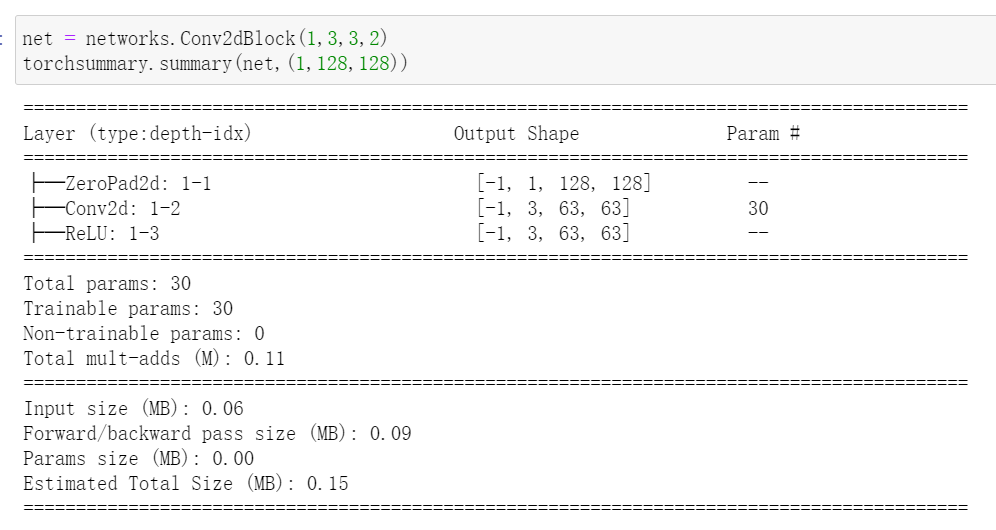
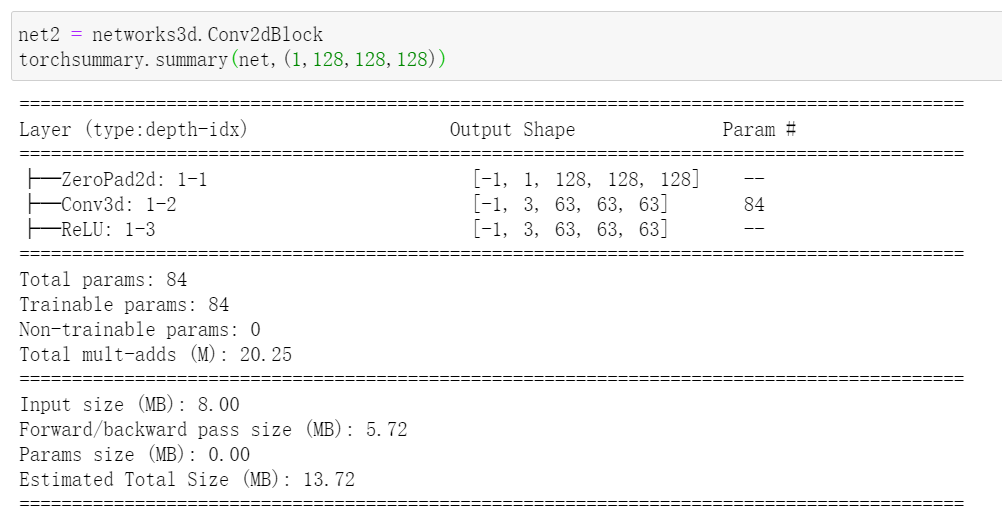
Conv2dBlock 🡪 Conv3dBlock

