

Train Lips Semantic Segmentation

Tensorflow Deeplab

<https://colab.research.google.com/drive/1yK6pYHecRbzLZ4Uflv59d-ajOJ6DxrGr?authuser=3#scrollTo=o0lpbTcpxfzS>

1. Check python version

- Method 1 (Deprecated as of December 17, 2022)

```
# Check python version
# It should be 3.7
# If it is 3.8, go to "Tools" > "Command Palette" > and search for "Use Fallback Runtime Version"
!pip --version

pip 21.1.3 from /usr/local/lib/python3.8/dist-packages/pip (python 3.8)
```

Figure 1.1: Wrong Python Version

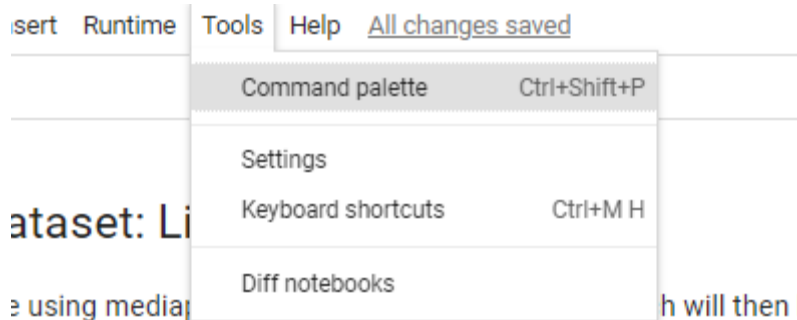


Figure 1.2: Go to Tools > Command palette

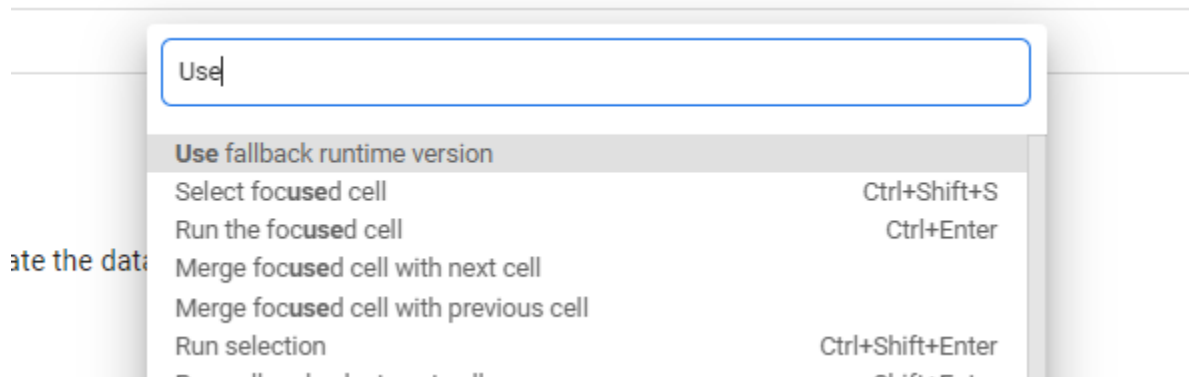


Figure 1.3: Search for "Use Fallback Runtime Version"

