Custom Semantic Image Segmentation with DeepLabv3+

Computational Intelligence Lab in the Department of Computer Science at Cal Poly Pomona

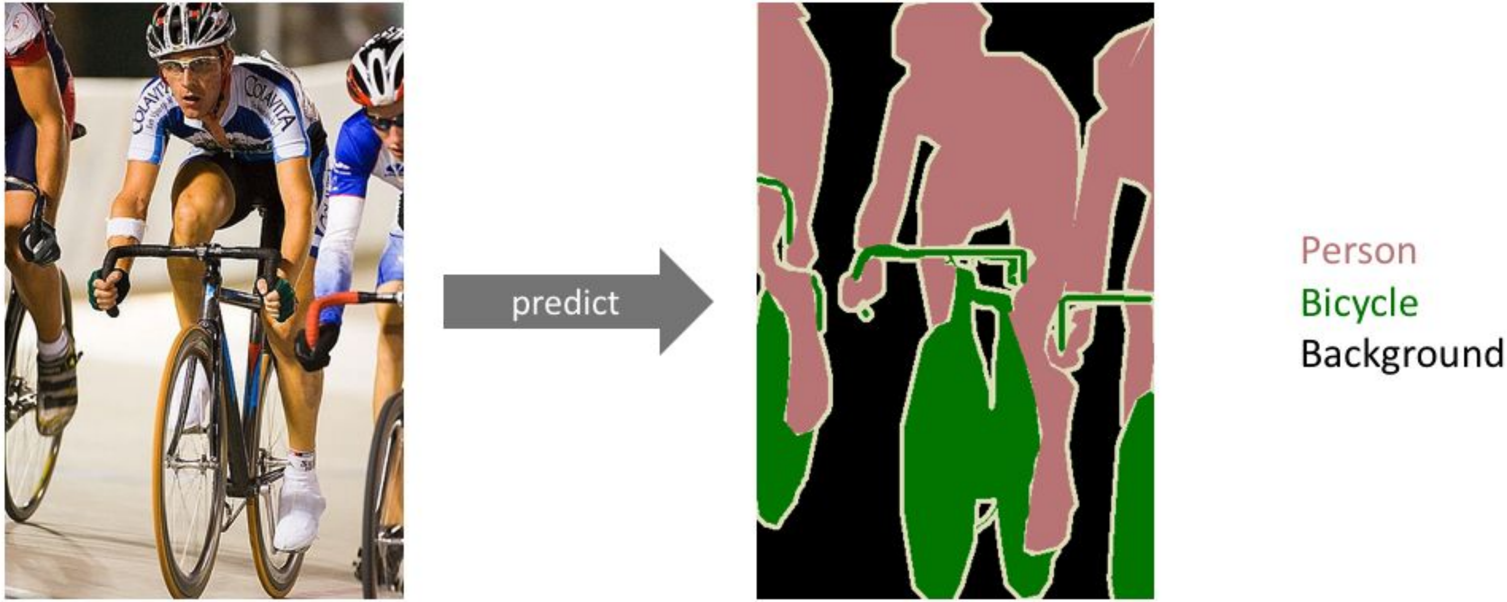
David Hughes

Faculty Mentor: Dr. Hao Ji

Resources

- Deeplabv3+ paper
  - https://arxiv.org/abs/1802.02611
- Tensorflow implementation of deeplab
  - https://github.com/tensorflow/models/tree/master/research/deeplab
- Github for custom training with Tensorflow's deeplab
  - https://github.com/David-Hughes3/deeplab_project

Semantic Image Segmentation



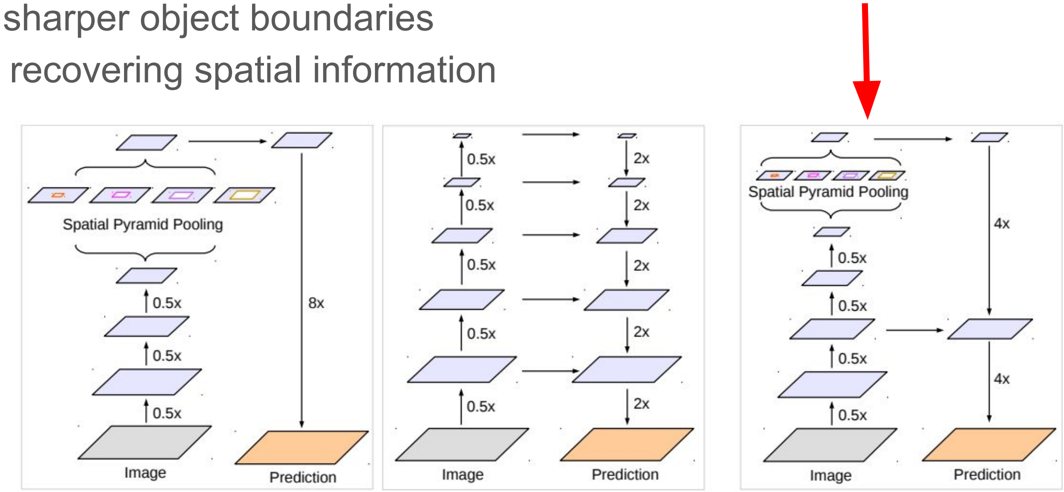
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/#devkit

