Computer Vision 2
WS 2018/19

Part 17 – CNNs for Video Analysis I
15.01.2019

Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de
Course Outline

• Single-Object Tracking
• Bayesian Filtering
• Multi-Object Tracking
• Visual Odometry
• Visual SLAM & 3D Reconstruction
  – Online SLAM methods
  – Full SLAM methods
• Deep Learning for Video Analysis
  – CNNs for video analysis
  – Optical flow
  – Video object segmentation
Topics of This Lecture

• Recap: Full SLAM methods

• CNNs for Video Analysis
  – Motivation
  – Example: Video classification

• CNN + RNN
  – RNN, LSTM
  – Example: Video captioning

• Matching and correspondence estimation
  – Metric learning
  – Correspondence networks
Recap: Full SLAM Approaches

• **SLAM graph optimization:**
  - Joint optimization for poses and map elements from image observations of map elements and control inputs

• **Pose graph optimization:**
  - Optimization of poses from relative pose constraints deduced from the image observations
  - Map recovered from the optimized poses
Pose Graph Optimization

• Optimization of poses
  – From relative pose constraints deduced from the image observations
  – Map recovered from the optimized poses

• Deduce relative constraints between poses from image observations, e.g.,
  – 8-point algorithm
  – Direct image alignment
Pose Graph Optimization Example

Dense Visual SLAM for RGB-D Cameras

Christian Kerl, Jürgen Sturm, Daniel Cremers

Computer Vision and Pattern Recognition Group
Department of Computer Science
Technical University of Munich

Kerl et al., Dense Visual SLAM for RGB-D Cameras, IROS 2013
Topics of This Lecture

• Recap: Full SLAM methods

• **CNNs for Video Analysis**
  – Motivation
  – Example: Video classification

• **CNN + RNN**
  – RNN, LSTM
  – Example: Video captioning

• Matching and correspondence estimation
  – Metric learning
  – Correspondence networks
Video Analysis with CNNs

• Modeling perspective
  – What **architecture** to use to best capture temporal patterns?

• Computational perspective
  – Video processing is expensive!
  – How to reduce **computation cost** without sacrificing accuracy
Large-Scale Video Classification with CNNs

- **Architecture**
  - Different ways to fuse features from multiple frames

---

**Single Frame**

- Conv layer

**Late Fusion**

- Conv layer
- Norm layer

**Early Fusion**

- Conv layer
- Norm layer
- Pooling layer

**Slow Fusion**

- Conv layer
- Norm layer
- Pooling layer

*Image source: Andrej Karpathy*
Large-Scale Video Classification with CNNs

- Computational cost
  - Reduce spatial dimension to reduce model complexity
  - Multi-resolution: low-res context + high-res foveate

Image source: Andrej Karpathy
Topics of This Lecture

• Recap: Full SLAM methods

• CNNs for Video Analysis
  – Motivation
  – Example: Video classification

• **CNN + RNN**
  – RNN, LSTM
  – Example: Video captioning

• Matching and correspondence estimation
  – Metric learning
  – Correspondence networks
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: RNNs

- RNNs are regular NNs whose hidden units have additional forward connections over time.
  - You can **unroll** them to create a network that extends over time.
  - When you do this, keep in mind that the weights for the hidden units are shared between temporal layers.
Extension: Long Short-Term Memory (LSTM)

- **LSTMs**
  - 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

---

Image source: Christopher Olah, [http://colah.github.io/posts/2015-08-Understanding-LSTMs/](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
Recap: RNNs for Text Generation

- RNN for text generation

10,001D class scores (Softmax over 10k words and a special <END> token)

\[ y_4 = W_{hy} h_4 \]

Hidden layer (e.g., 500D vectors)

\[ h_4 = \max \{ 0, W_{xh} x_4 + W_{hh} h_3 \} \]
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.

- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.

- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]
Recap: RNNs for Text Generation

• Training this on a lot of sentences would give us a language model.

• I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]

Slide credit: Andrej Karpathy, Fei-Fei Li
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.

- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]

Slide credit: Andrej Karpathy, Fei-Fei Li
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.
- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.

- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.
- I.e., a way to predict

\[ p(\text{next word} | \text{previous words}) \]
Recap: RNNs for Text Generation

• Training this on a lot of sentences would give us a language model.

• I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]

Slide credit: Andrej Karpathy, Fei-Fei Li
Recap: RNNs for Text Generation

- Training this on a lot of sentences would give us a language model.

- I.e., a way to predict

\[ p(\text{next word} \mid \text{previous words}) \]
Applications: 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.
Applications: Image Tagging

• Setup
  – Train on corpus of images with textual descriptions
  – E.g. Microsoft CoCo
    ▪ 120k images
    ▪ 5 sentences each
Results: Image Tagging

Spectacular results!

- A group of people standing around a room with remotes
  logprob: -9.17

- A young boy is holding a baseball bat
  logprob: -7.61

- A cow is standing in the middle of a street
  logprob: -8.84
Results: Image Tagging

- Wrong, but one can still see why those results were selected...
Application: Video to Text Description

Our LSTM network is connected to a CNN for RGB frames or a CNN for optical flow images.

Flow images

CNN - Action pretrained

CNN Outputs

LSTMs

Raw Frames

CNN - Object pretrained

A

man

is

cutting

a

bottle

<eos>

Source: Subhashini Venugopalan, ICCV'15
Video-to-Text Results

Correct descriptions.

S2VT: A man is doing stunts on his bike.

2VT: A herd of zebras are walking in a field.

S2VT: A young woman is doing her hair.

S2VT: A man is shooting a gun at a target.

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

S2VT: A man is cutting a piece of a pair of a paper.

S2VT: A cat is trying to get a small board.

S2VT: A man is spreading butter on a tortilla.

Irrelevant descriptions.

S2VT: A man is pouring liquid in a pan.

S2VT: A polar bear is walking on a hill.

S2VT: A man is doing a pencil.

S2VT: A black clip to walking through a path.

Source: Subhashini Venugopalan, ICCV’15
Topics of This Lecture

• Recap: Full SLAM methods

• CNNs for Video Analysis
  – Motivation
  – Example: Video classification

• CNN + RNN
  – RNN, LSTM
  – Example: Video captioning

• Matching and correspondence estimation
  – Metric learning
  – Correspondence networks
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
  – …

- 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.

\[ \| f(x_i^a) - f(x_i^p) \|_2^2 < \| f(x_i^a) - f(x_i^n) \|_2^2 \]

⇒ Used
• 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
Universal Correspondence Network

- Siamese architecture for matching patches

\[ I \]

\[ I' \]
Universal Correspondence Network

• UCN Training

\[ f(x_+) \]
\[ f(x'_+) \]
\[ f(x_-) \]
\[ f(x'_-) \]

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

• Contrastive loss

Slide credit: Christopher Choy
Semantic Correspondences with UCN

Ground truth

UCN

VGG Conv4

Slide credit: Christopher Choy
Exact Correspondences with UCN (Disparity Estimation)

C. Choy, J.Y. Gwak, S. Savarese, M. Chandraker, Universal Correspondence Network, NIPS’16
References and Further Reading

• RNNs

• LSTM