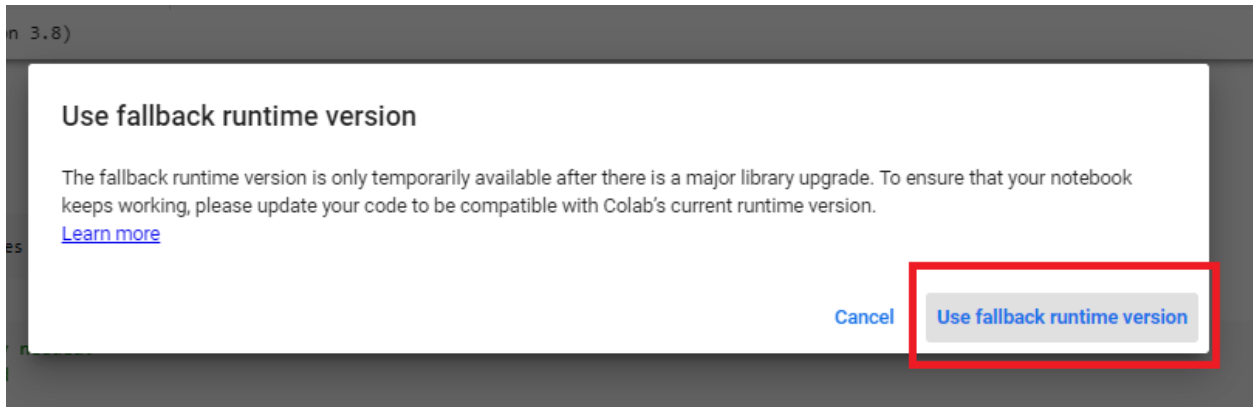


Figure 1.4: Use Fallback Runtime Version

```

# Check python version
# It should be 3.7
# If it is 3.8, go to "Tools" > "Command Palette" > and search for "Use Fallback Runtime Version"
!pip --version

pip 21.1.3 from /usr/local/lib/python3.7/dist-packages/pip (python 3.7)

```

Figure 1.5: Check the python version again

- Method 2 (Made on December 17, 2022)

change python interpreter (Working)

```

#install python 3.9 and dev utils
#you may not need all the dev libraries, but I haven't tested which aren't necessary.
!sudo apt-get update -y
!sudo apt-get install python3.7 python3.7-dev python3.7-distutils libpython3.7-dev

#change alternatives
!sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.8 1
!sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 2

# install pip
!curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
!python3 get-pip.py --force-reinstall

#install colab's dependencies
!python3 -m pip install ipython ipython_genutils ipykernel jupyter_console prompt_toolkit httplib2 astor

# link to the old google package
!ln -s /usr/local/lib/python3.8/dist-packages/google \
    /usr/local/lib/python3.7/dist-packages/google

Hit:7 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:10 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Get:11 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:13 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease

```

Figure 1.6: Install python 3.7 and its dependencies on google colab. It will also copy google dependencies from the default runtime python 3.8

```
#check python version
import sys
print(sys.version)
!python3 --version
!python --version
```

```
3.8.16 (default, Dec  7 2022, 01:12:13)
[GCC 7.5.0]
Python 3.7.16
Python 3.7.16
```

```
!python --version
#Python 3.7.15
```

```
Python 3.7.16
```

```
!pip --version
```

```
pip 22.3.1 from /usr/local/lib/python3.7/dist-packages/pip (python 3.7)
```

Figure 1.7: Check result

```
!pip install -q --upgrade ipython
!pip install -q --upgrade ipykernel
```

```
WARNING: Running pip as the 'root' user can result in |
WARNING: Running pip as the 'root' user can result in |
```

Figure 1.8: Install related packages (After running this cell, you should restart the runtime)

2. Install necessary packages and clone tensorflow repository

```
!git clone https://github.com/tensorflow/models.git

Cloning into 'models'...
remote: Enumerating objects: 79721, done.
remote: Counting objects: 100% (427/427), done.
remote: Compressing objects: 100% (233/233), done.
remote: Total 79721 (delta 239), reused 362 (delta 190), pack-reused 79294
Receiving objects: 100% (79721/79721), 594.12 MiB | 22.48 MiB/s, done.

!pip install tf-slim==1.0
!pip install tensorflow==1.15
```

