Video Object Segmentation using Deep Learning

Update Presentation, Week 3

Zack While

Advised by: Rui Hou, Dr. Chen Chen, and Dr. Mubarak Shah

June 2, 2017

Youngstown State University

Table of Contents

1. Previous Work

2. Current Work

3. Upcoming Work

Previous Work

Literature Review

Read the following:

- *Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos* by Hou, Chen, & Shah (arXiv, April 2017)
- *Learning Spatiotemporal Features with 3D Convolutional Networks* by Tran et al. (arXiv, October 2015)
- *The 2017 DAVIS Challenge on Video Object Segmentation* by Pont-Tuset et al. (arXiv, April 2017)
- *FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos* by Jain, Xiong, & Grauman (arXiv, April 2017)
- *Semantically-Guided Video Object Segmentation* by Perazzi et al. (arXiv, December 2016)
- *Learning Video Object Segmentation from Static Images* by Caelles et al. (arXiv, April 2017)

Suggestions

- Focusing on semantic segmentation, using the **DAVIS 2017** dataset to start.

- Working on the implementation.

Suggestions

- Focusing on semantic segmentation, using the **DAVIS 2017** dataset to start.

- Working on the implementation.

Current Work

Goals

- Finishing the initial literature review.
- Gaining a deeper understanding of the *SegNet*, *T-CNN*, and *C3D* papers.
- Understanding the current implementation.
- Becoming familiar with the **DAVIS 2017** dataset.

Goals

- Finishing the initial literature review.
- Gaining a deeper understanding of the *SegNet*, *T-CNN*, and *C3D* papers.
- Understanding the current implementation.
- Becoming familiar with the DAVIS 2017 dataset.

Goals

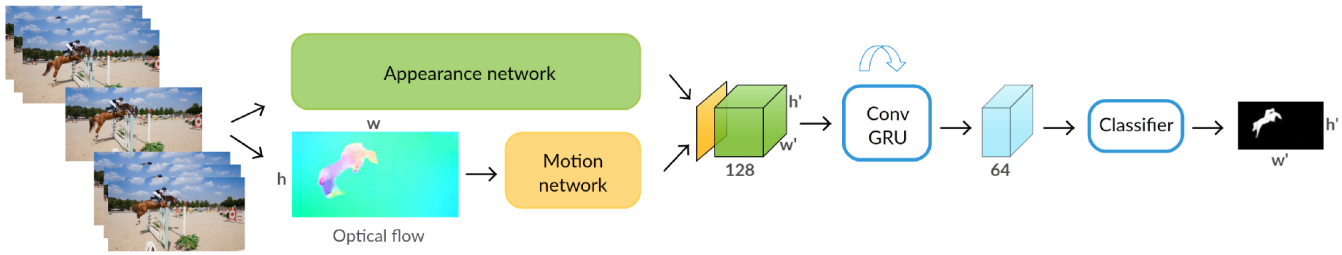
- Finishing the initial literature review.
- Gaining a deeper understanding of the *SegNet*, *T-CNN*, and *C3D* papers.
- Understanding the current implementation.
- Becoming familiar with the **DAVIS 2017** dataset.

Goals

- Finishing the initial literature review.
- Gaining a deeper understanding of the *SegNet*, *T-CNN*, and *C3D* papers.
- Understanding the current implementation.
- Becoming familiar with the **DAVIS 2017** dataset.

Literature Review

- ***Learning Video Object Segmentation with Visual Memory*** by Tokmakov et al. (arXiv, April 2017)
  - Uses a GRU as the *memory module* which continuously learns the appearance of the object(s) in the scene.



