A Closer Look at Spatiotemporal Convolutions for Action Recognition

by Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri

Presented by Jeffrey Wang, Kamran Alipour and Menghe Zhang
Introduction

- Highest score on Sports-1M
- One Simple Trick... R(2+1)D
Background

- Image Classification
- Video Classification
Convolutional Neural Networks
Convolutional Neural Networks
Convolutional Neural Networks

- The more layers, the more non-linearities
Long Short-Term Memory (LSTM) Networks

sequential input & output
Background

- Image Classification
  - Advances
    - Smaller spatial filters
    - Batch normalization
    - Residual learning
    - ...
  - Deepnet vs. handcrafted

- Video Classification
  - Not a lot of advances
    - Single frame works almost just as well
  - Handcrafted does almost just as well
Related Work

- CNNs + RNNs
- 3D CNNs
- Two-stream framework
Related Work

- CNNs + RNNs
- 3D CNNs
- Two-stream framework
Related Work

- CNNs + RNNs
- 3D CNNs
- Two-stream framework
Related Work

- Improved Dense Trajectories (iDT)
  - First account for camera motion
Related Work

- I3D

Inflated Inception-V1

Video → 7x7x7 Conv (stride 2) → 1x3x3 Max-Pool (stride 1,2,2) → 1x1x1 Conv → 3x3x3 Conv → 1x3x3 Max-Pool (stride 1,2,2) → Inc. → Rec. Field: 7,11,11 → Rec. Field: 11,27,27

Inc. → Inc. → Inc. → Inc. → 3x3x3 Max-Pool (stride 2) → Inc. → Rec. Field: 23,75,75

Inc. → 2x2x2 Max-Pool (stride 2) → Inc. → Inc. → 2x7x7 Avg-Pool → 1x1x1 Conv → Predictions → Rec. Field: 59,219,219 → Rec. Field: 99,539,539
Convolutional Blocks

- **R2D**: 2D convolutions over the entire clip
- **f-R2D**: 2D convolutions over frames
- **R3D**: 3D convolutions
- **MCX rMCX**: Mixed 3D-2D convolutions
- **R(2+1)D**: (2+1) D Convolutions
CNN Architectures

\[ z_i = z_{i-1} + F(z_{i-1}; \theta_i) \]

3xLxHxW

(a) R2D
(b) MCx
(c) rMCx
(d) R3D
(e) R(2+1)D
R2D: 2D Convolutions

- Ignore the temporal ordering
- The very first convolutional layer collapses the entire temporal in single-channel feature maps

Input Dim: \(3L \times H \times W\)

Output \(z_i\) : \(N_i \times H_i \times W_i\)

Filter : \(N_{i-1} \times d \times d\)
f-R2D: 2D Convolutions

- No temporal modeling in convolutional layers
- Fuse L outputs in global spatiotemporal pooling layer

**Input Dim:** $3 \times H \times W$

**Output $z_i$:** $N_i \times H_i \times W_i$

**Filter:** $N_{i-1} \times d \times d$
R3D: 3D Convolutions

- Preserve temporal information
- Propagate temporal information through the layers of network
- Convolved over both time and the space dimension

Input Dim: $3 \times L \times H \times W$

Output $z_i$: $N_i \times t_i \times H_i \times W_i$

Filter: $N_{i-1} \times t \times d \times d$
MCx : 3D+2D Convolutions

► Motion modeling may be particularly useful in the early layers, while at higher levels of semantic abstraction, motion or temporal modeling is not necessary.
rMCx: 2D+3D Convolutions

Input Dim: $3 \times L \times H \times W$
Output $z_i(3d)$: $N_i \times t_i \times H_i \times W_i$
Output $z_i(2d)$: $N_i \times H_i \times W_i$
Filter(3d): $N_{i-1} \times t \times d \times d$
Filter(2d): $N_{i-1} \times d \times d$

- Temporal modeling may be more beneficial in the deep layers, with early capturing appearance information via 2d convolutions.
R3D vs R(2+1)D

- **Kernel:** \( t \times d \times d \)
  Convolve spatial and temporal information together.

