Long-term Recurrent Convolutional Networks for Visual Recognition and Description

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach

Presented By: Harshul Gupta
LRCN is a class of models that is both spatially and temporally deep. It uses **ConvNet** (encoder) to encode deep spatial state vector and an **LSTM** (decoder) to generate natural language strings.

**ConvNet** : Time-invariant and independent at each timestep

✔ Enables training parallelizable over all timesteps of the input.
✔ Independent batch processing

**LSTM** : Allows modeling of sequential data of varying lengths.

LRCN architecture take a sequence of T inputs and generate a sequence of T outputs.
Activity Recognition

Sequential Input $< F_1, F_2, F_3, \ldots, F_t >$ (different visual frames)
Scalar Output $< Y >$ (final label)

How can LRCN model be used for Activity Recognition Task?

✔ Predict video class at each time step $< Y_1, Y_2, Y_3, \ldots, Y_t >$
✔ Average these predictions for final classification ($Y$)
Both RGB and Optical Flow are given as inputs and a weighted average of both is used for the final prediction.

**Advantage of using both RGB and Optical Flow models?**

- **Typing**: Strongly correlated with the presence of certain objects, such as a keyboard, and are thus best learned by the RGB model.
- **Juggling**: Include more generic objects which are frequently seen in other activities (such as ball, humans) and are thus best identified from class-specific motion cues.

Because RGB and flow signals are complementary, the best models take both into account.

Variants of the LRCN architecture are used:
- LSTM is placed after the first fully connected layer of the CNN (LRCN-fc6)
- LSTM is placed after the second fully connected layer of the CNN (LRCN-fc7).
Evaluation

Evaluation is done on the UCF-101 dataset which consists of over 12,000 videos categorized into 101 human action classes.

<table>
<thead>
<tr>
<th>Model</th>
<th>Single Input Type</th>
<th>Weighted Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RGB</td>
<td>Flow</td>
</tr>
<tr>
<td>Single frame (split-1)</td>
<td>69.00</td>
<td>72.20</td>
</tr>
<tr>
<td>LRCN-fc6 (split-1)</td>
<td>71.12</td>
<td>76.95</td>
</tr>
<tr>
<td>LRCN-fc7 (split-1)</td>
<td>70.68</td>
<td>69.36</td>
</tr>
<tr>
<td>Single frame (all splits)</td>
<td>67.70</td>
<td>72.19</td>
</tr>
<tr>
<td>LRCN-fc6 (all splits)</td>
<td>68.19</td>
<td>77.46</td>
</tr>
</tbody>
</table>

- The LRCN-fc6 network yields the best results for both RGB and flow.
- Since the flow network outperformed the RGB network, weighting the flow network higher lead to better accuracy.
- LRCN outperforms the baseline single-frame model by 3.82%.
**Image Description**

**Scalar Input** $< F >$

**Sequential Output** $< Y_1, Y_2, Y_3, \ldots, Y_t >$

How can LRCN model be used for Image Description Task?

✔ Simply duplicate the input $F$ at all $T$ timesteps

Inputs @ timestep $t$ (Factored)

- LSTM-1: embedded ground truth word (“one-hot”) from the previous timestep.
- LSTM-2: outputs of the first LSTM + image representation to produce a joint representation of the visual and language inputs

Can you think of any advantage this might offer?

It creates a sort of separation of responsibilities by ensuring that the hidden state of lower LSTM is conditionally independent of visual input. This forces all of the capacity of their hidden states to represent only the caption.
Evaluation

Datasets
Flickr30k (30,000 training images) and COCO 2014 (80,000 training images)

Metrics
Recall@K: Number of images for which a correct caption is retrieved within the top K results (higher is better)
Median Rank (Medr): Median of the first retrieved ground truth caption (lower is better)

Human Evaluator Rankings (lower is better)

<table>
<thead>
<tr>
<th>Method</th>
<th>Correctness</th>
<th>Grammar</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeTalk [23]</td>
<td>4.08</td>
<td>4.35</td>
<td>3.98</td>
</tr>
<tr>
<td>OxfordNet [18]</td>
<td>3.71</td>
<td>3.46</td>
<td>3.70</td>
</tr>
<tr>
<td>NN [18]</td>
<td><strong>3.44</strong></td>
<td>3.20</td>
<td><strong>3.49</strong></td>
</tr>
<tr>
<td>LRCN fc8 (ours)</td>
<td>3.74</td>
<td>3.19</td>
<td>3.72</td>
</tr>
<tr>
<td>LRCN FT (ours)</td>
<td><strong>3.47</strong></td>
<td><strong>3.01</strong></td>
<td><strong>3.50</strong></td>
</tr>
<tr>
<td>Captions</td>
<td>2.55</td>
<td>3.72</td>
<td>2.59</td>
</tr>
</tbody>
</table>

- LRCN with 2 layers LSTM (factored) performs better both in terms of Medr and Recall@K.
- Generated sentences were evaluated by Amazon Mechanical Turkers to rank them based on correctness, grammar and relevance.
Image Description Results

A close up of a hot dog on a bun.

A boat on a river with a bridge in the background.

A bathroom with a toilet and a bathtub.

A man that is standing in the dirt with a bat.

A white toilet sitting in a bathroom next to a trash can.

Black and white photograph of a woman sitting on a bench.
Video Description

**Sequential Input** \(< F_1, F_2, F_3, ..., F_t >\) (T inputs)

**Sequential Output** \(\{Y_1, Y_2, Y_3, ..., Y_{t'}\}\) (T’ outputs)

For different input and output lengths, an “**encoder-decoder**” approach is taken

**Encoder**: Maps the input sequence to a fixed-length vector

**Decoder**: Unrolls this vector to sequential outputs of arbitrary length

Under this model, the system as a whole may be thought of as having \(T + T'\) timesteps of input and output

Semantic representation of the video is obtained using the MAP of a CRF.

It is translated to a sentence using encoder-decoder LSTM architecture.
Similar to Image Description Task

Semantic representation is encoded as a single fixed length vector.

Hence, provide entire visual input representation at each time step to the LSTM decoder.

Similar to above but CRF max is replaced with probability
Evaluation

Datasets
TACoS (Textually Annotated Cooking Scenes) multilevel dataset, which has 44,762 video/sentence pairs (about 40,000 for training/validation).
It contains multiple sentence descriptions and longer videos, however they are restricted to the cooking scenario.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Input</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT [30]</td>
<td>CRF max</td>
<td>24.9</td>
</tr>
<tr>
<td>SMT [29]</td>
<td>CRF prob</td>
<td>26.9</td>
</tr>
<tr>
<td>(a) LSTM Encoder-Decoder (ours)</td>
<td>CRF max</td>
<td>25.3</td>
</tr>
<tr>
<td>(b) LSTM Decoder (ours)</td>
<td>CRF max</td>
<td>27.4</td>
</tr>
<tr>
<td>(c) LSTM Decoder (ours)</td>
<td>CRF prob</td>
<td>28.8</td>
</tr>
</tbody>
</table>

- LSTM outperforms an SMT-based approach to video description;
- Simpler decoder architecture (b) and (c) achieve better performance than (a) (BLEU-4 scores)
1) The person entered the kitchen.
2) The person took out a bag of pasta.
3) The person placed the pasta on the counter.
4) The person placed a pot on the stove.
5) The person filled the pot with water.
6) The person put the pot on the stove.
Conclusions

LRCN is a flexible framework for vision problems involving sequences

Able to handle:
✔ Sequences in the input (video)
✔ Sequences in the output (natural language description)