2D



3D

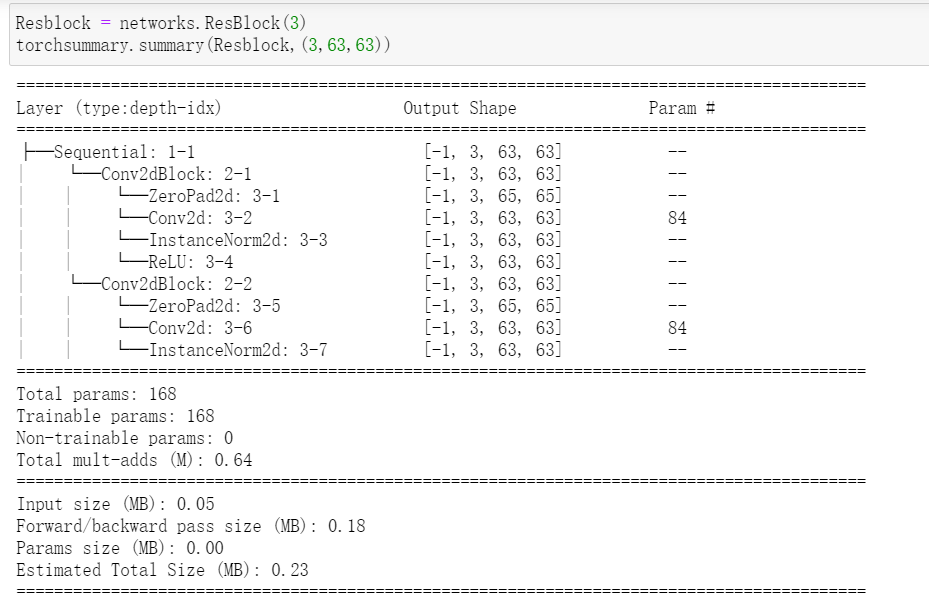


PS:

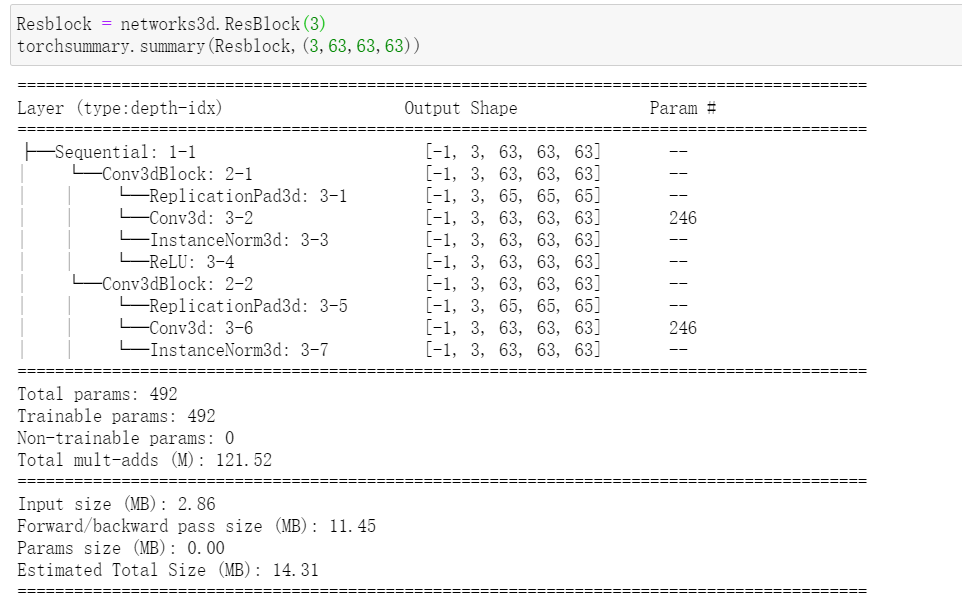
The padding functions in Conv2dBlock like, nn.ReflectionPad2d() have no correspondent pytorch function in 3D, the only pad function works is “replicate”

