Video Object Segmentation using Deep Learning

Update Presentation, Week 4

Zack While

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

June 9, 2017

Youngstown State University

Table of Contents

1. Previous Work

2. Current Work

3. Upcoming Work

Previous Work

Literature Review

- *Learning Video Object Segmentation with Visual Memory* by Tokmakov et al. (arXiv, April 2017)
- *A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation* by Perazzi et al. (CVPR, 2016)
- *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* by Badrinarayanan et al. (arXiv, Oct. 2016)
- *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)

Implementation

1. dataset_input.py
2. jhmdb_iter_size_train.py
3. jhmdb_to_records.py
4. train_net.py

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

Current Work

Goals

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

Goals

- Setting up the current code to work on the DAVIS 2017 dataset.

- Reading related material and doing tutorials as needed.

Converting to `.tfrecords`

- Contained in `davis-train_to_records.py`
- As of now, the training, validation, & test sets are separate files.
- Not all clips are the same resolution.
- Folders are set up differently from the JHMDB video set.
- DAVIS masks are a folder of `.png` files instead of a single `.mat` file (per clip).

Converting to `.tfrecords`

- Contained in `davis-train_to_records.py`
- As of now, the training, validation, & test sets are separate files.
- Not all clips are the same resolution.
- Folders are set up differently from the JHMDB video set.
- DAVIS masks are a folder of `.png` files instead of a single `.mat` file (per clip).

Converting to `.tfrecords`

- Contained in `davis-train_to_records.py`
- As of now, the training, validation, & test sets are separate files.
- Not all clips are the same resolution.
- Folders are set up differently from the JHMDB video set.
- DAVIS masks are a folder of `.png` files instead of a single `.mat` file (per clip).

Converting to `.tfrecords`

- Contained in `davis-train_to_records.py`
- As of now, the training, validation, & test sets are separate files.
- Not all clips are the same resolution.
- Folders are set up differently from the JHMDB video set.
- DAVIS masks are a folder of `.png` files instead of a single `.mat` file (per clip).

Converting to `.tfrecords`

- Contained in `davis-train_to_records.py`
- As of now, the training, validation, & test sets are separate files.
- Not all clips are the same resolution.
- Folders are set up differently from the JHMDB video set.
- DAVIS masks are a folder of `.png` files instead of a single `.mat` file (per clip).

Checking the `.tfrecords` Conversion (1)

- Need to calculate the **mean_frame** using `davis_480p_compute_mean_frame.py`.
  - Requires all the input frames to be the exact same resolution.
  - Now all images are converted to 240×320 in the beginning.
- Making sure `dataset_input.py` is correctly formatted for the new `.tfrecords` file.
- Setting up `davis_480p_test_input.py` to run correctly.

Checking the `.tfrecords` Conversion (1)

- Need to calculate the **mean_frame** using `davis_480p_compute_mean_frame.py`.
  - Requires all the input frames to be the exact same resolution.
  - Now all images are converted to 240×320 in the beginning.
- Making sure `dataset_input.py` is correctly formatted for the new `.tfrecords` file.
- Setting up `davis_480p_test_input.py` to run correctly.

Checking the `.tfrecords` Conversion (1)

- Need to calculate the **mean_frame** using `davis_480p_compute_mean_frame.py`.
  - Requires all the input frames to be the exact same resolution.
  - Now all images are converted to 240×320 in the beginning.

