# CNN Segmentation

In [1]:
```
%%shell
# download the Penn-Fudan dataset
wget https://www.cis.upenn.edu/~jshi/ped_html/PennFudanPed.zip

# extract it in the current folder
unzip PennFudanPed.zip
```

```
--2023-04-15 16:43:41--  https://www.cis.upenn.edu/~jshi/ped_html/PennFudanPed.zip
Resolving www.cis.upenn.edu (www.cis.upenn.edu)... 158.130.69.163, 2607:f470:8:64:5ea5::d
Connecting to www.cis.upenn.edu (www.cis.upenn.edu)|158.130.69.163|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53723336 (51M) [application/zip]
Saving to: ‘PennFudanPed.zip’

PennFudanPed.zip    100%[===================>]  51.23M   104MB/s    in 0.5s

2023-04-15 16:43:42 (104 MB/s) - ‘PennFudanPed.zip’ saved [53723336/53723336]

Archive:  PennFudanPed.zip
   creating: PennFudanPed/
  inflating: PennFudanPed/added-object-list.txt
   creating: PennFudanPed/Annotation/
  inflating: PennFudanPed/Annotation/FudanPed00001.txt
  inflating: PennFudanPed/Annotation/FudanPed00002.txt
  inflating: PennFudanPed/Annotation/FudanPed00003.txt
  inflating: PennFudanPed/Annotation/FudanPed00004.txt
  inflating: PennFudanPed/Annotation/FudanPed00005.txt
  inflating: PennFudanPed/Annotation/FudanPed00006.txt
  inflating: PennFudanPed/Annotation/FudanPed00007.txt
  inflating: PennFudanPed/Annotation/FudanPed00008.txt
  inflating: PennFudanPed/Annotation/FudanPed00009.txt
  inflating: PennFudanPed/Annotation/FudanPed00010.txt
  inflating: PennFudanPed/Annotation/FudanPed00011.txt
  inflating: PennFudanPed/Annotation/FudanPed00012.txt
  inflating: PennFudanPed/Annotation/FudanPed00013.txt
  inflating: PennFudanPed/Annotation/FudanPed00014.txt
  inflating: PennFudanPed/Annotation/FudanPed00015.txt
  inflating: PennFudanPed/Annotation/FudanPed00016.txt
  inflating: PennFudanPed/Annotation/FudanPed00017.txt
  inflating: PennFudanPed/Annotation/FudanPed00018.txt
  inflating: PennFudanPed/Annotation/FudanPed00019.txt
  inflating: PennFudanPed/Annotation/FudanPed00020.txt
  inflating: PennFudanPed/Annotation/FudanPed00021.txt
  inflating: PennFudanPed/Annotation/FudanPed00022.txt
  inflating: PennFudanPed/Annotation/FudanPed00023.txt
  inflating: PennFudanPed/Annotation/FudanPed00024.txt
  inflating: PennFudanPed/Annotation/FudanPed00025.txt
  inflating: PennFudanPed/Annotation/FudanPed00026.txt
  inflating: PennFudanPed/Annotation/FudanPed00027.txt
  inflating: PennFudanPed/Annotation/FudanPed00028.txt
  inflating: PennFudanPed/Annotation/FudanPed00029.txt
  inflating: PennFudanPed/Annotation/FudanPed00030.txt
  inflating: PennFudanPed/Annotation/FudanPed00031.txt
  inflating: PennFudanPed/Annotation/FudanPed00032.txt
  inflating: PennFudanPed/Annotation/FudanPed00033.txt
  inflating: PennFudanPed/Annotation/FudanPed00034.txt
  inflating: PennFudanPed/Annotation/FudanPed00035.txt
  inflating: PennFudanPed/Annotation/FudanPed00036.txt
  inflating: PennFudanPed/Annotation/FudanPed00037.txt
  inflating: PennFudanPed/Annotation/FudanPed00038.txt
  inflating: PennFudanPed/Annotation/FudanPed00039.txt
  inflating: PennFudanPed/Annotation/FudanPed00040.txt
  inflating: PennFudanPed/Annotation/FudanPed00041.txt
  inflating: PennFudanPed/Annotation/FudanPed00042.txt
  inflating: PennFudanPed/Annotation/FudanPed00043.txt
  inflating: PennFudanPed/Annotation/FudanPed00044.txt
  inflating: PennFudanPed/Annotation/FudanPed00045.txt
  inflating: PennFudanPed/Annotation/FudanPed00046.txt
  inflating: PennFudanPed/Annotation/FudanPed00047.txt
  inflating: PennFudanPed/Annotation/FudanPed00048.txt
  inflating: PennFudanPed/Annotation/FudanPed00049.txt
  inflating: PennFudanPed/Annotation/FudanPed00050.txt
  inflating: PennFudanPed/Annotation/FudanPed00051.txt
  inflating: PennFudanPed/Annotation/FudanPed00052.txt
  inflating: PennFudanPed/Annotation/FudanPed00053.txt
  inflating: PennFudanPed/Annotation/FudanPed00054.txt
  inflating: PennFudanPed/Annotation/FudanPed00055.txt
  inflating: PennFudanPed/Annotation/FudanPed00056.txt
  inflating: PennFudanPed/Annotation/FudanPed00057.txt
  inflating: PennFudanPed/Annotation/FudanPed00058.txt
  inflating: PennFudanPed/Annotation/FudanPed00059.txt
  inflating: PennFudanPed/Annotation/FudanPed00060.txt
  inflating: PennFudanPed/Annotation/FudanPed00061.txt
  inflating: PennFudanPed/Annotation/FudanPed00062.txt
  inflating: PennFudanPed/Annotation/FudanPed00063.txt
  inflating: PennFudanPed/Annotation/FudanPed00064.txt
  inflating: PennFudanPed/Annotation/FudanPed00065.txt
  inflating: PennFudanPed/Annotation/FudanPed00066.txt
  inflating: PennFudanPed/Annotation/FudanPed00067.txt
  inflating: PennFudanPed/Annotation/FudanPed00068.txt
  inflating: PennFudanPed/Annotation/FudanPed00069.txt
  inflating: PennFudanPed/Annotation/FudanPed00070.txt
  inflating: PennFudanPed/Annotation/FudanPed00071.txt
  inflating: PennFudanPed/Annotation/FudanPed00072.txt
  inflating: PennFudanPed/Annotation/FudanPed00073.txt
  inflating: PennFudanPed/Annotation/FudanPed00074.txt
  inflating: PennFudanPed/Annotation/PennPed00001.txt
  inflating: PennFudanPed/Annotation/PennPed00002.txt
  inflating: PennFudanPed/Annotation/PennPed00003.txt
  inflating: PennFudanPed/Annotation/PennPed00004.txt
```

```
inflating: PennFudanPed/Annotation/PennPed00005.txt
inflating: PennFudanPed/Annotation/PennPed00006.txt
inflating: PennFudanPed/Annotation/PennPed00007.txt
inflating: PennFudanPed/Annotation/PennPed00008.txt
inflating: PennFudanPed/Annotation/PennPed00009.txt
inflating: PennFudanPed/Annotation/PennPed00010.txt
inflating: PennFudanPed/Annotation/PennPed00011.txt
inflating: PennFudanPed/Annotation/PennPed00012.txt
inflating: PennFudanPed/Annotation/PennPed00013.txt
inflating: PennFudanPed/Annotation/PennPed00014.txt
inflating: PennFudanPed/Annotation/PennPed00015.txt
inflating: PennFudanPed/Annotation/PennPed00016.txt
inflating: PennFudanPed/Annotation/PennPed00017.txt
inflating: PennFudanPed/Annotation/PennPed00018.txt
inflating: PennFudanPed/Annotation/PennPed00019.txt
inflating: PennFudanPed/Annotation/PennPed00020.txt
inflating: PennFudanPed/Annotation/PennPed00021.txt
inflating: PennFudanPed/Annotation/PennPed00022.txt
inflating: PennFudanPed/Annotation/PennPed00023.txt
inflating: PennFudanPed/Annotation/PennPed00024.txt
inflating: PennFudanPed/Annotation/PennPed00025.txt
inflating: PennFudanPed/Annotation/PennPed00026.txt
inflating: PennFudanPed/Annotation/PennPed00027.txt
inflating: PennFudanPed/Annotation/PennPed00028.txt
inflating: PennFudanPed/Annotation/PennPed00029.txt
inflating: PennFudanPed/Annotation/PennPed00030.txt
inflating: PennFudanPed/Annotation/PennPed00031.txt
inflating: PennFudanPed/Annotation/PennPed00032.txt
inflating: PennFudanPed/Annotation/PennPed00033.txt
inflating: PennFudanPed/Annotation/PennPed00034.txt
inflating: PennFudanPed/Annotation/PennPed00035.txt
inflating: PennFudanPed/Annotation/PennPed00036.txt
inflating: PennFudanPed/Annotation/PennPed00037.txt
inflating: PennFudanPed/Annotation/PennPed00038.txt
inflating: PennFudanPed/Annotation/PennPed00039.txt
inflating: PennFudanPed/Annotation/PennPed00040.txt
inflating: PennFudanPed/Annotation/PennPed00041.txt
inflating: PennFudanPed/Annotation/PennPed00042.txt
inflating: PennFudanPed/Annotation/PennPed00043.txt
inflating: PennFudanPed/Annotation/PennPed00044.txt
inflating: PennFudanPed/Annotation/PennPed00045.txt
inflating: PennFudanPed/Annotation/PennPed00046.txt
inflating: PennFudanPed/Annotation/PennPed00047.txt
inflating: PennFudanPed/Annotation/PennPed00048.txt
inflating: PennFudanPed/Annotation/PennPed00049.txt
inflating: PennFudanPed/Annotation/PennPed00050.txt
inflating: PennFudanPed/Annotation/PennPed00051.txt
inflating: PennFudanPed/Annotation/PennPed00052.txt
inflating: PennFudanPed/Annotation/PennPed00053.txt
inflating: PennFudanPed/Annotation/PennPed00054.txt
inflating: PennFudanPed/Annotation/PennPed00055.txt
inflating: PennFudanPed/Annotation/PennPed00056.txt
inflating: PennFudanPed/Annotation/PennPed00057.txt
inflating: PennFudanPed/Annotation/PennPed00058.txt
inflating: PennFudanPed/Annotation/PennPed00059.txt
inflating: PennFudanPed/Annotation/PennPed00060.txt
inflating: PennFudanPed/Annotation/PennPed00061.txt
inflating: PennFudanPed/Annotation/PennPed00062.txt
inflating: PennFudanPed/Annotation/PennPed00063.txt
inflating: PennFudanPed/Annotation/PennPed00064.txt
inflating: PennFudanPed/Annotation/PennPed00065.txt
inflating: PennFudanPed/Annotation/PennPed00066.txt
inflating: PennFudanPed/Annotation/PennPed00067.txt
inflating: PennFudanPed/Annotation/PennPed00068.txt
inflating: PennFudanPed/Annotation/PennPed00069.txt
inflating: PennFudanPed/Annotation/PennPed00070.txt
inflating: PennFudanPed/Annotation/PennPed00071.txt
inflating: PennFudanPed/Annotation/PennPed00072.txt
inflating: PennFudanPed/Annotation/PennPed00073.txt
inflating: PennFudanPed/Annotation/PennPed00074.txt
inflating: PennFudanPed/Annotation/PennPed00075.txt
inflating: PennFudanPed/Annotation/PennPed00076.txt
inflating: PennFudanPed/Annotation/PennPed00077.txt
inflating: PennFudanPed/Annotation/PennPed00078.txt
inflating: PennFudanPed/Annotation/PennPed00079.txt
inflating: PennFudanPed/Annotation/PennPed00080.txt
inflating: PennFudanPed/Annotation/PennPed00081.txt
inflating: PennFudanPed/Annotation/PennPed00082.txt
inflating: PennFudanPed/Annotation/PennPed00083.txt
inflating: PennFudanPed/Annotation/PennPed00084.txt
inflating: PennFudanPed/Annotation/PennPed00085.txt
inflating: PennFudanPed/Annotation/PennPed00086.txt
inflating: PennFudanPed/Annotation/PennPed00087.txt
inflating: PennFudanPed/Annotation/PennPed00088.txt
inflating: PennFudanPed/Annotation/PennPed00089.txt
inflating: PennFudanPed/Annotation/PennPed00090.txt
inflating: PennFudanPed/Annotation/PennPed00091.txt
inflating: PennFudanPed/Annotation/PennPed00092.txt
inflating: PennFudanPed/Annotation/PennPed00093.txt
inflating: PennFudanPed/Annotation/PennPed00094.txt
inflating: PennFudanPed/Annotation/PennPed00095.txt
inflating: PennFudanPed/Annotation/PennPed00096.txt
 creating: PennFudanPed/PedMasks/
```

```
  inflating: PennFudanPed/PedMasks/FudanPed00001_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00002_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00003_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00004_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00005_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00006_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00007_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00008_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00009_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00010_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00011_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00012_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00013_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00014_mask.png
 extracting: PennFudanPed/PedMasks/FudanPed00015_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00016_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00017_mask.png
 extracting: PennFudanPed/PedMasks/FudanPed00018_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00019_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00020_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00021_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00022_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00023_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00024_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00025_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00026_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00027_mask.png
 extracting: PennFudanPed/PedMasks/FudanPed00028_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00029_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00030_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00031_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00032_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00033_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00034_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00035_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00036_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00037_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00038_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00039_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00040_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00041_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00042_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00043_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00044_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00045_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00046_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00047_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00048_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00049_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00050_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00051_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00052_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00053_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00054_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00055_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00056_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00057_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00058_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00059_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00060_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00061_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00062_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00063_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00064_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00065_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00066_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00067_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00068_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00069_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00070_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00071_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00072_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00073_mask.png
  inflating: PennFudanPed/PedMasks/FudanPed00074_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00001_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00002_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00003_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00004_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00005_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00006_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00007_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00008_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00009_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00010_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00011_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00012_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00013_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00014_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00015_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00016_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00017_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00018_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00019_mask.png
```

```
  inflating: PennFudanPed/PedMasks/PennPed00020_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00021_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00022_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00023_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00024_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00025_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00026_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00027_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00028_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00029_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00030_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00031_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00032_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00033_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00034_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00035_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00036_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00037_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00038_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00039_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00040_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00041_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00042_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00043_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00044_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00045_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00046_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00047_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00048_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00049_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00050_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00051_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00052_mask.png
extracting: PennFudanPed/PedMasks/PennPed00053_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00054_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00055_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00056_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00057_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00058_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00059_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00060_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00061_mask.png
extracting: PennFudanPed/PedMasks/PennPed00062_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00063_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00064_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00065_mask.png
extracting: PennFudanPed/PedMasks/PennPed00066_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00067_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00068_mask.png
extracting: PennFudanPed/PedMasks/PennPed00069_mask.png
extracting: PennFudanPed/PedMasks/PennPed00070_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00071_mask.png
extracting: PennFudanPed/PedMasks/PennPed00072_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00073_mask.png
extracting: PennFudanPed/PedMasks/PennPed00074_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00075_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00076_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00077_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00078_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00079_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00080_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00081_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00082_mask.png
extracting: PennFudanPed/PedMasks/PennPed00083_mask.png
extracting: PennFudanPed/PedMasks/PennPed00084_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00085_mask.png
extracting: PennFudanPed/PedMasks/PennPed00086_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00087_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00088_mask.png
extracting: PennFudanPed/PedMasks/PennPed00089_mask.png
extracting: PennFudanPed/PedMasks/PennPed00090_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00091_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00092_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00093_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00094_mask.png
  inflating: PennFudanPed/PedMasks/PennPed00095_mask.png
extracting: PennFudanPed/PedMasks/PennPed00096_mask.png
   creating: PennFudanPed/PNGImages/
  inflating: PennFudanPed/PNGImages/FudanPed00001.png
  inflating: PennFudanPed/PNGImages/FudanPed00002.png
  inflating: PennFudanPed/PNGImages/FudanPed00003.png
  inflating: PennFudanPed/PNGImages/FudanPed00004.png
  inflating: PennFudanPed/PNGImages/FudanPed00005.png
  inflating: PennFudanPed/PNGImages/FudanPed00006.png
  inflating: PennFudanPed/PNGImages/FudanPed00007.png
  inflating: PennFudanPed/PNGImages/FudanPed00008.png
  inflating: PennFudanPed/PNGImages/FudanPed00009.png
  inflating: PennFudanPed/PNGImages/FudanPed00010.png
  inflating: PennFudanPed/PNGImages/FudanPed00011.png
  inflating: PennFudanPed/PNGImages/FudanPed00012.png
  inflating: PennFudanPed/PNGImages/FudanPed00013.png
  inflating: PennFudanPed/PNGImages/FudanPed00014.png
  inflating: PennFudanPed/PNGImages/FudanPed00015.png
```

```
inflating: PennFudanPed/PNGImages/FudanPed00016.png
inflating: PennFudanPed/PNGImages/FudanPed00017.png
inflating: PennFudanPed/PNGImages/FudanPed00018.png
inflating: PennFudanPed/PNGImages/FudanPed00019.png
inflating: PennFudanPed/PNGImages/FudanPed00020.png
inflating: PennFudanPed/PNGImages/FudanPed00021.png
inflating: PennFudanPed/PNGImages/FudanPed00022.png
inflating: PennFudanPed/PNGImages/FudanPed00023.png
inflating: PennFudanPed/PNGImages/FudanPed00024.png
inflating: PennFudanPed/PNGImages/FudanPed00025.png
inflating: PennFudanPed/PNGImages/FudanPed00026.png
inflating: PennFudanPed/PNGImages/FudanPed00027.png
inflating: PennFudanPed/PNGImages/FudanPed00028.png
inflating: PennFudanPed/PNGImages/FudanPed00029.png
inflating: PennFudanPed/PNGImages/FudanPed00030.png
inflating: PennFudanPed/PNGImages/FudanPed00031.png
inflating: PennFudanPed/PNGImages/FudanPed00032.png
inflating: PennFudanPed/PNGImages/FudanPed00033.png
inflating: PennFudanPed/PNGImages/FudanPed00034.png
inflating: PennFudanPed/PNGImages/FudanPed00035.png
inflating: PennFudanPed/PNGImages/FudanPed00036.png
inflating: PennFudanPed/PNGImages/FudanPed00037.png
inflating: PennFudanPed/PNGImages/FudanPed00038.png
inflating: PennFudanPed/PNGImages/FudanPed00039.png
inflating: PennFudanPed/PNGImages/FudanPed00040.png
inflating: PennFudanPed/PNGImages/FudanPed00041.png
inflating: PennFudanPed/PNGImages/FudanPed00042.png
inflating: PennFudanPed/PNGImages/FudanPed00043.png
inflating: PennFudanPed/PNGImages/FudanPed00044.png
inflating: PennFudanPed/PNGImages/FudanPed00045.png
inflating: PennFudanPed/PNGImages/FudanPed00046.png
inflating: PennFudanPed/PNGImages/FudanPed00047.png
inflating: PennFudanPed/PNGImages/FudanPed00048.png
inflating: PennFudanPed/PNGImages/FudanPed00049.png
inflating: PennFudanPed/PNGImages/FudanPed00050.png
inflating: PennFudanPed/PNGImages/FudanPed00051.png
inflating: PennFudanPed/PNGImages/FudanPed00052.png
inflating: PennFudanPed/PNGImages/FudanPed00053.png
inflating: PennFudanPed/PNGImages/FudanPed00054.png
inflating: PennFudanPed/PNGImages/FudanPed00055.png
inflating: PennFudanPed/PNGImages/FudanPed00056.png
inflating: PennFudanPed/PNGImages/FudanPed00057.png
inflating: PennFudanPed/PNGImages/FudanPed00058.png
inflating: PennFudanPed/PNGImages/FudanPed00059.png
inflating: PennFudanPed/PNGImages/FudanPed00060.png
inflating: PennFudanPed/PNGImages/FudanPed00061.png
inflating: PennFudanPed/PNGImages/FudanPed00062.png
inflating: PennFudanPed/PNGImages/FudanPed00063.png
inflating: PennFudanPed/PNGImages/FudanPed00064.png
inflating: PennFudanPed/PNGImages/FudanPed00065.png
inflating: PennFudanPed/PNGImages/FudanPed00066.png
inflating: PennFudanPed/PNGImages/FudanPed00067.png
inflating: PennFudanPed/PNGImages/FudanPed00068.png
inflating: PennFudanPed/PNGImages/FudanPed00069.png
inflating: PennFudanPed/PNGImages/FudanPed00070.png
inflating: PennFudanPed/PNGImages/FudanPed00071.png
inflating: PennFudanPed/PNGImages/FudanPed00072.png
inflating: PennFudanPed/PNGImages/FudanPed00073.png
inflating: PennFudanPed/PNGImages/FudanPed00074.png
inflating: PennFudanPed/PNGImages/PennPed00001.png
inflating: PennFudanPed/PNGImages/PennPed00002.png
inflating: PennFudanPed/PNGImages/PennPed00003.png
inflating: PennFudanPed/PNGImages/PennPed00004.png
inflating: PennFudanPed/PNGImages/PennPed00005.png
inflating: PennFudanPed/PNGImages/PennPed00006.png
inflating: PennFudanPed/PNGImages/PennPed00007.png
inflating: PennFudanPed/PNGImages/PennPed00008.png
inflating: PennFudanPed/PNGImages/PennPed00009.png
inflating: PennFudanPed/PNGImages/PennPed00010.png
inflating: PennFudanPed/PNGImages/PennPed00011.png
inflating: PennFudanPed/PNGImages/PennPed00012.png
inflating: PennFudanPed/PNGImages/PennPed00013.png
inflating: PennFudanPed/PNGImages/PennPed00014.png
inflating: PennFudanPed/PNGImages/PennPed00015.png
inflating: PennFudanPed/PNGImages/PennPed00016.png
inflating: PennFudanPed/PNGImages/PennPed00017.png
inflating: PennFudanPed/PNGImages/PennPed00018.png
inflating: PennFudanPed/PNGImages/PennPed00019.png
inflating: PennFudanPed/PNGImages/PennPed00020.png
inflating: PennFudanPed/PNGImages/PennPed00021.png
inflating: PennFudanPed/PNGImages/PennPed00022.png
inflating: PennFudanPed/PNGImages/PennPed00023.png
inflating: PennFudanPed/PNGImages/PennPed00024.png
inflating: PennFudanPed/PNGImages/PennPed00025.png
inflating: PennFudanPed/PNGImages/PennPed00026.png
inflating: PennFudanPed/PNGImages/PennPed00027.png
inflating: PennFudanPed/PNGImages/PennPed00028.png
inflating: PennFudanPed/PNGImages/PennPed00029.png
inflating: PennFudanPed/PNGImages/PennPed00030.png
inflating: PennFudanPed/PNGImages/PennPed00031.png
inflating: PennFudanPed/PNGImages/PennPed00032.png
inflating: PennFudanPed/PNGImages/PennPed00033.png
inflating: PennFudanPed/PNGImages/PennPed00034.png
```

```
inflating: PennFudanPed/PNGImages/PennPed00035.png
inflating: PennFudanPed/PNGImages/PennPed00036.png
inflating: PennFudanPed/PNGImages/PennPed00037.png
inflating: PennFudanPed/PNGImages/PennPed00038.png
inflating: PennFudanPed/PNGImages/PennPed00039.png
inflating: PennFudanPed/PNGImages/PennPed00040.png
inflating: PennFudanPed/PNGImages/PennPed00041.png
inflating: PennFudanPed/PNGImages/PennPed00042.png
inflating: PennFudanPed/PNGImages/PennPed00043.png
inflating: PennFudanPed/PNGImages/PennPed00044.png
inflating: PennFudanPed/PNGImages/PennPed00045.png
inflating: PennFudanPed/PNGImages/PennPed00046.png
inflating: PennFudanPed/PNGImages/PennPed00047.png
inflating: PennFudanPed/PNGImages/PennPed00048.png
inflating: PennFudanPed/PNGImages/PennPed00049.png
inflating: PennFudanPed/PNGImages/PennPed00050.png
inflating: PennFudanPed/PNGImages/PennPed00051.png
inflating: PennFudanPed/PNGImages/PennPed00052.png
inflating: PennFudanPed/PNGImages/PennPed00053.png
inflating: PennFudanPed/PNGImages/PennPed00054.png
inflating: PennFudanPed/PNGImages/PennPed00055.png
inflating: PennFudanPed/PNGImages/PennPed00056.png
inflating: PennFudanPed/PNGImages/PennPed00057.png
inflating: PennFudanPed/PNGImages/PennPed00058.png
inflating: PennFudanPed/PNGImages/PennPed00059.png
inflating: PennFudanPed/PNGImages/PennPed00060.png
inflating: PennFudanPed/PNGImages/PennPed00061.png
inflating: PennFudanPed/PNGImages/PennPed00062.png
inflating: PennFudanPed/PNGImages/PennPed00063.png
inflating: PennFudanPed/PNGImages/PennPed00064.png
inflating: PennFudanPed/PNGImages/PennPed00065.png
inflating: PennFudanPed/PNGImages/PennPed00066.png
inflating: PennFudanPed/PNGImages/PennPed00067.png
inflating: PennFudanPed/PNGImages/PennPed00068.png
inflating: PennFudanPed/PNGImages/PennPed00069.png
inflating: PennFudanPed/PNGImages/PennPed00070.png
inflating: PennFudanPed/PNGImages/PennPed00071.png
inflating: PennFudanPed/PNGImages/PennPed00072.png
inflating: PennFudanPed/PNGImages/PennPed00073.png
inflating: PennFudanPed/PNGImages/PennPed00074.png
inflating: PennFudanPed/PNGImages/PennPed00075.png
inflating: PennFudanPed/PNGImages/PennPed00076.png
inflating: PennFudanPed/PNGImages/PennPed00077.png
inflating: PennFudanPed/PNGImages/PennPed00078.png
inflating: PennFudanPed/PNGImages/PennPed00079.png
inflating: PennFudanPed/PNGImages/PennPed00080.png
inflating: PennFudanPed/PNGImages/PennPed00081.png
inflating: PennFudanPed/PNGImages/PennPed00082.png
inflating: PennFudanPed/PNGImages/PennPed00083.png
inflating: PennFudanPed/PNGImages/PennPed00084.png
inflating: PennFudanPed/PNGImages/PennPed00085.png
inflating: PennFudanPed/PNGImages/PennPed00086.png
inflating: PennFudanPed/PNGImages/PennPed00087.png
inflating: PennFudanPed/PNGImages/PennPed00088.png
inflating: PennFudanPed/PNGImages/PennPed00089.png
inflating: PennFudanPed/PNGImages/PennPed00090.png
inflating: PennFudanPed/PNGImages/PennPed00091.png
inflating: PennFudanPed/PNGImages/PennPed00092.png
inflating: PennFudanPed/PNGImages/PennPed00093.png
inflating: PennFudanPed/PNGImages/PennPed00094.png
inflating: PennFudanPed/PNGImages/PennPed00095.png
inflating: PennFudanPed/PNGImages/PennPed00096.png
inflating: PennFudanPed/readme.txt
```

