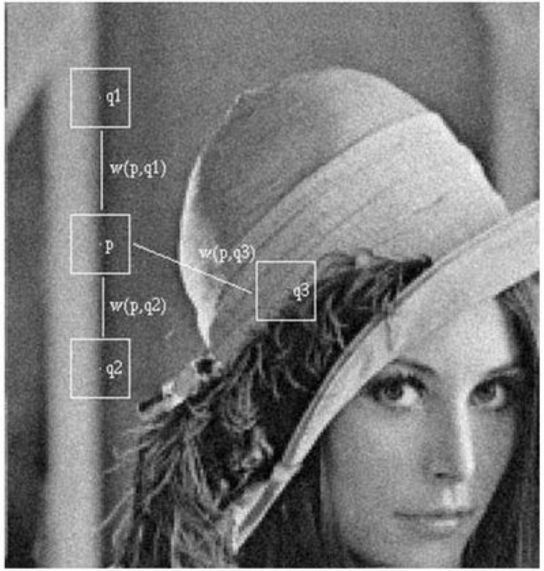
- A classical algorithm for doing image de-noising.
- *"Local mean"* filters take the mean value of pixel values in a local neighborhood of the target pixel for smoothing the image.
- *"Non-local means"* filtering involves taking a mean of all the pixels in the image, weighted by how similar they are to the target pixel.

← Result: less loss of detail in filtered image.

Image source: Ryosuke Ueda, Hiroyuki Kudo, Jian Dong, "Applications of compressed sensing image reconstruction to sparse view phase tomography," Proc. SPIE 10391, Developments in X-Ray Tomography XI, 103910H (3 October 2017)

How does non-local means work?

Weights are high for q1 and q2, but not for q3.



- Consider the area, Ω, of the image and two points, *p* and *q*. The filtered value *u(p)* of the image at pixel *p* is then:

$$u(p) = \frac{1}{C(p)} \int_{\Omega} v(q) f(p,q) dq.$$

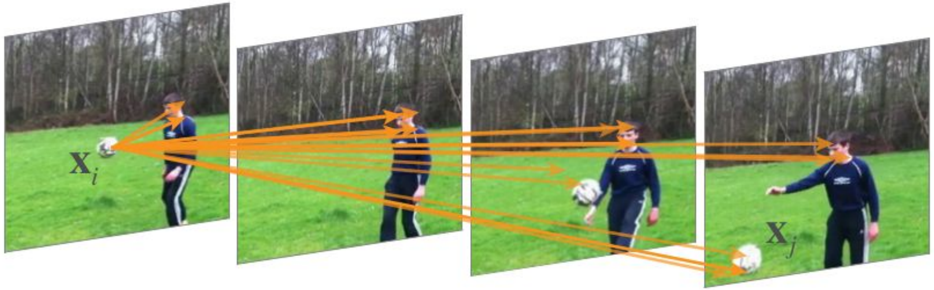
- Here, *v(q)* is the unfiltered pixel value at point *q*, and *f(p,q)* is the weighting function, which is often a Gaussian. The integral is taken over all *q*.
- *C(p)* is a normalizing factor:

$$C(p) = \int_{\Omega} f(p,q) dq.$$

Image and formulas courtesy of Buades and Morel, 2005.

Generalizing non-local means

- A non-local operation, in general, “computes the response at a position as a weighted sum of the features at all positions in the input feature maps”.
  - “Set of positions” applies in space (images), time (sequences), and spacetime (videos).



A “spacetime” non-local operation example is displayed here.
The response at position *xi* is computed using a weighted average of features at all positions *xj*. The highest weighted ones are shown.

(Application: Video classification from the Kinetics data set.)

Image courtesy of Wang et al.

