kmaninis / **OSVOS-caffe**



One-Shot Video Object Segmentation http://vision.ee.ethz.ch/~cvlsegmenta…

**15** commits **1** branch **0** releases **2** contributors



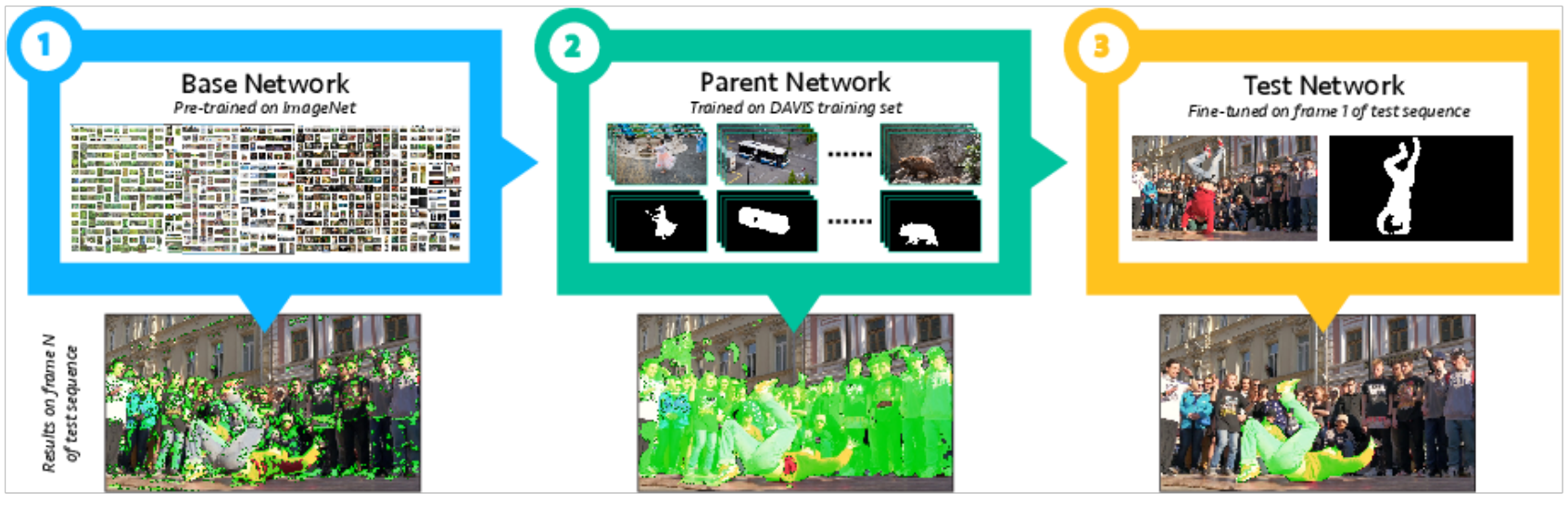
kmaninis parent net small bug fix Latest commit 368862a 11 days ago

| | | |
|---|---|---|
| DAVIS | first commit | 4 months ago |
| caffe-osvos | first commit | 4 months ago |
| doc/ims | first commit | 4 months ago |
| lib | first commit | 4 months ago |
| models | - bug fix | 3 months ago |
| src | parent net small bug fix | 11 days ago |
| .gitignore | gitignore | 4 months ago |
| README.md | clarify DAVIS -> DAVIS 2016 | 2 months ago |
| build.m | Small fix for c++11 compilation in some computers | 3 months ago |
| demo.m | minor change | 4 months ago |
| osvos_root.m | first commit | 4 months ago |
| set_params.m | first commit | 4 months ago |
| test_all.m | first commit | 4 months ago |

**README.md**

# One-Shot Video Object Segmentation (OSVOS)

Visit our project page for accessing the paper, and the pre-computed results.




This is the implementation of our work `One-Shot Video Object Segmentation (OSVOS)`, for semi-supervised video object segmentation. OSVOS is based on a fully convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object of the test sequence (hence one-shot). Experiments on DAVIS 2016 show that OSVOS is faster than currently available techniques and improves the state of the art by a significant margin (79.8% vs 68.0%).

While the results of the paper were obtained by this code, we also provide a TensorFlow implementation of OSVOS:

OSVOS-TensorFlow.

## Installation:

1. Clone the OSVOS-caffe repository

```
git clone https://github.com/kmaninis/OSVOS-caffe.git
```

2. Install the Caffe version under `caffe-osvos/` along with standard dependencies, pycaffe and matcaffe. Caffe would need to be built with support for Python layers, in case you would like to use the Python API *(TODO)*. cuDNN is not necessary.

```
# In your Makefile.config, make sure to have this line uncommented
WITH_PYTHON_LAYER := 1
```

3. Download the parent model from here (55 MB) and put it under `models/` .
4. Optionally download the contour model for contour snapping from here (55 MB) and put it under `models/` .
5. If you want to use the contour snapping step (a.k.a you downloaded the model of step 4.), run `build.m` from within MATLAB.
6. All the steps to re-train OSVOS are provided in this repository. In case you would like to test with the pre-trained models, you can download them from here (1GB) and put it under `models/` .

## Demo online training and testing

1. Edit in file `set_params.m` the parameters of the code (eg. useGPU, gpu_id, etc.).
2. Run `demo.m` .
3. You can test all sequences of DAVIS 2016 validation set, by running `test_all.m` , once the pre-trained models are available under `models/` .

It is possible to work with all sequences of DAVIS 2016 just by creating a soft link ( `ln -s /path/to/DAVIS/` ) in the root folder of the project.

## Training the parent network (optional)

1. All necessary files are under `src/parent` . So, `cd src/parent` .
2. Download the pre-trained vgg model by running `./download_pretrained_vgg.sh`
3. Augment the data. In the paper we used flipping and scaling into 0.5, 0.8 and 1.0 of the original scale. Your image and ground truth pairs are specified in `solvers/train_pair.txt` .
4. Under `solvers` edit the `data_root_dir` of `train_val*.prototxt` .
5. Finally, train the parent model with `python solve_cluster.py` . You need pycaffe for this step, so don't forget to `make pycaffe` when installing Caffe.

Enjoy! :)

## Citation

If you use this code, please consider citing the following paper:

```
@Inproceedings{Cae+17,
  Title          = {One-Shot Video Object Segmentation},
  Author         = {S. Caelles and K.K. Maninis and J. Pont-Tuset and L. Leal-Taix\'e and D. Cremers and L.
  Booktitle      = {Computer Vision and Pattern Recognition (CVPR)},
  Year           = {2017}
}
```

If you encounter any problems with the code, want to report bugs, etc. please contact me at kmaninis[at]vision[dot]ee[dot]ethz[dot]ch.