- **Kernel:** \( 1 \times d \times d + t \times 1 \times 1 \)
  Decompose spatial and temporal convolution

- **Complexity is increased**: Double the number of nonlinearities
- **Easier Optimization**: Force the 3D convolution into separate spatial and temporal components.
R(2+1)D: (2+1)D Convolutions

2d spatial

Intermediate output

1d temporal

Temporal convolutional filter size:

\[ M_i \times t \times 1 \times 1 \]

\[ M_i = \left[ \frac{t d^2 N_{i-1} N_i}{d^2 N_{i-1} + t N_i} \right] \]

Approximately equal to implementation of full 3D convolution
# (2+1)D vs 3D Convolution

<table>
<thead>
<tr>
<th>Net</th>
<th># params (in millions)</th>
<th>Clip@1 (%)</th>
<th>Video (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3D</td>
<td>33.4</td>
<td>49.3</td>
<td>61.2</td>
</tr>
<tr>
<td>R(2+1)D</td>
<td>33.3</td>
<td>52.4</td>
<td>63.9</td>
</tr>
</tbody>
</table>

Accuracy comparison between R3D and R(2+1)D

R(2+1)D has roughly the same cost as R3D but it yields higher accuracy
(2+1)D vs 3D Convolution

What makes (2+1)D convolutions better than 3D?

Training and testing errors for R3D and R(2+1)D
Datasets

- Sports-1M (trained on)
- Kinetics (trained on)
- UCF101
- HMDB51
Datasets

Sports-1M

- 1,133,158 video URLs
- 487 labels
- Annotated via YouTube Topics API

Labels: tumbling (gymnastics), trampolining

Labels: boomerang
Datasets

Kinetics

- ~240,000 video clips
- Each clip around 10 secs
- Each clip labeled with one class
- 400 human action classes
- At least 400 video clips for each class
Datasets

UCF101

- 13320 videos
- 101 categories
- Very diverse in terms of:
  - Actions
  - Camera motions
  - Object pose & scale
  - Illumination
Datasets

HMDB51

- Various sources, mostly movies
- 6849 clips
- 51 action categories
- Each category minimum of 101 clips
- Action categories:
  - General facial actions: smile, laugh, ...
  - Facial actions with objects: smoke, drink, ...
  - General Body movements: run, jump, ...
  - Body movements with object: catch, draw, ...
## Comparison with the State-of-the-art

Action recognition accuracy on the Kinetics validation set

<table>
<thead>
<tr>
<th>Net</th>
<th># params</th>
<th>Clip@1</th>
<th>Video@1</th>
<th>Clip@1</th>
<th>Video@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td>8×112×112</td>
<td>16×112×112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2D</td>
<td>11.4M</td>
<td>46.7</td>
<td>59.5</td>
<td>47.0</td>
<td>58.9</td>
</tr>
<tr>
<td>f-R2D</td>
<td>11.4M</td>
<td>48.1</td>
<td>59.4</td>
<td>50.3</td>
<td>60.5</td>
</tr>
<tr>
<td>R3D</td>
<td>33.4M</td>
<td>49.4</td>
<td>61.8</td>
<td>52.5</td>
<td>64.2</td>
</tr>
<tr>
<td>MC2</td>
<td>11.4M</td>
<td>50.2</td>
<td>62.5</td>
<td>53.1</td>
<td>64.2</td>
</tr>
<tr>
<td>MC3</td>
<td>11.7M</td>
<td>50.7</td>
<td>62.9</td>
<td>53.7</td>
<td>64.7</td>
</tr>
<tr>
<td>MC4</td>
<td>12.7M</td>
<td>50.5</td>
<td>62.5</td>
<td>53.7</td>
<td>65.1</td>
</tr>
<tr>
<td>MC5</td>
<td>16.9M</td>
<td>50.3</td>
<td>62.5</td>
<td>53.7</td>
<td>65.1</td>
</tr>
<tr>
<td>rMC2</td>
<td>33.3M</td>
<td>49.8</td>
<td>62.1</td>
<td>53.1</td>
<td>64.9</td>
</tr>
<tr>
<td>rMC3</td>
<td>33.0M</td>
<td>49.8</td>
<td>62.3</td>
<td>53.2</td>
<td>65.0</td>
</tr>
<tr>
<td>rMC4</td>
<td>32.0M</td>
<td>49.9</td>
<td>62.3</td>
<td>53.4</td>
<td>65.1</td>
</tr>
<tr>
<td>rMC5</td>
<td>27.9M</td>
<td>49.4</td>
<td>61.2</td>
<td>52.1</td>
<td>63.1</td>
</tr>
<tr>
<td>R(2+1)D</td>
<td>33.3M</td>
<td><strong>52.8</strong></td>
<td><strong>64.8</strong></td>
<td><strong>56.8</strong></td>
<td><strong>68.0</strong></td>
</tr>
</tbody>
</table>
Comparison with the State-of-the-art