Out[1]:

In [2]:
```
import os
import numpy as np
import torch
from PIL import Image

import torch
import torch.nn as nn
import torch.nn.functional as F
import random
import sys
import os
from optparse import OptionParser
from torch import optim
from torch.autograd import Function, Variable
import matplotlib.pyplot as plt
import torchvision.transforms.functional as TF
import torch.optim.lr_scheduler as lr_scheduler
import torchvision.transforms as transforms
import torchvision.datasets as datasets
from torch.utils.data import Dataset, DataLoader, SubsetRandomSampler
```

In [3]:
```
class PennFudanDataset(torch.utils.data.Dataset):
    def __init__(self, root, transforms):
        self.root = root
        self.transforms = transforms
        # load all image files, sorting them to
```

```
        # ensure that they are aligned
        self.imgs = list(sorted(os.listdir(os.path.join(root, "PNGImages"))))
        self.masks = list(sorted(os.listdir(os.path.join(root, "PedMasks"))))

    def __getitem__(self, idx):
        # load images and masks
        img_path = os.path.join(self.root, "PNGImages", self.imgs[idx])
        mask_path = os.path.join(self.root, "PedMasks", self.masks[idx])
        img = Image.open(img_path).convert("RGB")
        # print(img.size)
        newsize = ((128, 128))
        img = img.resize(newsize)
        # print("new: ", img.size)

        mask = Image.open(mask_path)
        # print(img.size)
        newsize = ((128, 128))
        mask = mask.resize(newsize)

        mask = np.array(mask)
        mask[mask >= 1] = 255    #Late Submission due to this. In assignment, it given to assign value 1 but with 1 it

        if self.transforms is not None:     #Data Augmentation
            img, mask = self.transforms(img, mask)

        return img, mask

    def __len__(self):
        return len(self.imgs)
```

In [4]:
```
%%shell

# Download TorchVision repo to use some files from
# references/detection
# git clone https://github.com/pytorch/vision.git
# cd vision
# git checkout v0.8.2
```

Out[4]:

# Data Augmentation

In [5]:
```
class MyCompose(object):
    def __init__(self, transforms):
        self.transforms = transforms

    def __call__(self, img, tar):
        for t in self.transforms:
            if isinstance(t, transforms.transforms.RandomHorizontalFlip):
              if random.random() > 0.5:
                img = TF.hflip(img)
                tar = TF.hflip(tar)
            elif isinstance(t, transforms.transforms.RandomVerticalFlip):
              if random.random() > 0.5:
                img = TF.vflip(img)
                tar = TF.vflip(tar)
            elif isinstance(t, transforms.transforms.ToTensor):
              img = TF.to_tensor(img)
              tar = TF.to_tensor(tar)
        return img, tar
```

In [6]:
```
import torchvision.transforms as T

def get_transform(train):
    transforms = []
    transforms.append(T.ToTensor())
    if train:
        transforms.append(T.RandomHorizontalFlip(0.5))
        transforms.append(T.RandomVerticalFlip(0.5))
    return MyCompose(transforms)
```

# UNet Architecture

In [7]:
```
class DoubleConvBlock(nn.Module):
  def __init__(self, in_ch, out_ch):
      super(DoubleConvBlock, self).__init__()
      self.in_ch = in_ch
      self.out_ch = out_ch
      self.conv1 = nn.Sequential(
          nn.Conv2d(self.in_ch, self.out_ch, kernel_size =3, padding = 1),
          nn.BatchNorm2d(out_ch),
          nn.ReLU()
      )
      self.conv2 = nn.Sequential(
          nn.Conv2d(self.out_ch, self.out_ch, kernel_size =3, padding = 1),
          nn.BatchNorm2d(out_ch),
          nn.ReLU()
```

```
        )

    def forward(self, x):
        x = self.conv1(x)
        x = self.conv2(x)
        return x
```

