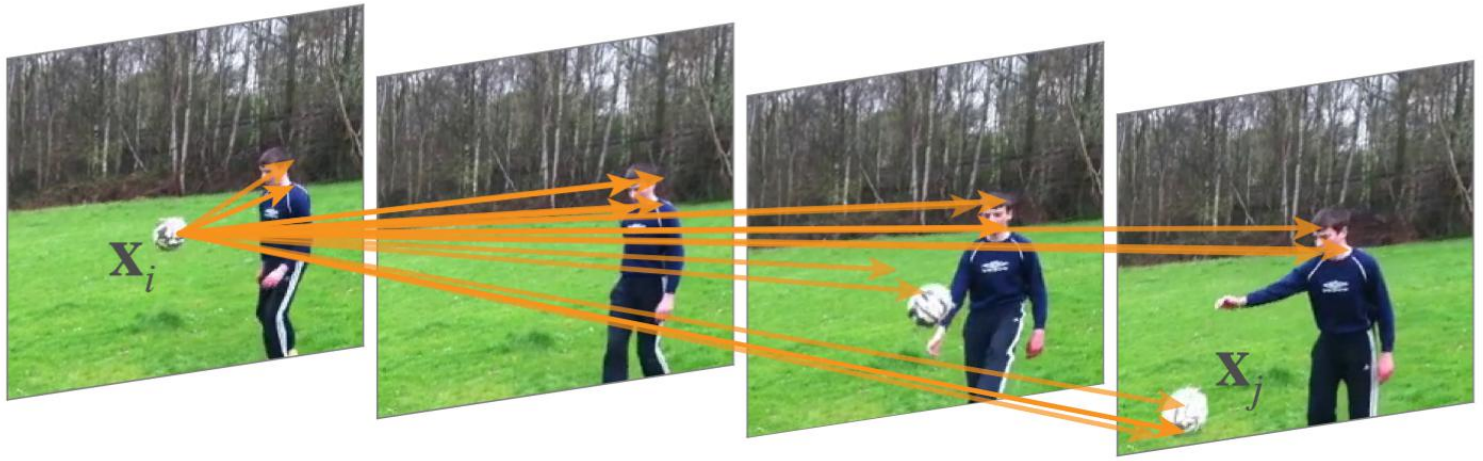
Non-local Neural Networks

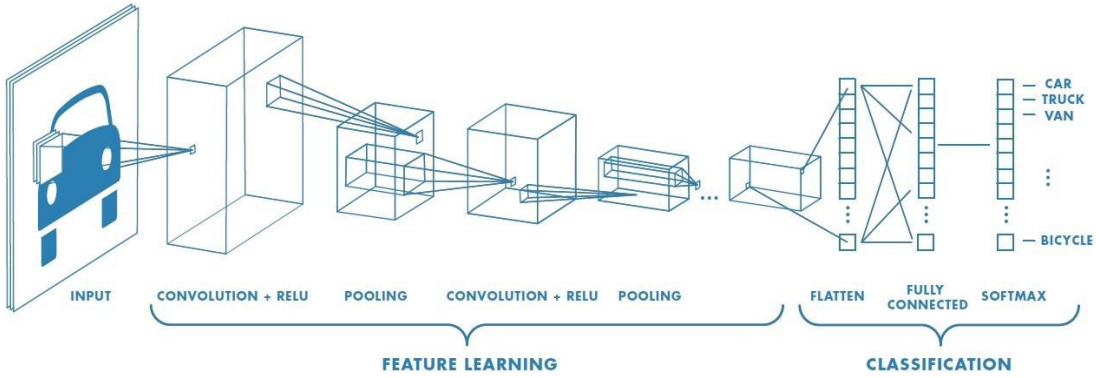
Xiaolong Wang
Ross Girshick
Abhinav Gupta
Kaiming He