Video classification:
a quick overview of the datasets

Kinetics Dataset

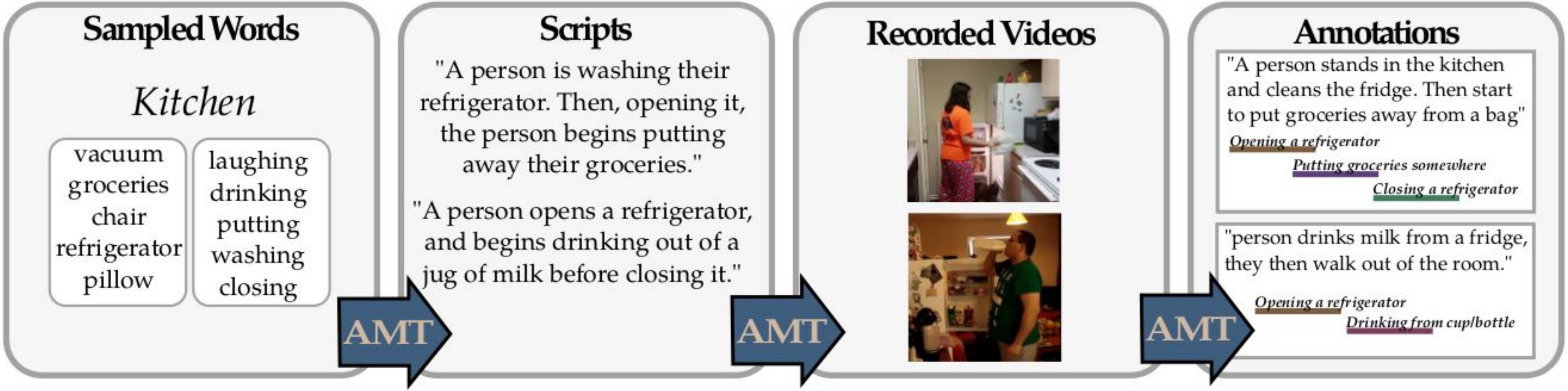
- Discussed by Vikash!

Charades Dataset

- Objective: to gather a ton of videos representing “boring” activities in daily life (rather than niche activities like sports)
- *Charades* dataset composed by actors who record themselves in their own homes, acting out casual everyday activities
    - Videos around 30s each
    - 9,848 annotated videos with 267 actors, 157 actions classes and 46 object classes
- AMT workers generate scripts, act out scenes, and perform annotation verification

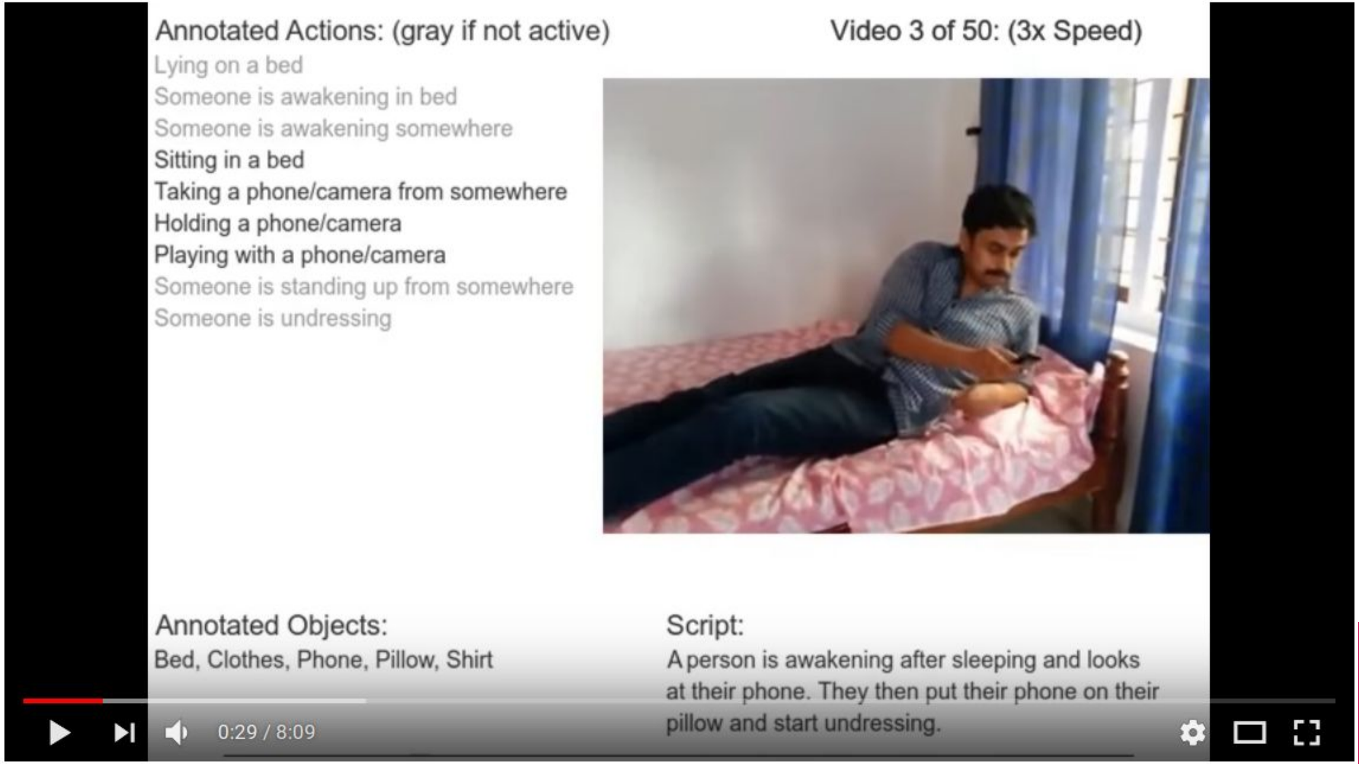
Images courtesy of Sigurdsson et al., “Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding”, 2016