In [8]:
```
class DownSample(nn.Module):
    def __init__(self, in_ch, out_ch):
        super(DownSample, self).__init__()
        self.down = nn.Sequential(
          nn.MaxPool2d(kernel_size=2, stride=2),
          DoubleConvBlock(in_ch, out_ch)
        )

    def forward(self, x):
        x = self.down(x)
        return x
```

In [9]:
```
class UpSample(nn.Module):
    def __init__(self, in_ch, out_ch):
        super(UpSample, self).__init__()
        self.up = nn.Upsample(scale_factor=2, mode='bilinear')
        self.conv = DoubleConvBlock(in_ch, out_ch)

    def forward(self, x1, x2):
        x1 = self.up(x1)

        # diffY = x2.size()[2] - x1.size()[2]
        # diffX = x2.size()[3] - x1.size()[3]
        # x1 = F.pad(x1, (diffX // 2, diffX - diffX//2,
        #                 diffY // 2, diffY - diffY//2))

        x = torch.cat([x2, x1], dim=1)
        x = self.conv(x)
        return x
```

In [10]:
```
class SingleConv(nn.Module):
    def __init__(self, in_ch, out_ch):
        super(SingleConv, self).__init__()
        self.conv = nn.Conv2d(in_ch, out_ch, kernel_size = 3, padding = 1)

    def forward(self, x):
        return torch.sigmoid(self.conv(x))
```

In [11]:
```
class UNet(nn.Module):
    def __init__(self, n_channels):
        super(UNet, self).__init__()

        #Encoder
        self.initial = DoubleConvBlock(n_channels, 16)
        self.down1 = DownSample(16, 32)
        self.down2 = DownSample(32, 32)

        #Decoder
        self.up1 = UpSample(64, 16)
        self.up2 = UpSample(32, 16)
        self.end = SingleConv(16, 1)

    def forward(self, x):
        x1 = self.initial(x)
        x2 = self.down1(x1)
        x3 = self.down2(x2)

        x = self.up1(x3, x2)
        x = self.up2(x, x1)
        x = self.end(x)
        return x
```

# Split Data into train, val and test as 80, 10 and 10

In [12]:
```
dataset = PennFudanDataset('PennFudanPed', get_transform(train=True))
dataset_test = PennFudanDataset('PennFudanPed', get_transform(train=False))

dataset_length = len(dataset)

# define split sizes
train_size = int(0.8 * dataset_length)
val_size = int(0.1 * dataset_length)
test_size = dataset_length - train_size - val_size

# define samplers for each split
indices = list(range(dataset_length))
np.random.shuffle(indices)
train_indices = indices[:train_size]
val_indices = indices[train_size:(train_size+val_size)]
test_indices = indices[(train_size+val_size):]

train_sampler = SubsetRandomSampler(train_indices)
```

```
val_sampler = SubsetRandomSampler(val_indices)
test_sampler = SubsetRandomSampler(test_indices)

# create data loaders for each split
train_loader = DataLoader(dataset, batch_size=8, sampler=train_sampler)
val_loader = DataLoader(dataset_test, batch_size=8, sampler=val_sampler)
test_loader = DataLoader(dataset_test, batch_size=8, sampler=test_sampler)
```

In [13]:
```
class SoftDiceLoss(nn.Module):
    def __init__(self, eps):
        super(SoftDiceLoss, self).__init__()
        self.eps = eps

    def forward(self, outputs, targets):
        outputs = outputs.view(outputs.size(0), -1).float()
        targets = targets.view(targets.size(0), -1).float()
        intersection = torch.sum(outputs * targets, dim =1)

        dice = ((2.*intersection + self.eps)/(torch.sum(outputs + targets, dim = 1) + self.eps))
        loss = 1 - dice

        return loss.mean()
```

In [14]:
```
def DiceScore(outputs, targets, eps = 1):
  outputs = outputs.view(outputs.size(0), -1).float()
  targets = targets.view(targets.size(0), -1).float()
  intersection = torch.sum(outputs * targets, dim =1)
  dice = ((2.*intersection + eps)/(torch.sum(outputs + targets, dim = 1) + eps))
  return dice.mean().item()
```

In [15]:
```
from torchsummary import summary
model = UNet(3)
# summary(model, (3, 128, 128))
model
```

Out[15]:
```
UNet(
  (initial): DoubleConvBlock(
    (conv1): Sequential(
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
    )
    (conv2): Sequential(
      (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU()
    )
  )
  (down1): DownSample(
    (down): Sequential(
      (0): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (1): DoubleConvBlock(
        (conv1): Sequential(
          (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU()
        )
        (conv2): Sequential(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU()
        )
      )
    )
  )
  (down2): DownSample(
    (down): Sequential(
      (0): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (1): DoubleConvBlock(
        (conv1): Sequential(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU()
        )
        (conv2): Sequential(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU()
        )
      )
    )
  )
  (up1): UpSample(
    (up): Upsample(scale_factor=2.0, mode='bilinear')
    (conv): DoubleConvBlock(
      (conv1): Sequential(
        (0): Conv2d(64, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
      (conv2): Sequential(
        (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
    )
  )
  (up2): UpSample(
    (up): Upsample(scale_factor=2.0, mode='bilinear')
    (conv): DoubleConvBlock(
      (conv1): Sequential(
        (0): Conv2d(32, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
      (conv2): Sequential(
        (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU()
      )
    )
  )
  (end): SingleConv(
    (conv): Conv2d(16, 1, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  )
)
```

In [16]:
```
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
# scheduler = lr_scheduler.LinearLR(optimizer, start_factor=1e-3, end_factor=1e-5, total_iters=300)
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
criterion = SoftDiceLoss(1)
# criterion = SoftDICELoss()
# criterion = nn.BCELoss()
# criterion_dice = SoftDiceLoss(1)
model = model.to(device)
criterion = criterion.to(device)
# criterion_dice = criterion.to(device)
```

```
import torchsummary
def count_parameters(model):
    return sum(p.numel() for p in model.parameters() if p.requires_grad)

count_parameters(model)
```

Out[16]:
```
54241
```

In [18]:
```
import numpy as np

def train_epoch(model, iterator, optimizer, criterion, device):

    epoch_loss = 0
    epoch_acc = 0
    # epoch_dice_loss = 0

    model.train()

    for (x, y) in iterator:

        x = x.to(device)
        y = y.to(device)


        y_pred = model(x)
        optimizer.zero_grad()
        loss = criterion(y_pred, y)

        loss.backward()
        optimizer.step()
        epoch_loss += loss.item()
        return epoch_loss / len(iterator)
```

In [19]:
```
def evaluate(model, iterator, criterion, device):

    epoch_loss = 0
    epoch_acc = 0
    # epoch_dice_loss = 0
    epoch_dice_score = 0

    model.eval()

    with torch.no_grad():

        for (x, y) in iterator:

            x = x.to(device)
            y = y.to(device)

            y_pred = model(x)
            loss = criterion(y_pred, y)

            epoch_loss += loss.item()

            epoch_dice_score += DiceScore(y_pred, y)


    return epoch_loss / len(iterator),  epoch_dice_score / len(iterator)
```