- ***A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation*** by Perazzi et al. (CVPR, 2016)
  - **DAVIS 2016** paper
  - More detail in describing metrics, necessity of dataset

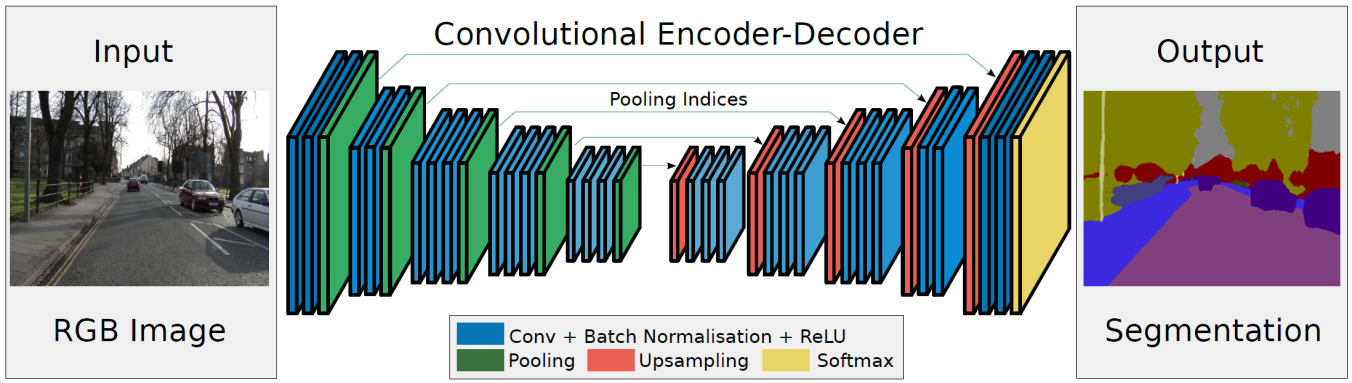
Literature Review

- *Learning Video Object Segmentation with Visual Memory* by Tokmakov et al. (arXiv, April 2017)
  - Uses a GRU as the *memory module* which continuously learns the appearance of the object(s) in the scene.

- *A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation* by Perazzi et al. (CVPR, 2016)
  - **DAVIS 2016** paper
  - More detail in describing metrics, necessity of dataset

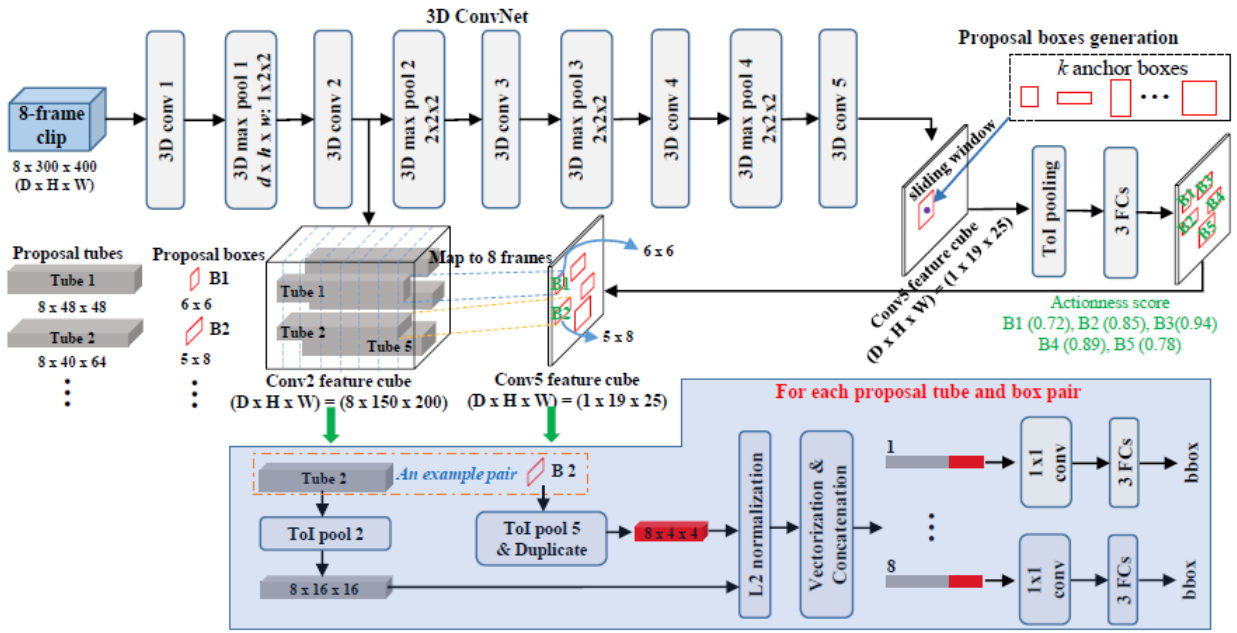
SegNet

- *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* by Badrinarayanan et al. (arXiv, Oct. 2016)
  - Encoder network is first 13 convolutional layers from VGG-16
  - Uses pooling indices from max-pooling step when upsampling



T-CNN and C3D

- *Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos* by Hou, Chen, & Shah (arXiv, April 2017)
  - Uses C3D to gain spatio-temporal information.

