TensorFlow DeepLab Model Zoo

We provide deeplab models pretrained several datasets, including (1) PASCAL VOC 2012, (2) Cityscapes, and (3) ADE20K for reproducing our results, as well as some checkpoints that are only pretrained on ImageNet for training your own models.

DeepLab models trained on PASCAL VOC 2012

Un-tar'ed directory includes:

- a frozen inference graph (`frozen_inference_graph.pb`). All frozen inference graphs by default use output stride of 8, a single eval scale of 1.0 and no left-right flips, unless otherwise specified. MobileNet-v2 based models do not include the decoder module.
- a checkpoint (`model.ckpt.data-00000-of-00001` , `model.ckpt.index`)

Model details

We provide several checkpoints that have been pretrained on VOC 2012 train_aug set or train_aug + trainval set. In the former case, one could train their model with smaller batch size and freeze batch normalization when limited GPU memory is available, since we have already fine-tuned the batch normalization for you. In the latter case, one could directly evaluate the checkpoints on VOC 2012 test set or use this checkpoint for demo. Note *MobileNet-v2* based models do not employ ASPP and decoder modules for fast computation.

Checkpoint name	Network backbone	Pretrained dataset	ASPP	Decoder
mobilenetv2_dm05_coco_voc_trainaug	MobileNet-v2 Depth-Multiplier = 0.5	ImageNet MS-COCO VOC 2012 train_aug set	N/A	N/A
mobilenetv2_dm05_coco_voc_trainval	MobileNet-v2 Depth-Multiplier = 0.5	ImageNet MS-COCO VOC 2012 train_aug + trainval sets	N/A	N/A
mobilenetv2_coco_voc_trainaug	MobileNet-v2	ImageNet MS-COCO VOC 2012 train_aug set	N/A	N/A
mobilenetv2_coco_voc_trainval	MobileNet-v2	ImageNet MS-COCO VOC 2012 train_aug + trainval sets	N/A	N/A
xception65_coco_voc_trainaug	Xception_65	ImageNet MS-COCO VOC 2012 train_aug set	[6,12,18] for OS=16 [12,24,36] for OS=8	OS = 4

xception65_coco_voc_trainval	Xception_65	ImageNet MS-COCO VOC 2012 train_aug + trainval sets	[6,12,18] for OS=16 [12,24,36] for OS=8	OS = 4

In the table, **OS** denotes output stride.

Checkpoint name	Eval OS	Eval scales	Left-right Flip	Multiply-Adds	Runtime (sec)	PASCA mIOU
[mobilenetv2_dm05_coco_voc_trainaug]()	16	[1.0]	No	0.88B	-	70.19% (val)
[mobilenetv2_dm05_coco_voc_trainval]()	8	[1.0]	No	2.84B	-	71.83% (test)
[mobilenetv2_coco_voc_trainaug]()	16 8	[1.0] [0.5:0.25:1.75]	No Yes	2.75B 152.59B	0.1 26.9	75.32% (val) 77.33 (val)
[mobilenetv2_coco_voc_trainval]()	8	[0.5:0.25:1.75]	Yes	152.59B	26.9	80.25% (**test**)
[xception65_coco_voc_trainaug]()	16 8	[1.0] [0.5:0.25:1.75]	No Yes	54.17B 3055.35B	0.7 223.2	82.20% (val) 83.58% (val)
[xception65_coco_voc_trainval]()	8	[0.5:0.25:1.75]	Yes	3055.35B	223.2	87.80% (**test**)

In the table, we report both computation complexity (in terms of Multiply-Adds and CPU Runtime) and segmentation performance (in terms of mIOU) on the PASCAL VOC val or test set. The reported runtime is calculated by tfprof on a workstation with CPU E5-1650 v3 @ 3.50GHz and 32GB memory. Note that applying multi-scale inputs and left-right flips increases the segmentation performance but also significantly increases the computation and thus may not be suitable for real-time applications.

DeepLab models trained on Cityscapes

Model details

We provide several checkpoints that have been pretrained on Cityscapes train_fine set. Note *MobileNet-v2* based model has been pretrained on MS-COCO dataset and does not employ ASPP and decoder modules for fast computation.