In [20]:
```
EPOCHS = 350

best_validation_loss = float('inf')
val_dice = []
train_loss_epoch = []
val_loss_epoch = []

for epoch in range(EPOCHS):
  train_loss = train_epoch(model, train_loader, optimizer, criterion, device)
  validation_loss, validation_dice_score = evaluate(model, val_loader, criterion, device)
  val_dice.append(validation_dice_score)
  train_loss_epoch.append(train_loss)
  val_loss_epoch.append(validation_loss)
  # scheduler.step()

  if validation_loss<best_validation_loss:
    torch.save(model, "best_model.pt")
    best_validation_loss=validation_loss

  print (f"Epoch: {epoch+1} \ Training Loss={train_loss:.6f}")
  print (f"Epoch: {epoch+1} \ Validation Loss={validation_loss:.6f}" )
```

```
Epoch: 1 \ Training Loss=0.045445
Epoch: 1 \ Validation Loss=0.722169
Epoch: 2 \ Training Loss=0.040170
Epoch: 2 \ Validation Loss=0.751290
Epoch: 3 \ Training Loss=0.041301
Epoch: 3 \ Validation Loss=0.770028
Epoch: 4 \ Training Loss=0.044263
Epoch: 4 \ Validation Loss=0.707547
Epoch: 5 \ Training Loss=0.040462
Epoch: 5 \ Validation Loss=0.682091
Epoch: 6 \ Training Loss=0.039891
Epoch: 6 \ Validation Loss=0.730319
Epoch: 7 \ Training Loss=0.039566
Epoch: 7 \ Validation Loss=0.723730
Epoch: 8 \ Training Loss=0.034846
Epoch: 8 \ Validation Loss=0.766135
Epoch: 9 \ Training Loss=0.036314
Epoch: 9 \ Validation Loss=0.762760
Epoch: 10 \ Training Loss=0.042168
Epoch: 10 \ Validation Loss=0.768130
Epoch: 11 \ Training Loss=0.039686
Epoch: 11 \ Validation Loss=0.758401
Epoch: 12 \ Training Loss=0.039586
Epoch: 12 \ Validation Loss=0.767358
Epoch: 13 \ Training Loss=0.037132
Epoch: 13 \ Validation Loss=0.698912
Epoch: 14 \ Training Loss=0.037293
Epoch: 14 \ Validation Loss=0.708754
Epoch: 15 \ Training Loss=0.033901
Epoch: 15 \ Validation Loss=0.784987
Epoch: 16 \ Training Loss=0.039053
Epoch: 16 \ Validation Loss=0.733913
Epoch: 17 \ Training Loss=0.036054
Epoch: 17 \ Validation Loss=0.738816
Epoch: 18 \ Training Loss=0.039116
Epoch: 18 \ Validation Loss=0.724627
Epoch: 19 \ Training Loss=0.033758
Epoch: 19 \ Validation Loss=0.737409
Epoch: 20 \ Training Loss=0.032923
Epoch: 20 \ Validation Loss=0.742369
Epoch: 21 \ Training Loss=0.035295
Epoch: 21 \ Validation Loss=0.790289
Epoch: 22 \ Training Loss=0.036050
Epoch: 22 \ Validation Loss=0.743395
Epoch: 23 \ Training Loss=0.036117
Epoch: 23 \ Validation Loss=0.766087
Epoch: 24 \ Training Loss=0.030284
Epoch: 24 \ Validation Loss=0.784676
Epoch: 25 \ Training Loss=0.033734
Epoch: 25 \ Validation Loss=0.795389
Epoch: 26 \ Training Loss=0.033107
Epoch: 26 \ Validation Loss=0.802256
Epoch: 27 \ Training Loss=0.027899
Epoch: 27 \ Validation Loss=0.820116
Epoch: 28 \ Training Loss=0.028924
Epoch: 28 \ Validation Loss=0.798464
Epoch: 29 \ Training Loss=0.028623
Epoch: 29 \ Validation Loss=0.801036
Epoch: 30 \ Training Loss=0.029641
Epoch: 30 \ Validation Loss=0.800850
Epoch: 31 \ Training Loss=0.029021
Epoch: 31 \ Validation Loss=0.816819
Epoch: 32 \ Training Loss=0.030073
Epoch: 32 \ Validation Loss=0.752029
Epoch: 33 \ Training Loss=0.030895
Epoch: 33 \ Validation Loss=0.782547
Epoch: 34 \ Training Loss=0.030736
Epoch: 34 \ Validation Loss=0.753331
Epoch: 35 \ Training Loss=0.031198
Epoch: 35 \ Validation Loss=0.647486
Epoch: 36 \ Training Loss=0.031970
Epoch: 36 \ Validation Loss=0.660632
Epoch: 37 \ Training Loss=0.028062
Epoch: 37 \ Validation Loss=0.611385
Epoch: 38 \ Training Loss=0.029210
Epoch: 38 \ Validation Loss=0.682858
Epoch: 39 \ Training Loss=0.033606
Epoch: 39 \ Validation Loss=0.655456
Epoch: 40 \ Training Loss=0.032077
Epoch: 40 \ Validation Loss=0.682054
Epoch: 41 \ Training Loss=0.034747
Epoch: 41 \ Validation Loss=0.573574
Epoch: 42 \ Training Loss=0.028907
Epoch: 42 \ Validation Loss=0.655038
Epoch: 43 \ Training Loss=0.033445
Epoch: 43 \ Validation Loss=0.571773
Epoch: 44 \ Training Loss=0.028297
Epoch: 44 \ Validation Loss=0.543617
Epoch: 45 \ Training Loss=0.026246
Epoch: 45 \ Validation Loss=0.636340
Epoch: 46 \ Training Loss=0.028729
Epoch: 46 \ Validation Loss=0.649630
Epoch: 47 \ Training Loss=0.025989
```

```
Epoch: 47 \ Validation Loss=0.619764
Epoch: 48 \ Training Loss=0.030037
Epoch: 48 \ Validation Loss=0.620844
Epoch: 49 \ Training Loss=0.033717
Epoch: 49 \ Validation Loss=0.572335
Epoch: 50 \ Training Loss=0.028378
Epoch: 50 \ Validation Loss=0.485998
Epoch: 51 \ Training Loss=0.029864
Epoch: 51 \ Validation Loss=0.489634
Epoch: 52 \ Training Loss=0.026190
Epoch: 52 \ Validation Loss=0.578032
Epoch: 53 \ Training Loss=0.026749
Epoch: 53 \ Validation Loss=0.516618
Epoch: 54 \ Training Loss=0.022408
Epoch: 54 \ Validation Loss=0.538934
Epoch: 55 \ Training Loss=0.030622
Epoch: 55 \ Validation Loss=0.520703
Epoch: 56 \ Training Loss=0.026240
Epoch: 56 \ Validation Loss=0.400309
Epoch: 57 \ Training Loss=0.025018
Epoch: 57 \ Validation Loss=0.508227
Epoch: 58 \ Training Loss=0.023171
Epoch: 58 \ Validation Loss=0.447494
Epoch: 59 \ Training Loss=0.032459
Epoch: 59 \ Validation Loss=0.471080
Epoch: 60 \ Training Loss=0.023971
Epoch: 60 \ Validation Loss=0.474417
Epoch: 61 \ Training Loss=0.027531
Epoch: 61 \ Validation Loss=0.538789
Epoch: 62 \ Training Loss=0.023725
Epoch: 62 \ Validation Loss=0.418527
Epoch: 63 \ Training Loss=0.030262
Epoch: 63 \ Validation Loss=0.399670
Epoch: 64 \ Training Loss=0.028506
Epoch: 64 \ Validation Loss=0.392124
Epoch: 65 \ Training Loss=0.027353
Epoch: 65 \ Validation Loss=0.379696
Epoch: 66 \ Training Loss=0.027456
Epoch: 66 \ Validation Loss=0.435089
Epoch: 67 \ Training Loss=0.026729
Epoch: 67 \ Validation Loss=0.428497
Epoch: 68 \ Training Loss=0.024936
Epoch: 68 \ Validation Loss=0.421832
Epoch: 69 \ Training Loss=0.025178
Epoch: 69 \ Validation Loss=0.413964
Epoch: 70 \ Training Loss=0.024586
Epoch: 70 \ Validation Loss=0.492546
Epoch: 71 \ Training Loss=0.027162
Epoch: 71 \ Validation Loss=0.510254
Epoch: 72 \ Training Loss=0.021592
Epoch: 72 \ Validation Loss=0.407703
Epoch: 73 \ Training Loss=0.016407
Epoch: 73 \ Validation Loss=0.508081
Epoch: 74 \ Training Loss=0.021498
Epoch: 74 \ Validation Loss=0.373698
Epoch: 75 \ Training Loss=0.025989
Epoch: 75 \ Validation Loss=0.520348
Epoch: 76 \ Training Loss=0.025346
Epoch: 76 \ Validation Loss=0.458871
Epoch: 77 \ Training Loss=0.019010
Epoch: 77 \ Validation Loss=0.497524
Epoch: 78 \ Training Loss=0.022783
Epoch: 78 \ Validation Loss=0.478387
Epoch: 79 \ Training Loss=0.023391
Epoch: 79 \ Validation Loss=0.468633
Epoch: 80 \ Training Loss=0.018910
Epoch: 80 \ Validation Loss=0.511367
Epoch: 81 \ Training Loss=0.023917
Epoch: 81 \ Validation Loss=0.433899
Epoch: 82 \ Training Loss=0.018614
Epoch: 82 \ Validation Loss=0.456825
Epoch: 83 \ Training Loss=0.024093
Epoch: 83 \ Validation Loss=0.448565
Epoch: 84 \ Training Loss=0.021159
Epoch: 84 \ Validation Loss=0.485732
Epoch: 85 \ Training Loss=0.020273
Epoch: 85 \ Validation Loss=0.427085
Epoch: 86 \ Training Loss=0.021436
Epoch: 86 \ Validation Loss=0.500588
Epoch: 87 \ Training Loss=0.022252
Epoch: 87 \ Validation Loss=0.425212
Epoch: 88 \ Training Loss=0.029037
Epoch: 88 \ Validation Loss=0.420918
Epoch: 89 \ Training Loss=0.024319
Epoch: 89 \ Validation Loss=0.438870
Epoch: 90 \ Training Loss=0.019727
Epoch: 90 \ Validation Loss=0.450768
Epoch: 91 \ Training Loss=0.020653
Epoch: 91 \ Validation Loss=0.394233
Epoch: 92 \ Training Loss=0.021819
Epoch: 92 \ Validation Loss=0.477889
Epoch: 93 \ Training Loss=0.023517
Epoch: 93 \ Validation Loss=0.423853
```

```
Epoch: 94 \ Training Loss=0.017001
Epoch: 94 \ Validation Loss=0.384067
Epoch: 95 \ Training Loss=0.021316
Epoch: 95 \ Validation Loss=0.414489
Epoch: 96 \ Training Loss=0.020584
Epoch: 96 \ Validation Loss=0.410956
Epoch: 97 \ Training Loss=0.021496
Epoch: 97 \ Validation Loss=0.439098
Epoch: 98 \ Training Loss=0.018896
Epoch: 98 \ Validation Loss=0.472003
Epoch: 99 \ Training Loss=0.024994
Epoch: 99 \ Validation Loss=0.416973
Epoch: 100 \ Training Loss=0.022319
Epoch: 100 \ Validation Loss=0.373571
Epoch: 101 \ Training Loss=0.017420
Epoch: 101 \ Validation Loss=0.413679
Epoch: 102 \ Training Loss=0.017923
Epoch: 102 \ Validation Loss=0.386694
Epoch: 103 \ Training Loss=0.023415
Epoch: 103 \ Validation Loss=0.371156
Epoch: 104 \ Training Loss=0.019554
Epoch: 104 \ Validation Loss=0.506182
Epoch: 105 \ Training Loss=0.019642
Epoch: 105 \ Validation Loss=0.392924
Epoch: 106 \ Training Loss=0.020394
Epoch: 106 \ Validation Loss=0.424838
Epoch: 107 \ Training Loss=0.020282
Epoch: 107 \ Validation Loss=0.370090
Epoch: 108 \ Training Loss=0.024602
Epoch: 108 \ Validation Loss=0.389490
Epoch: 109 \ Training Loss=0.017377
Epoch: 109 \ Validation Loss=0.324842
Epoch: 110 \ Training Loss=0.017569
Epoch: 110 \ Validation Loss=0.411819
Epoch: 111 \ Training Loss=0.018064
Epoch: 111 \ Validation Loss=0.412094
Epoch: 112 \ Training Loss=0.018800
Epoch: 112 \ Validation Loss=0.378611
Epoch: 113 \ Training Loss=0.018048
Epoch: 113 \ Validation Loss=0.394850
Epoch: 114 \ Training Loss=0.021441
Epoch: 114 \ Validation Loss=0.416389
Epoch: 115 \ Training Loss=0.020703
Epoch: 115 \ Validation Loss=0.425448
Epoch: 116 \ Training Loss=0.017164
Epoch: 116 \ Validation Loss=0.458954
Epoch: 117 \ Training Loss=0.021024
Epoch: 117 \ Validation Loss=0.419092
Epoch: 118 \ Training Loss=0.016876
Epoch: 118 \ Validation Loss=0.417663
Epoch: 119 \ Training Loss=0.021051
Epoch: 119 \ Validation Loss=0.412679
Epoch: 120 \ Training Loss=0.025487
Epoch: 120 \ Validation Loss=0.464578
Epoch: 121 \ Training Loss=0.020031
Epoch: 121 \ Validation Loss=0.437716
Epoch: 122 \ Training Loss=0.021970
Epoch: 122 \ Validation Loss=0.445134
Epoch: 123 \ Training Loss=0.021777
Epoch: 123 \ Validation Loss=0.524354
Epoch: 124 \ Training Loss=0.027444
Epoch: 124 \ Validation Loss=0.408205
Epoch: 125 \ Training Loss=0.018636
Epoch: 125 \ Validation Loss=0.540653
Epoch: 126 \ Training Loss=0.018442
Epoch: 126 \ Validation Loss=0.403018
Epoch: 127 \ Training Loss=0.021413
Epoch: 127 \ Validation Loss=0.406536
Epoch: 128 \ Training Loss=0.022301
Epoch: 128 \ Validation Loss=0.455194
Epoch: 129 \ Training Loss=0.016834
Epoch: 129 \ Validation Loss=0.467367
Epoch: 130 \ Training Loss=0.022736
Epoch: 130 \ Validation Loss=0.592005
Epoch: 131 \ Training Loss=0.022708
Epoch: 131 \ Validation Loss=0.511217
Epoch: 132 \ Training Loss=0.024211
Epoch: 132 \ Validation Loss=0.488557
Epoch: 133 \ Training Loss=0.020181
Epoch: 133 \ Validation Loss=0.417032
Epoch: 134 \ Training Loss=0.020818
Epoch: 134 \ Validation Loss=0.339230
Epoch: 135 \ Training Loss=0.019296
Epoch: 135 \ Validation Loss=0.423581
Epoch: 136 \ Training Loss=0.023280
Epoch: 136 \ Validation Loss=0.573360
Epoch: 137 \ Training Loss=0.018537
Epoch: 137 \ Validation Loss=0.413448
Epoch: 138 \ Training Loss=0.019257
Epoch: 138 \ Validation Loss=0.401994
Epoch: 139 \ Training Loss=0.019007
Epoch: 139 \ Validation Loss=0.372294
Epoch: 140 \ Training Loss=0.020022
```

```
Epoch: 140 \ Validation Loss=0.518514
Epoch: 141 \ Training Loss=0.020293
Epoch: 141 \ Validation Loss=0.664319
Epoch: 142 \ Training Loss=0.020922
Epoch: 142 \ Validation Loss=0.532302
Epoch: 143 \ Training Loss=0.022501
Epoch: 143 \ Validation Loss=0.543485
Epoch: 144 \ Training Loss=0.017300
Epoch: 144 \ Validation Loss=0.481857
Epoch: 145 \ Training Loss=0.025244
Epoch: 145 \ Validation Loss=0.387890
Epoch: 146 \ Training Loss=0.022464
Epoch: 146 \ Validation Loss=0.389828
Epoch: 147 \ Training Loss=0.017680
Epoch: 147 \ Validation Loss=0.295274
Epoch: 148 \ Training Loss=0.018387
Epoch: 148 \ Validation Loss=0.351327
Epoch: 149 \ Training Loss=0.018331
Epoch: 149 \ Validation Loss=0.409704
Epoch: 150 \ Training Loss=0.020559
Epoch: 150 \ Validation Loss=0.454662
Epoch: 151 \ Training Loss=0.021428
Epoch: 151 \ Validation Loss=0.365003
Epoch: 152 \ Training Loss=0.020433
Epoch: 152 \ Validation Loss=0.438665
Epoch: 153 \ Training Loss=0.018838
Epoch: 153 \ Validation Loss=0.355951
Epoch: 154 \ Training Loss=0.018725
Epoch: 154 \ Validation Loss=0.403164
Epoch: 155 \ Training Loss=0.016619
Epoch: 155 \ Validation Loss=0.324612
Epoch: 156 \ Training Loss=0.022945
Epoch: 156 \ Validation Loss=0.355467
Epoch: 157 \ Training Loss=0.022349
Epoch: 157 \ Validation Loss=0.388688
Epoch: 158 \ Training Loss=0.018742
Epoch: 158 \ Validation Loss=0.355932
Epoch: 159 \ Training Loss=0.019197
Epoch: 159 \ Validation Loss=0.453076
Epoch: 160 \ Training Loss=0.023365
Epoch: 160 \ Validation Loss=0.348507
Epoch: 161 \ Training Loss=0.013870
Epoch: 161 \ Validation Loss=0.323203
Epoch: 162 \ Training Loss=0.015792
Epoch: 162 \ Validation Loss=0.453650
Epoch: 163 \ Training Loss=0.017337
Epoch: 163 \ Validation Loss=0.358142
Epoch: 164 \ Training Loss=0.017063
Epoch: 164 \ Validation Loss=0.393672
Epoch: 165 \ Training Loss=0.015424
Epoch: 165 \ Validation Loss=0.376506
Epoch: 166 \ Training Loss=0.016091
Epoch: 166 \ Validation Loss=0.385078
Epoch: 167 \ Training Loss=0.015481
Epoch: 167 \ Validation Loss=0.425733
Epoch: 168 \ Training Loss=0.017362
Epoch: 168 \ Validation Loss=0.323471
Epoch: 169 \ Training Loss=0.017612
Epoch: 169 \ Validation Loss=0.383775
Epoch: 170 \ Training Loss=0.020815
Epoch: 170 \ Validation Loss=0.316351
Epoch: 171 \ Training Loss=0.019230
Epoch: 171 \ Validation Loss=0.297861
Epoch: 172 \ Training Loss=0.016501
Epoch: 172 \ Validation Loss=0.350708
Epoch: 173 \ Training Loss=0.016174
Epoch: 173 \ Validation Loss=0.354419
Epoch: 174 \ Training Loss=0.019547
Epoch: 174 \ Validation Loss=0.385268
Epoch: 175 \ Training Loss=0.015942
Epoch: 175 \ Validation Loss=0.459194
Epoch: 176 \ Training Loss=0.017932
Epoch: 176 \ Validation Loss=0.349080
Epoch: 177 \ Training Loss=0.017446
Epoch: 177 \ Validation Loss=0.378419
Epoch: 178 \ Training Loss=0.017726
Epoch: 178 \ Validation Loss=0.339379
Epoch: 179 \ Training Loss=0.015714
Epoch: 179 \ Validation Loss=0.314773
Epoch: 180 \ Training Loss=0.018585
Epoch: 180 \ Validation Loss=0.335382
Epoch: 181 \ Training Loss=0.018285
Epoch: 181 \ Validation Loss=0.345353
Epoch: 182 \ Training Loss=0.017504
Epoch: 182 \ Validation Loss=0.377297
Epoch: 183 \ Training Loss=0.014919
Epoch: 183 \ Validation Loss=0.336582
Epoch: 184 \ Training Loss=0.014564
Epoch: 184 \ Validation Loss=0.308044
Epoch: 185 \ Training Loss=0.017785
Epoch: 185 \ Validation Loss=0.465487
Epoch: 186 \ Training Loss=0.014921
Epoch: 186 \ Validation Loss=0.336813
```

```
Epoch: 187 \ Training Loss=0.015786
Epoch: 187 \ Validation Loss=0.316578
Epoch: 188 \ Training Loss=0.017270
Epoch: 188 \ Validation Loss=0.330079
Epoch: 189 \ Training Loss=0.017385
Epoch: 189 \ Validation Loss=0.436908
Epoch: 190 \ Training Loss=0.017184
Epoch: 190 \ Validation Loss=0.332114
Epoch: 191 \ Training Loss=0.014950
Epoch: 191 \ Validation Loss=0.374539
Epoch: 192 \ Training Loss=0.021173
Epoch: 192 \ Validation Loss=0.366167
Epoch: 193 \ Training Loss=0.015245
Epoch: 193 \ Validation Loss=0.375583
Epoch: 194 \ Training Loss=0.016730
Epoch: 194 \ Validation Loss=0.318709
Epoch: 195 \ Training Loss=0.014981
Epoch: 195 \ Validation Loss=0.287563
Epoch: 196 \ Training Loss=0.019906
Epoch: 196 \ Validation Loss=0.465944
Epoch: 197 \ Training Loss=0.017481
Epoch: 197 \ Validation Loss=0.311204
Epoch: 198 \ Training Loss=0.016315
Epoch: 198 \ Validation Loss=0.295460
Epoch: 199 \ Training Loss=0.015693
Epoch: 199 \ Validation Loss=0.310499
Epoch: 200 \ Training Loss=0.020229
Epoch: 200 \ Validation Loss=0.375161
Epoch: 201 \ Training Loss=0.019034
Epoch: 201 \ Validation Loss=0.302183
Epoch: 202 \ Training Loss=0.019369
Epoch: 202 \ Validation Loss=0.388304
Epoch: 203 \ Training Loss=0.020571
Epoch: 203 \ Validation Loss=0.405770
Epoch: 204 \ Training Loss=0.018799
Epoch: 204 \ Validation Loss=0.304450
Epoch: 205 \ Training Loss=0.016561
Epoch: 205 \ Validation Loss=0.309041
Epoch: 206 \ Training Loss=0.009684
Epoch: 206 \ Validation Loss=0.310447
Epoch: 207 \ Training Loss=0.014702
Epoch: 207 \ Validation Loss=0.458230
Epoch: 208 \ Training Loss=0.017340
Epoch: 208 \ Validation Loss=0.315591
Epoch: 209 \ Training Loss=0.016766
Epoch: 209 \ Validation Loss=0.300169
Epoch: 210 \ Training Loss=0.019554
Epoch: 210 \ Validation Loss=0.347219
Epoch: 211 \ Training Loss=0.011741
Epoch: 211 \ Validation Loss=0.308891
Epoch: 212 \ Training Loss=0.016662
Epoch: 212 \ Validation Loss=0.457663
Epoch: 213 \ Training Loss=0.017049
Epoch: 213 \ Validation Loss=0.313088
Epoch: 214 \ Training Loss=0.015398
Epoch: 214 \ Validation Loss=0.365379
Epoch: 215 \ Training Loss=0.017435
Epoch: 215 \ Validation Loss=0.326434
Epoch: 216 \ Training Loss=0.019187
Epoch: 216 \ Validation Loss=0.370010
Epoch: 217 \ Training Loss=0.014779
Epoch: 217 \ Validation Loss=0.401464
Epoch: 218 \ Training Loss=0.019651
Epoch: 218 \ Validation Loss=0.488498
Epoch: 219 \ Training Loss=0.019490
Epoch: 219 \ Validation Loss=0.332317
Epoch: 220 \ Training Loss=0.015391
Epoch: 220 \ Validation Loss=0.301711
Epoch: 221 \ Training Loss=0.017765
Epoch: 221 \ Validation Loss=0.258205
Epoch: 222 \ Training Loss=0.012703
Epoch: 222 \ Validation Loss=0.356655
Epoch: 223 \ Training Loss=0.017417
Epoch: 223 \ Validation Loss=0.333729
Epoch: 224 \ Training Loss=0.018741
Epoch: 224 \ Validation Loss=0.430465
Epoch: 225 \ Training Loss=0.022951
Epoch: 225 \ Validation Loss=0.299548
Epoch: 226 \ Training Loss=0.020321
Epoch: 226 \ Validation Loss=0.284114
Epoch: 227 \ Training Loss=0.016179
Epoch: 227 \ Validation Loss=0.417564
Epoch: 228 \ Training Loss=0.022883
Epoch: 228 \ Validation Loss=0.340542
Epoch: 229 \ Training Loss=0.016656
Epoch: 229 \ Validation Loss=0.395258
Epoch: 230 \ Training Loss=0.014851
Epoch: 230 \ Validation Loss=0.338092
Epoch: 231 \ Training Loss=0.026675
Epoch: 231 \ Validation Loss=0.379583
Epoch: 232 \ Training Loss=0.019721
Epoch: 232 \ Validation Loss=0.305516
Epoch: 233 \ Training Loss=0.018010
```