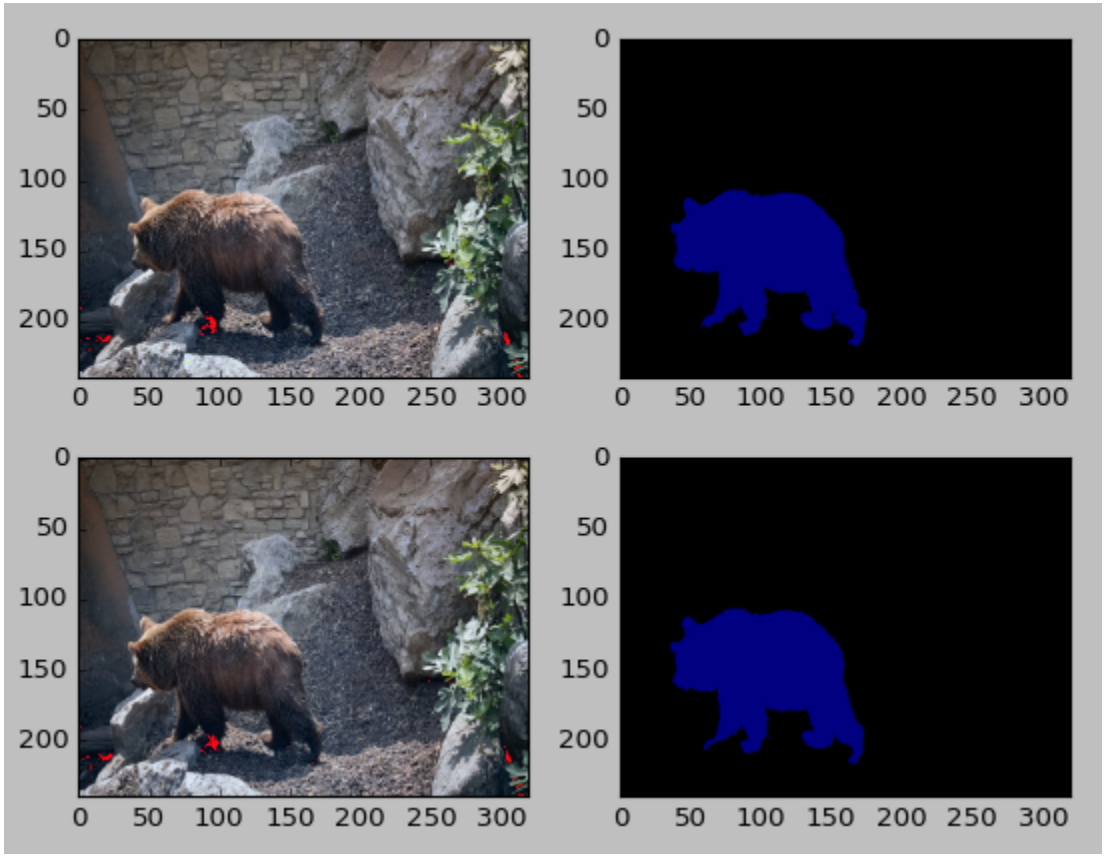
- Making sure `dataset_input.py` is correctly formatted for the new `.tfrecords` file.

- Setting up `davis_480p_test_input.py` to run correctly.

Checking the `.tfrecords` Conversion (1)

- Need to calculate the **mean_frame** using `davis_480p_compute_mean_frame.py`.
  - Requires all the input frames to be the exact same resolution.
  - Now all images are converted to 240×320 in the beginning.
- Making sure `dataset_input.py` is correctly formatted for the new `.tfrecords` file.
- Setting up `davis_480p_test_input.py` to run correctly.

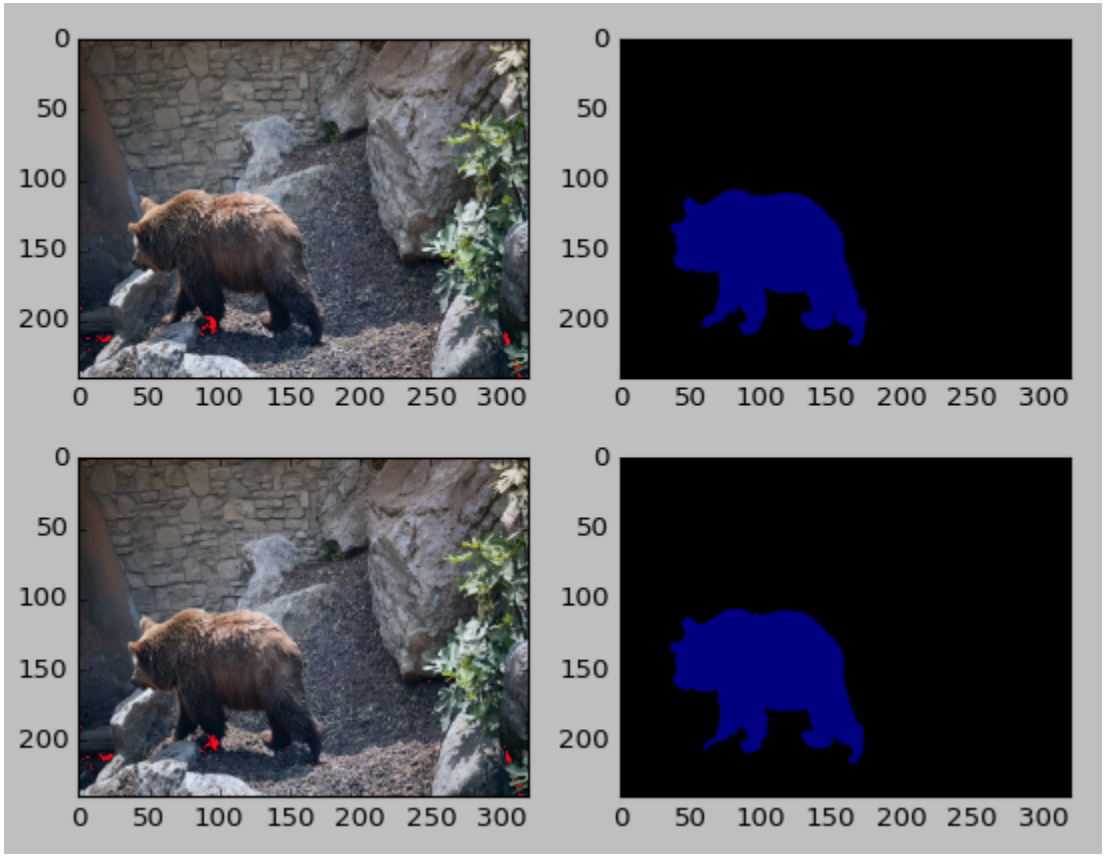
Checking the `.tfrecords` Conversion (2)