Charades Dataset: AMT Workflow



Images courtesy of Sigurdsson et al., "Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding", 2016

Charades Dataset



https://youtu.be/x9AhZLDkbyc?t=35

Non-local operations

Non-local operations: Formulation

- Recall the classical non-local mean operation (discrete version shown here):

$$u(p) = \frac{1}{C(p)} \sum_{q \in \Omega} v(q) f(p, q) \qquad \text{where} \quad C(p) = \sum_{q \in \Omega} f(p, q)$$

- A non-local operation in a deep neural network is defined by:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j)$$

Here, *x* is the input signal (image, text, video), *y* is the output signal, *i* is the position of interest, and *j* enumerates all possible positions.

Non-local operations: Formulation

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j)$$

- The pairwise function *f* computes a scalar (affinity measure), while *g* is the input signal at position *j*.
- Non-local because we're summing over *j*. Compare to:
  - Convolutional operation: sums up the weighted input in a local neighborhood
    $(i - 1 \leq j \leq i + 1)$
  - Recurrent operation: based on current or previous time steps
    $j = i$ or $j = i - 1$
  - Fully connected layers: relationships between positions are based on learned weights. So the relationship is not a function of the input.
- This formulation also supports inputs of varying size.

Some Instantiations (choosing *g* and *f* functions)

- Only one choice of *g* was considered; that of a linear embedding, where W_g is learned:

$$g(\mathbf{x}_j) = W_g \mathbf{x}_j$$

  - Can be implemented as a 1x1 convolution in space, or a 1x1x1 convolution in spacetime.

- Many more choices for the affinity function *f*; but experiments demonstrate that the models are not very sensitive to the choice of function.

Some Instantiations of *f*

- Gaussian: $f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^T \mathbf{x}_j}$
- Embedded Gaussian: $f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}$
    - Similar to Gaussian, except embedding parameters are learned:

$$\theta(\mathbf{x}_i) = W_\theta \mathbf{x}_i \text{ and } \phi(\mathbf{x}_j) = W_\phi \mathbf{x}_j$$

- Dot product: $f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$
- Concatenation: $f(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{ReLU}(\mathbf{w}_f^T[\theta(\mathbf{x}_i), \ \phi(\mathbf{x}_j)])$
    - Here, [□ , □] denotes concatenation. This formulation is used in Relation Networks:

      A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap.
      A simple neural network module for relational reasoning.
      In Neural Information Processing Systems (NIPS), 2017.

