Topics of This Lecture

• Recap: CNNs for Video Analysis
  - Matching and correspondence estimation
    - Metric learning
    - Spatial Transformer Networks
    - Correspondence networks
  - Optical Flow Estimation
    - FlowNet
    - FlowNet2

Recap: Recurrent Networks

• Feed-forward networks
  - Simple neural network structure: 1-to-1 mapping of inputs to outputs
• Recurrent Neural Networks
  - Generalize this to arbitrary mappings

Recap: Long Short-Term Memory (LSTM)

- Inspired by the design of memory cells
- Each module has 4 layers, interacting in a special way.
- Effect: LSTMs can learn longer dependencies (~100 steps) than RNNs

Recap: Image Tagging

- Simple combination of CNN and RNN
  - Use CNN to define initial state \( h_0 \) of an RNN.
  - Use RNN to produce text description of the image.
Recap: Video to Text Description

Topics of This Lecture
- Recap: CNNs for Video Analysis
- Matching and correspondence estimation
  - Metric learning
    - Spatial Transformer Networks
  - Correspondence networks
- Optical Flow Estimation
  - FlowNet
  - FlowNet2

Learning Similarity Functions
- Siamese Network
  - Present the two stimuli to two identical copies of a network (with shared parameters)
  - Train them to output similar values if the inputs are (semantically) similar.
- Used for many matching tasks
  - Face identification
  - Stereo estimation
  - Optical flow
  - ...

Metric Learning: Contrastive Loss
- Mapping an image to a metric embedding space
  - Metric space: distance relationship = class membership

\[ \| f(x) - f(x_+) \| \to 0 \]
\[ \| f(x) - f(x_-) \| \geq m \]

Yi et al., LIFT: Learned Invariant Feature Transform, ECCV 16

Metric Learning: Triplet Loss
- Learning a discriminative embedding
  - Present the network with triplets of examples
  - Apply triplet loss to learn an embedding \( f(\cdot) \) that groups the positive example closer to the anchor than the negative one.

\[ \| f(x^+_{\text{positive}}) - f(x^-_{\text{negative}}) \|_2 < \| f(x^+_{\text{positive}}) - f(x^-_{\text{negative}}) \|_2 \]

\[ \Rightarrow \text{Used} \]

Patch Normalization with Spatial Transformer Nets
- Patch Normalization
  - Key component of local feature matching
  - Finding the scale and rotation
  - Invariant to perspective transformation

- Spatial Transformer Network
  - Adaptively apply transformation
Universal Correspondence Network

- Computing a patch descriptor

![Diagram](image1.png)

Universal Correspondence Network

- Siamese architecture for matching patches

![Diagram](image2.png)

Universal Correspondence Network

- UCN Training

![Diagram](image3.png)

Universal Correspondence Network

- Contrastive loss

\[ \|f(x_+) - f(x'_+))\| \to 0 \]
\[ \|f(x_-) - f(x'_-)\| > m \]

Semantic Correspondences with UCN

- Ground truth
- UCN
- VGG Conv4

Exact Correspondences with UCN (Disparity Estimation)

Topics of This Lecture

- Recap: CNNs for Video Analysis
- Matching and correspondence estimation
  - Metric learning
  - Spatial Transformer Networks
  - Correspondence networks
- Optical Flow Estimation
  - FlowNet
  - FlowNet2
Recap: Estimating Optical Flow

- **Optical Flow**
  - Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them.
- **Key assumptions**
  - Brightness constancy: projection of the same point looks the same in every frame.
  - Small motion: points do not move very far.
  - Spatial coherence: points move like their neighbors.

Recap: Iterative LK Refinement

- Estimate velocity at each pixel using one iteration of LK estimation.
- Warp one image toward the other using the estimated flow field.
- Refine estimate by repeating the process.
- Iterative procedure
  - Results in subpixel accurate localization.
  - Converges for small displacements.