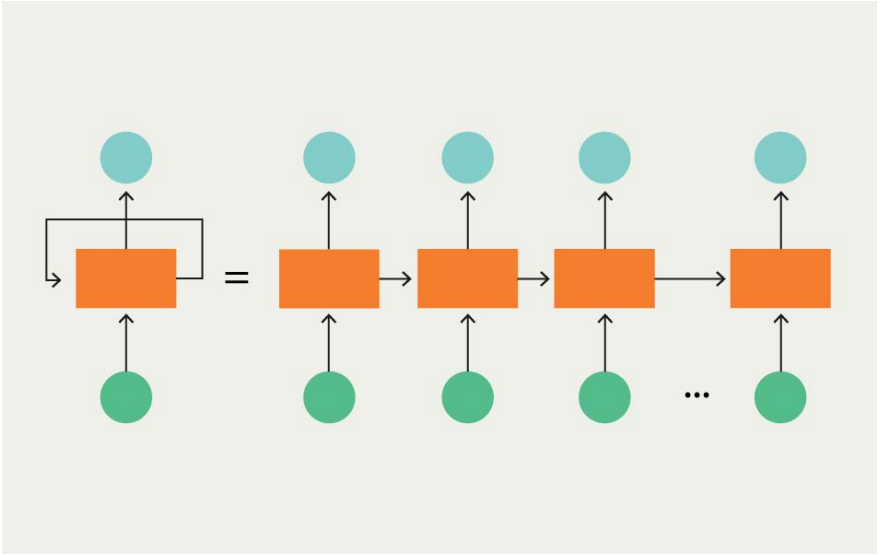


Wang, X. et al. "Non-local Neural Networks." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 7794-7803.

Current Approach

- In image processing, CNNs process a local neighborhood in convolutional layers
- In sequential data, recurrent operations are also applied to local data
- Signals are then propagated through the network by repeated application of these operations
- However, repeated application can be
  - Computationally expensive
  - Causes optimization difficulties
  - Make multi-hop dependency modeling difficult





The Concept

- Proposal: Non-local operations
  - Efficient, simple, and generic operation for capturing long-range dependencies in deep networks
  - Computes the response at a position as a weights sum of the features at all positions in the input feature maps
  - ‘Positions’ can be in time or space

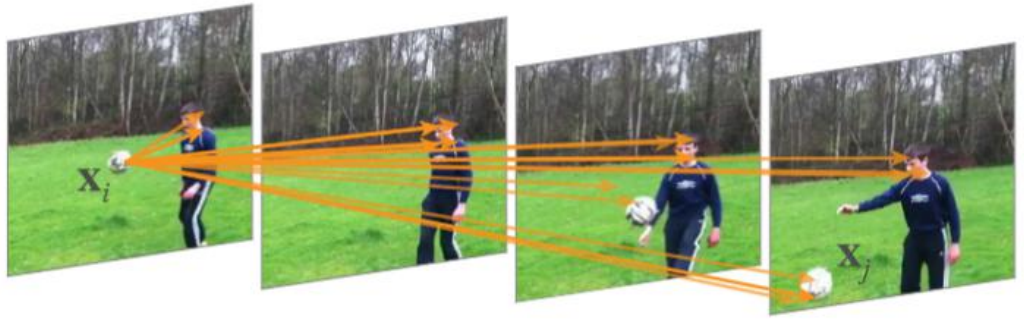


Figure 1. A spacetime ***non-local*** operation in our network trained for video classification in Kinetics. A position $\mathbf{x}_i$'s response is computed by the weighted average of the features of *all* positions $\mathbf{x}_j$ (only the highest weighted ones are shown here). In this example computed by our model, note how it relates the ball in the first frame to the ball in the last two frames. More examples are in Figure 3.

Advantages

- While convolutional and recurrent operations progress information through the network, non-local operations capture long-range dependencies directly – they calculate interactions between two positions regardless of distance
- Non-local operations are more efficient than multiple convolutions or recurrent operations
- Non-local operations maintain variable input sizes, and can be combined with other operations

