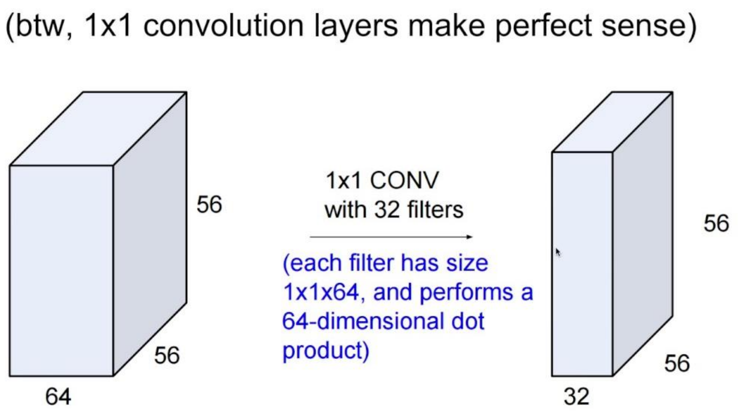
AI Benchmark:

Model: Task3 Inception ResNet v1 (Face Recognition)

1.Special Operation: 1x1 conv

Detail: Fuse the channels, decrease the number of channels



(btw, 1x1 convolution layers make perfect sense)

(each filter has size 1x1x64, and performs a 64-dimensional dot product)

2. Special Operation: 1x3 conv or 3x1 conv

Detail: Guessing this kind of conv is doing gradient for each channel, where 1x3 is for x direction and 3x1 is for y direction gradient for each channel. Instead of using 3x3, this method can save computation time.

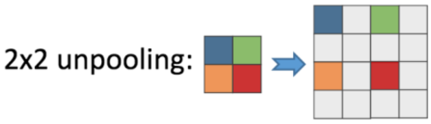
3. Special Operation: 1x7 conv or 7x1 conv

Detail: Guessing this kind of conv is doing gradient for each channel, where 1x7 is for x direction and 7x1 is for y direction gradient for each channel. Instead of using 7x7, this method can save computation time.

Model: Task 6 ICNet

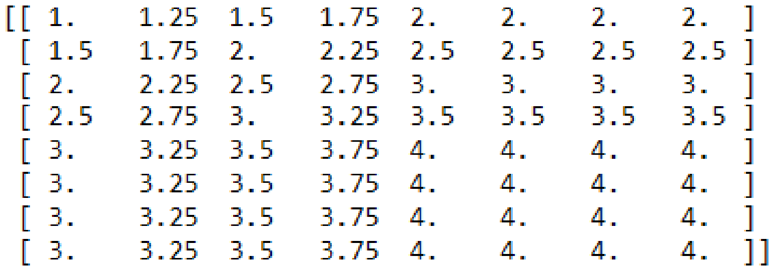
1.Special Operation: Upsampling2D

Detail: padding zero in between and use bilinear interpolation way to calculate value to replace the in-between zero.

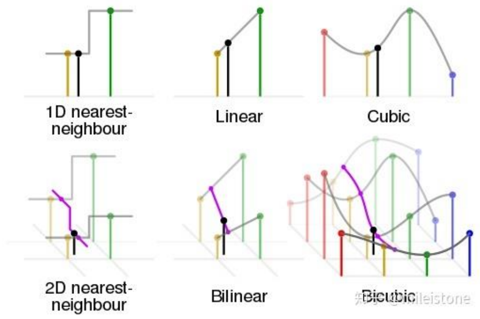


```
[[1 2]
 [3 4]]
```

Bilinear x 4



```
[[ 1.    1.25  1.5   1.75  2.    2.    2.    2.  ]
 [ 1.5   1.75  2.    2.25  2.5   2.5   2.5   2.5 ]
 [ 2.    2.25  2.5   2.75  3.    3.    3.    3.  ]
 [ 2.5   2.75  3.    3.25  3.5   3.5   3.5   3.5 ]
 [ 3.    3.25  3.5   3.75  4.    4.    4.    4.  ]
 [ 3.    3.25  3.5   3.75  4.    4.    4.    4.  ]
 [ 3.    3.25  3.5   3.75  4.    4.    4.    4.  ]
 [ 3.    3.25  3.5   3.75  4.    4.    4.    4.  ]]
```



2. Special Operation: Dilated convolution

Detail: the kernel size will increase by inserting zeros as the dilated rate getting larger.

Model: Task SRGAN

1. Special Operation: (Input Image Normalization)

   Detail: Image / 255

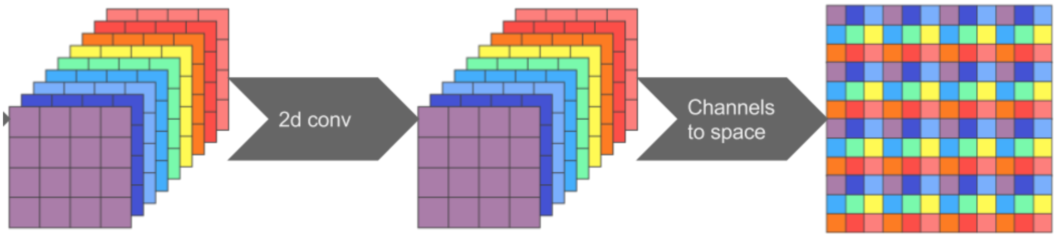
2. Special Operation: (Output Image DeNormalization)

