

In [1]:

```
1 import os
2 import json
3 #import cv2
4 from tqdm import tqdm
5 import numpy as np
6 from glob import glob
7 import tensorflow as tf
8 import matplotlib.pyplot as plt
9 import skimage.io as io
10
11 from tensorflow.keras.callbacks import Callback
12 from tensorflow.keras.callbacks import TensorBoard, ModelCheckpoint, EarlyStopping
13 #from deeplab import DeepLabV3Plus
14
15 print('TensorFlow', tf.__version__)
```

TensorFlow 2.0.0

In [2]:

```
1 img1 = tf.io.read_file("C:/Users/eadhaw/Desktop/1221/5.png")
2 img1 = tf.image.decode_png(img1)
3 print(img1.shape)
```

(333, 500, 3)

In [3]:

```
1 VOC_COLORMAP = [[0, 0, 0], [128, 0, 0], [0, 128, 0], [128, 128, 0],
2                 [0, 0, 128], [128, 0, 128], [0, 128, 128], [128, 128, 128],
3                 [64, 0, 0], [192, 0, 0], [64, 128, 0], [192, 128, 0],
4                 [64, 0, 128], [192, 0, 128], [64, 128, 128], [192, 128, 128],
5                 [0, 64, 0], [128, 64, 0], [0, 192, 0], [128, 192, 0],
6                 [0, 64, 128]]
7 colormap2label = np.zeros(256 ** 3, dtype=np.uint8)
8
```

In [4]:

```
1 for i, colormap in enumerate(VOC_COLORMAP):
2     colormap2label[(colormap[0] * 256 + colormap[1]) * 256 + colormap[2]] = i
3 colormap2label = tf.convert_to_tensor(colormap2label)
4
5 # 本函数已保存在d2lzh_pytorch中方便以后使用(没)
6 def voc_label_indices(colormap, colormap2label):
7     """
8     convert colormap (tf image) to colormap2label (uint8 tensor).
9     """
10    colormap = tf.cast(colormap, dtype=tf.int32)
11    idx = tf.add(tf.multiply(colormap[:, :, 0], 256), colormap[:, :, 1])
12    idx = tf.add(tf.multiply(idx, 256), colormap[:, :, 2])
13    idx = tf.add(idx, colormap[:, :, 2])
14
15    # print(tf.gather_nd(colormap2label, tf.expand_dims(idx, -1)))
16    return tf.gather_nd(colormap2label, tf.expand_dims(idx, -1))
```

In [5]:

```
1 y = voc_label_indices(img1, colormap2label)
```

```
In [6]:
```

```
1 | img1[230:240, 325:330, :]
```

```
<tf.Tensor: id=34, shape=(10, 5, 3), dtype=uint8, numpy=
```

```
array([[ [ 0,  0, 128],  
        [ 0,  0, 128],  
        [ 0,  0, 128],  
        [ 0,  0, 128],  
        [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128],  
   [ 0,  0, 128]],
```

```
[[ [ 0,  0, 128],  
   [ 0,  0, 128],
```

```
[ 0,  0, 128],  
[ 0,  0, 128],  
[ 0,  0, 128]],  
  
[[ 0,  0, 128],  
 [ 0,  0, 128],  
 [ 0,  0, 128],  
 [ 0,  0, 128],  
 [ 0,  0, 128]], dtype=uint8)>
```

```
In [7]:
```

```
1 | y[230:240, 325:330]
```

```
<tf.Tensor: id=38, shape=(10, 5), dtype=uint8, numpy=  
array([[0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0],  
       [0, 0, 0, 0, 0]], dtype=uint8)>
```