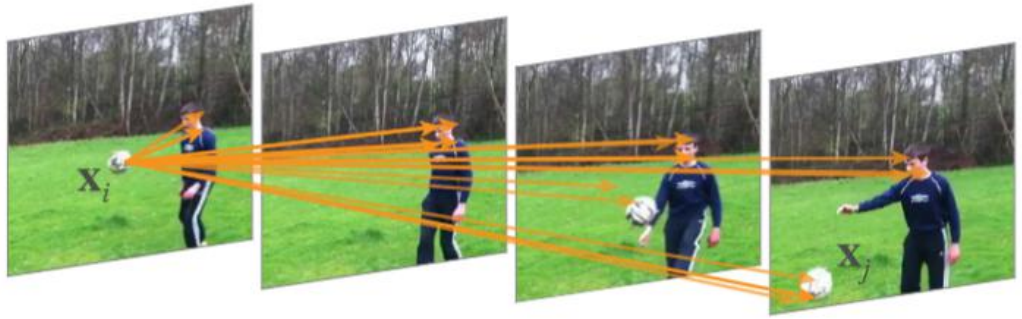
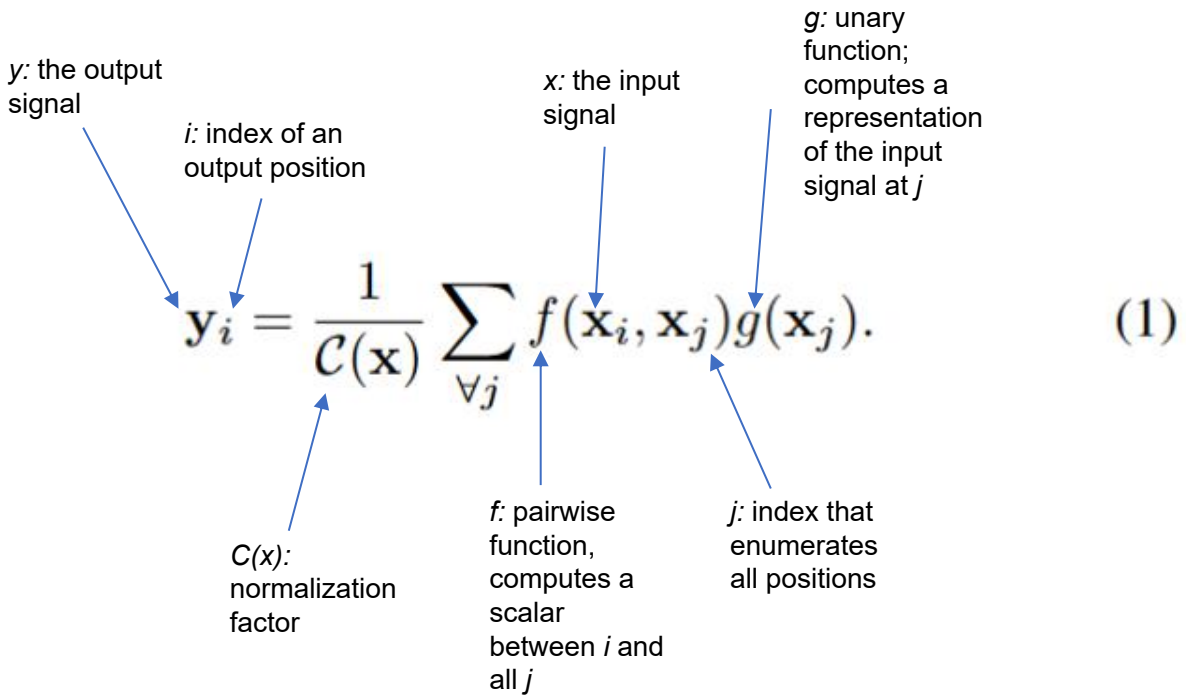


Figure 1. A spacetime ***non-local*** operation in our network trained for video classification in Kinetics. A position $\mathbf{x}_i$'s response is computed by the weighted average of the features of *all* positions $\mathbf{x}_j$ (only the highest weighted ones are shown here). In this example computed by our model, note how it relates the ball in the first frame to the ball in the last two frames. More examples are in Figure 3.

Formulation

- A generic non-local operation in deep neural networks is given as (1)
- Here, *j* enumerates all positions in the input signal
  - Compare this to a convolutional operation, that sums up weighted input in a local neighborhood
  - Or, to a recurrent operation that is often based on preceding and subsequent time steps
- *NOT* a FC layer – here, responses are computed based on relationships between locations

$$f\left(\sum_i w_i x_i + b\right)$$



Instantiations

- What to use for *f* and *g*?