Figure 2.1: Cloning and installation

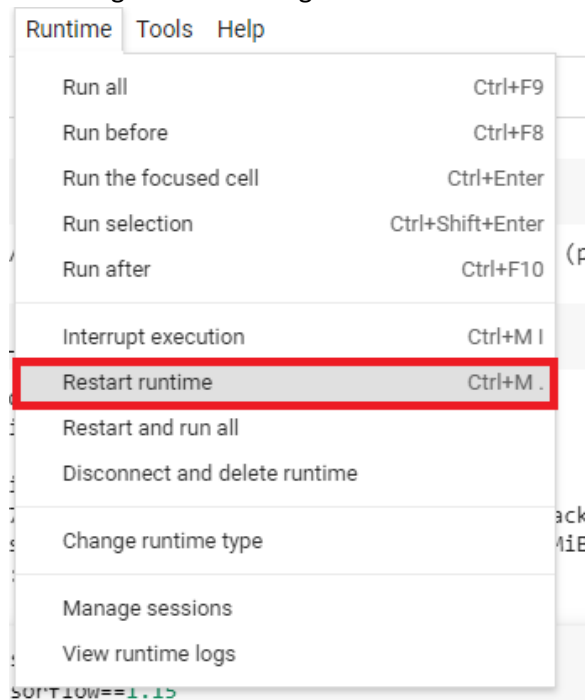


Figure 2.2: Restart runtime after installing tensorflow 1.15

3. Initialization

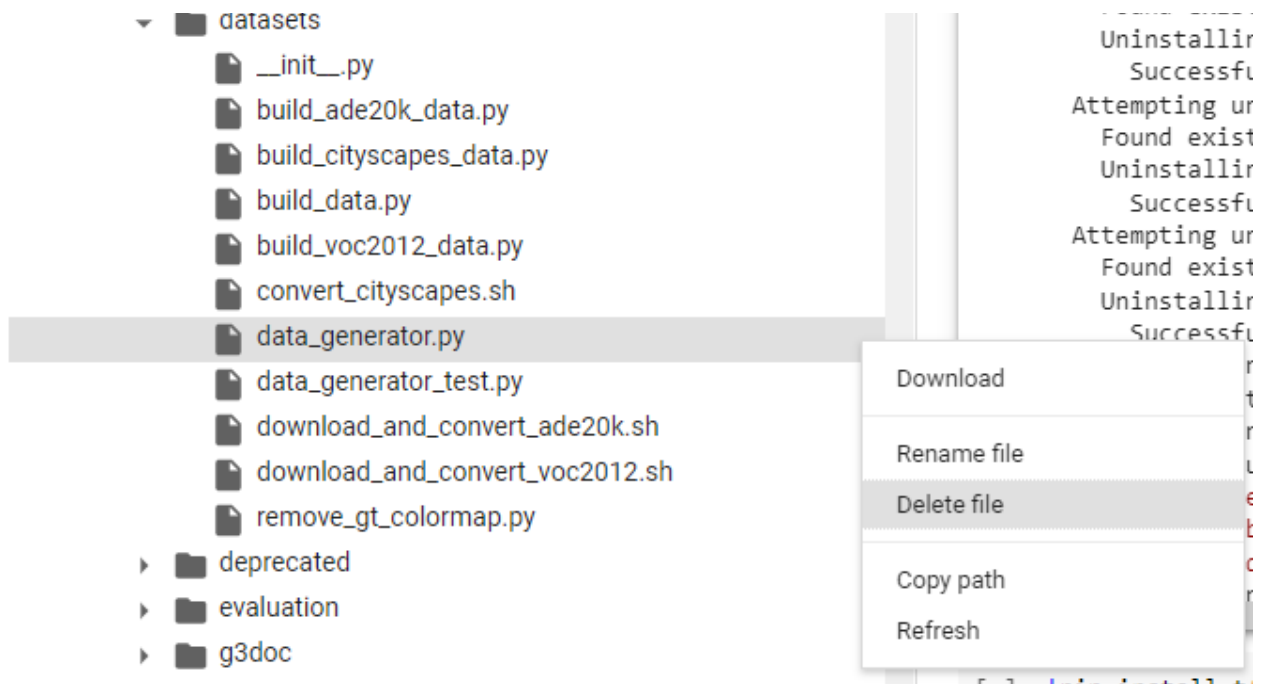


Figure 3.1: Delete data_generator
(/content/models/research/deeplab/datasets/data_generator.py)

```
[1] import os

import tensorflow as tf
from PIL import Image
from tqdm import tqdm
import numpy as np

import os, shutil

from google.colab import drive
drive.mount('/content/drive', force_remount= True)

%env PYTHONPATH=/env/python/:/content/models/research/:/content/models/research/slim:/content/models/research/deeplab
```

Figure 3.2: Import packages, mount gdrive, and set paths

```
[ ] !wget http://download.tensorflow.org/models/deeplab_mv3_small_cityscapes_trainfine_2019_11_15.tar.gz

--2022-12-15 00:00:35-- http://download.tensorflow.org/models/deeplab_mv3_small_cityscapes_trainfine_2019_11_15.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 74.125.204.128, 2404:6800:4008:c04::80
Connecting to download.tensorflow.org (download.tensorflow.org)|74.125.204.128|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5400383 (5.1M) [application/x-tar]
Saving to: 'deeplab_mv3_small_cityscapes_trainfine_2019_11_15.tar.gz'

deeplab_mv3_small_ 100%[=====] 5.15M 11.7MB/s in 0.4s

2022-12-15 00:00:37 (11.7 MB/s) - 'deeplab_mv3_small_cityscapes_trainfine_2019_11_15.tar.gz' saved [5400383/5400383]

[ ] !tar -xf /content/deeplab_mv3_small_cityscapes_trainfine_2019_11_15.tar.gz
```