Recap: Coarse-to-fine Optical Flow Estimation

- Run iterative LK
- Warp & upsample

CNNs for Optical Flow Estimation

- How can we achieve this with Deep Networks?
  - Intuition: need to match local image patches
  - CNNs can capture local context, so spatial smoothing should not be necessary
  - But iterative and coarse-to-fine estimation may be necessary.

FlowNet: FlowNetSimple Design

- Simple initial design
  - Simply stack two sequential images together and feed them through the network.
  - In order to compute flow, the network has to compare image patches
  - But it has to figure out on its own how to do that...
FlowNet: FlowNetCorr Design

- Correlation network
  - Central idea: compute a correlation score between two feature maps
    \[ c(x_1, x_2) = \sum_i (f_1(x_1 + \alpha_i), f_2(x_2 + \alpha_i)) \]
  - Then refine the correlation scores and turn them into flow predictions

FlowNet

- Flow refinement stage (both network designs)
  - After series of conv and pooling layers, the resolution has been reduced
  - Refine the coarse pooled representation by upconvolution layers (unpooling + upconvolution)
  - Skip connections to preserve high-res information from early layers

FlowNet: Training

- Training on FlyingChairs dataset
  - Synthetic dataset with known ground-truth
- Example prediction
  - Both networks can capture fine details

FlowNet: Comparing the two designs

<table>
<thead>
<tr>
<th>Method</th>
<th>Train/CE</th>
<th>Train Final</th>
<th>KITTI test</th>
<th>Middlebury test</th>
<th>Middlebury test</th>
<th>Train test</th>
<th>Test test</th>
<th>Time</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepFlow [11]</td>
<td>3.35</td>
<td>4.59</td>
<td>7.49</td>
<td>2.03</td>
<td>2.03</td>
<td>2.03</td>
<td>2.03</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EPF (1)</td>
<td>4.03</td>
<td>5.16</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6.2</td>
</tr>
<tr>
<td>FlowNetS [12]</td>
<td>4.50</td>
<td>5.49</td>
<td>8.64</td>
<td>2.70</td>
<td>2.70</td>
<td>2.70</td>
<td>2.70</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC [12]</td>
<td>4.50</td>
<td>5.49</td>
<td>8.64</td>
<td>2.70</td>
<td>2.70</td>
<td>2.70</td>
<td>2.70</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetS (ft)</td>
<td>5.08</td>
<td>6.07</td>
<td>8.05</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC (ft)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetS (v)</td>
<td>5.08</td>
<td>6.07</td>
<td>8.05</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC (v)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetS (ft+v)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC (ft+v)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetS (ft+ft)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC (ft+ft)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetS (ft+ft+v)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>FlowNetC (ft+ft+v)</td>
<td>4.98</td>
<td>5.97</td>
<td>8.35</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>2.23</td>
<td>0.2</td>
<td>6</td>
</tr>
</tbody>
</table>

FlowNet: Results

FlowNet: Learning Optical Flow with Convolutional Networks

FlowNet 2.0: Improved Design

- Stacked architecture
  - Several instances of FlowNetC and FlowNetS stacked together to estimate large-displacement flow
  - Sub-network specialized on small motions
  - Fusion layer
FlowNet 2.0: Detailed View

- Stacked FlowNets
  - Estimates large motion in a coarse-to-fine approach
  - Second image is warped at each level with the intermediate optical flow
  - Intermediate flow and (warped brightness) error are concatenated
  \[ \Rightarrow \text{Difficulty of the learning task is reduced at each level} \]

- Small Displacement Module and Fusion
  - For small displacements, FlowNet2-CSS is not accurate
  - Separate FlowNet2-SD module replaces 5x5 and 7x7 by multiple 3x3 kernels and assumes a stride 1 instead of stride 2 at the first layer
  - Small and simple network to fuse the outputs

Image source: Ilg et al., CVPR'17

FlowNet 2.0: Comparison

- Comparison (avg endpoint errors)
  - Similar accuracy as best pre-CNN methods (but much faster)

References and Further Reading

- RNNs

- LSTM
  - C. Olah, Understanding LSTM Networks, blog post, August 2015.

- Optical Flow