ResBlock

2D



3D



PS：

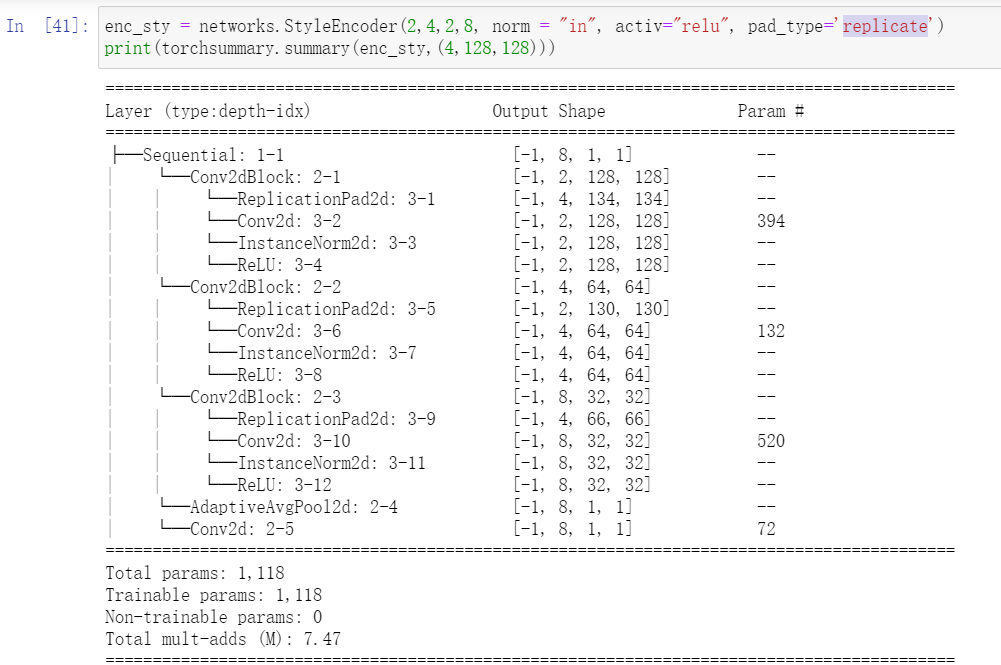
Since ResBlock takes Conv3dBlock, we still need to figure out padding problems in 3D.

Here the input for resBlock (3,63,63,63) is the output of Conv3dBlock

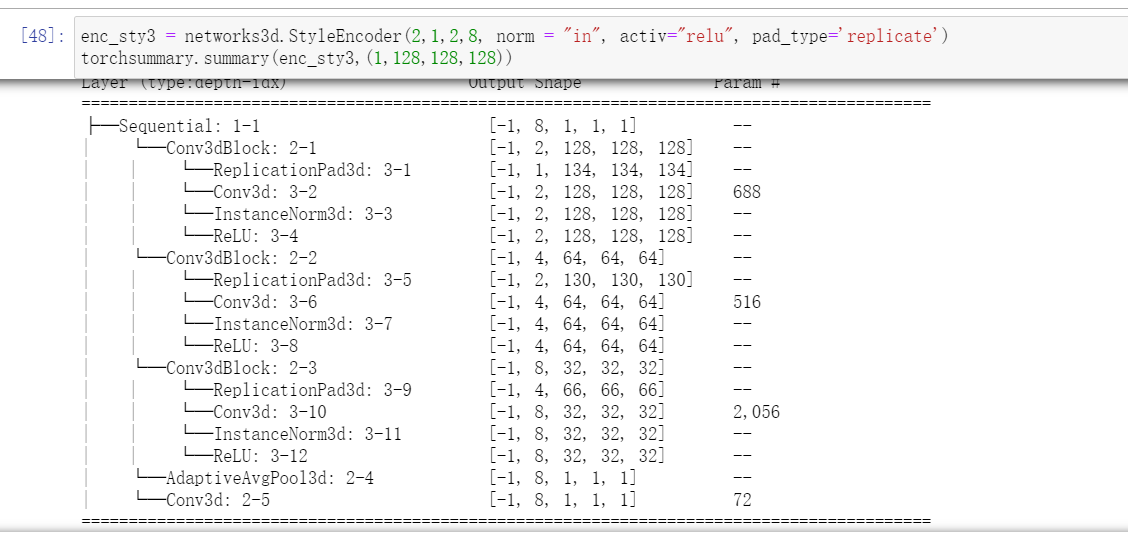
But as long as Conv3dBlock works fine it should be fine

StyleEncoder 🡪 3D

2D (with output style dimension to be 8, input dimension is random set)