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

- **Gaussian**

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^T \mathbf{x}_j}$$

- **Embedded Gaussian**

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}.$$

$$\theta(\mathbf{x}_i) = W_\theta \mathbf{x}_i$$

$$\phi(\mathbf{x}_j) = W_\phi \mathbf{x}_j$$

- **Dot Product**

$$f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j).$$

- **Concatenation**

$$f(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{x}_i), \phi(\mathbf{x}_j)]).$$

Non-local Block

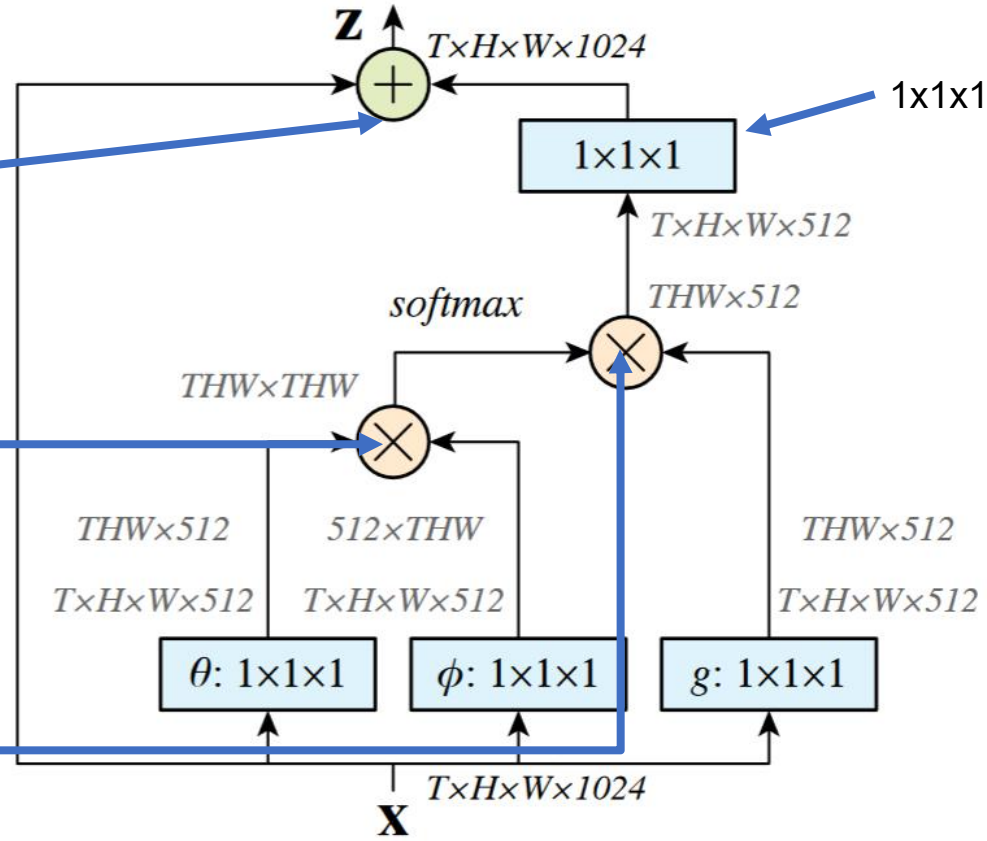
- Equation (1) is wrapped in a non-local block for incorporation into existing architectures

Residual connection

$$\mathbf{z}_i = W_z \mathbf{y}_i + \mathbf{x}_i,$$

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

Figure 2. A spacetime **non-local block**. The feature maps are shown as the shape of their tensors, *e.g.*, $T \times H \times W \times 1024$ for 1024 channels (proper reshaping is performed when noted). "$\otimes$" denotes matrix multiplication, and "$\oplus$" denotes element-wise sum. The softmax operation is performed on each row. The blue boxes denote $1 \times 1 \times 1$ convolutions. Here we show the embedded Gaussian version, with a bottleneck of 512 channels. The vanilla Gaussian version can be done by removing θ and ϕ, and the dot-product version can be done by replacing softmax with scaling by $1/N$.