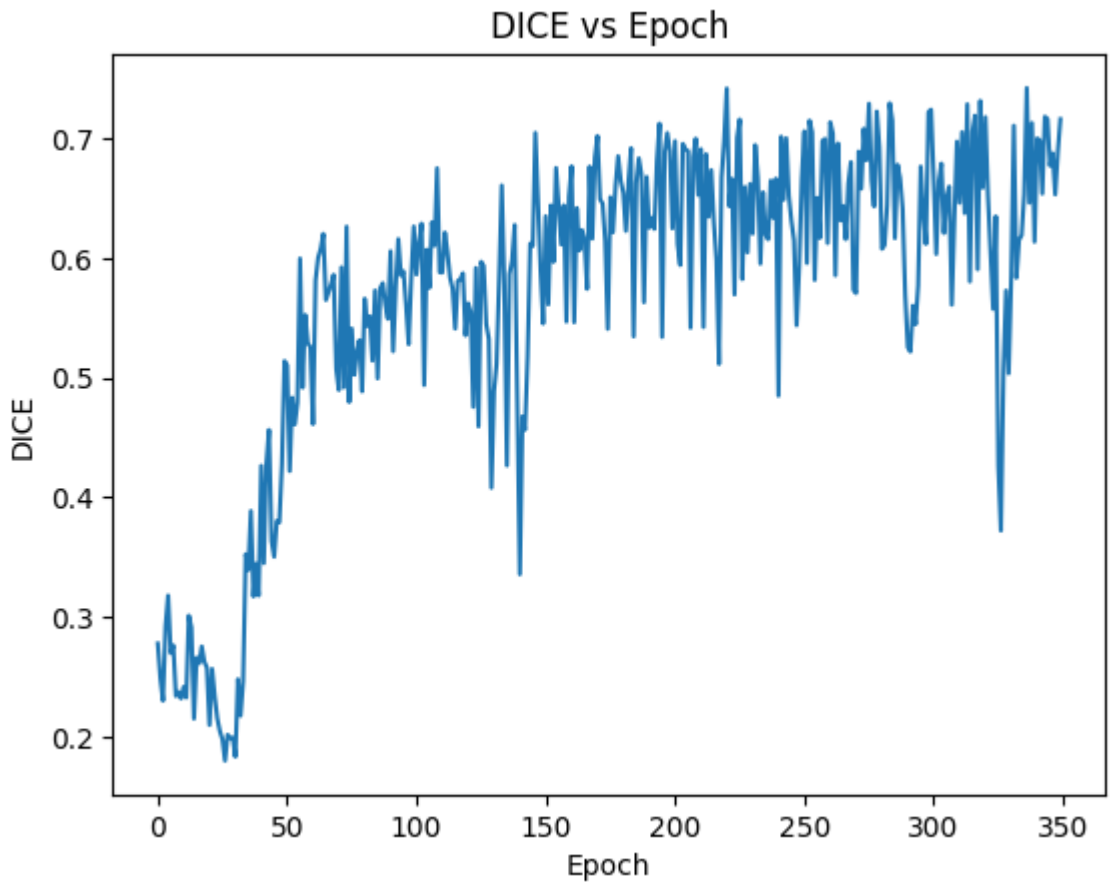
```
Epoch: 233 \ Validation Loss=0.332995
Epoch: 234 \ Training Loss=0.016347
Epoch: 234 \ Validation Loss=0.404654
Epoch: 235 \ Training Loss=0.017474
Epoch: 235 \ Validation Loss=0.344923
Epoch: 236 \ Training Loss=0.021054
Epoch: 236 \ Validation Loss=0.380017
Epoch: 237 \ Training Loss=0.019876
Epoch: 237 \ Validation Loss=0.384299
Epoch: 238 \ Training Loss=0.016420
Epoch: 238 \ Validation Loss=0.335224
Epoch: 239 \ Training Loss=0.020054
Epoch: 239 \ Validation Loss=0.366344
Epoch: 240 \ Training Loss=0.015745
Epoch: 240 \ Validation Loss=0.333552
Epoch: 241 \ Training Loss=0.016119
Epoch: 241 \ Validation Loss=0.515041
Epoch: 242 \ Training Loss=0.017549
Epoch: 242 \ Validation Loss=0.298486
Epoch: 243 \ Training Loss=0.014408
Epoch: 243 \ Validation Loss=0.351206
Epoch: 244 \ Training Loss=0.016390
Epoch: 244 \ Validation Loss=0.299756
Epoch: 245 \ Training Loss=0.015320
Epoch: 245 \ Validation Loss=0.347949
Epoch: 246 \ Training Loss=0.014240
Epoch: 246 \ Validation Loss=0.368895
Epoch: 247 \ Training Loss=0.020925
Epoch: 247 \ Validation Loss=0.385043
Epoch: 248 \ Training Loss=0.017659
Epoch: 248 \ Validation Loss=0.456172
Epoch: 249 \ Training Loss=0.019534
Epoch: 249 \ Validation Loss=0.406651
Epoch: 250 \ Training Loss=0.016662
Epoch: 250 \ Validation Loss=0.333532
Epoch: 251 \ Training Loss=0.015429
Epoch: 251 \ Validation Loss=0.293986
Epoch: 252 \ Training Loss=0.020094
Epoch: 252 \ Validation Loss=0.404172
Epoch: 253 \ Training Loss=0.015749
Epoch: 253 \ Validation Loss=0.285046
Epoch: 254 \ Training Loss=0.015789
Epoch: 254 \ Validation Loss=0.294798
Epoch: 255 \ Training Loss=0.013915
Epoch: 255 \ Validation Loss=0.418295
Epoch: 256 \ Training Loss=0.016158
Epoch: 256 \ Validation Loss=0.349633
Epoch: 257 \ Training Loss=0.016922
Epoch: 257 \ Validation Loss=0.383218
Epoch: 258 \ Training Loss=0.016361
Epoch: 258 \ Validation Loss=0.301912
Epoch: 259 \ Training Loss=0.020826
Epoch: 259 \ Validation Loss=0.300363
Epoch: 260 \ Training Loss=0.015716
Epoch: 260 \ Validation Loss=0.387445
Epoch: 261 \ Training Loss=0.013748
Epoch: 261 \ Validation Loss=0.286386
Epoch: 262 \ Training Loss=0.013676
Epoch: 262 \ Validation Loss=0.295815
Epoch: 263 \ Training Loss=0.016407
Epoch: 263 \ Validation Loss=0.414429
Epoch: 264 \ Training Loss=0.013089
Epoch: 264 \ Validation Loss=0.304166
Epoch: 265 \ Training Loss=0.017538
Epoch: 265 \ Validation Loss=0.368479
Epoch: 266 \ Training Loss=0.013068
Epoch: 266 \ Validation Loss=0.356416
Epoch: 267 \ Training Loss=0.016252
Epoch: 267 \ Validation Loss=0.384197
Epoch: 268 \ Training Loss=0.010996
Epoch: 268 \ Validation Loss=0.337790
Epoch: 269 \ Training Loss=0.011066
Epoch: 269 \ Validation Loss=0.319744
Epoch: 270 \ Training Loss=0.016049
Epoch: 270 \ Validation Loss=0.426539
Epoch: 271 \ Training Loss=0.015647
Epoch: 271 \ Validation Loss=0.429299
Epoch: 272 \ Training Loss=0.017863
Epoch: 272 \ Validation Loss=0.310931
Epoch: 273 \ Training Loss=0.012790
Epoch: 273 \ Validation Loss=0.341975
Epoch: 274 \ Training Loss=0.015974
Epoch: 274 \ Validation Loss=0.291873
Epoch: 275 \ Training Loss=0.017025
Epoch: 275 \ Validation Loss=0.318286
Epoch: 276 \ Training Loss=0.018081
Epoch: 276 \ Validation Loss=0.270886
Epoch: 277 \ Training Loss=0.018317
Epoch: 277 \ Validation Loss=0.335060
Epoch: 278 \ Training Loss=0.014750
Epoch: 278 \ Validation Loss=0.356800
Epoch: 279 \ Training Loss=0.014573
Epoch: 279 \ Validation Loss=0.277478
```

```
Epoch: 280 \ Training Loss=0.014689
Epoch: 280 \ Validation Loss=0.307420
Epoch: 281 \ Training Loss=0.011403
Epoch: 281 \ Validation Loss=0.392228
Epoch: 282 \ Training Loss=0.011328
Epoch: 282 \ Validation Loss=0.389154
Epoch: 283 \ Training Loss=0.013380
Epoch: 283 \ Validation Loss=0.356217
Epoch: 284 \ Training Loss=0.016860
Epoch: 284 \ Validation Loss=0.270533
Epoch: 285 \ Training Loss=0.014482
Epoch: 285 \ Validation Loss=0.284888
Epoch: 286 \ Training Loss=0.011748
Epoch: 286 \ Validation Loss=0.383344
Epoch: 287 \ Training Loss=0.018128
Epoch: 287 \ Validation Loss=0.321964
Epoch: 288 \ Training Loss=0.014801
Epoch: 288 \ Validation Loss=0.333053
Epoch: 289 \ Training Loss=0.018221
Epoch: 289 \ Validation Loss=0.358809
Epoch: 290 \ Training Loss=0.019442
Epoch: 290 \ Validation Loss=0.431028
Epoch: 291 \ Training Loss=0.013921
Epoch: 291 \ Validation Loss=0.474448
Epoch: 292 \ Training Loss=0.018710
Epoch: 292 \ Validation Loss=0.478174
Epoch: 293 \ Training Loss=0.013737
Epoch: 293 \ Validation Loss=0.439776
Epoch: 294 \ Training Loss=0.016514
Epoch: 294 \ Validation Loss=0.455328
Epoch: 295 \ Training Loss=0.018026
Epoch: 295 \ Validation Loss=0.422798
Epoch: 296 \ Training Loss=0.015289
Epoch: 296 \ Validation Loss=0.323466
Epoch: 297 \ Training Loss=0.017392
Epoch: 297 \ Validation Loss=0.361712
Epoch: 298 \ Training Loss=0.015048
Epoch: 298 \ Validation Loss=0.388186
Epoch: 299 \ Training Loss=0.022789
Epoch: 299 \ Validation Loss=0.277833
Epoch: 300 \ Training Loss=0.015669
Epoch: 300 \ Validation Loss=0.275894
Epoch: 301 \ Training Loss=0.011526
Epoch: 301 \ Validation Loss=0.332883
Epoch: 302 \ Training Loss=0.014183
Epoch: 302 \ Validation Loss=0.396599
Epoch: 303 \ Training Loss=0.019609
Epoch: 303 \ Validation Loss=0.335719
Epoch: 304 \ Training Loss=0.013331
Epoch: 304 \ Validation Loss=0.320964
Epoch: 305 \ Training Loss=0.015369
Epoch: 305 \ Validation Loss=0.379099
Epoch: 306 \ Training Loss=0.017089
Epoch: 306 \ Validation Loss=0.348512
Epoch: 307 \ Training Loss=0.014357
Epoch: 307 \ Validation Loss=0.340231
Epoch: 308 \ Training Loss=0.014000
Epoch: 308 \ Validation Loss=0.439002
Epoch: 309 \ Training Loss=0.011789
Epoch: 309 \ Validation Loss=0.366368
Epoch: 310 \ Training Loss=0.012589
Epoch: 310 \ Validation Loss=0.302586
Epoch: 311 \ Training Loss=0.014998
Epoch: 311 \ Validation Loss=0.353798
Epoch: 312 \ Training Loss=0.013649
Epoch: 312 \ Validation Loss=0.294674
Epoch: 313 \ Training Loss=0.012352
Epoch: 313 \ Validation Loss=0.362474
Epoch: 314 \ Training Loss=0.016509
Epoch: 314 \ Validation Loss=0.271350
Epoch: 315 \ Training Loss=0.013113
Epoch: 315 \ Validation Loss=0.419572
Epoch: 316 \ Training Loss=0.014223
Epoch: 316 \ Validation Loss=0.294248
Epoch: 317 \ Training Loss=0.014215
Epoch: 317 \ Validation Loss=0.280916
Epoch: 318 \ Training Loss=0.014963
Epoch: 318 \ Validation Loss=0.409353
Epoch: 319 \ Training Loss=0.016311
Epoch: 319 \ Validation Loss=0.268548
Epoch: 320 \ Training Loss=0.017045
Epoch: 320 \ Validation Loss=0.341210
Epoch: 321 \ Training Loss=0.013630
Epoch: 321 \ Validation Loss=0.282127
Epoch: 322 \ Training Loss=0.017399
Epoch: 322 \ Validation Loss=0.348423
Epoch: 323 \ Training Loss=0.013703
Epoch: 323 \ Validation Loss=0.393903
Epoch: 324 \ Training Loss=0.018688
Epoch: 324 \ Validation Loss=0.442539
Epoch: 325 \ Training Loss=0.015615
Epoch: 325 \ Validation Loss=0.365069
Epoch: 326 \ Training Loss=0.016759
```

```
Epoch: 326 \ Validation Loss=0.573811
Epoch: 327 \ Training Loss=0.013628
Epoch: 327 \ Validation Loss=0.627768
Epoch: 328 \ Training Loss=0.014690
Epoch: 328 \ Validation Loss=0.492734
Epoch: 329 \ Training Loss=0.017252
Epoch: 329 \ Validation Loss=0.427110
Epoch: 330 \ Training Loss=0.015243
Epoch: 330 \ Validation Loss=0.496360
Epoch: 331 \ Training Loss=0.013851
Epoch: 331 \ Validation Loss=0.414979
Epoch: 332 \ Training Loss=0.017254
Epoch: 332 \ Validation Loss=0.289285
Epoch: 333 \ Training Loss=0.015764
Epoch: 333 \ Validation Loss=0.416455
Epoch: 334 \ Training Loss=0.016070
Epoch: 334 \ Validation Loss=0.384790
Epoch: 335 \ Training Loss=0.016291
Epoch: 335 \ Validation Loss=0.380885
Epoch: 336 \ Training Loss=0.013452
Epoch: 336 \ Validation Loss=0.346807
Epoch: 337 \ Training Loss=0.014776
Epoch: 337 \ Validation Loss=0.257758
Epoch: 338 \ Training Loss=0.016181
Epoch: 338 \ Validation Loss=0.353804
Epoch: 339 \ Training Loss=0.011628
Epoch: 339 \ Validation Loss=0.287011
Epoch: 340 \ Training Loss=0.013888
Epoch: 340 \ Validation Loss=0.386332
Epoch: 341 \ Training Loss=0.011195
Epoch: 341 \ Validation Loss=0.299918
Epoch: 342 \ Training Loss=0.010629
Epoch: 342 \ Validation Loss=0.301501
Epoch: 343 \ Training Loss=0.010960
Epoch: 343 \ Validation Loss=0.346107
Epoch: 344 \ Training Loss=0.011457
Epoch: 344 \ Validation Loss=0.281804
Epoch: 345 \ Training Loss=0.014579
Epoch: 345 \ Validation Loss=0.283658
Epoch: 346 \ Training Loss=0.012321
Epoch: 346 \ Validation Loss=0.322721
Epoch: 347 \ Training Loss=0.015576
Epoch: 347 \ Validation Loss=0.312460
Epoch: 348 \ Training Loss=0.012496
Epoch: 348 \ Validation Loss=0.346831
Epoch: 349 \ Training Loss=0.009891
Epoch: 349 \ Validation Loss=0.311428
Epoch: 350 \ Training Loss=0.015163
Epoch: 350 \ Validation Loss=0.283818
```