Deeplabv3+

Goal of Deeplabv3+

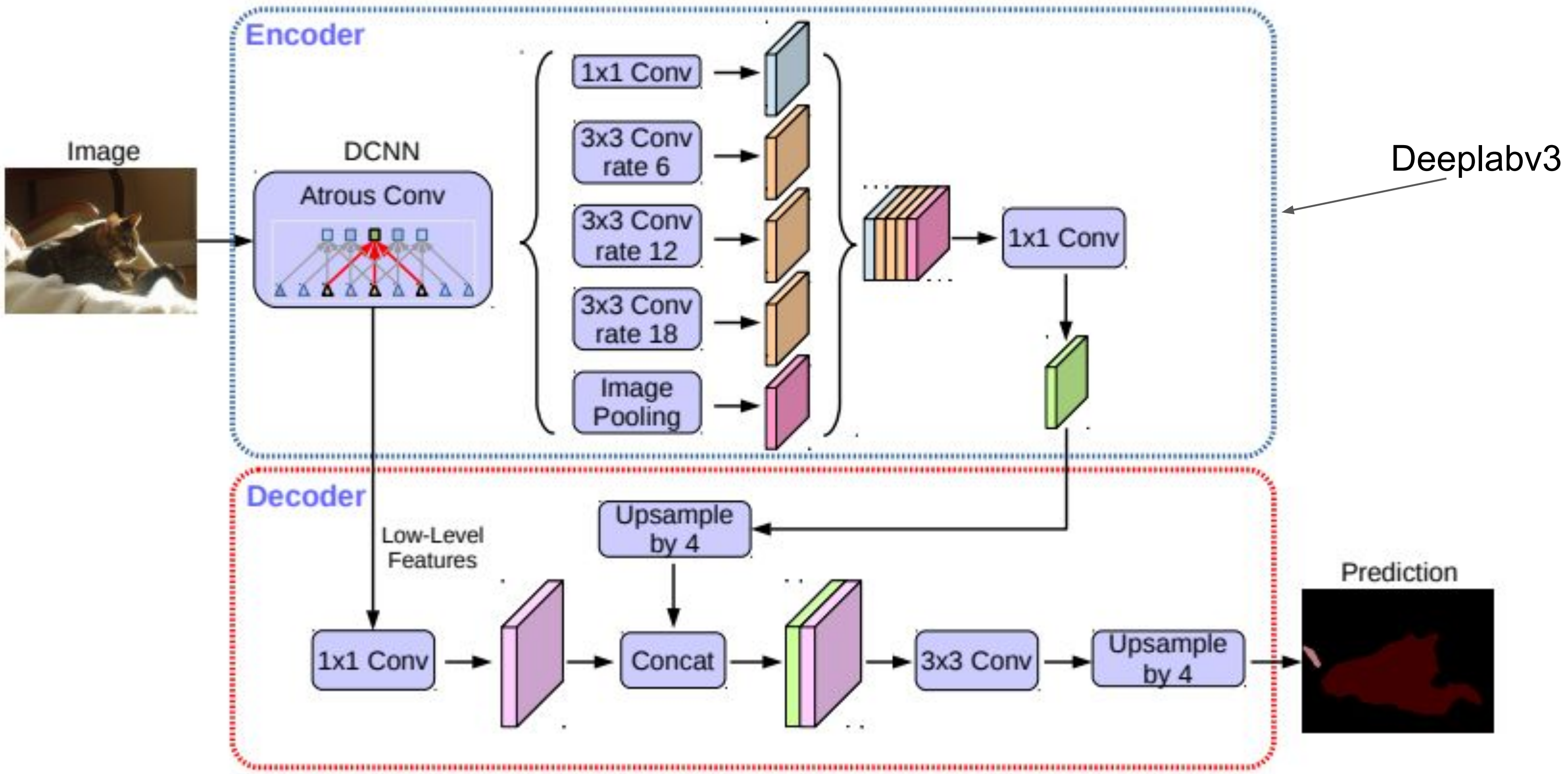
Combine the advantages of:

- Spatial pyramid pooling module
    - Encode multi-scale contextual information
        - By probing incoming features with filters or pooling ops at multiple rates and effective FOVs
- Encoder-decoder structure
    - Capture sharper object boundaries
        - By recovering spatial information



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

[1]



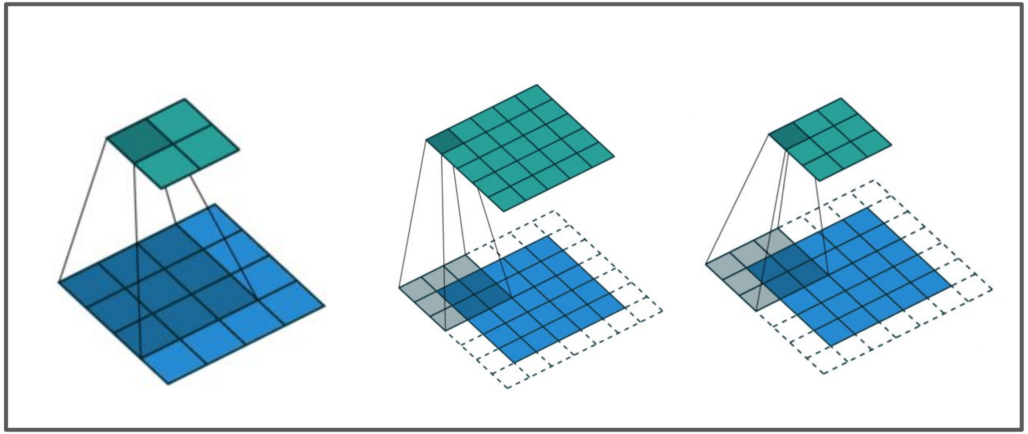
Deeplabv3+ Structure

[1]

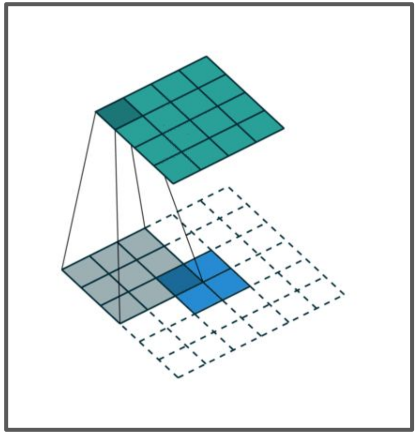
Encoder = DeepLabv3

Basic Types of Convolutions

Blue maps are inputs, and cyan maps are outputs.