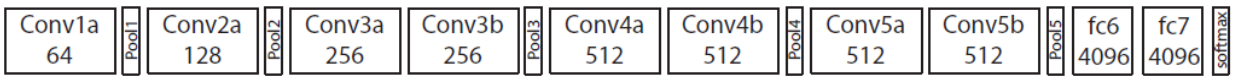


- *Learning Spatiotemporal Features with 3D Convolutional Networks* by Tran et al. (arXiv, October 2015)
  - Better than 2D CNN architectures on various benchmarks.

T-CNN and C3D

- *Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos* by Hou, Chen, & Shah (arXiv, April 2017)
  - Uses C3D to gain spatio-temporal information.

- ***Learning Spatiotemporal Features with 3D Convolutional Networks*** by Tran et al. (arXiv, October 2015)
  - Better than 2D CNN architectures on various benchmarks.



Implementation

So far, there are 4 core files:

1. dataset_input.py
    - Reads/decodes data from TFRecords file and provides data batches.
2. jhmdb_iter_size_train.py
    - Trains the JHMDB (*Joint-annotated Human Motion Data Base*) dataset using iter_size, originally from Caffe.
3. jhmdb_to_records.py
    - Converts items in JHMDB dataset to the TFRecords file format.
4. train_net.py
    - Implements the C3D pipeline but replaces fully-connected layers with convolution-transpose layers for semantic segmentation.
    - Also provides visual summaries for activation functions, loss functions, etc.

Implementation

So far, there are 4 core files:

1. dataset_input.py
   - Reads/decodes data from TFRecords file and provides data batches.
2. jhmdb_iter_size_train.py
   - Trains the JHMDB (*Joint-annotated Human Motion Data Base*) dataset using iter_size, originally from Caffe.
3. jhmdb_to_records.py
   - Converts items in JHMDB dataset to the TFRecords file format.
4. train_net.py
   - Implements the C3D pipeline but replaces fully-connected layers with convolution-transpose layers for semantic segmentation.
   - Also provides visual summaries for activation functions, loss functions, etc.

Implementation

So far, there are 4 core files:

1. dataset_input.py
   - Reads/decodes data from TFRecords file and provides data batches.
2. jhmdb_iter_size_train.py
   - Trains the JHMDB (*Joint-annotated Human Motion Data Base*) dataset using iter_size, originally from Caffe.
3. jhmdb_to_records.py
   - Converts items in JHMDB dataset to the TFRecords file format.
4. train_net.py
   - Implements the C3D pipeline but replaces fully-connected layers with convolution-transpose layers for semantic segmentation.
   - Also provides visual summaries for activation functions, loss functions, etc.

Implementation

So far, there are 4 core files:

1. dataset_input.py
   - Reads/decodes data from TFRecords file and provides data batches.
2. jhmdb_iter_size_train.py
   - Trains the JHMDB (*Joint-annotated Human Motion Data Base*) dataset using iter_size, originally from Caffe.
3. jhmdb_to_records.py
   - Converts items in JHMDB dataset to the TFRecords file format.
4. train_net.py
   - Implements the C3D pipeline but replaces fully-connected layers with convolution-transpose layers for semantic segmentation.
   - Also provides visual summaries for activation functions, loss functions, etc.

DAVIS 2017 Dataset

Offered in both 480p and full-resolution, the set includes:

- Folder of annotations for the first frame of each sequence
- .txt files of each video sequence's label
- Folders of the sequence frames

Upcoming Work

Plan for Next Week

- Setting up the current code to work on the DAVIS 2017 dataset.
- Reading related material and doing tutorials as needed.

Plan for Next Week

- Setting up the current code to work on the DAVIS 2017 dataset.
- Reading related material and doing tutorials as needed.

Questions?