Recurrent Neural Networks for Person Re-identification Revisited

Jean-Baptiste Boin
Stanford University
jbboin@stanford.edu

André Araujo
Google AI
andrearaujo@gmail.com

Bernd Girod
Stanford University
bgirod@stanford.edu
Person video re-identification

- Goal: associate person video tracks from different cameras
- Applications:
  - Video surveillance
  - Home automation
  - Crowd dynamics understanding

Image credit: PRID2011 dataset [Hirzer et al., 2011]
Person video re-identification: challenges

- Lighting variations
- Viewpoint changes
- Clothing similarity
- Background clutter and occlusions

Credit: iLIDS-VID dataset [Wang et al., 2014]
Framework: re-identification by retrieval

Database (Camera A)

Query (Camera B)

Sequence feature extraction

Sequence feature extraction

Sequence feature extraction

Sequence feature extraction

Sequence feature extraction

Sequence feature extraction

Sequence feature extraction

Sequence matching by feature similarity
Related work

- Most common setup
  - Frame feature extraction: CNN
  - Sequence processing: RNN
  - Temporal pooling: mean pooling
  - [McLaughlin et al., 2016], [Yan et al., 2016], [Wu et al., 2016]
Related work

- Most common setup
  - Frame feature extraction: CNN
  - Sequence processing: RNN
  - Temporal pooling: mean pooling
    - [McLaughlin et al., 2016], [Yan et al., 2016], [Wu et al., 2016]

- Extensions
  - Bi-directional RNNs [Zhang et al., 2017]
  - Multi-scale + attention pooling [Xu et al., 2017]
  - Fusion of CNN+RNN features [Chen et al., 2017]

See review paper [Zheng et al., 2016]
Outline

- Feed-forward RNN approximation with similar representational power
- New training protocol to leverage multiple video tracks within a mini-batch
- Experimental evaluation
- Conclusions
RNN setup

- $f^{(t)}$: inputs of sequence processing stage (frame descriptors)
- $o^{(t)}$: outputs of sequence processing stage
  \[
  o^{(t)} = W_i f^{(t)} + W_s \tanh(o^{(t-1)})
  \]
- $v_s = \frac{1}{T} \sum_{t=1}^{T} o^{(t)}$: sequence feature (output of temporal pooling stage)
Proposed feed-forward approximation (1/2)

- “Short-term dependency” approximation

  Disregard terms from step $(t-2)$ in output from step $(t)$

\[
o^{(t)} = W_if^{(t)} + W_s \tanh(o^{(t-1)}) \\
\approx W_if^{(t)} + W_s \tanh(W_if^{(t-1)})
\]
Proposed feed-forward approximation (2/2)

- “Long sequence” approximation

\[ v_s = \frac{1}{T} \sum_{t=1}^{T} \sigma^{(t)} \]

\[ \approx \frac{1}{T} \sum_{t=1}^{T} \left( W_if^{(t)} + W_s \tanh(W_if^{(t-1)}) \right) \]

\[ = \frac{1}{T} \sum_{t=1}^{T} W_if^{(t)} + \frac{1}{T} \sum_{t=0}^{T-1} W_s \tanh(W_if^{(t)}) \]

\[ \approx \frac{1}{T} \sum_{t=1}^{T} \left. \left( W_if^{(t)} + W_s \tanh(W_if^{(t)}) \right) \right|_{\sigma^{(t)}} \]

Using approximation from previous slide

Disregard edge cases (first and last frame) since videos are long
Proposed feed-forward approximation: new block

- Same memory footprint
- Direct mapping between RNN and FNN parameters
Training pipeline

- Training data

- Video tracks (camera A)

- Video tracks (camera B)

Frames
Training pipeline: RNN baseline

- **SEQ**: load sequences of consecutive frames in mini-batch
Proposed FNN training pipeline

- **FRM**: load independent frames
- Load images from many more identities in a mini-batch (same memory/computational cost)
Data and experimental protocol

- Dataset 1: PRID2011 [Hirzer et al., 2011]
  - 200 identities, average length: 100 frames / track
- Dataset 2: iLIDS-VID [Wang et al., 2014]
  - 300 identities, average length: 71 frames / track
- Data splits
  - Train/test set with half of the identities each
  - Performance averaged over 20 splits
- Evaluation metric: CMC (equivalent to mean accuracy at rank k)
Experiment: Influence of the recurrent connection