Convolution



Transposed Convolution

https://github.com/vdumoulin/conv_arithmetic

Atrous Convolution / Hole Algorithm/ Dilated Conv

$$y[i] = \sum_{k} x[i + r \cdot k] w[k]$$

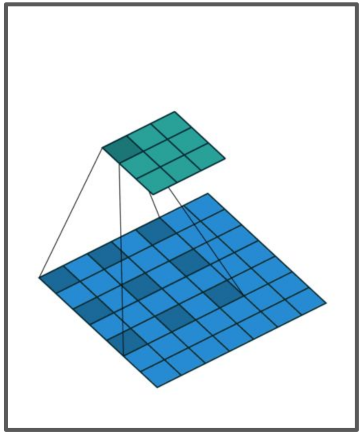
Atrous Convolution

- Location i on output feature map y
- Conv filter w
- Applied over the input feature map x
- Atrous rate r is stride to sample input signal

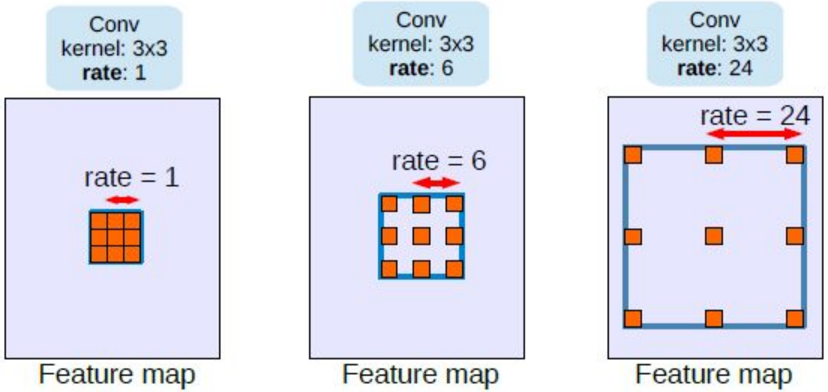
- r=1, standard convolution
- Holes = introducing zeros between filter values
  - “convolving the input x with upsampled filters produced by inserting r-1 zeros between two consecutive filter values along each spatial dimension.” source
- **Filter’s field-of-view can be changed by modifying rate**
  - Small = accurate localization
  - Large = context assimilation

[3]

Atrous Convolution



Dilated-
Convolution



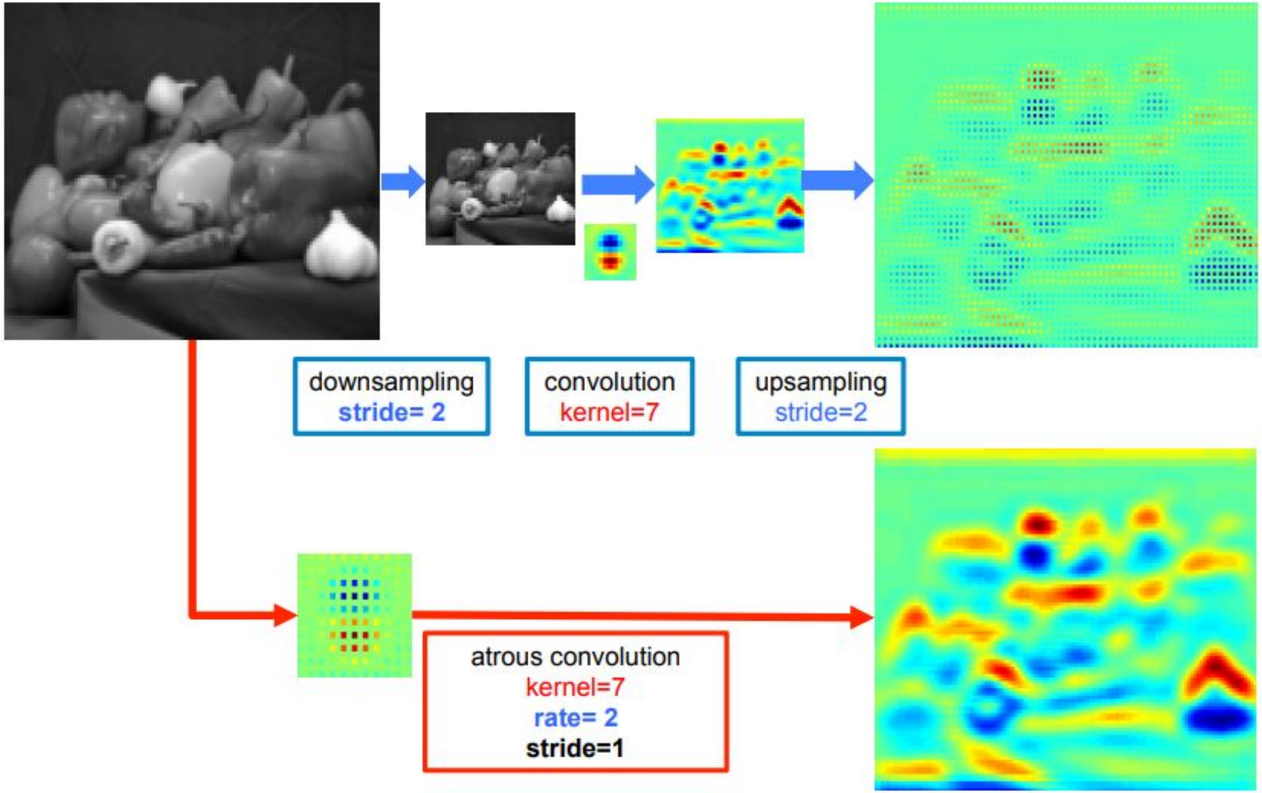
Atrous Convolution with Different Rates r