Figure 3.3: If you want to use tensorflow's pretrained model, you can run this (Optional)

```
# Take the data_generator.py that I have modified
!cp /content/drive/SharedDrives/MachineLearning/Machine_Learning/Lips_Datasets/Lips_Dataset_257/data_generator.py /content/models/research/deeplab/datasets
```

Figure 3.4: Copy the data_generator.py (Note: If you have a different dataset, you should modify this first)

```
#bk 2
# To make Backup, run this. Don't forget to change folder names to makes
# sure that the files will not be replaced.
!cp -r /content/drive/MyDrive/PQR_Checkpoints_257/mnv3_small_170k_pretrained_b4_lr0000125_adam /content/drive/MyDrive/PQR_Checkpoints_257/backups
```

Figure 3.5: Create backups (Optional) (Note: Modify this depending on your directory)

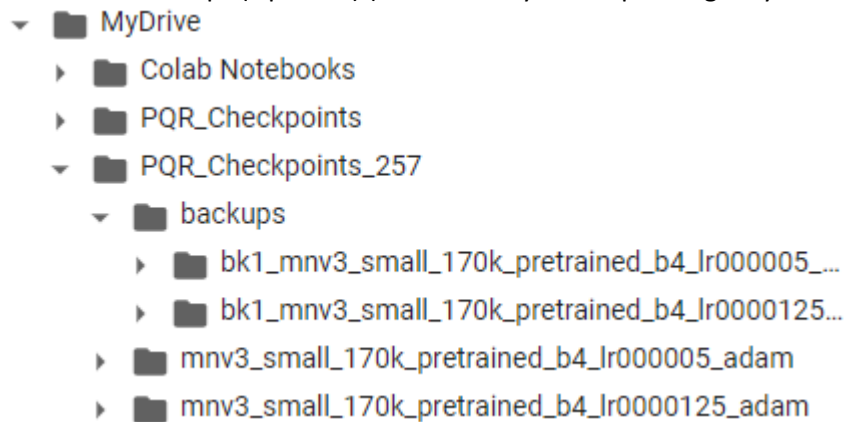


Figure 3.6: Sample directory (Reference)

```
#257
# Copy the tfrecords that we need
!cp -r /content/drive/SharedDrives/MachineLearning/Machine_Learning/Lips_Datasets/Lips_Dataset_257/tfrecord /content/
```

Figure 3.7: Copy the dataset that I have generated

4. Training the model

```
%cd /content/models/research
!pwd

/content/models/research
/content/models/research
```

Figure 4.1: Go to the necessary research directory

```

#--tf_initial_checkpoint="/content/deeplab_mnv3_small_cityscapes_trainfine/model.ckpt" \
#--initialize_last_layer=false \
#--last_layers_contain_logits_only=False \

# global step 4300: loss = 0.0111
# step 5953: loss = 0.0071

python3 deeplab/train.py \
--logtostderr \
--train_split="train" \
--model_variant="mobilenet_v3_small_seg" \
--train_crop_size="257,257" \
--train_batch_size=4 \
--dataset="pascal_voc_seg" \
--training_number_of_steps="200000" \
--fine_tune_batch_norm=false \
--base_learning_rate=0.0000125 \
--optimizer="adam" \
--tf_initial_checkpoint="/content/drive/MyDrive/PQR_Checkpoints_257/mnv3_small_170k_pretrained_b4_lr0000125_adam" \
--train_logdir="/content/drive/MyDrive/PQR_Checkpoints_257/mnv3_small_170k_pretrained_b4_lr0000125_adam" \
--dataset_dir="/content/tfrecord" \
--image_pooling_crop_size=257,257 \
--image_pooling_stride=4,5 \
--add_image_level_feature=1 \
--aspp_convs_filters=128 \
--aspp_with_concat_projection=0 \
--aspp_with_squeeze_and_excitation=1 \
--decoder_use_sum_merge=1 \
--decoder_filters=2 \
--decoder_output_is_logits=1 \
--image_se_uses_qsigmoid=1 \
--decoder_output_stride=8 \
--output_stride=32 \
--image_pyramid=1 \
--initialize_last_layer=true \
--last_layers_contain_logits_only=True \

```

Figure 4.2: Start training. Change the training flags to suit your needs

```

#--tf_initial_checkpoint="/content/deeplab_mnv3_small_cityscapes_trainfine/model.ckpt" \
#--initialize_last_layer=false \
#--last_layers_contain_logits_only=False \

```

Figure 4.3: IF you are using a different pretrained model, change to this flags.