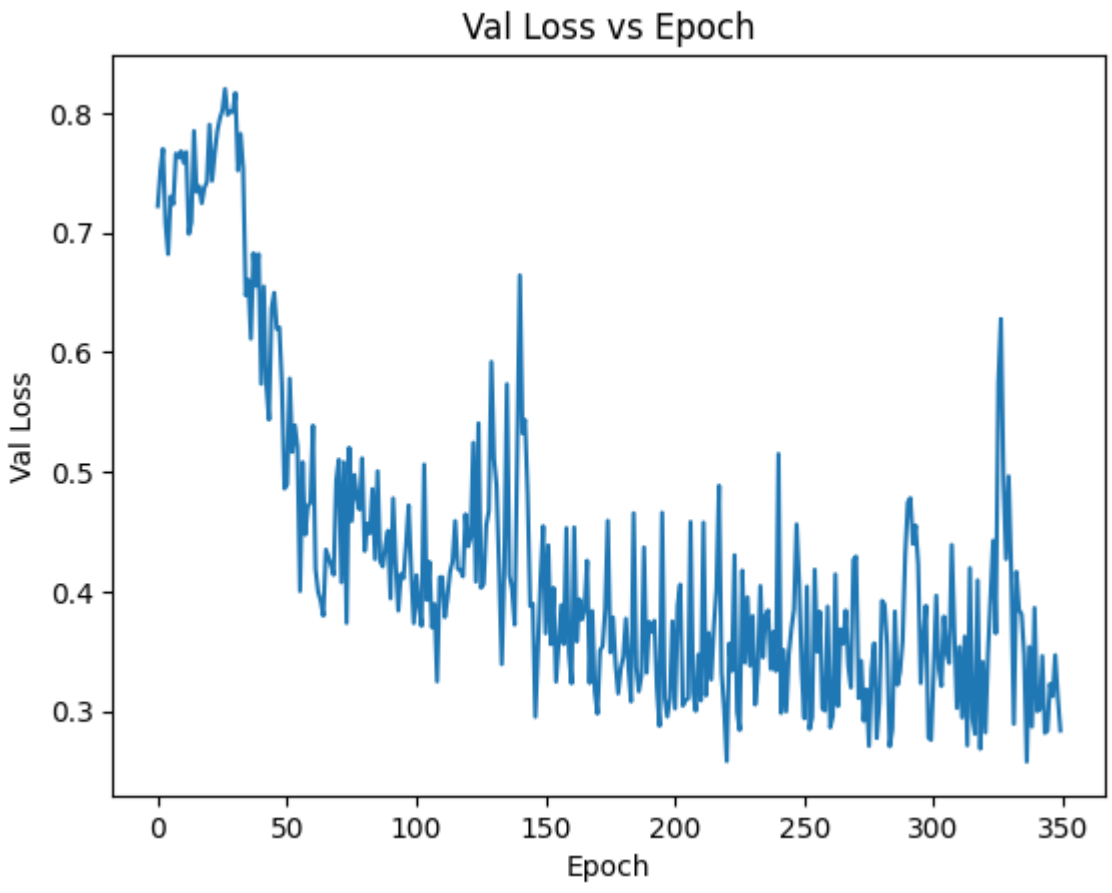
## We can see that at the end of 200 epoch, val loss decreased continously along with training loss. Hence no overfitting.

In [21]:
```
import matplotlib.pyplot as plt
import numpy as np

epoch_x = np.arange(0, 350, 1)
plt.plot(epoch_x, val_dice)
plt.title("DICE vs Epoch")
plt.xlabel('Epoch')
# naming the y axis
plt.ylabel('DICE')
plt.show()
```

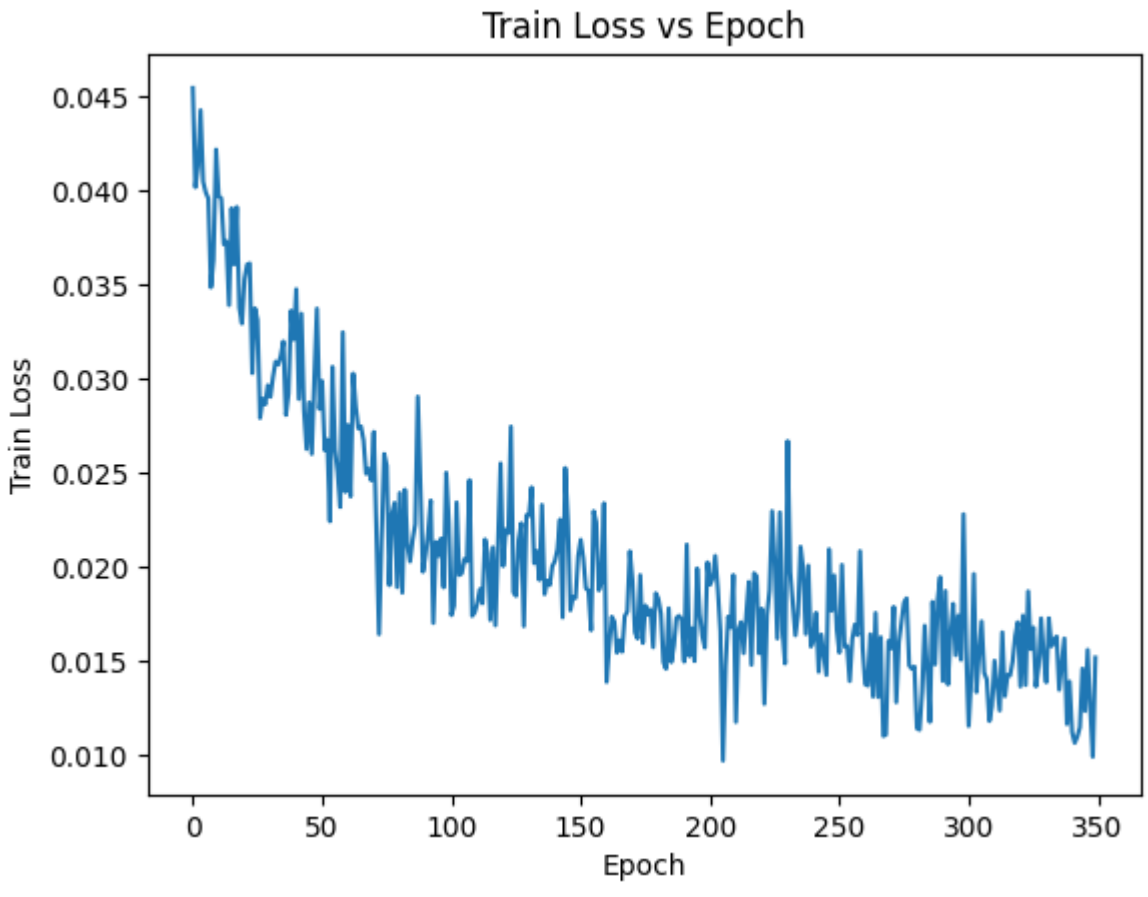



If we analyze below loss curves, we can see that in between epoch 100 and 150, model was overfitted and similarly around epoch 325. But later we can see that validation loss decreased as training loss dereases.

In [22]:
```
plt.plot(epoch_x, val_loss_epoch)
plt.title("Val Loss vs Epoch")
plt.xlabel('Epoch')
# naming the y axis
plt.ylabel('Val Loss')
plt.show()
```




In [23]:
```
plt.plot(epoch_x, train_loss_epoch)
plt.title("Train Loss vs Epoch")
plt.xlabel('Epoch')
# naming the y axis
plt.ylabel('Train Loss')
plt.show()
```




## Avg DICE socre of test data => 0.7398249308268229

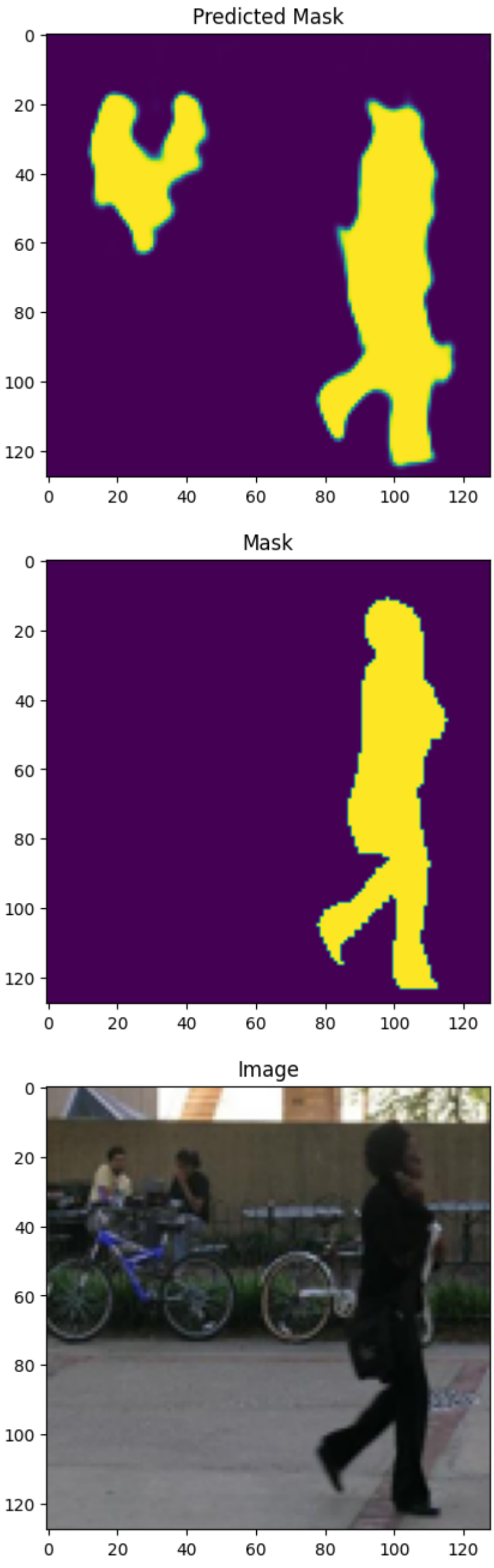
In [24]:
```
epoch_dice_score = 0
model = torch.load('best_model.pt')
model.eval()
with torch.no_grad():
  for (x, y) in test_loader:

      x = x.to(device)
      y = y.to(device)

      y_pred = model(x)
      loss = DiceScore(y_pred, y)
      epoch_dice_score += DiceScore(y_pred, y)

print(epoch_dice_score/len(test_loader))
```

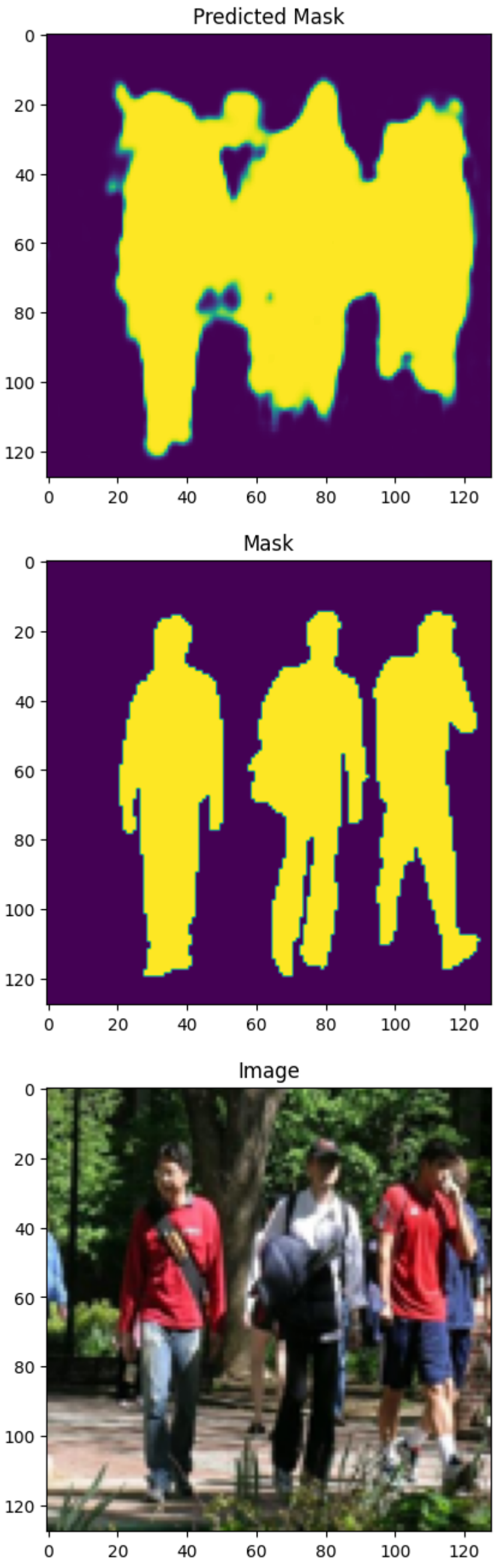
```
0.7398249308268229
```

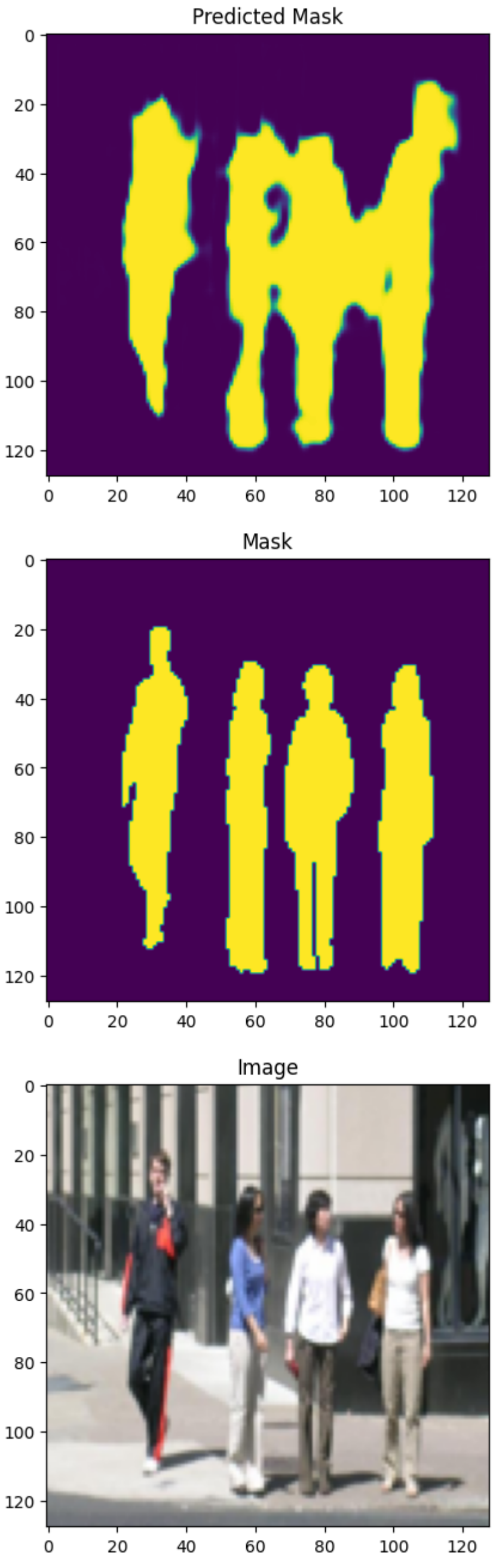
## We can see below that on test data, model was able to output the mask with good accuracy. In one of the test data, network was able to identify both legs and space between both legs and was properly able to segement those area.

