56 lines (42 sloc) | 1.74 KB •••

```
1
2   # -*- coding: utf-8 -*-
3   """
4   Created on Fri Jun  7 19:51:21 2019
5
6   @author: ai
7   """
8   from torch import nn
9   from torch.nn import functional as F
10  from torchsummary import summary
11  from efficientnet_pytorch import Effic
12
13
14  params_dict = {
15          # Coefficients:   width,depth,
16          'efficientnet-b0': (1.0, 1.0,
17          'efficientnet-b1': (1.0, 1.1,
18          'efficientnet-b2': (1.1, 1.2,
19          'efficientnet-b3': (1.2, 1.4,
20          'efficientnet-b4': (1.4, 1.8,
21          'efficientnet-b5': (1.6, 2.2,
22          'efficientnet-b6': (1.8, 2.6,
23          'efficientnet-b7': (2.0, 3.1,
24      }
25  class Effect_netI(nn.Module):
26      def __init__(self, num_classes=5,
27          super().__init__()
28          assert num_channels == 3
29          self.model_name= model_name
30          self.device = device
```

```
29            self.model_name= model_name
30            self.device = device
31            #device = torch.device("cuda"
32            self.Efficient = EfficientNet.
33
34
35
36            self.logit = nn.Linear(1000, n
37        def forward(self, inputs):
38            """ Calls extract_features to
39
40            # Convolution layers
41            x = self.Efficient(inputs)
42
43            x = self.logit(x)
44            return x
45    """
46
47
48    device = torch.device("cuda" if torch.
49    Efficient = EfficientNet.from_pretrain
50    feature   = Efficient.extract_features
51    model = Effect_netI(num_classes=196, n
52    summary(model,(3, 300, 300))
53
54    """
55
```