5. Evaluate model

```

# mobilenet_v3_small_seg
!python deeplab/eval.py \
  --logtostderr \
  --model_variant="mobilenet_v3_small_seg" \
  --dataset="pascal_voc_seg" \
  --checkpoint_dir="/content/drive/MyDrive/PQR_Checkpoints/mnv3_small_170k_pretrained_b4" \
  --dataset_dir="/content/Tfrecord" \
  --eval_logdir="/content/eval_logdir" \
  --eval_split='train' \
  --eval_crop_size=513,513 \
  --image_pooling_stride=4,5 \
  --add_image_level_feature=1 \
  --aspp_convs_filters=128 \
  --aspp_with_concat_projection=0 \
  --aspp_with_squeeze_and_excitation=1 \
  --decoder_use_sum_merge=1 \
  --decoder_filters=2 \
  --decoder_output_is_logits=1 \
  --image_se_uses_qsigmoid=1 \
  --decoder_output_stride=8 \
  --output_stride=32

```

Figure 5.1: Evaluate model (Note: The time it takes to evaluate depends on the model and the size of dataset used. You can change the flags to suit your needs)

```

INFO:tensorflow:Starting evaluation at 2022-12-04-04:41:01
I1204 04:41:01.108534 139912754358144 evaluation.py:450] Starting evaluation at 2022-12-04-04:41:01
eval/miou_1.0_class_0[0.994508862]
eval/miou_1.0_class_1[0.536162496]
eval/miou_1.0_overall[0.765334487]
INFO:tensorflow:Waiting for new checkpoint at /content/drive/MyDrive/PQR_Checkpoints/mnv3_small_170k_pretr

```

Figure 5.2: Sample Output of Evaluation (Reference)

6. Visualize model

```

!python deeplab/vis.py \
  --logtostderr \
  --model_variant="mobilenet_v3_small_seg" \
  --dataset="pascal_voc_seg" \
  --checkpoint_dir="/content/drive/MyDrive/PQR_Checkpoints/mnv3_small_170k_pretrained_b4_lr0000125_adam" \
  --dataset_dir="/content/Tfrecord" \
  --vis_logdir="/content/drive/MyDrive/vis_lr0000125" \
  --vis_split='train' \
  --vis_crop_size=513,513 \
  --image_pooling_stride=4,5 \
  --add_image_level_feature=1 \
  --aspp_convs_filters=128 \
  --aspp_with_concat_projection=0 \
  --aspp_with_squeeze_and_excitation=1 \
  --decoder_use_sum_merge=1 \
  --decoder_filters=2 \
  --decoder_output_is_logits=1 \
  --image_se_uses_qsigmoid=1 \
  --decoder_output_stride=8 \
  --output_stride=32

```

Figure 6.1: Visualize model (Note: This will generate a folder depending on vis_logdir which contains the visualized segmentations. Change the flags to suit your needs)

7. Export model

```
!python deeplab/export_model.py \
--checkpoint_path='/content/drive/MyDrive/PQR_Checkpoints/mnv3_small_170k_pretrained_b4_lr0000125_adam/model.ckpt-106722' \
--export_path='/content/drive/MyDrive/PQR_Checkpoints/graphs/mnv3_small_b4_lr0000125_adam_s106722_loss0052.pb' \
--model_variant="mobilenet_v3_small_seg" \
--num_classes=2 \
--crop_size=513 \
--crop_size=513 \
--output_stride=32 \
--inference_scales=1.0 \
--image_pooling_stride=4,5 \
--add_image_level_feature=1 \
--aspp_convs_filters=128 \
--aspp_with_concat_projection=0 \
--aspp_with_squeeze_and_excitation=1 \
--decoder_use_sum_merge=1 \
--decoder_filters=2 \
--decoder_output_is_logits=1 \
--image_se_uses_qsigmoid=1 \
--decoder_output_stride=8 \
```

Figure 7.1: Export model in pb format (Note: Change the flags as you see fit.)

8.