Animation: https://github.com/vdumoulin/conv_arithmetic
Image: [2]

Reasoning

- repeated combination of max-pooling and striding at consecutive layers of these networks reduces significantly the spatial resolution of the resulting feature maps
- A partial remedy is to use ‘deconvolutional’/transpose layers
    - However requires additional memory and time.
- Atrous convolution allows computing the responses of any layer at any desirable resolution

[3]

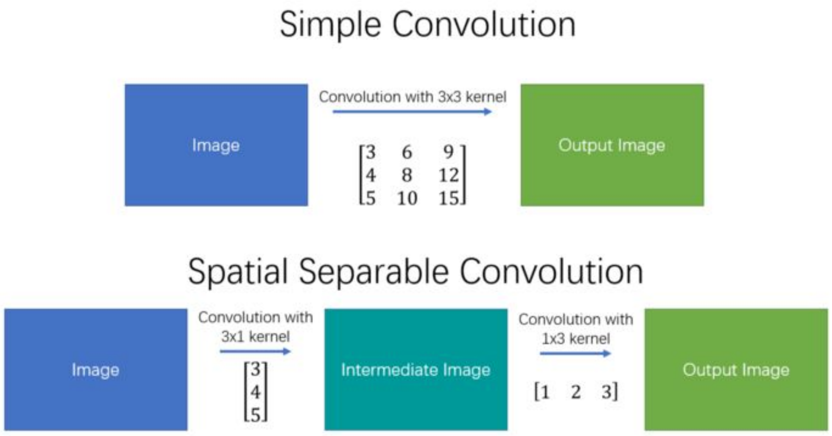

downsampling
stride= 2
convolution
kernel=7
upsampling
stride=2
atrous convolution
kernel=7
rate= 2
stride=1

[3]

Depthwise separable convolution

First, spatial separable convolutions

- Instead of doing one conv with 9 multiplications
- Do two convolutions with 3 mults each (6 total)
- Less mults = less comp complexity



https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Depthwise separable convolution

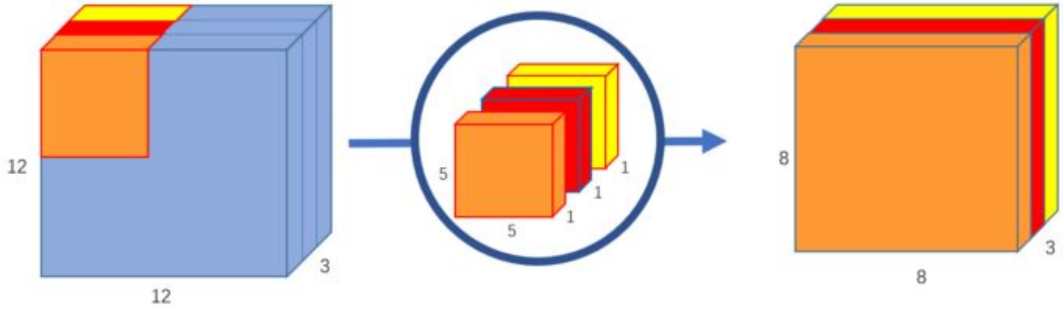
- Not all kernels can be factored into two smaller kernels
  - Depthwise separable convolution work either way = more common
- Ex: keras.layers.SeparableConv2D or tf.layers.separable_conv2d

- Like spatial separable conv, depthwise splits a kernel into 2 separate kernels
  - ***Depthwise convolution kernel***
  - ***Pointwise convolution kernel***

https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Depthwise Convolution Kernel

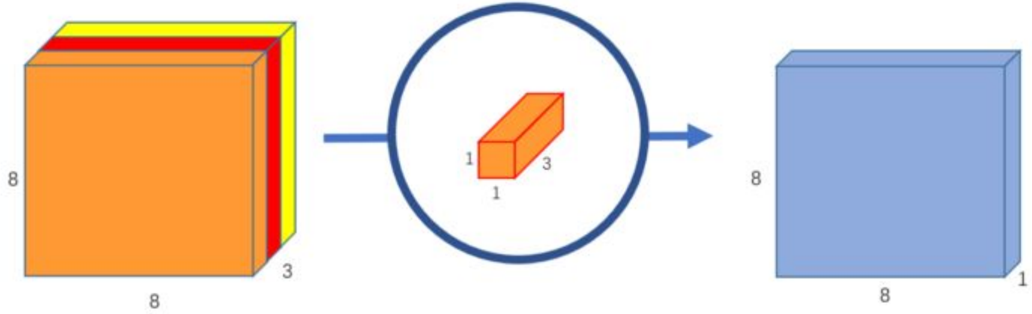
- Performs a spatial convolution independently for each channel
  - Apply a single filter for each input channel



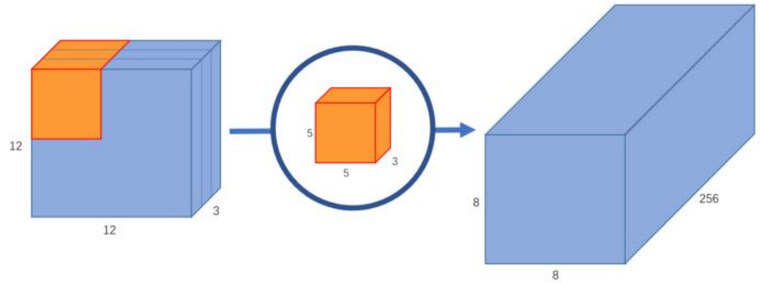
https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Pointwise convolution kernel