### Image, Mask and predicted mask of test data

In [31]:
```
model = torch.load('best_model.pt')
model.eval()
num = 0
with torch.no_grad():
  for (x, y) in test_loader:
    num = num + 1
    x = x.to(device)
    y = y.to(device)

    y_pred = model(x)

    plt.imshow(y_pred[0].permute(1,2,0).detach().numpy())
    plt.title("Predicted Mask")
    plt.show()

    plt.imshow(y[0].permute(1,2,0).detach().numpy())
    plt.title("Mask")
    plt.show()

    plt.imshow(x[0].permute(1,2,0).detach().numpy())
    plt.title("Image")
    plt.show()

    if num == 3:
      break
```


Predicted Mask
0
20
40
60
80
100
120
Mask
Image

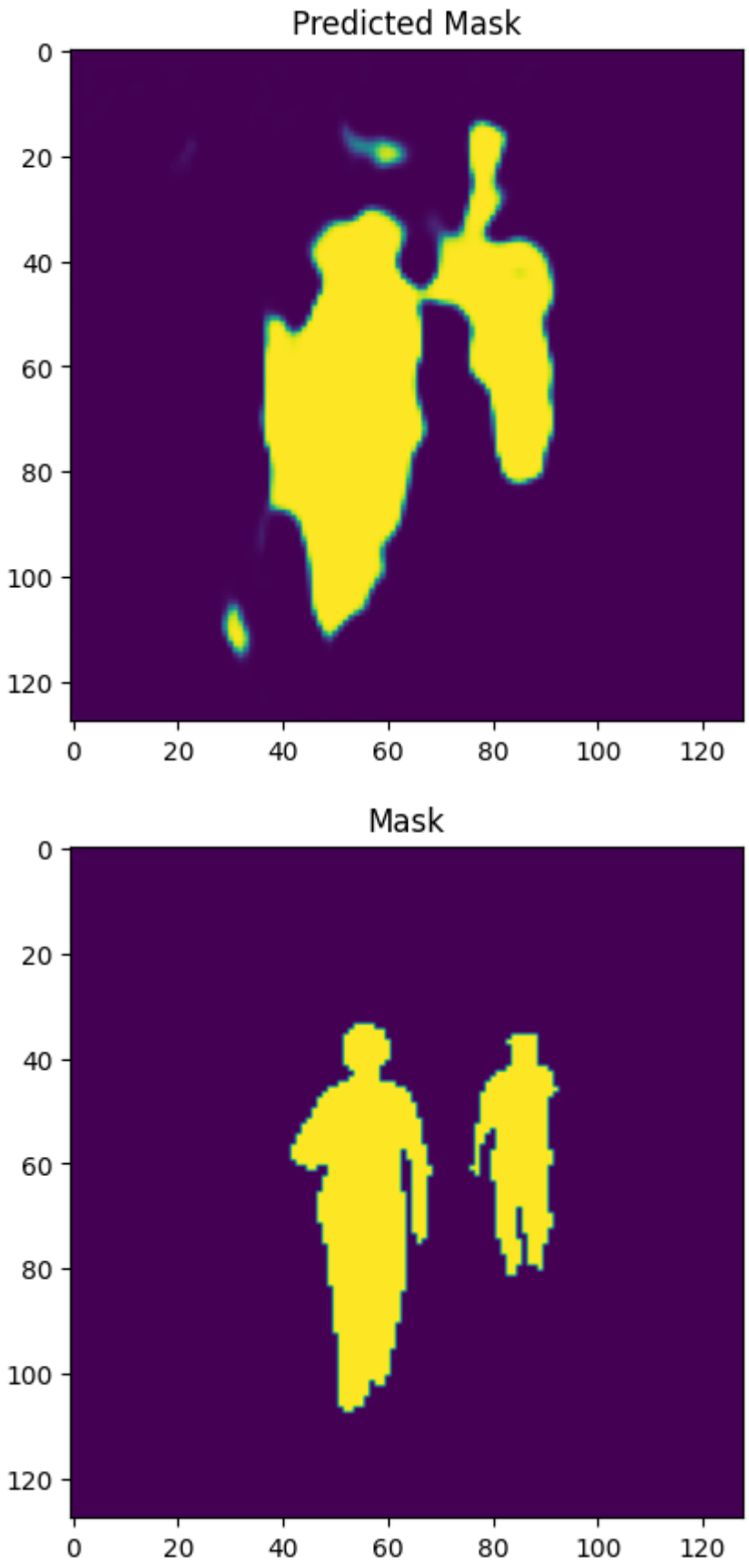

Predicted Mask
0
20
40
60
80
100
120
Mask
Image




## Image, Mask and predicted mask of Training data

In [32]:
```
model = torch.load('best_model.pt')
model.eval()
num = 0
with torch.no_grad():
  for (x, y) in train_loader:
    num = num + 1
    x = x.to(device)
```

```
    y = y.to(device)

    y_pred = model(x)

    plt.imshow(y_pred[0].permute(1,2,0).detach().numpy())
    plt.title("Predicted Mask")
    plt.show()

    plt.imshow(y[0].permute(1,2,0).detach().numpy())
    plt.title("Mask")
    plt.show()

    plt.imshow(x[0].permute(1,2,0).detach().numpy())
    plt.title("Image")
    plt.show()

    if num == 3:
      break
```



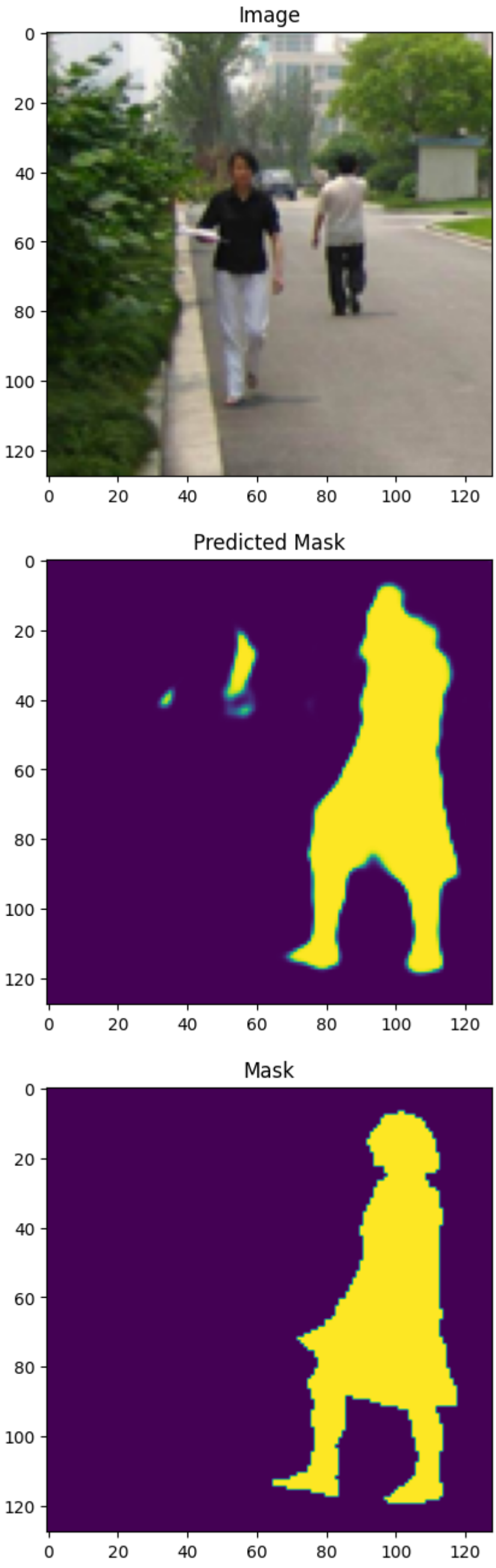


Image
Predicted Mask
Mask
0
20
40
60
80
100
120

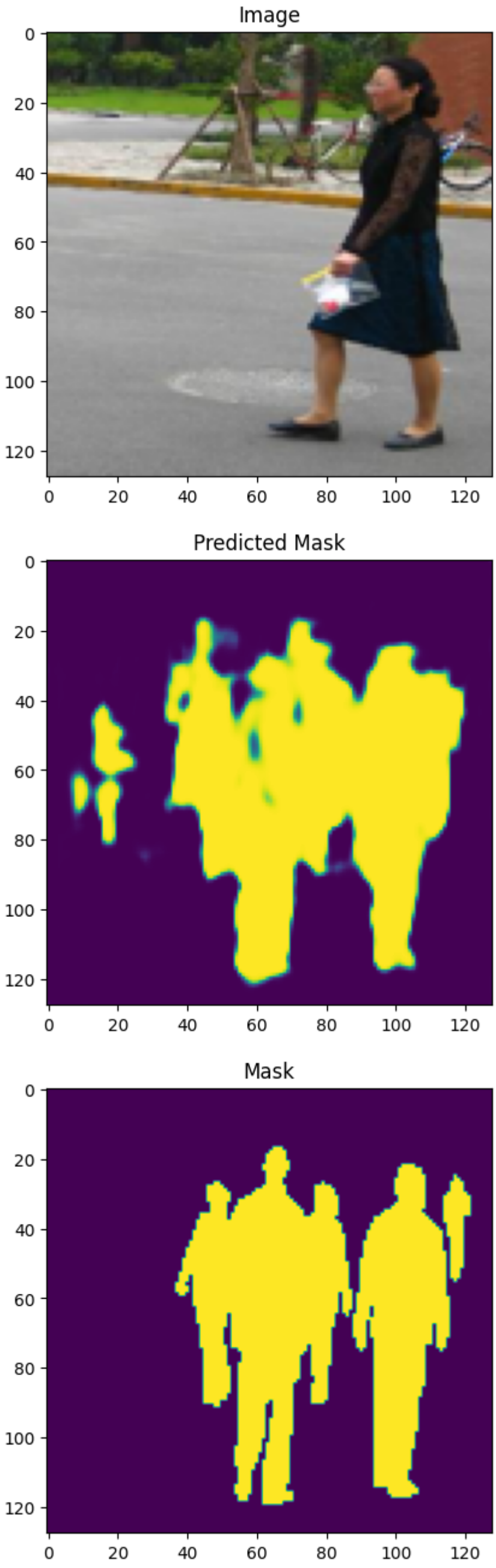
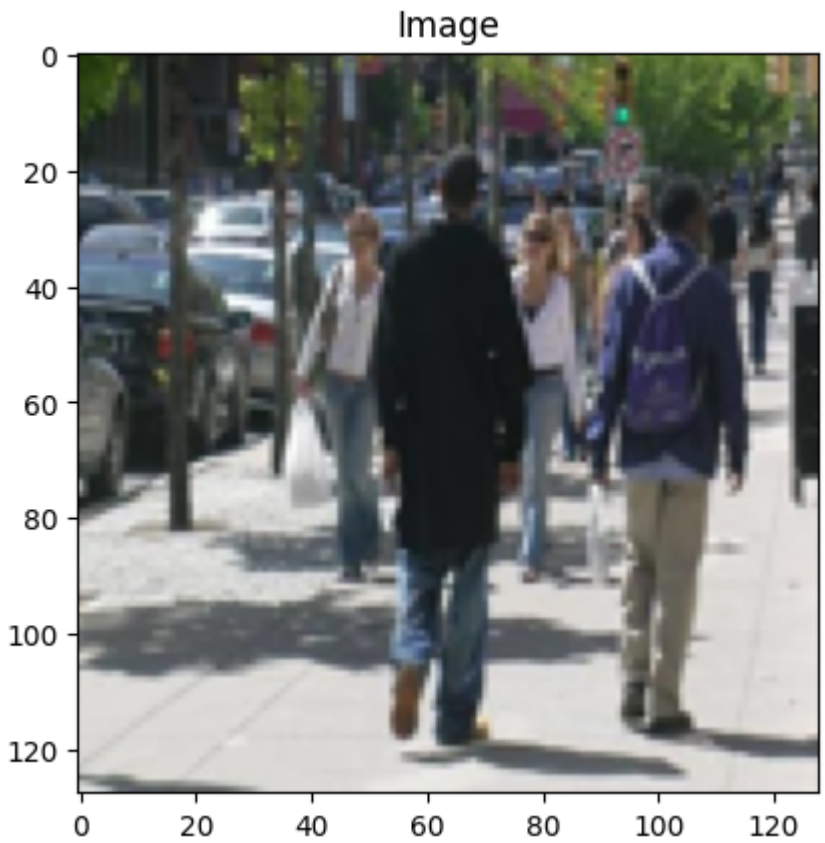


Image
0
20
40
60
80
100
120
Predicted Mask
Mask




# Model output on different image other than pennfundan dataset

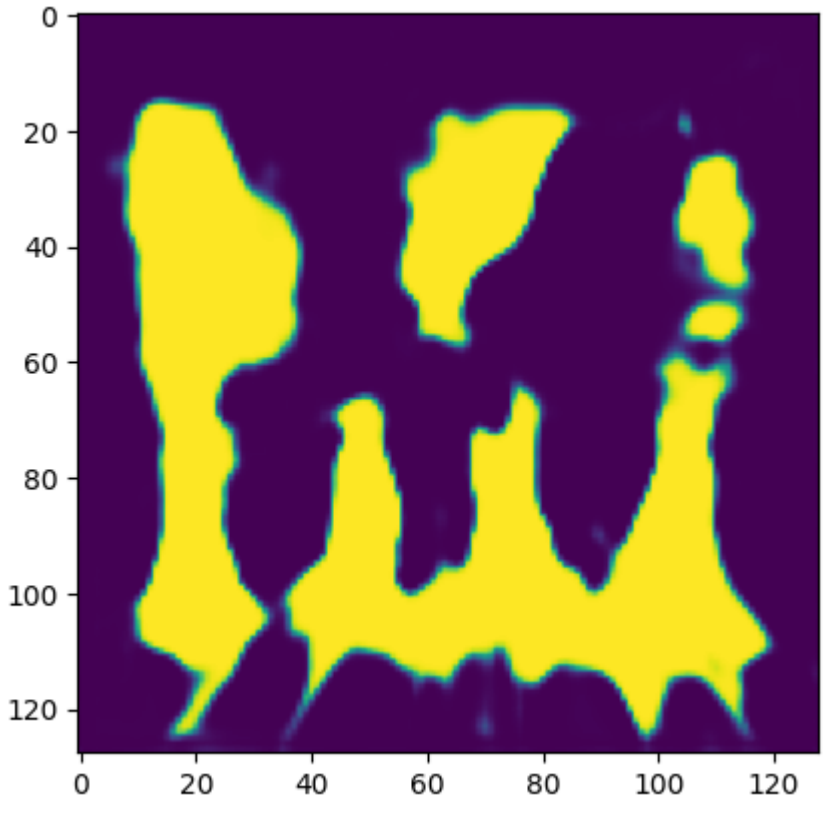
In [28]:
```
import torchvision.transforms as T

model = torch.load('best_model.pt')

# Test the beatles image
img2 = Image.open('Beatles.jpeg').convert("RGB")
newsize = ((128, 128))
img2 = img2.resize(newsize)
convert_tensor = transforms.ToTensor()
img2 = convert_tensor(img2)
img2 = img2.unsqueeze(0)

# put the model in evaluation mode
model.eval()
with torch.no_grad():
  prediction = model(img2.to(device))

print(prediction.squeeze(0).size())

Image.fromarray(img2.squeeze(0).permute(1, 2, 0).mul(255).byte().numpy())
```

```
torch.Size([1, 128, 128])
```

