



# Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

milesial / Pytorch-UNet

[Code](#) [Issues 35](#) [Pull requests 1](#) [Discussions](#) [Actions](#) [Security](#) [Insights](#)

master [Pytorch-UNet / unet / unet\\_model.py](#) [Jump to](#)

[Go to file](#)



hushenghao optimize the passed in parameters in the case of bilinear

Latest commit 6234bd1 on Apr 17, 2020 [History](#)

2 contributors

38 lines (32 sloc) | 1.13 KB

[Raw](#) [Blame](#)

```
1  """ Full assembly of the parts to form the complete network """
2
3  import torch.nn.functional as F
4
5  from .unet_parts import *
6
7
8  class UNet(nn.Module):
9      def __init__(self, n_channels, n_classes, bilinear=True):
10         super(UNet, self).__init__()
11         self.n_channels = n_channels
12         self.n_classes = n_classes
13         self.bilinear = bilinear
14
15         self.inc = DoubleConv(n_channels, 64)
16         self.down1 = Down(64, 128)
17         self.down2 = Down(128, 256)
18         self.down3 = Down(256, 512)
19         factor = 2 if bilinear else 1
20         self.down4 = Down(512, 1024 // factor)
21         self.up1 = Up(1024, 512 // factor, bilinear)
22         self.up2 = Up(512, 256 // factor, bilinear)
23         self.up3 = Up(256, 128 // factor, bilinear)
24         self.up4 = Up(128, 64, bilinear)
25         self.outc = OutConv(64, n_classes)
26
27     def forward(self, x):
28         x1 = self.inc(x)
29         x2 = self.down1(x1)
30         x3 = self.down2(x2)
31         x4 = self.down3(x3)
32         x5 = self.down4(x4)
33         x = self.up1(x5, x4)
34         x = self.up2(x, x3)
35         x = self.up3(x, x2)
36         x = self.up4(x, x1)
37         logits = self.outc(x)
38         return logits
```