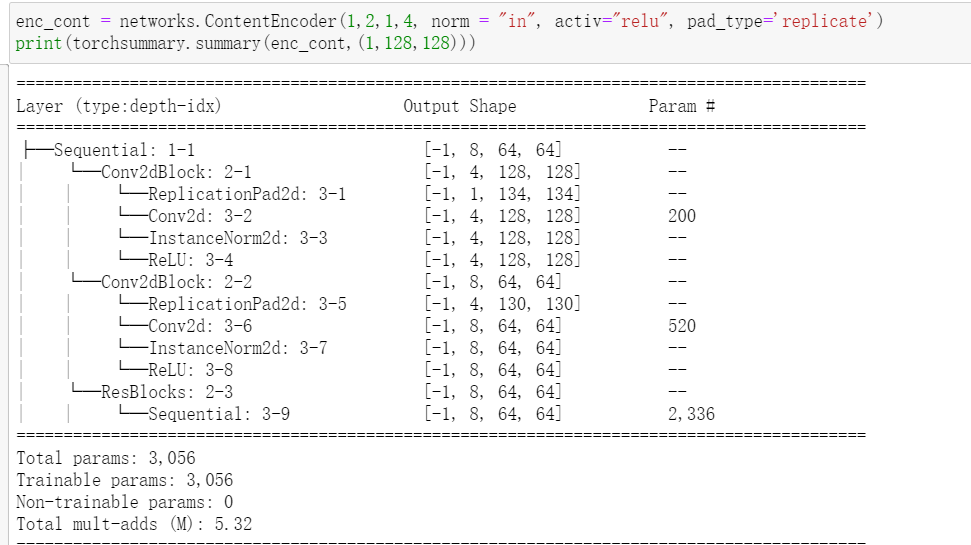
3D



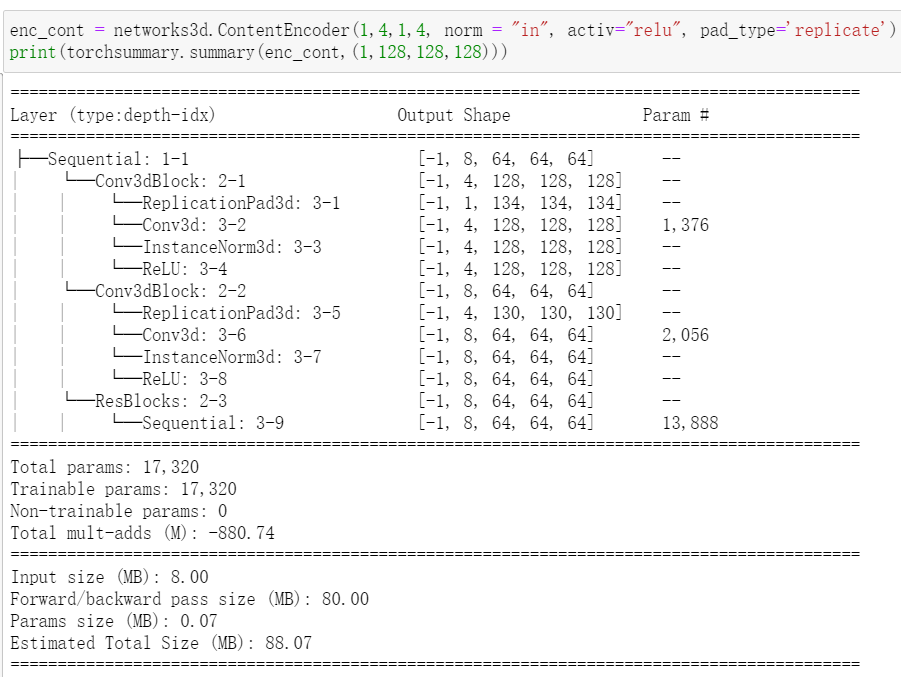
PS: In 2D the output of the content is 8\*1\*1, thus take it for granted that for 3D is 8\*1\*1\*1, one dim higher.