Out[28]:

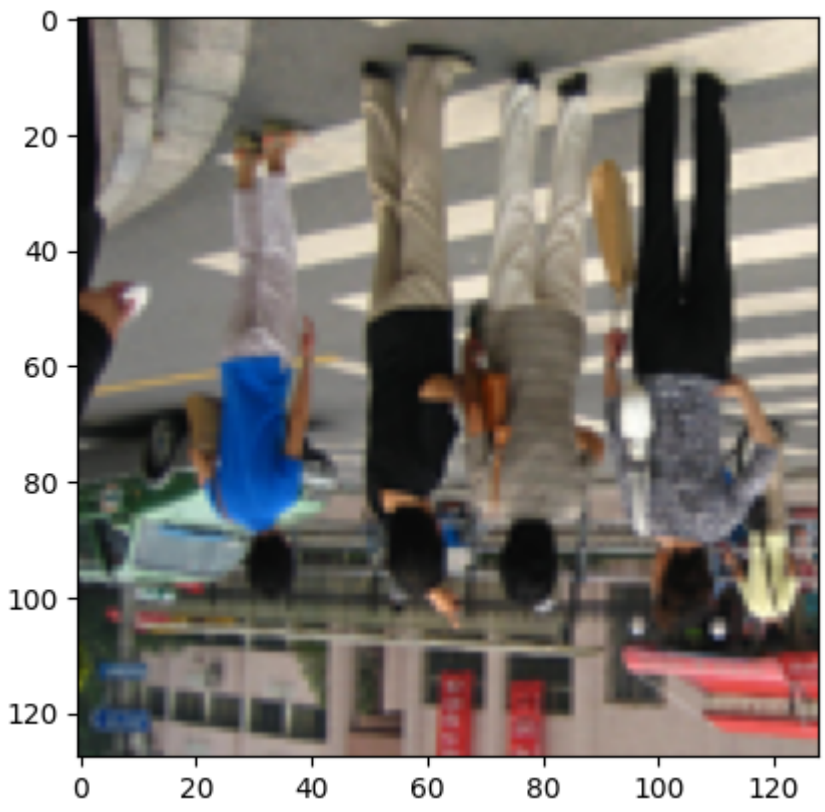
# We can see below that on different image (other than that of dataset), network was able to identify person although it identified other areas which are not persons but at least all correct persons were segmented.

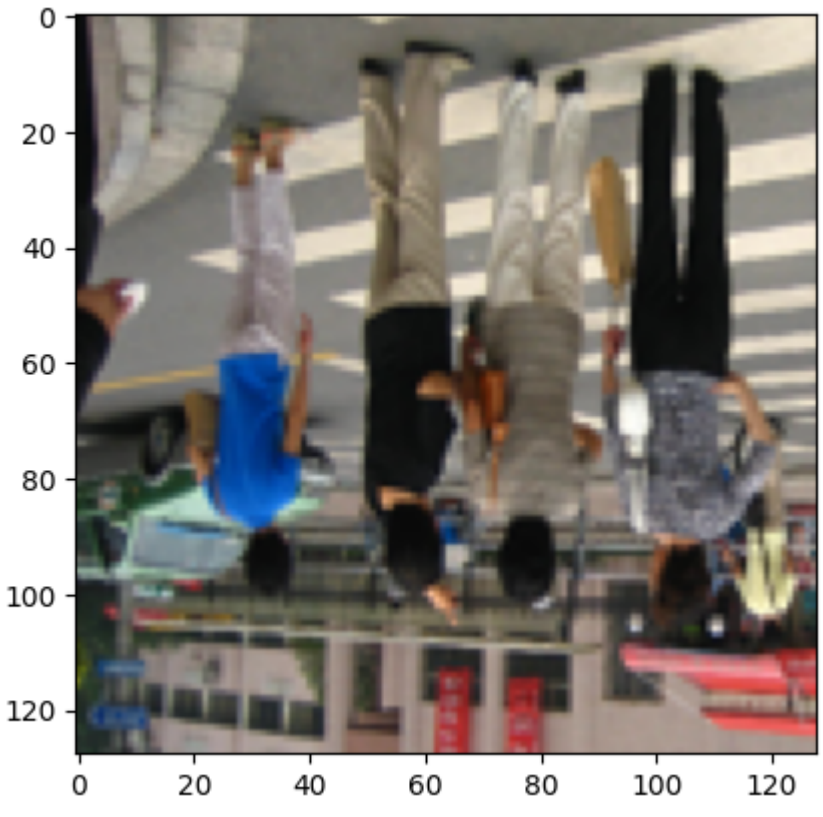
In [29]:
```
plt.imshow(prediction[0].permute(1,2,0).detach().numpy())
plt.show()
```

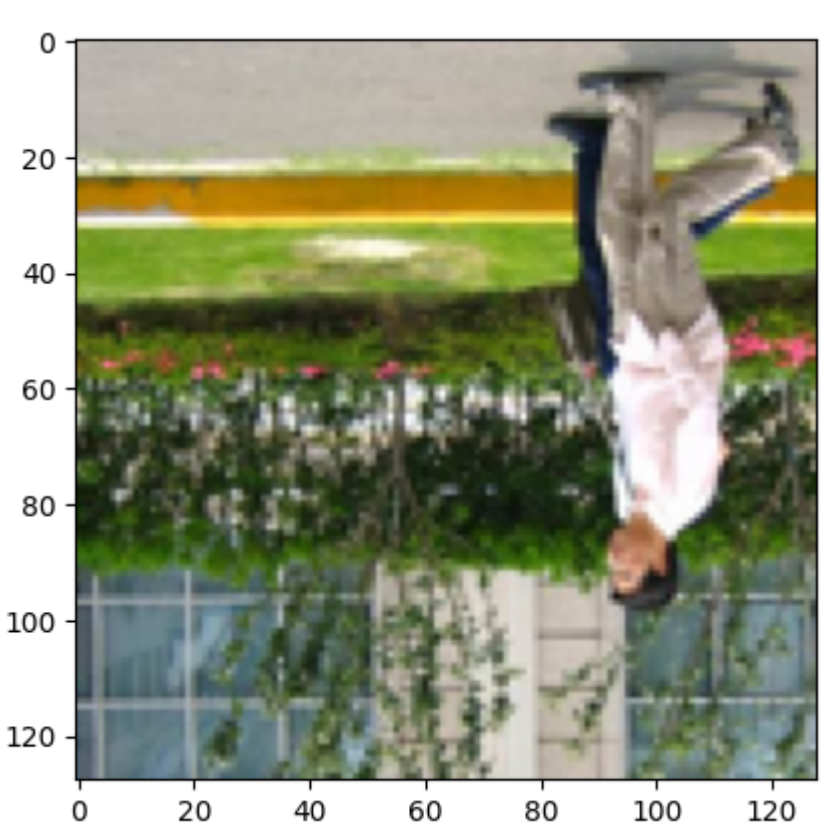


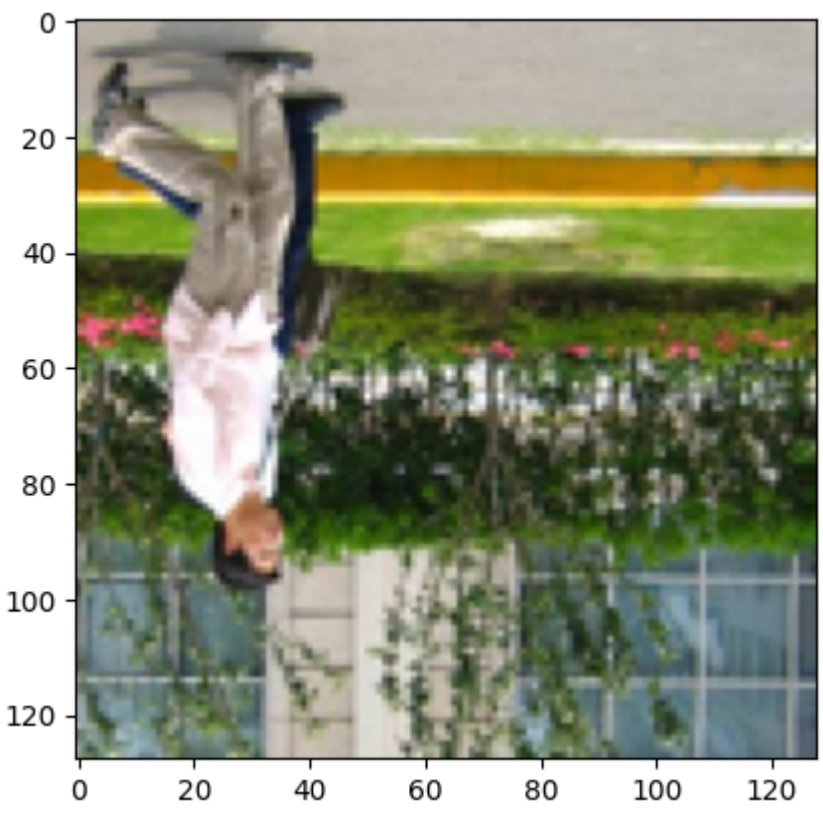

## Example of Data Augmentation

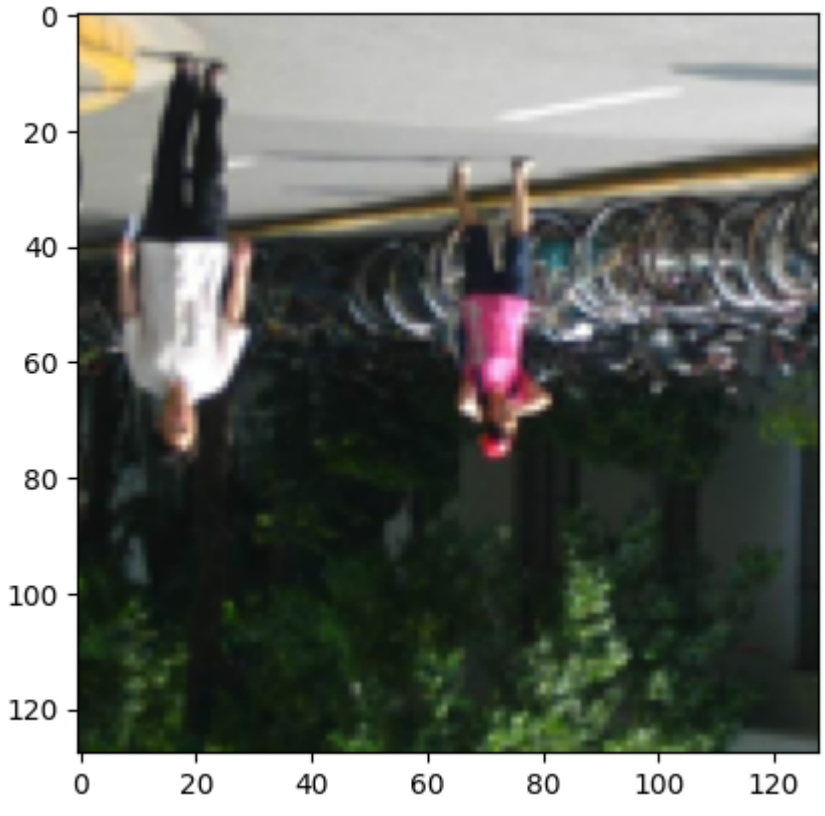
In [30]:
```
num = 0
for (x, y) in train_loader:
  num = num + 1
  x = x.to(device)
  y = y.to(device)

  plt.imshow(x[0].permute(1,2,0).detach().numpy())
  plt.show()

  convert_tensor = transforms.RandomHorizontalFlip(0.5)
  x = convert_tensor(x)

  plt.imshow(x[0].permute(1,2,0).detach().numpy())
  plt.show()

  if num == 3:
    break
```

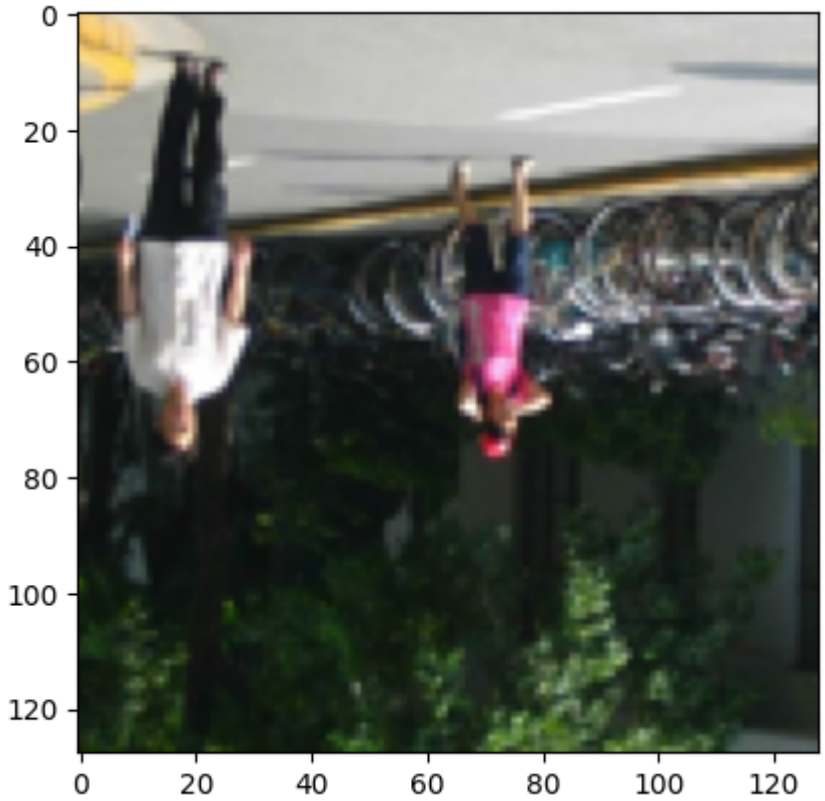



0
20
40
60
80
100
120
0
20
40
60
80
100
120


0
20
40
60
80
100
120
0
20
40
60
80
100
120


0
20
40
60
80
100
120
0
20
40
60
80
100
120


0
20
40
60
80
100
120
0
20
40
60
80
100
120


0
20
40
60
80
100
120
0
20
40
60
80
100
120