Batchsize = 128

Learning rate = 0.002

Epoch = 150

Total time elapsed: 2618.70 seconds

final training accuracy: 0.747297384311805 final training loss: 0.05836192194960619

final validation accuracy: 0.6243192055831255 final validation loss: 0.1387075879252874

final test accuracy: 0.6385904853493467 final test loss: 0.13577914696473342

class Autoencoder\_1\_0(nn.Module):

def \_\_init\_\_(self):

super(Autoencoder\_1\_0, self).\_\_init\_\_()

self.encoder = nn.Sequential(

nn.Conv2d(3, 8, 16, stride=2, padding=1),

nn.LeakyReLU(True),

nn.MaxPool2d(2, stride=2, padding=0),

nn.Conv2d(8, 16, 3, stride=2, padding=1),

nn.LeakyReLU(True),

nn.MaxPool2d(2, stride=2, padding=0),

nn.Conv2d(16, 32, 3, stride=2, padding=1),

nn.LeakyReLU(True),

nn.Conv2d(32, 64, 3),

nn.MaxPool2d(2, stride=2, padding=0),

)

self.decoder = nn.Sequential(

nn.Upsample(scale\_factor=2, mode='nearest'),

nn.ConvTranspose2d(64, 32, 3),

nn.LeakyReLU(True),

nn.ConvTranspose2d(32, 16, 3, stride=2, padding=1, output\_padding=0),

nn.Upsample(scale\_factor=2, mode='nearest'),

nn.LeakyReLU(True),

nn.ConvTranspose2d(16, 8, 3, stride=2, padding=1, output\_padding=1),

nn.Upsample(scale\_factor=2, mode='nearest'),

nn.LeakyReLU(True),

nn.ConvTranspose2d(8, 3, 17, stride=2, padding=0, output\_padding=1),

nn.LeakyReLU(True),

nn.Sigmoid(),

)

def forward(self, x):

x = self.encoder(x)

x = self.decoder(x)

return x