- Combines the output from the depthwise convolution

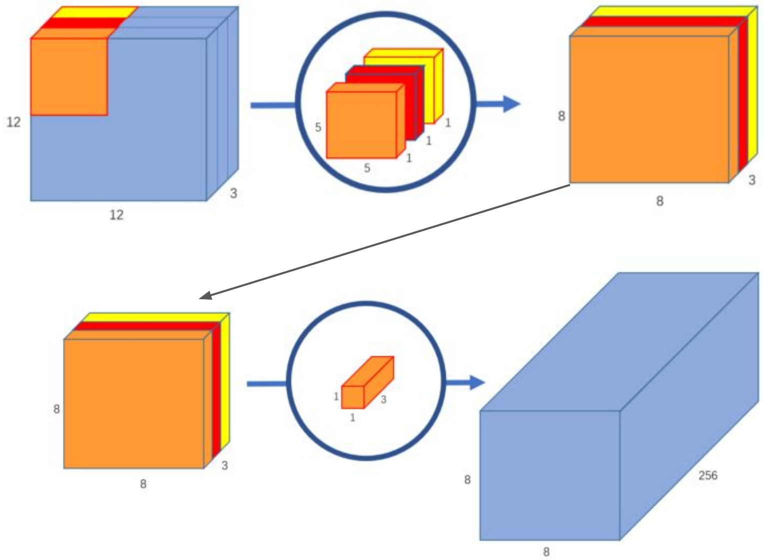


https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728



Normal Convolution:

- 12x12x3 image with 256 (5x5x3) kernels -> 8x8x1 image
- 256 (5x5x3) kernels move 8x8 times
    - 256x3x5x5x8x8=1,228,800 multiplications.



Depthwise Convolution (conv without changing depth):

- 12x12x3 image with 3 (5x5x1) kernels
- 3 (5x5x1) kernels that move 8x8 times.
    - That’s 3x5x5x8x8 = 4,800 multiplications

Pointwise Convolution:

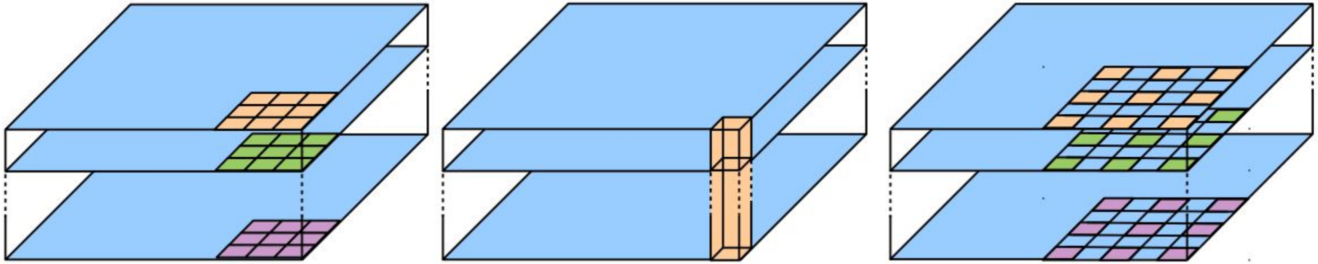
- 256 (1x1x3) kernels that output a 8x8x1 image each to get a final image of shape 8x8x256.
- 256 (1x1x3) kernels that move 8x8 times.
    - That’s 256x1x1x3x8x8=49,152 multiplications.

https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

49,152 + 4,800 = 53,952 << 1,228,800

Atrous Separable Convolution

DeepLabv3+: Encoder-Decoder with Atrous Separable Convolution 5



(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

Fig. 3. 3×3 Depthwise separable convolution decomposes a standard convolution into (a) a depthwise convolution (applying a single filter for each input channel) and (b) a pointwise convolution (combining the outputs from depthwise convolution across channels). In this work, we explore *atrous separable convolution* where atrous convolution is adopted in the depthwise convolution, as shown in (c) with $rate = 2$.

[1]

Atrous Spatial Pyramid Pooling (ASPP) module

- probes convolutional features at multiple scales by applying atrous convolution with different rates