Neural network implementation

Non-local neural networks

- To implement the non-local operations, we wrap them in a “non-local block”:

$$\boxed{\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j)} \Rightarrow \boxed{\mathbf{z}_i = W_z \mathbf{y}_i + \mathbf{x}_i}$$

Non-local operation

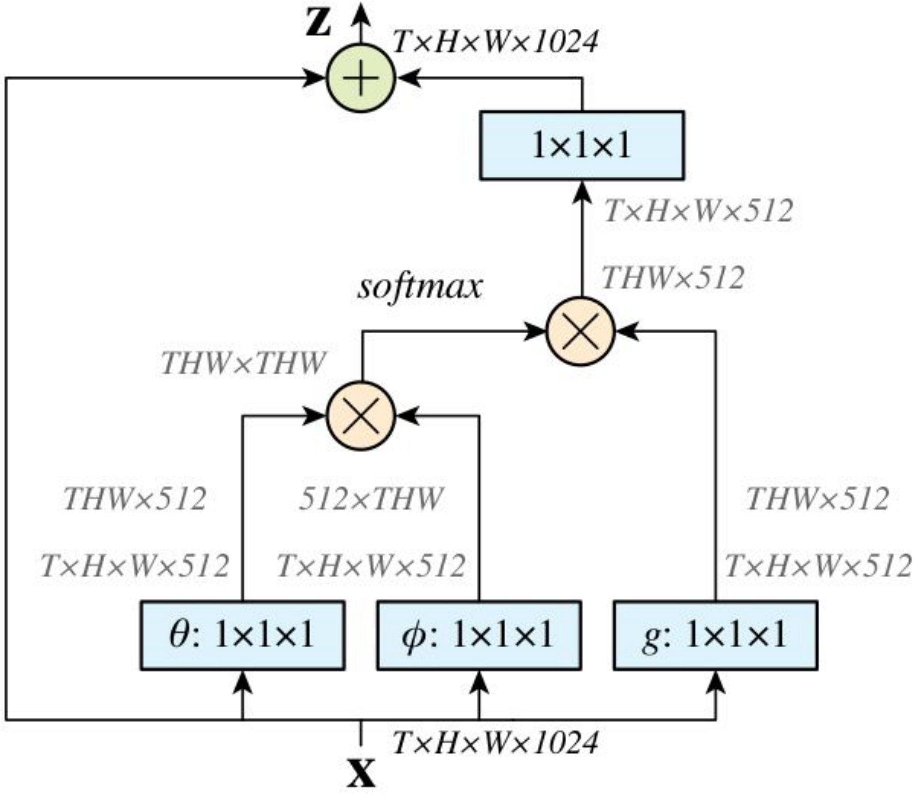
Non-local block

This denotes a residual connection

- The residual connection allows us to insert the non-local block into pre-trained networks: it won’t fail even if all weights are zero.

An example non-local block

$$\mathbf{y} = softmax(\mathbf{x}^T W_\theta^T W_\phi \mathbf{x}) g(\mathbf{x})$$



Other Implementation Details

- Number of channel in embeddings: bottlenecks to reduce weights to learn
- Subsampling trick (pooling) in the spatial domain to reduce computation:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\hat{\mathbf{x}})} \sum_{\forall j} f(\mathbf{x}_i, \hat{\mathbf{x}}_j) g(\hat{\mathbf{x}}_j)$$

$\hat{\mathbf{x}}$ is the subsampled version of $\mathbf{x}$

  - Makes the computation sparser. Implemented using a max pooling layer.

Video Classification Models

- 2D ConvNet baseline test: →
  - Temporal dimension trivially addressed (pooling layers only)
  - Implemented to isolate temporal effects in non-local block extensions
- (Inflated) 3D ConvNet baseline test:
  - Extend the 2D network by inflating kernels to handle the time dimension:
    E.g., a 2D *k* x *k* kernel becomes a *t* x *k* x *k* kernel.
- Non-local neural networks
  - Inserting (1, 5, or 10) non-local blocks into the above networks and comparing performance.