- Red spots due to conversion from **np.float32** to **np.uint8** in `davis_480p_test_input.py` file.

- Can be further extended to check other photos.

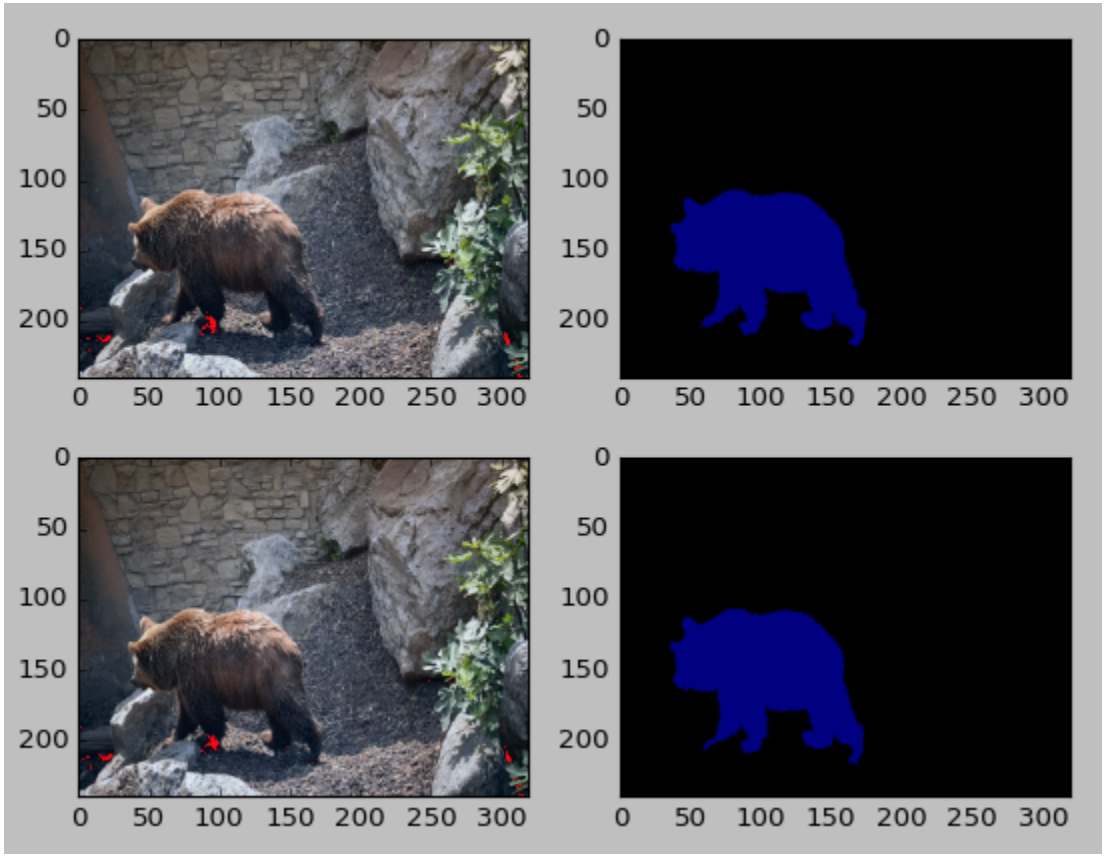
Checking the `.tfrecords` Conversion (2)



- Red spots due to conversion from **np.float32** to **np.uint8** in `davis_480p_test_input.py` file.

- Can be further extended to check other photos.

Checking the `.tfrecords` Conversion (2)



- Red spots due to conversion from **np.float32** to **np.uint8** in `davis_480p_test_input.py` file.

- Can be further extended to check other photos.

Upcoming Work

Plan for Next Week

- Using the new `.tfrecords` with some basic models (to be decided).
- Updating the previously-mentioned files as needed.

Plan for Next Week

- Using the new `.tfrecords` with some basic models (to be decided).
- Updating the previously-mentioned files as needed.

Questions?