- Train weights on RNN-SEQ (RNN architecture, SEQ training protocol)
- Evaluate on RNN and FNN using the weights directly (**no re-training**)
- Same performance obtained

*PRID2011 dataset*
Experiment: Comparison with baseline

- FNN-FRM (ours) outperforms RNN-SEQ
- More diversity in mini-batches allows for a much better training
Comparison with baseline (comprehensive)

- Our method outperforms the baseline for all ranks in both datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PRID2011</th>
<th>iLIDS-VID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>1 5 10 20</td>
<td>1 5 10 20</td>
</tr>
<tr>
<td>RNN [12]</td>
<td>70 90 95</td>
<td>97</td>
</tr>
<tr>
<td>– (reproduced)</td>
<td>71.6 92.8</td>
<td>96.6 98.5</td>
</tr>
<tr>
<td>FNN-SEQ (ours)</td>
<td>72.3 92.9</td>
<td>96.4 98.4</td>
</tr>
<tr>
<td>FNN-FRM (ours)</td>
<td><strong>76.4</strong></td>
<td><strong>95.3</strong></td>
</tr>
</tbody>
</table>

CMC values (in %)
Comparison with state-of-the-art RNN methods

- Our method is considerably simpler than the other state-of-the-art RNN methods compared but still achieves comparable performance results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PRID2011</th>
<th></th>
<th></th>
<th></th>
<th>iLIDS-VID</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>RNN [12]</td>
<td></td>
<td>70</td>
<td>90</td>
<td>95</td>
<td>97</td>
<td>58</td>
<td>84</td>
<td>91</td>
</tr>
<tr>
<td>– (reproduced)</td>
<td></td>
<td>71.6</td>
<td>92.8</td>
<td>96.6</td>
<td>98.5</td>
<td>57.1</td>
<td>83.4</td>
<td>91.8</td>
</tr>
<tr>
<td>RFA-Net [18]</td>
<td></td>
<td>58.2</td>
<td>85.8</td>
<td>93.4</td>
<td>97.9</td>
<td>49.3</td>
<td>76.8</td>
<td>85.3</td>
</tr>
<tr>
<td>Deep RCN [15]</td>
<td></td>
<td>69.0</td>
<td>88.4</td>
<td>93.2</td>
<td>96.4</td>
<td>46.1</td>
<td>76.8</td>
<td>89.7</td>
</tr>
<tr>
<td>Zhou et al. [25]</td>
<td></td>
<td><strong>79.4</strong></td>
<td><strong>94.4</strong></td>
<td>-</td>
<td><strong>99.3</strong></td>
<td>55.2</td>
<td><strong>86.5</strong></td>
<td>-</td>
</tr>
<tr>
<td>BRNN [20]</td>
<td></td>
<td>72.8</td>
<td>92.0</td>
<td>95.1</td>
<td>97.6</td>
<td>55.3</td>
<td>85.0</td>
<td>91.7</td>
</tr>
<tr>
<td>ASTPN [17]</td>
<td></td>
<td><strong>77</strong></td>
<td><strong>95</strong></td>
<td><strong>99</strong></td>
<td>99</td>
<td><strong>62</strong></td>
<td>86</td>
<td><strong>94</strong></td>
</tr>
<tr>
<td>Chen et al. [2]</td>
<td></td>
<td>77</td>
<td>93</td>
<td>95</td>
<td>98</td>
<td>61</td>
<td>85</td>
<td><strong>94</strong></td>
</tr>
<tr>
<td>FNN-FRM (ours)</td>
<td></td>
<td>76.4</td>
<td><strong>95.3</strong></td>
<td><strong>98.0</strong></td>
<td><strong>99.1</strong></td>
<td>58.0</td>
<td><strong>87.5</strong></td>
<td><strong>93.7</strong></td>
</tr>
</tbody>
</table>

CMC values (in %)
Conclusions

- Simple feed-forward RNN approximation with similar representational power
- New training protocol to leverage multiple video sequences within a mini-batch
- Results significantly and consistently improved compared to baseline
- Results on par or better than other published work based on RNNs, with a much simpler technique
- Faster model training compared to RNN baseline
Questions?