layer		output size
conv_1	7×7, 64, stride 2, 2, 2	$16\times112\times112$
pool_1	$3\times3\times3$ max, stride 2, 2, 2	$8\times56\times56$
res_2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$8\times56\times56$
pool_2	$3\times1\times1$ max, stride 2, 1, 1	$4\times56\times56$
res_3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$4\times28\times28$
res_4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$4\times14\times14$
res_5	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$4\times7\times7$
global average pool, fc		$1\times1\times1$

Experiments on Kinetics

One NL block added to C2D

- One block is added after the res-4 layer of 2D ConvNet baseline
- Not much difference between *f* instantiations

model, R50	top-1	top-5
C2D baseline	71.8	89.7
Gaussian	72.5	90.2
Gaussian, embed	72.7	**90.5**
dot-product	**72.9**	90.3
concatenation	72.8	**90.5**

→

	layer		output size
conv_1	7×7, 64, stride 2, 2, 2		$16\times112\times112$
pool_1	$3\times3\times3$ max, stride 2, 2, 2		$8\times56\times56$
res_2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}$	$\times3$	$8\times56\times56$
pool_2	$3\times1\times1$ max, stride 2, 1, 1		$4\times56\times56$
res_3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}$	$\times4$	$4\times28\times28$
res_4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}$	$\times6$	$4\times14\times14$
res_5	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}$	$\times3$	$4\times7\times7$
global average pool, fc			$1\times1\times1$

When to add NL blocks?

- res5 might be too small to support much spatial information

model, R50	top-1	top-5
baseline	71.8	89.7
res_2	72.7	90.3
res_3	**72.9**	90.4
res_4	72.7	**90.5**
res_5	72.3	90.1

	layer	output size
$conv_1$	7×7, 64, stride 2, 2, 2	$16\times112\times112$
$pool_1$	$3\times3\times3$ max, stride 2, 2, 2	$8\times56\times56$
res_2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$8\times56\times56$
$pool_2$	$3\times1\times1$ max, stride 2, 1, 1	$4\times56\times56$
res_3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$4\times28\times28$
res_4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$4\times14\times14$
res_5	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$4\times7\times7$
global average pool, fc		$1\times1\times1$