Networks and Implementation Details

2D ConvNet Baseline (C2D)

layer		output size
$conv_1$	7×7, 64, stride 2, 2, 2	16×112×112
$pool_1$	3×3×3 max, stride 2, 2, 2	8×56×56
res_2	[1×1, 64; 3×3, 64; 1×1, 256] ×3	8×56×56
$pool_2$	3×1×1 max, stride 2, 1, 1	4×56×56
res_3	[1×1, 128; 3×3, 128; 1×1, 512] ×4	4×28×28
res_4	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	4×14×14
res_5	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	4×7×7
global average pool, fc		1×1×1

Table 1. Our *baseline* ResNet-50 C2D model for video. The dimensions of 3D output maps and filter kernels are in T×H×W (2D kernels in H×W), with the number of channels following. The input is 32×224×224. Residual blocks are shown in brackets.

Inflated 3D ConvNet (I3D)

- Kernels are inflated to third dimension (k*k becomes t*k*k, spanning t frames)

Non-local Network

- Described non-local blocks are inserted into C2D or I3D
- Models trained on ImageNet using 32-frame input clips

Results

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

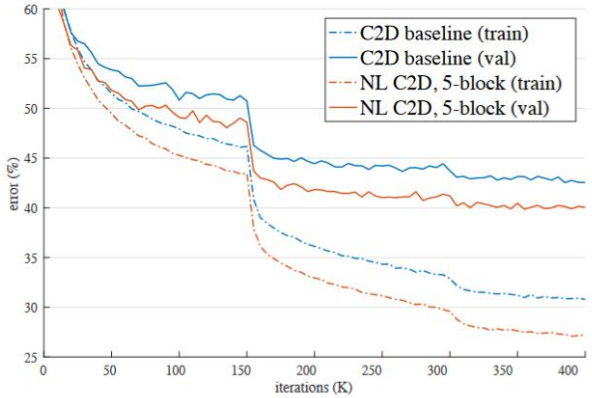


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

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**

(a) **Instantiations**: 1 non-local block of different types is added into the C2D baseline. All entries are with ResNet-50.

Instantiations

- Different instantiations are compared
- Adding 1 non-local block improves performance over baseline
- As can be seen, the addition of a non-local block is insensitive to instantiations

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

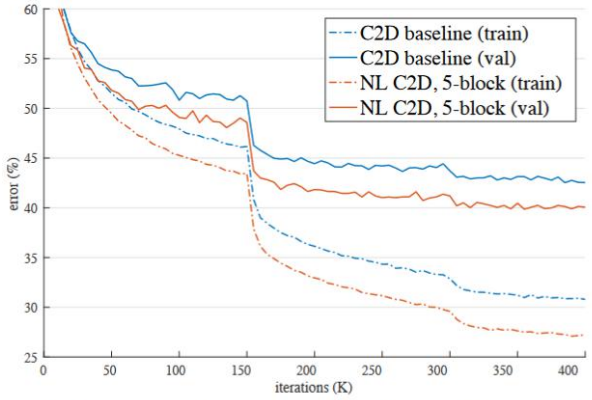


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

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

(b) **Stages**: 1 non-local block is added into different stages. All entries are with ResNet-50.

Stages

- A single non-local block was added to different instantiations
- Additions early in the network show similar improvements
- A somewhat smaller improvement is shown when applied to res_5, likely due to small spatial size (7x7)

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

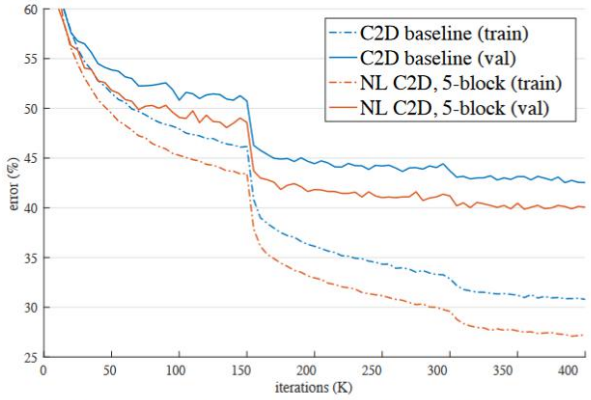


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

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

(c) **Deeper non-local models**: we compare 1, 5, and 10 non-local blocks added to the C2D baseline. We show ResNet-50 (top) and ResNet-101 (bottom) results.