Checkpoint name	Network backbone	Pretrained dataset	ASPP	Decoder
mobilenetv2_coco_cityscapes_trainfine	MobileNet-v2	ImageNet MS-COCO	N/A	N/A

		Cityscapes train_fine set		
mobilenetv3_large_cityscapes_trainfine	MobileNet-v3 Large	Cityscapes train_fine set (No ImageNet)	N/A	OS = 8
mobilenetv3_small_cityscapes_trainfine	MobileNet-v3 Small	Cityscapes train_fine set (No ImageNet)	N/A	OS = 8
xception65_cityscapes_trainfine	Xception_65	ImageNet Cityscapes train_fine set	[6, 12, 18] for OS=16 [12, 24, 36] for OS=8	OS = 4
xception71_dpc_cityscapes_trainfine	Xception_71	ImageNet MS-COCO Cityscapes train_fine set	Dense Prediction Cell	OS = 4
xception71_dpc_cityscapes_trainval	Xception_71	ImageNet MS-COCO Cityscapes trainval_fine and coarse set	Dense Prediction Cell	OS = 4

In the table, **OS** denotes output stride.

Note for mobilenet v3 models, we use additional commandline flags as follows:

```
--model_variant={ mobilenet_v3_large_seg | mobilenet_v3_small_seg }
--image_pooling_crop_size=769,769
--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=19
--decoder_output_is_logits=1
--image_se_uses_qsigmoid=1
--decoder_output_stride=8
--output_stride=32
```

Checkpoint name	Eval OS	Eval scales	Left-right Flip	Multiply-Adds	Runtime (sec)	Citys ml
mobilenetv2_coco_cityscapes_trainfine	16 8	[1.0] [0.75:0.25:1.25]	No Yes	21.27B 433.24B	0.8 51.12	70. (v 73. (v

mobilenetv3_large_cityscapes_trainfine	32	[1.0]	No	15.95B	0.6	72. (v
mobilenetv3_small_cityscapes_trainfine	32	[1.0]	No	4.63B	0.4	68. (v
xception65_cityscapes_trainfine	16 8	[1.0] [0.75:0.25:1.25]	No Yes	418.64B 8677.92B	5.0 422.8	78. (v 80. (v
xception71_dpc_cityscapes_trainfine	16	[1.0]	No	502.07B	-	80. (v
xception71_dpc_cityscapes_trainval	8	[0.75:0.25:2]	Yes	-	-	82. (**te**

EdgeTPU-DeepLab models on Cityscapes

EdgeTPU is Google's machine learning accelerator architecture for edge devices (exists in Coral devices and Pixel4's Neural Core). Leveraging nerual architecture search (NAS, also named as Auto-ML) algorithms, EdgeTPU-Mobilenet has been released which yields higher hardware utilization, lower latency, as well as better accuracy over Mobilenet-v2/v3. We use EdgeTPU-Mobilenet as the backbone and provide checkpoints that have been pretrained on Cityscapes train_fine set. We named them as EdgeTPU-DeepLab models.

Checkpoint name	Network backbone	Pretrained dataset	ASPP	Decoder
EdgeTPU-DeepLab	EdgeMobilenet-1.0	ImageNet	N/A	N/A
EdgeTPU-DeepLab-slim	EdgeMobilenet-0.75	ImageNet	N/A	N/A

For EdgeTPU-DeepLab-slim, the backbone feature extractor has depth multiplier = 0.75 and aspp_convs_filters = 128. We do not employ ASPP nor decoder modules to further reduce the latency. We employ the same train/eval flags used for MobileNet-v2 DeepLab model. Flags changed for EdgeTPU-DeepLab model are listed here.

```
--decoder_output_stride=''
--aspp_convs_filters=256
--model_variant=mobilenet_edgetpu
```

For EdgeTPU-DeepLab-slim, also include the following flags.

```
--depth_multiplier=0.75
--aspp_convs_filters=128
```

Checkpoint name	Eval OS	Eval scales	Cityscapes mIOU	Multiply-Adds	Simulator latency on Pixel 4 EdgeTPU
EdgeTPU-DeepLab	32 16	[1.0]	70.6% (val) 74.1% (val)	5.6B 7.1B	13.8 ms 17.5 ms
EdgeTPU-DeepLab-slim	32 16	[1.0]	70.0% (val) 73.2% (val)	3.5B 4.3B	9.9 ms 13.2 ms

DeepLab models trained on ADE20K

Model details

We provide some checkpoints that have been pretrained on ADE20K training set. Note that the model has only been pretrained on ImageNet, following the dataset rule.