Action recognition accuracy on the Kinetics validation set

1. Motion modeling is important for action recognition
2. Decomposing into spatial and temporal convolutions is better than mixed 3D-2D convolutions.
3. Spatiotemporal modeling is particularly beneficial in the early layers.
4. The spatiotemporal decomposition renders easier optimization
Comparison of spatiotemporal convolutions

Accuracy vs computational complexity for different types of convolution on Kinetics
Comparison of spatiotemporal convolutions

- Varol et. al (2017):

  Longer Clips \(\rightarrow\) LTC \(\rightarrow\) More Accuracy
What causes the differences in video-level accuracies?

Experiments:

1- Model trained on 8-frames, tested on 32-frames:
   - Clip accuracy drop 1.2%
   - Video accuracy drop 5.8%

2- Model initialized on 8 frames, fine tuned on 32 frames:
   - Results as good as learning on 32-frames from scratch
   - Gain 7% over 8 frame model
   - Shorter training

training on longer clips yields better clip-level models, as the filters learn longer-term temporal patterns
## Training time/accuracy trade-off

<table>
<thead>
<tr>
<th>train</th>
<th>finetune</th>
<th>test</th>
<th>training time</th>
<th>Clip@1</th>
<th>Video@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>clip length (in frames)</td>
<td></td>
<td></td>
<td>hours</td>
<td>(%)</td>
<td>(%)</td>
</tr>
<tr>
<td>8</td>
<td>none</td>
<td>8</td>
<td>11.8</td>
<td>52.8</td>
<td>64.8</td>
</tr>
<tr>
<td>8</td>
<td>none</td>
<td>32</td>
<td>11.8</td>
<td>51.6</td>
<td>59.0</td>
</tr>
<tr>
<td>32</td>
<td>none</td>
<td>32</td>
<td>59.8</td>
<td>60.1</td>
<td>69.4</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>32</td>
<td>20.5</td>
<td>59.8</td>
<td>68.0</td>
</tr>
</tbody>
</table>
### Action recognition with a 34-layer R(2+1)D

<table>
<thead>
<tr>
<th>method</th>
<th>Clip@1</th>
<th>Video@1</th>
<th>Video@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepVideo [16]</td>
<td>41.9</td>
<td>60.9</td>
<td>80.2</td>
</tr>
<tr>
<td>C3D [36]</td>
<td>46.1</td>
<td>61.1</td>
<td>85.2</td>
</tr>
<tr>
<td>2D Resnet-152 [13]</td>
<td>46.5*</td>
<td>64.6*</td>
<td>86.4*</td>
</tr>
<tr>
<td>Conv pooling [42]</td>
<td>-</td>
<td>71.7</td>
<td>90.4</td>
</tr>
<tr>
<td>P3D [25]</td>
<td>47.9*</td>
<td>66.4*</td>
<td>87.4*</td>
</tr>
<tr>
<td>R3D-RGB-8frame</td>
<td>53.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R(2+1)D-RGB-8frame</td>
<td>56.1</td>
<td>72.0</td>
<td>91.2</td>
</tr>
<tr>
<td>R(2+1)D-Flow-8frame</td>
<td>44.5</td>
<td>65.5</td>
<td>87.2</td>
</tr>
<tr>
<td>R(2+1)D-Two-Stream-8frame</td>
<td>-</td>
<td>72.2</td>
<td>91.4</td>
</tr>
<tr>
<td>R(2+1)D-RGB-32frame</td>
<td>57.0</td>
<td>73.0</td>
<td>91.5</td>
</tr>
<tr>
<td>R(2+1)D-Flow-32frame</td>
<td>46.4</td>
<td>68.4</td>
<td>88.7</td>
</tr>
<tr>
<td>R(2+1)D-Two-Stream-32frame</td>
<td>-</td>
<td>73.3</td>
<td>91.9</td>
</tr>
</tbody>
</table>
Action recognition with a 34-layer R(2+1)D