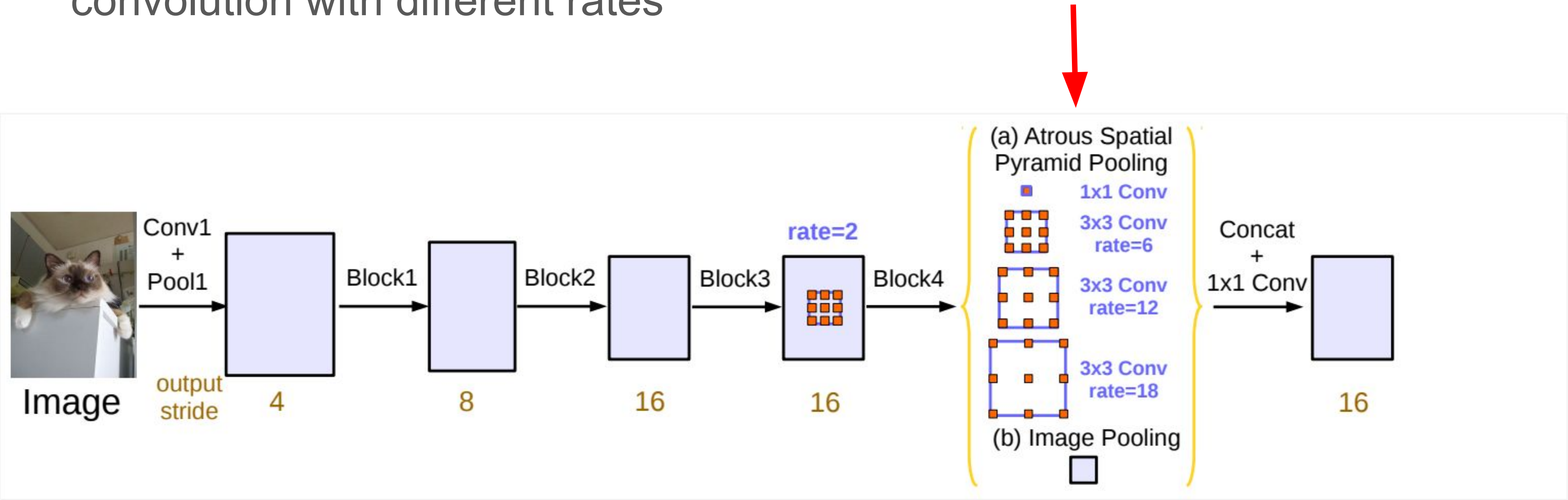


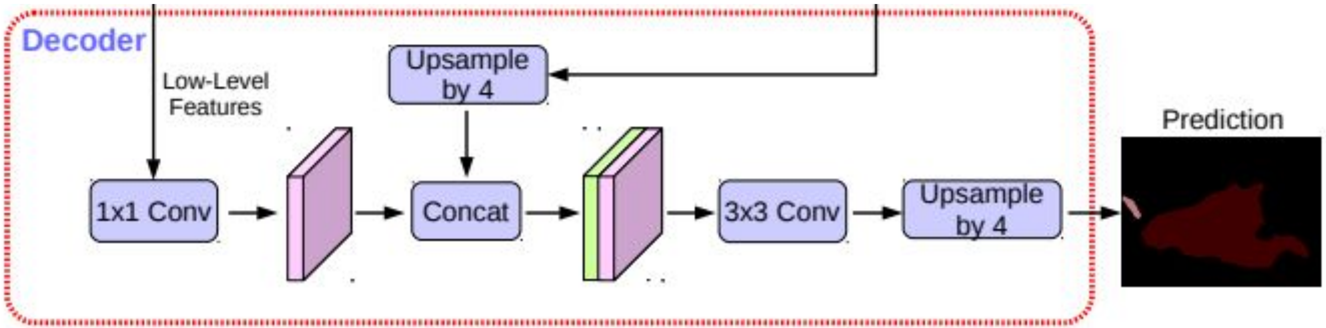
Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

[2]

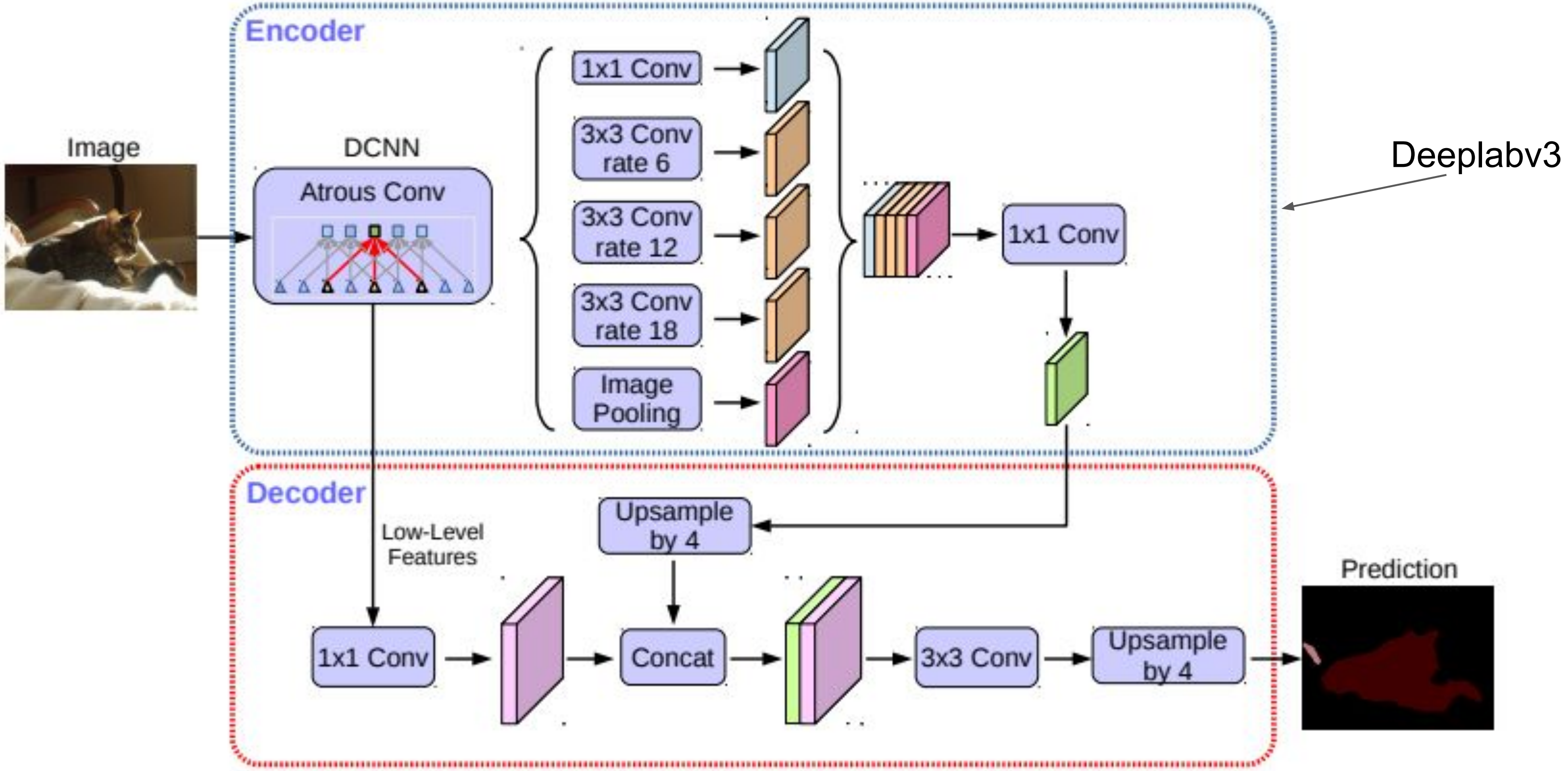
Decoder

Simple Decoder

- very simple yet effective decoder module:
  - concatenation of
    - DeepLabv3 feature map
    - channel-reduced Conv2 feature map
  - refined by two [3 × 3, 256] operations.