ContentEncoder 🡪 3D

2D:

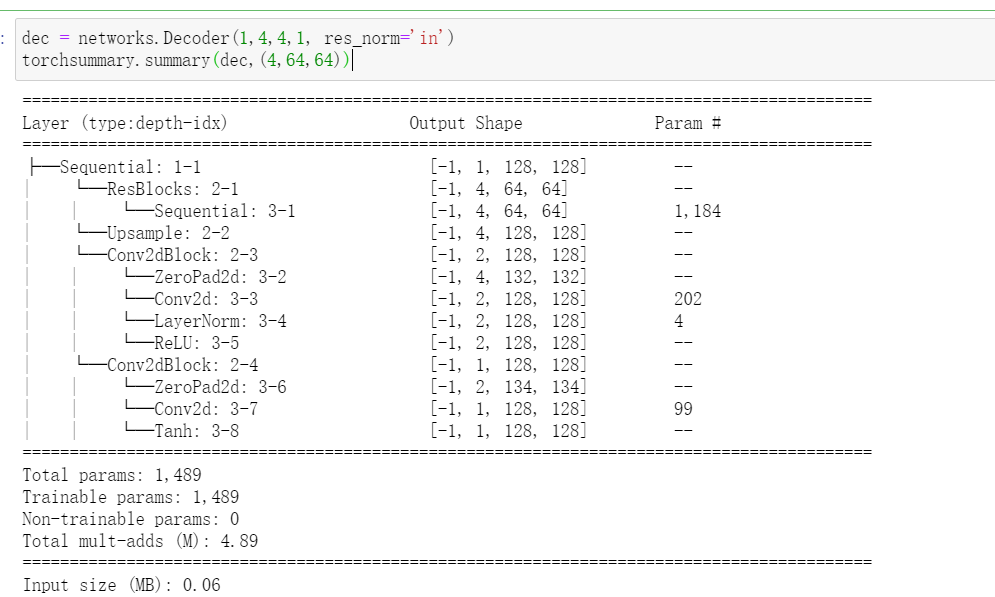


3D

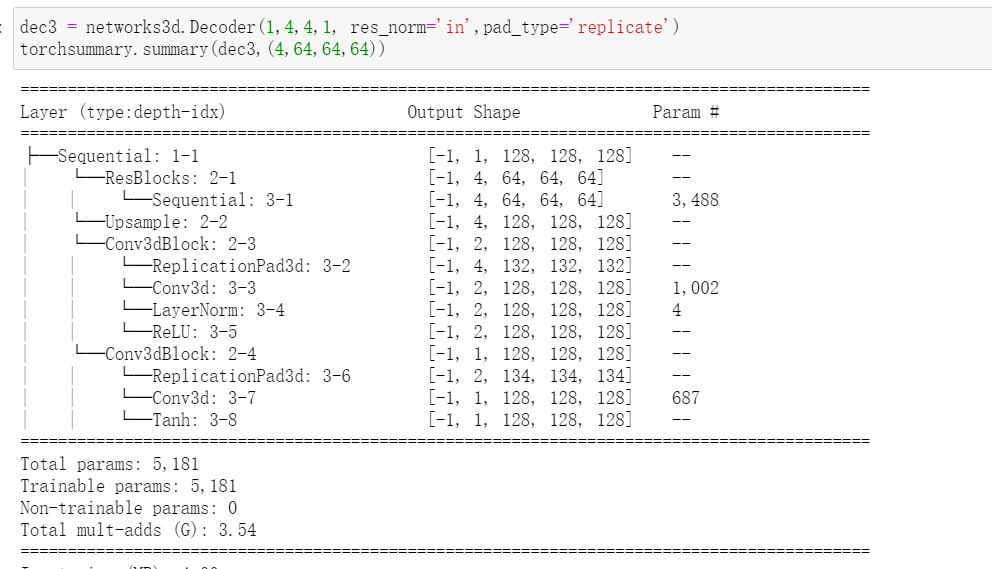


Decoder 🡪 3D

2D



3D:



PS: “Adain” model not modified, thus “res\_norm” is instance normalization, need to change later