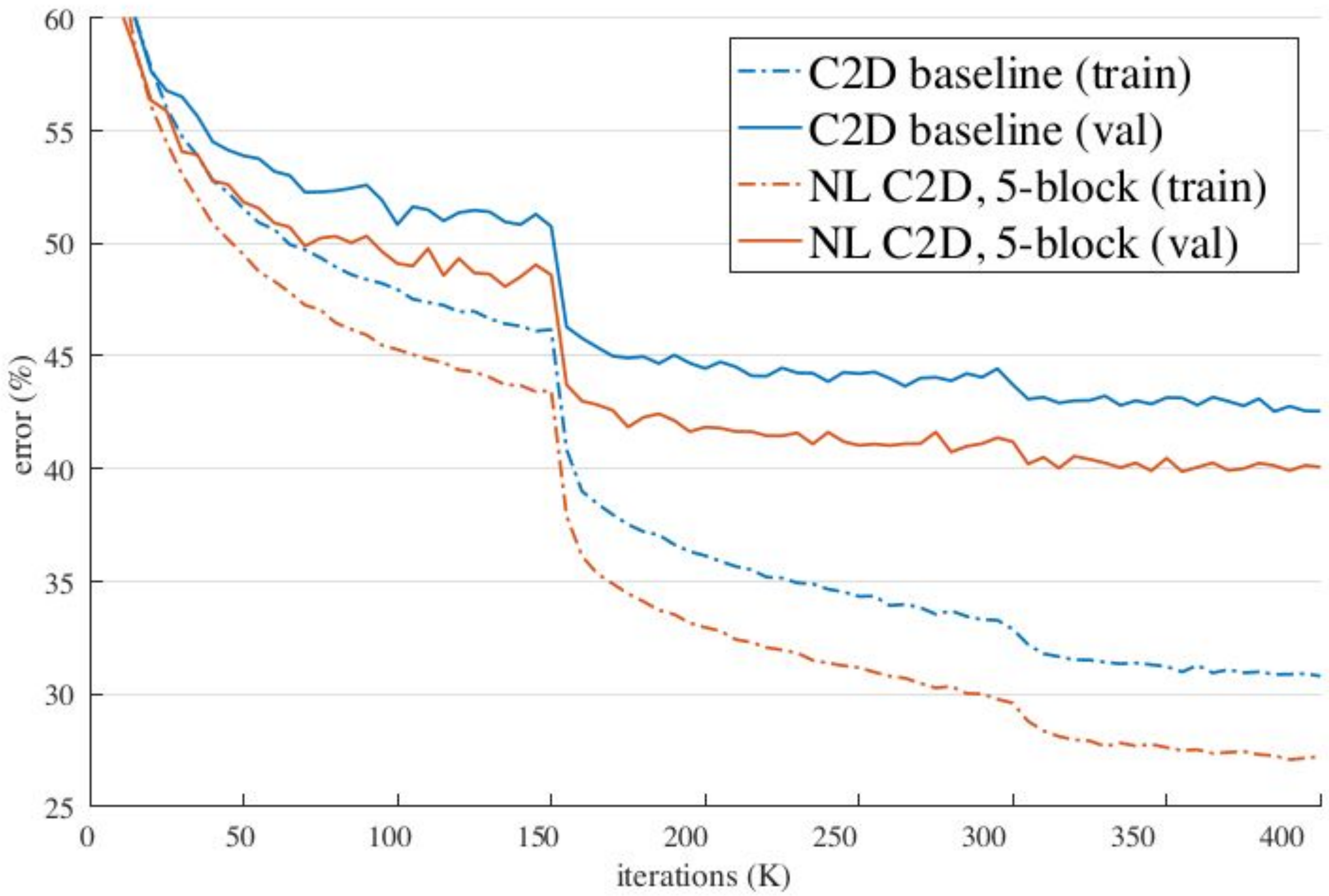
Going deeper with NL blocks

- Add 1 block (to res 4), 5 blocks (3 to res 4 and 2 to res 3 , to every other residual block), and 10 blocks (to every residual block in res 3 and res 4)

model		top-1	top-5
R50	baseline	71.8	89.7
	1-block	72.7	90.5
	5-block	73.8	91.0
	10-block	**74.3**	**91.2**
R101	baseline	73.1	91.0
	1-block	74.3	91.3
	5-block	**75.1**	**91.7**
	10-block	**75.1**	91.6

- More blocks is generally better, but, not just because of depth:

  5-block ResNet-50 has about 70% parameters and 80% FLOPS of baseline ResNet-101, and is shallower.


C2D baseline (train)
C2D baseline (val)
NL C2D, 5-block (train)
NL C2D, 5-block (val)
error (%)
iterations (K)
60
55
50
45
40
35
30
25
0
50
100
150
200
250
300
350
400

Space, Time, or Spacetime?

- Comparing various types of NL blocks shows that spacetime performs better.

model		top-1	top-5
R50	baseline	71.8	89.7
	space-only	72.9	90.8
	time-only	73.1	90.5
	spacetime	**73.8**	**91.0**
R101	baseline	73.1	91.0
	space-only	74.4	91.3
	time-only	74.4	90.5
	spacetime	**75.1**	**91.7**

(Inflated) 3D ConvNet vs. NL

- NL blocks can be more effective than 3D convolutions when used alone:

model, R101	params	FLOPs	top-1	top-5
C2D baseline	1×	1×	73.1	91.0
I3D$_{3\times3\times3}$	1.5×	1.8×	74.1	91.2
I3D$_{3\times1\times1}$	**1.2**×	1.5×	74.4	91.1
NL C2D, 5-block	**1.2**×	**1.2**×	**75.1**	**91.7**

Non-local 3D ConvNet extension

- NL blocks and 3D convolutions are complementary:

	model	top-1	top-5
R50	C2D baseline	71.8	89.7
	I3D	73.3	90.7
	NL I3D	**74.9**	**91.6**
R101	C2D baseline	73.1	91.0
	I3D	74.4	91.1
	NL I3D	**76.0**	**92.1**

Longer video clips

- Using 128-frame clips versus 32-frame clips. Models have better results on longer sequences.