Kinetics

- Learning rate: 0.001
- Fine-tuning in 15 Epochs
- I3D-two-stream outperforms

<table>
<thead>
<tr>
<th>method</th>
<th>pretraining dataset</th>
<th>top1</th>
<th>top5</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D-RGB [4]</td>
<td>none</td>
<td>67.5</td>
<td>87.2</td>
</tr>
<tr>
<td>I3D-RGB [4]</td>
<td>ImageNet</td>
<td>72.1</td>
<td>90.3</td>
</tr>
<tr>
<td>I3D-Flow [4]</td>
<td>ImageNet</td>
<td>65.3</td>
<td>86.2</td>
</tr>
<tr>
<td>I3D-Two-Stream [4]</td>
<td>ImageNet</td>
<td>75.7</td>
<td>92.0</td>
</tr>
<tr>
<td>R(2+1)D-RGB</td>
<td>none</td>
<td>72.0</td>
<td>90.0</td>
</tr>
<tr>
<td>R(2+1)D-Flow</td>
<td>none</td>
<td>67.5</td>
<td>87.2</td>
</tr>
<tr>
<td>R(2+1)D-Two-Stream</td>
<td>none</td>
<td>73.9</td>
<td>90.9</td>
</tr>
<tr>
<td>R(2+1)D-RGB</td>
<td>Sports-1M</td>
<td>74.3</td>
<td>91.4</td>
</tr>
<tr>
<td>R(2+1)D-Flow</td>
<td>Sports-1M</td>
<td>68.5</td>
<td>88.1</td>
</tr>
<tr>
<td>R(2+1)D-Two-Stream</td>
<td>Sports-1M</td>
<td>75.4</td>
<td>91.9</td>
</tr>
</tbody>
</table>
## Transferring models to UCF101 and HMDB51

<table>
<thead>
<tr>
<th>method</th>
<th>pretraining dataset</th>
<th>UCF101</th>
<th>HMDB51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-Stream [29]</td>
<td>ImageNet</td>
<td>88.0</td>
<td>59.4</td>
</tr>
<tr>
<td>Action Transf. [40]</td>
<td>ImageNet</td>
<td>92.4</td>
<td>62.0</td>
</tr>
<tr>
<td>Conv Pooling [42]</td>
<td>Sports-1M</td>
<td>88.6</td>
<td>-</td>
</tr>
<tr>
<td>$F_{STCN}$ [33]</td>
<td>ImageNet</td>
<td>88.1</td>
<td>59.1</td>
</tr>
<tr>
<td>Two-Stream Fusion [10]</td>
<td>ImageNet</td>
<td>92.5</td>
<td>65.4</td>
</tr>
<tr>
<td>Spatiotemp. ResNet [9]</td>
<td>ImageNet</td>
<td>93.4</td>
<td>66.4</td>
</tr>
<tr>
<td>Temp. Segm. Net [39]</td>
<td>ImageNet</td>
<td>94.2</td>
<td>69.4</td>
</tr>
<tr>
<td>P3D [25]</td>
<td>ImageNet+Sports1M</td>
<td>88.6</td>
<td>-</td>
</tr>
<tr>
<td>I3D-RGB [4]</td>
<td>ImageNet+Kinetics</td>
<td>95.6</td>
<td>74.8</td>
</tr>
<tr>
<td>I3D-Flow [4]</td>
<td>ImageNet+Kinetics</td>
<td>96.7</td>
<td>77.1</td>
</tr>
<tr>
<td>I3D-Two-Stream [4]</td>
<td>ImageNet+Kinetics</td>
<td><strong>98.0</strong></td>
<td><strong>80.7</strong></td>
</tr>
<tr>
<td>R(2+1)D-RGB</td>
<td>Sports1M</td>
<td>93.6</td>
<td>66.6</td>
</tr>
<tr>
<td>R(2+1)D-Flow</td>
<td>Sports1M</td>
<td>93.3</td>
<td>70.1</td>
</tr>
<tr>
<td>R(2+1)D-TwoStream</td>
<td>Sports1M</td>
<td>95.0</td>
<td>72.7</td>
</tr>
<tr>
<td>R(2+1)D-RGB</td>
<td>Kinetics</td>
<td>96.8</td>
<td>74.5</td>
</tr>
<tr>
<td>R(2+1)D-Flow</td>
<td>Kinetics</td>
<td>95.5</td>
<td>76.4</td>
</tr>
<tr>
<td>R(2+1)D-TwoStream</td>
<td>Kinetics</td>
<td>97.3</td>
<td>78.7</td>
</tr>
</tbody>
</table>
Future Work

- Expand beyond ResNet
- Investigate clip to video methods beyond just averaging
Summary

- R2D vs. R3D: 58.9% vs 64.2% accuracy
- MCx vs rMCx: first capture temporal information with 3D convolutions
- R(2+1)D is an effective architecture for video classification
  - Temporal information is crucial; decomposition doubles the non-linearities and eases optimization