A Deep Learning Based Image Segmentation Tool

Ali Emre Kavur
emrekavur@gmail.com
17/05/2019

I. Introduction

This document explains how to use this tool that adopts a pretrained deep learning model for segmentation of colour images. This tool needs both Python and Matlab installations on the computer. Also, some additional Python libraries are necessary.

The pretrained model is selected as DeepLabv3+ [1] which is a TensorFlow model. It is open-source and shared on Github [2]. DeepLab: Deep Labelling for Semantic Image Segmentation [3], [4] is a state of the art deep learning model for semantic image segmentation, where the goal is to assign semantic labels to every pixel in the input image.

The selected model is based on MobileNetV2 [5] architecture and is pretrained with Microsoft COCO dataset [6]. The model can segment and identify the objects: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, and tv.

II. Installation

The main file of the tool is a MATLAB script. It is assumed that MATLAB is already installed on the system. Any version of MATLAB after R2015a can be used. In addition, the following installation process for Windows 10 operating system is explained below:

1) Download the latest 64-bit version of Python 3.X from `www.python.org/downloads/windows/` website (Download *Windows x86-64 executable installer* under the "Stable Releases" section)
2) On the install window, make the changes below;
   a) Select `Add Python 3.X to PATH`
   b) Click on `Customize installation` and select `pip`
   c) Click on next button and select `Install for all users`
   d) On `customize install location` line write `C:\python3X`
3) After installation is completed, open a command window and check your Python version with `python --version` command. If the output is *"Python 3.X.Y"*, Python is installed correctly. If you get *" 'python' is not recognized as an internal or external command, operable program or batch file."* message, there is a problem with Python installation.
4) Check the pip installation by executing `pip --version` command. If the output is similar with *"pip X from..."*, pip installed correctly.
5) Now it is possible to install the necessary Python libraries with the code below: (Run these lines separately.)
   - `pip install numpy`
   - `pip install scipy`
   - `pip install Pillow`
   - `pip install tensorflow`
6) Note: If you have CUDA-compatible NVIDIA graphic card, you may install Tensorflow-GPU. However it is necessary to follow the instructions on TersorFlow web page for installing.

III. Usage

The tool is packed in *"semantic-segmenter-tool-master.zip"*. It includes the following files:

- `segment_image.m` (main file)
- `segmenter.py`
- `deeplabv3_mnv2...2018_01_29.tar.gz`
- `autoArrangeFigures.m`

Unpack zip file to any location in the computer. Open MATLAB and navigate to the chosen folder. The main file of the tool is ready to run. The main file of the tool is `segment_image.m` which is called from MATLAB. The only necessary modification is to define the name of the input image in the script. After that the script can be run. The working principle of the tool is illustrated in Fig.1

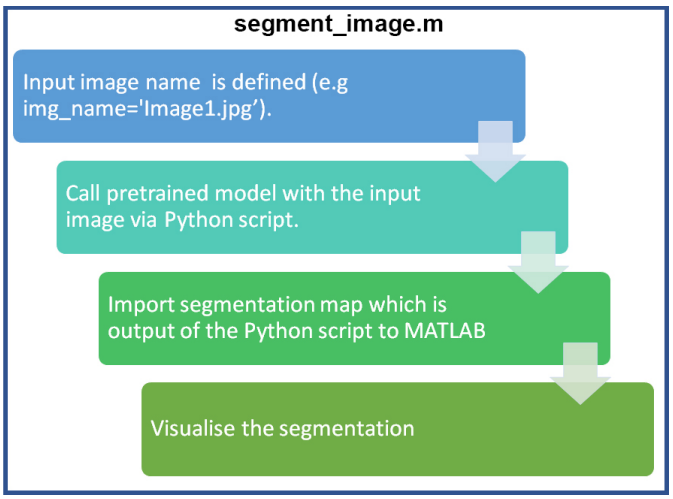

Fig. 1. Working principle of the tool.

Fig. 2. Example of a segmentation task. Original image at the left, segmentation map at the middle, visualisation is at the right.

`python segmenter.py "ImageName.jpg"` command will be called from script. If the segmentation is successful, the segmentation map will be saved into `seg_map.mat` file. This file will be imported to MATLAB. After that, the segmentation map will be colourized by the defined colour map. The segmentation will be visualised on to the original image. Label(s) of the detected object(s) will be determined. Finally, the original image and the segmented image with its labels will be shown on the screen. An example output of the segmentation tool is presented in Fig.2.

REFERENCES

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation" in arXiv: 1706.05587, 2017.

[2] https://github.com/tensorflow/models/tree/master/research/deeplab

[3] Liang-Chieh Chen*, George Papandreou*, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs" in arXiv:1606.00915, 2016

[4] Liang-Chieh Chen+, George Papandreou+, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected " in ICLR 2015, arXiv:1412.7062, 2015.

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks" in https://arxiv.org/abs/1801.04381, 2018

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, "Microsoft COCO: Common Objects in Context" in https://arxiv.org/abs/1405.0312, 2014