Deeper with Non-Local Blocks

- Multiple non-local blocks were added to the baseline model
- In general, it was found that more non-local blocks lead to better results
- Results show this is not solely due to increased model depth

Results

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

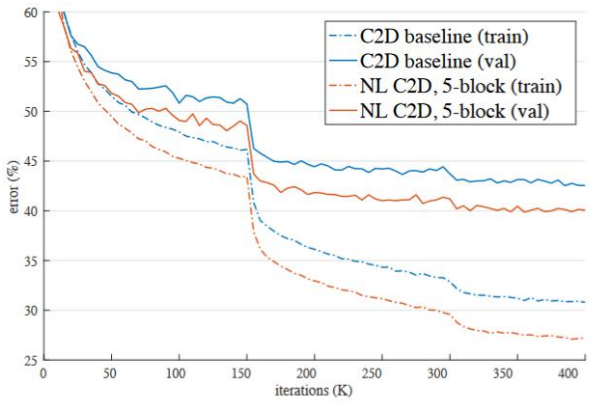


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

	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**

(d) **Space *vs.* time *vs.* spacetime**: we compare non-local operations applied along space, time, and spacetime dimensions respectively. 5 non-local blocks are used.

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

Non-local Blocks in Spacetime

- In space only, non-local blocks only considers single-frame (only sum over *j* for frame *i*)
- In time-only, reverse
- In general, space and time better than baseline, but worse than spacetime

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

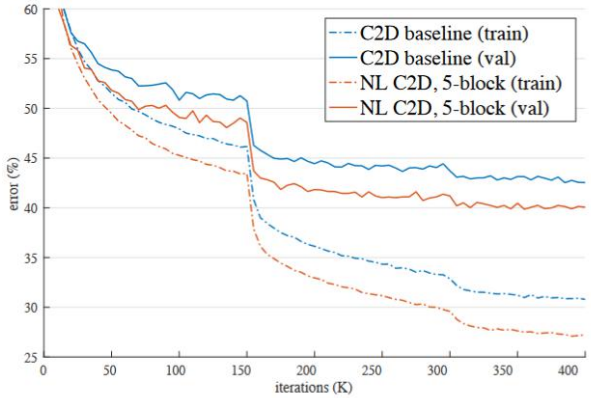


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

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**

(e) **Non-local *vs.* 3D Conv**: A 5-block non-local C2D *vs.* inflated 3D ConvNet (I3D) [7]. All entries are with ResNet-101. The numbers of parameters and FLOPs are relative to the C2D baseline (43.2M and 34.2B).

Non-local Net vs. 3D ConvNet

- Non-local nets and 3D ConvNets are both ways to extend models to temporal dimension
- Non-local blocks found to be more accurate than 3D ConvNet, with less computational requirement

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

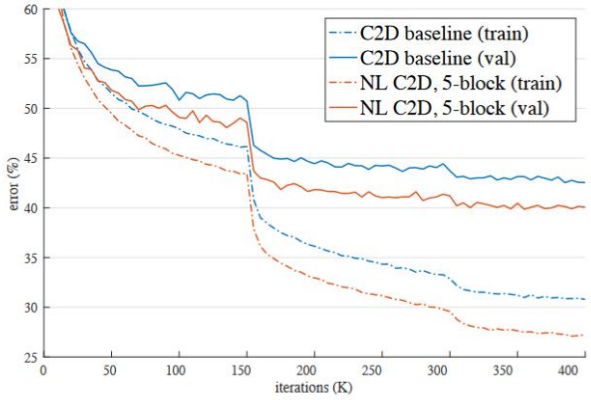


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

	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**

(f) **Non-local 3D ConvNet**: 5 non-local blocks are added on top of our best I3D models. These results show that non-local operations are complementary to 3D convolutions.