   Detail: (Image + 1) * 127.5

3. Special Operation: tf.Depth_to_Space(factor)

Detail: 1. Apply factor on conv filter first to enlarge channel

2. Use this Depth_to_Space function to do upsampling

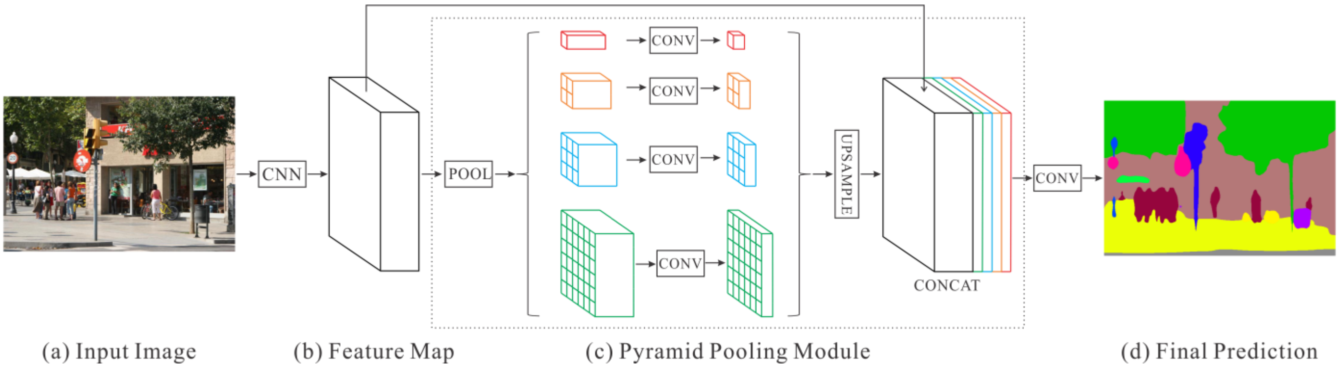


Segmentation:

Model: PSPNet

Special Operation: tf.image.resize

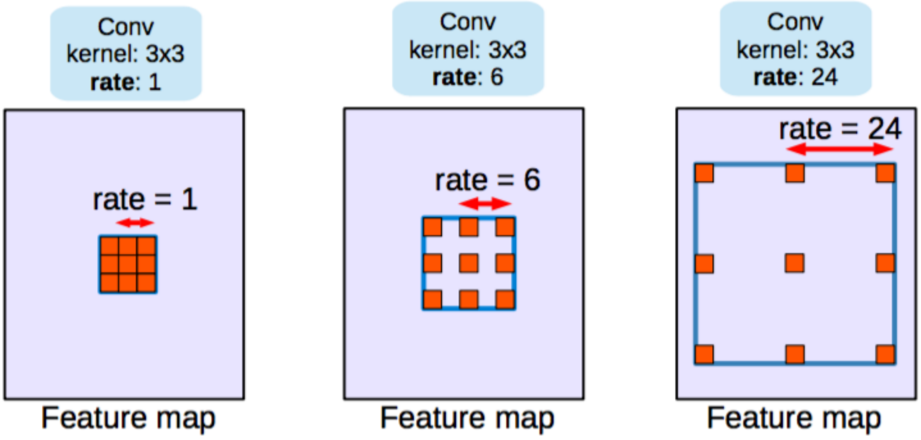
Detail: Use resize (bilinear interpolation) here in order to make sure each pooling level has the same output feature shape before concatenating them.



Model: Deeplabv3_plus

Special Operation: Dilated convolution

Detail: the kernel size will increase by inserting zeros as the dilated rate getting larger.



Super Resolution:

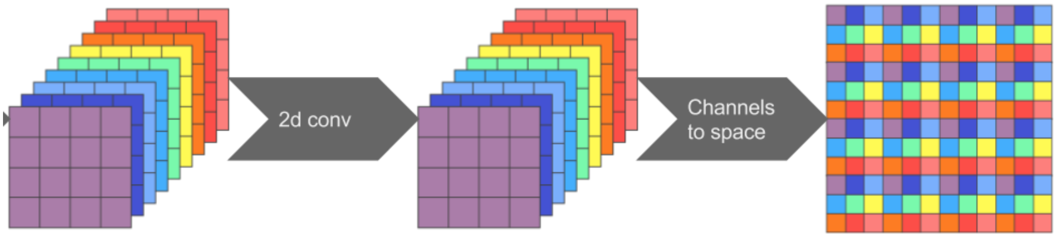
Model: EDSR

1. Special Operation: (Input Image Normalization)
   Detail: (Image – RGB_MEAN)/127.5

2. Special Operation: (Output Image DeNormalization)
   Detail: Image * 127.5 + RGB_MEAN

3. Special Operation: tf.Depth_to_Space(factor)

Detail: 1. Apply factor on conv filter first to enlarge channel
2. Use this Depth_to_Space function to do upsampling



Object Detection:

Model: Detectron Head

Special Operation: Transposed Convolution

Detail: Pad input (blue rectangle) with zeros first, and then do regular convolution.