Video autoencoder with geometric priors

Viorica Pătrăucean vp344@cam.ac.uk

https://github.com/viorik/
Scene understanding from videos

Road-scene understanding for self-driving cars

Understanding geometry and semantics for large scale (outdoor) scenes
Deep learning for videos

- **Convnets are successful, but ...**
  - no labelled video dataset for training (Camvid, Cityscapes videos are not fully labelled)
  - not able to exploit temporal information (redundancy)

- **Focus: reduce supervision effort**
  - synthetic training data
  - unsupervised learning (video autoencoder)
Related work (1):

a. Unsupervised learning of video representations using LSTMs (Srivastava et al, ICML 2015)
b. Deep multi-scale video prediction beyond mean square error (Mathieu et al, ICLR 2016)
Spatio-temporal video autoencoder (ICLRw2016)
V. Pătrașcean, A. Handa, R. Cipolla

Preliminary considerations:

- Video = spatial component + temporal component; video ≠ volume
- Classic video encoders: changes-centred
  - ability to generate a frame given a reference frame and the differences
  - end-to-end setup: learn to generate next frame given current frame and motion
- Biology: visual short-term memory handles movement/light changes
- Objectness: points moving together belong to the same object
Related work (1):

a. Unsupervised learning of video representations using LSTMs (Srivastava et al, ICML 2015)
b. Deep multi-scale video prediction beyond mean square error (Mathieu et al, ICLR 2016)

Related work (2):

a. Supervised optical flow estimation (DeepFlow, FlowNet etc.)
b. Learning to see by moving (Agrawal et al, ICCV 2015)
Spatio-temporal video autoencoder (ICLRw2016)
E, D - classic convolutional image encoder/decoder (conv+tanh+down(up)sample)
**LSTM**: Convolutional LSTM

- temporal encoder
- reduced number of parameters
- more adapted to local image structures
- preserves spatial information

\[
i_t = \sigma(x_t \cdot \theta_{xi} + h_{t-1} \cdot \theta_{hi} + \theta_{ibias}),
\]
\[
f_t = \sigma(x_t \cdot \theta_{xf} + h_{t-1} \cdot \theta_{hf} + \theta_{fbias}),
\]
\[
\tilde{c}_t = \tanh(x_t \cdot \theta_{xc} + h_{t-1} \cdot \theta_{hc} + \theta_{cbias}),
\]
\[
c_t = \tilde{c}_t \odot i_t + c_{t-1} \odot f_t,
\]
\[
o_t = \sigma(x_t \cdot \theta_{xo} + h_{t-1} \cdot \theta_{ho} + \theta_{obias}),
\]
\[
h_t = o_t \odot \tanh(c_t)
\]

**Spatio-temporal video autoencoder (ICLRw2016)**
Temporal decoder: less parameters than the encoder

\( \Theta \)- regression (convolutional) module; generates dense flow map

\( H \) - Huber penalty: motion locally smooth
Spatio-temporal video autoencoder (ICLRw2016)

GG, S – customised STN modules (NIPS2015)

GG - grid generator for dense transform map
S - sampler

\[
\begin{pmatrix}
x_s \\
y_s \\
\end{pmatrix} = \mathcal{T}(x_o, y_o) \begin{pmatrix}
x_o \\
y_o \\
1 \\
\end{pmatrix}, \mathcal{T}(\cdot, \cdot) = \begin{pmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
\end{pmatrix}
\]
Spatio-temporal video autoencoder (ICLRw2016)

Reconstruction error: $$\mathcal{L}_t = \|\hat{Y}_{t+1} - Y_{t+1}\|^2_2 + w_H \mathcal{H}(\nabla \mathcal{T})$$
Spatio-temporal video autoencoder (ICLRw2016)

Training: rmsprop with momentum (lr starts at 1e-4; decays by 0.1 every 5 epochs)

Initialisation: ConvLSTM weights U(-0.08, 0.08), bias 0; other conv layers: xavier init

Results

Synthetic dataset: Moving MNIST

Architectures:

a) AE-Conv

b) (AE-fcLSTM) AE-ConvLSTM
### Results

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Number of trainable parameters</th>
<th>Cross-entropy test error</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE-Conv</td>
<td>109,073</td>
<td>0.0948</td>
</tr>
<tr>
<td>AE-fcLSTM</td>
<td>33,623,649</td>
<td>0.0650</td>
</tr>
<tr>
<td>AE-ConvLSTM</td>
<td>1,256,305</td>
<td>0.0197</td>
</tr>
<tr>
<td>AE-ConvLSTM-flow</td>
<td>1,035,067</td>
<td>0.0439</td>
</tr>
</tbody>
</table>

![Diagram a) AE-Conv](image1.png)

![Diagram b) (AE-fcLSTM) AE-ConvLSTM](image2.png)
<table>
<thead>
<tr>
<th>$t - 3$</th>
<th>$t - 2$</th>
<th>$t - 1$</th>
<th>$t$</th>
<th>$t + 1$</th>
<th>conv</th>
<th>fcLSTM</th>
<th>cLSTM</th>
<th>cLSTM-flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
<td>07</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
</tbody>
</table>
Application: video segmentation with label propagation
### Application: video segmentation with label propagation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet</td>
<td>89.2</td>
<td>65.4</td>
<td>22.6</td>
<td>92.0</td>
<td>70.7</td>
<td>44.8</td>
<td>14.9</td>
<td>5.38</td>
<td>60.5</td>
<td>25.6</td>
<td>2.73</td>
<td>44.9</td>
<td>75.3</td>
</tr>
<tr>
<td>SegNet-flow</td>
<td>92.9</td>
<td>73.4</td>
<td>3.22</td>
<td>97.2</td>
<td>62.9</td>
<td>52.3</td>
<td>2.37</td>
<td>2.35</td>
<td>63.3</td>
<td>14.9</td>
<td>1.70</td>
<td>42.4</td>
<td>76.9</td>
</tr>
</tbody>
</table>
ConvLSTM improvements

- Untied first step
- Training on batches
- Training with variable sequence length
- Recurrent batch normalisation (Tim Cooijmans et al, 2016)

\[
\begin{array}{l}
\left( \begin{array}{c}
\tilde{f}_t \\
\tilde{i}_t \\
\tilde{o}_t \\
\tilde{g}_t \\
\end{array} \right) = \text{BN}(W_h h_{t-1}; \gamma_h, \beta_h) + \text{BN}(W_x x_t; \gamma_x, \beta_x) + b \\
c_t = \sigma(\tilde{f}_t) \odot c_{t-1} + \sigma(\tilde{i}_t) \odot \text{tanh}(\tilde{g}_t) \\
h_t = \sigma(\tilde{o}_t) \odot \text{tanh}(\text{BN}(c_t; \gamma_c, \beta_c))
\end{array}
\]
Conclusion

**Proposed:**
- Recurrent network for *self-supervised* motion understanding and next frame prediction
- Convolutional LSTM as artificial VSTM
- Application: weakly-supervised video segmentation with label propagation

**Next steps:**
- Motion drives attention
- Long(er)-term memory (e.g. for loop closure)