	model	top-1	top-5
R50	C2D baseline	71.8	89.7
	I3D	73.3	90.7
	NL I3D	**74.9**	**91.6**
R101	C2D baseline	73.1	91.0
	I3D	74.4	91.1
	NL I3D	**76.0**	**92.1**

32-frame clips

	model	top-1	top-5
R50	C2D baseline	73.8	91.2
	I3D	74.9	91.7
	NL I3D	**76.5**	**92.6**
R101	C2D baseline	75.3	91.8
	I3D	76.4	92.7
	NL I3D	**77.7**	**93.3**

128-frame clips

Comparison with other methods

model	backbone	modality	top-1 val	top-5 val	top-1 test	top-5 test	avg test†
I3D in [7]	Inception	RGB	72.1	90.3	71.1	89.3	80.2
2-Stream I3D in [7]	Inception	RGB + flow	75.7	92.0	74.2	91.3	82.8
RGB baseline in [3]	Inception-ResNet-v2	RGB	73.0	90.9	-	-	-
3-stream late fusion [3]	Inception-ResNet-v2	RGB + flow + audio	74.9	91.6	-	-	-
3-stream LSTM [3]	Inception-ResNet-v2	RGB + flow + audio	77.1	93.2	-	-	-
3-stream SATT [3]	Inception-ResNet-v2	RGB + flow + audio	77.7	93.2	-	-	-
NL I3D [ours]	ResNet-50	RGB	76.5	92.6	-	-	-
	ResNet-101	RGB	**77.7**	**93.3**	-	-	**83.8**

Experiments on Charades

Comparison to other methods: Charades

model	modality	*train/val*	*trainval/test*
2-Stream [43]	RGB + flow	18.6	-
2-Stream +LSTM [43]	RGB + flow	17.8	-
Asyn-TF [43]	RGB + flow	22.4	-
I3D [7]	RGB	32.9	34.4
I3D [ours]	RGB	35.5	37.2
NL I3D [ours]	RGB	**37.5**	**39.5**

- ResNet-101 with 5 NL blocks used.

Experiments on COCO

Comparison to other methods: COCO

method		AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
R50	baseline	38.0	59.6	41.0	34.6	56.4	36.5
	+1 NL	**39.0**	**61.1**	**41.9**	**35.5**	**58.0**	**37.4**
R101	baseline	39.5	61.4	42.9	36.0	58.1	38.3
	+1 NL	**40.8**	**63.1**	**44.5**	**37.1**	**59.9**	**39.2**
X152	baseline	44.1	66.4	48.4	39.7	63.2	42.2
	+1 NL	**45.0**	**67.8**	**48.9**	**40.3**	**64.4**	**42.8**

- COCO **object detection** and **instance segmentation**: Augmenting Mask R-CNN with one NL block.
- Results suggest non-local info not sufficiently captured
- X152 = ResNeXt-152 architecture

Comparison to other methods: COCO

model	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
R101 baseline	65.1	86.8	70.4
NL, +4 in head	66.0	87.1	71.7
NL, +4 in head, +1 in backbone	**66.5**	**87.3**	**72.8**

- COCO human pose estimation (**keypoint detection**): Insert 4 NL blocks after every 2 convolutional layers.
- Stronger localization performance.

Conclusions

- Non-local blocks are a generic family of building blocks for capturing long-range dependencies
- The response at a position is computed as a weighted sum of features at all positions

- The specific affinity function (*f*) doesn't seem to matter much
- NL blocks can be added anywhere into existing architectures
- More blocks is generally better
- NL blocks more effective than 3D convolutions when used alone; but together they can be complementary.
    - Future work: figure out how to use NL blocks in conjunction with with other network blocks
- Models work well on longer video clips

Questions?

Backup Slides

Non-local means Gaussian function example

$$f(p,q) = e^{-\frac{|B(q)-B(p)|^2}{h^2}}$$

Where *h* is the filtering parameter (standard deviation) and *B(p)* is the local mean value of the image point values surrounding *p*. This establishes a normal distribution with mean *B(p)*.