Checkpoint name	Network backbone	Pretrained dataset	ASPP	Decoder	Input size
mobilenetv2_ade20k_train	MobileNet-v2	ImageNet ADE20K training set	N/A	OS = 4	257x257
xception65_ade20k_train	Xception_65	ImageNet ADE20K training set	[6, 12, 18] for OS=16 [12, 24, 36] for OS=8	OS = 4	513x513

The input dimensions of ADE20K have a huge amount of variation. We resize inputs so that the longest size is 257 for MobileNet-v2 (faster inference) and 513 for Xception_65 (better performation). Note that we also include the decoder module in the MobileNet-v2 checkpoint.

Checkpoint name	Eval OS	Eval scales	Left-right Flip	mIOU	Pixel-wise Accuracy	File Size
mobilenetv2_ade20k_train	16	[1.0]	No	32.04% (val)	75.41% (val)	24.8MB
xception65_ade20k_train	8	[0.5:0.25:1.75]	Yes	45.65% (val)	82.52% (val)	439MB

Checkpoints pretrained on ImageNet

Un-tar'ed directory includes:

- model checkpoint (`model.ckpt.data-00000-of-00001` , `model.ckpt.index`).

Model details

We also provide some checkpoints that are pretrained on ImageNet and/or COCO (as post-fixed in the model name) so that one could use this for training your own models.

- mobilenet_v2: We refer the interested users to the TensorFlow open source MobileNet-V2 for details.
- xception_{41,65,71}: We adapt the original Xception model to the task of semantic segmentation with the following changes: (1) more layers, (2) all max pooling operations are replaced by strided (atrous) separable convolutions, and (3) extra batch-norm and ReLU after each 3x3 depthwise convolution are added. We provide three Xception model variants with different network depths.
- resnet_v1_{50,101}_beta: We modify the original ResNet-101 [10], similar to PSPNet [11] by replacing the first 7x7 convolution with three 3x3 convolutions. See resnet_v1_beta.py for more details.

Model name	File Size
xception_41_imagenet	288MB
xception_65_imagenet	447MB
xception_65_imagenet_coco	292MB
xception_71_imagenet	474MB
resnet_v1_50_beta_imagenet	274MB
resnet_v1_101_beta_imagenet	477MB

References

1. **Mobilenets: Efficient convolutional neural networks for mobile vision applications**
   Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam
   [link]. arXiv:1704.04861, 2017.

2. **Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation**
   Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen
   [link]. arXiv:1801.04381, 2018.

3. **Xception: Deep Learning with Depthwise Separable Convolutions**
   François Chollet
   [link]. In the Proc. of CVPR, 2017.

4. **Deformable Convolutional Networks -- COCO Detection and Segmentation Challenge 2017 Entry**
   Haozhi Qi, Zheng Zhang, Bin Xiao, Han Hu, Bowen Cheng, Yichen Wei, Jifeng Dai
   [link]. ICCV COCO Challenge Workshop, 2017.

5. **The Pascal Visual Object Classes Challenge: A Retrospective**
   Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, Andrew Zisserman
   [link]. IJCV, 2014.

6. **Semantic Contours from Inverse Detectors**
   Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, Jitendra Malik
   [link]. In the Proc. of ICCV, 2011.

7. **The Cityscapes Dataset for Semantic Urban Scene Understanding**
   Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele.
   [link]. In the Proc. of CVPR, 2016.

8. **Microsoft COCO: Common Objects in Context**
   Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar
   [link]. In the Proc. of ECCV, 2014.

9. **ImageNet Large Scale Visual Recognition Challenge**
   Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej

Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei
[link]. IJCV, 2015.

10. **Deep Residual Learning for Image Recognition**
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
[link]. CVPR, 2016.

11. **Pyramid Scene Parsing Network**
Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia
[link]. In CVPR, 2017.

12. **Scene Parsing through ADE20K Dataset**
Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, Antonio Torralba
[link]. In CVPR,

2017.

13. **Searching for MobileNetV3**
Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam
[link]. In ICCV, 2019.