[1]



Deeplabv3+ Structure

[1]

Deeplabv3+ Paper results

Method	mIOU
Deep Layer Cascade (LC) [82]	82.7
TuSimple [77]	83.1
Large_Kernel_Matters [60]	83.6
Multipath-RefineNet [58]	84.2
ResNet-38_MS_COCO [83]	84.9
PSPNet [24]	85.4
IDW-CNN [84]	86.3
CASIA_IVA_SDN [63]	86.6
DIS [85]	86.8
DeepLabv3 [23]	85.7
DeepLabv3-JFT [23]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

Table 6. PASCAL VOC 2012 *test* set results with top-performing models.

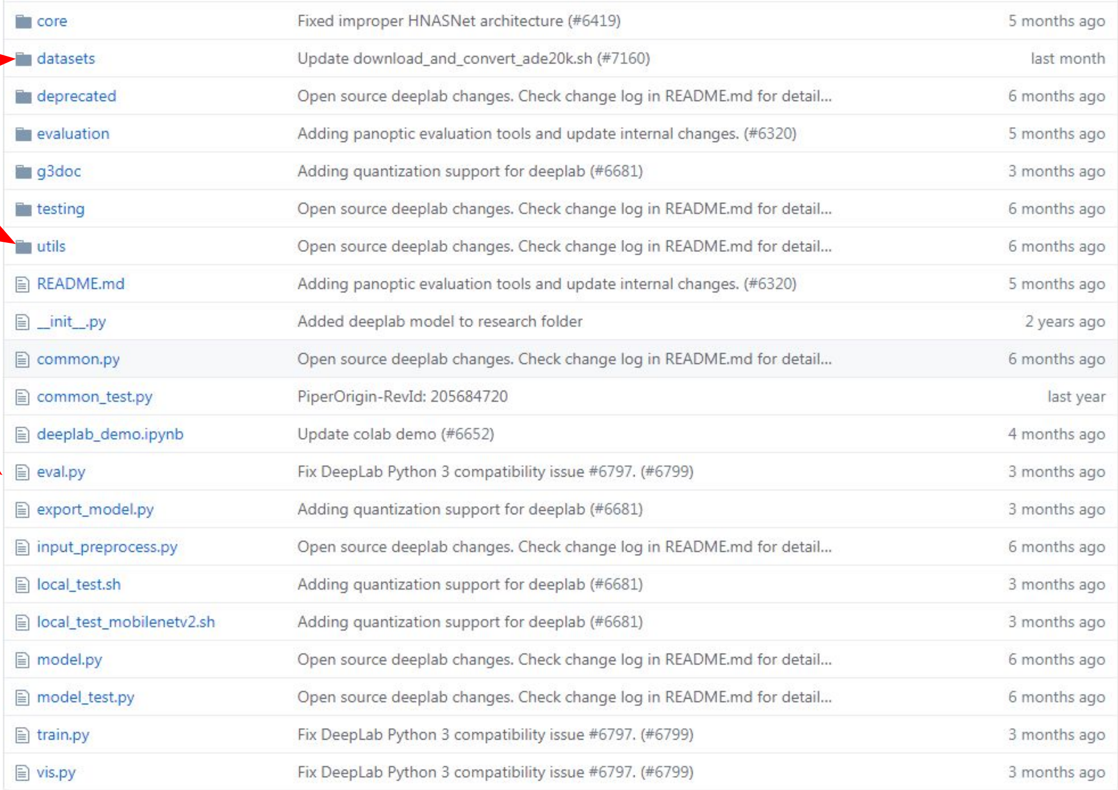
[1]

Custom training

Custom Training

Tensorflow Files that must be modified for custom dataset training

- train_utils.py
- data_generator.py
- Eval.py



core	Fixed improper HNASNet architecture (#6419)	5 months ago
datasets	Update download_and_convert_ade20k.sh (#7160)	last month
deprecated	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
evaluation	Adding panoptic evaluation tools and update internal changes. (#6320)	5 months ago
g3doc	Adding quantization support for deeplab (#6681)	3 months ago
testing	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
utils	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
README.md	Adding panoptic evaluation tools and update internal changes. (#6320)	5 months ago
__init__.py	Added deeplab model to research folder	2 years ago
common.py	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
common_test.py	PiperOrigin-RevId: 205684720	last year
deeplab_demo.ipynb	Update colab demo (#6652)	4 months ago
eval.py	Fix DeepLab Python 3 compatibility issue #6797. (#6799)	3 months ago
export_model.py	Adding quantization support for deeplab (#6681)	3 months ago
input_preprocess.py	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
local_test.sh	Adding quantization support for deeplab (#6681)	3 months ago
local_test_mobilenetv2.sh	Adding quantization support for deeplab (#6681)	3 months ago
model.py	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
model_test.py	Open source deeplab changes. Check change log in README.md for detail...	6 months ago
train.py	Fix DeepLab Python 3 compatibility issue #6797. (#6799)	3 months ago
vis.py	Fix DeepLab Python 3 compatibility issue #6797. (#6799)	3 months ago

Steps

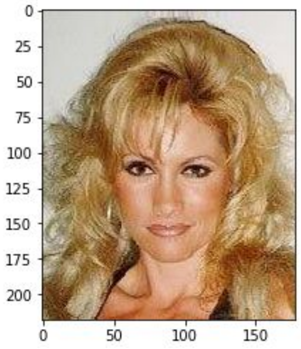
1. Modify tensorflow files with your dataset information
2. Convert masks to color-indexed images
3. Create
   a. 'train.txt' with training image filenames
   b. 'val.txt' with validation image filenames
   c. 'trainval.txt' with both
   d. NOTE: training images and masks should have the same filename
4. Create a tfrecord of your dataset using 'build_voc2012_data.py'
   a. Tfrecord = a Tensorflow binary storage format
5. If using a pre-trained model (transfer learning)
   a. https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md
6. Run 'train.py'
7. Run 'eval.py' - custom file has miou per class and overall miou
8. Run 'vis.py' - to get segmentation masks
9. Run 'export_model.py' to save your trained 'frozen_inference_graph.pb'
10. Use 'deeplab_demo.ipynb' with your checkpoint for future inference