Non-local 3D ConvNet

- Non-local blocks were then added to the 3D ConvNet architecture
- Again, increased performance was observed

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

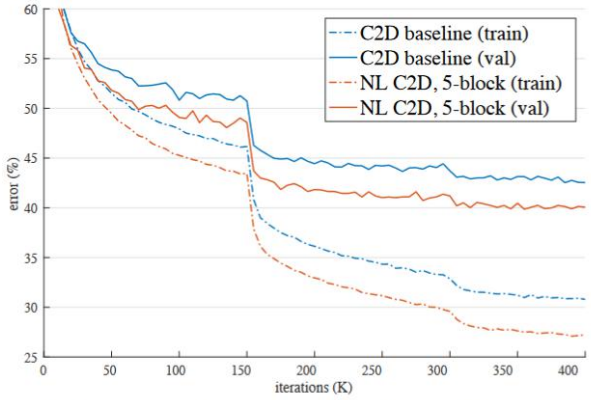


Figure 4. Curves of the training procedure on Kinetics for the ResNet-50 C2D baseline (blue) *vs.* non-local C2D with 5 blocks (red). We show the top-1 training error (dash) and validation error (solid). The validation error is computed in the same way as the training error (so it is 1-clip testing with the same random jittering at training time); the final results are in Table 2c (R50, 5-block).

	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**

(g) **Longer clips**: we fine-tune and test the models in Table 2f on the 128-frame clips. The gains of our non-local operations are consistent.

Longer Sequences

- Longer input video sequences are examined
- 128 frames
- All models have better results on longer inputs
- NL I3D maintains advantage over baselines

Results

Experiments on Video Classification

Kinetics

- ~246k training videos, ~20k validation videos with 400 human action categories

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**

Table 3. Comparisons with state-of-the-art results in **Kinetics**, reported on the val and test sets. We include the Kinetics 2017 competition winner's results [3], but their best results exploited audio signals (marked in gray) so were not vision-only solutions. †: "avg" is the average of top-1 and top-5 accuracy; individual top-1 or top-5 numbers are not available from the test server at the time of submitting this manuscript.

Comparison to SOTA

- NL method surpasses existing methods

Figure 3. Examples of the behavior of a non-local block in res_3 computed by a 5-block non-local model trained on Kinetics. These examples are from held-out validation videos. The starting point of arrows represents one $\mathbf{x}_i$, and the ending points represent $\mathbf{x}_j$. The 20 highest weighted arrows for each $\mathbf{x}_i$ are visualized. The 4 frames are from a 32-frame input, shown with a stride of 8 frames. These visualizations show how the model finds related clues to support its prediction.

Experiments on COCO

COCO

- Static image recognition; object detection/segmentation and human pose estimation (keypoint detection)

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**

Table 5. Adding 1 non-local block to Mask R-CNN for COCO **object detection** and **instance segmentation**. The backbone is ResNet-50/101 or ResNeXt-152 [53], both with FPN [32].

Object Detection and Instance Segmentation

- NL block added to Mask R-CNN model
- 3 different backbones tested
- Addition of non-local block improved performance in all cases, in both boxing and masking
- NL blocks are complementary to increasing model capacity
- Gain is at a small cost (<5% more additional computation)

Experiments on COCO

COCO

- Static image recognition; object detection/segmentation and human pose estimation (keypoint detection)

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**

Table 6. Adding non-local blocks to Mask R-CNN for COCO **keypoint detection**. The backbone is ResNet-101 with FPN [32].

Keypoint Detection

- Four non-local blocks inserted into Mask R-CNN model (one NL block after every 2 convolutional layers)
- Performance is again increased in all categories

Conclusion

- Long-range dependencies can be captured using the described non-local operations
- Non-local operations in video classification, object detection/segmentation, and keypoint estimation can increase performance at small computational cost

Acknowledgement:

This work was partially supported by ONR MURI N000141612007, Sloan, Okawa Fellowship to AG and NVIDIA Fellowship
to XW. We would also like to thank Haoqi Fan, Du Tran, Heng Wang, Georgia Gkioxari and Piotr Dollar for many helpful discussions.

Authors:

- Xiaolong Wang
- Ross Girshick
- Abhinav Gupta
- Kaiming He

Supported By:

Carnegie Mellon University

Facebook AI Research (FAIR)

Code available at https://github.com/facebookresearch/video-nonlocal-net