Custom Training - Datasets

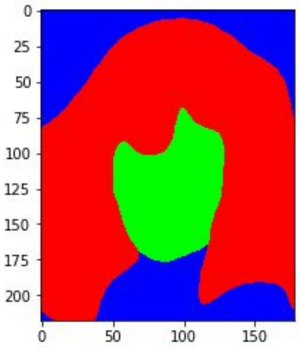
- Labeled Faces in the Wild - Parts Dataset
- CelebA segmentation masks & CelebA

Preprocessing needs to be done to convert segmentation maps to color-indexed images.

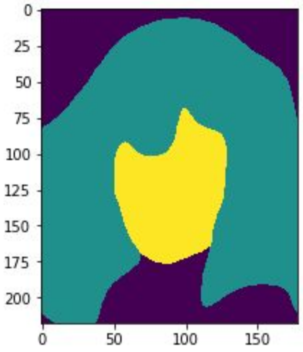
0 = background
1 = hair
2 = skin



LFW image



LFW - parts mask



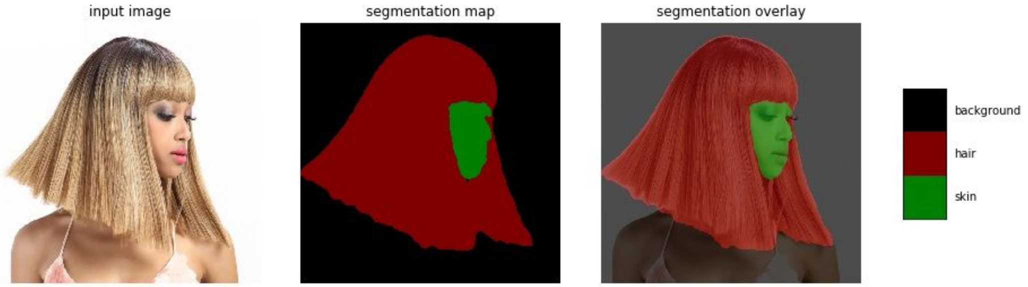
color-indexed mask

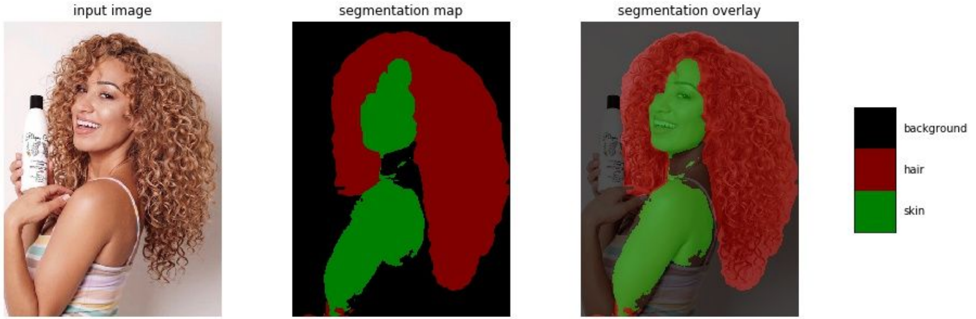
Custom Training Results

Weights for classes:
1, background
10, hair
5, face

10000 iterations: miou
class_0: 0.91722
class_1: 0.66235
class_2: 0.83431
mean_iou: 0.8046266666666666

20000 iterations: miou
class_0: 0.92631
class_1: 0.67995
class_2: 0.85498
mean_iou: 0.8204133333333333





Bibtex format References

[1] @inproceedings{deeplabv3plus2018,
  title={Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation},
  author={Liang-Chieh Chen and Yukun Zhu and George Papandreou and Florian Schroff and Hartwig Adam},
  booktitle={ECCV},
  year={2018}
}
[2] @article{DBLP:journals/corr/ChenPSA17,
  author   = {Liang{-}Chieh Chen and George Papandreou and Florian Schroff and Hartwig Adam},
  title   = {Rethinking Atrous Convolution for Semantic Image Segmentation},
  journal   = {CoRR},
  volume   = {abs/1706.05587},
  year   = {2017},
  url   = {http://arxiv.org/abs/1706.05587},
  archivePrefix = {arXiv},
  eprint   = {1706.05587},
  timestamp = {Mon, 13 Aug 2018 16:48:07 +0200},
  biburl   = {https://dblp.org/rec/bib/journals/corr/ChenPSA17},
  bibsource = {dblp computer science bibliography, https://dblp.org}
}
[3] @article{DBLP:journals/corr/ChenPK0Y16,
  author   = {Liang{-}Chieh Chen and George Papandreou and Iasonas Kokkinos and Kevin Murphy and Alan L. Yuille},
  title   = {DeepLab: Semantic Image Segmentation with Deep Convolutional Nets,
               Atrous Convolution, and Fully Connected CRFs},
  journal   = {CoRR},
  volume   = {abs/1606.00915},
  year   = {2016},
  url   = {http://arxiv.org/abs/1606.00915},
  archivePrefix = {arXiv},
  eprint   = {1606.00915},
  timestamp = {Mon, 13 Aug 2018 16:46:13 +0200},
  biburl   = {https://dblp.org/rec/bib/journals/corr/ChenPK0Y16},
  bibsource = {dblp computer science bibliography, https://dblp.org}
}