CNN Segmentation

```
%%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
```

```
--2023-04-14 23:23:42--
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  27.8MB/s    in
1.8s

2023-04-14 23:23:44 (27.8 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
```

```python
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 transforms
import torchvision.transforms as transforms
import torchvision.datasets as datasets
from torch.utils.data import Dataset, DataLoader, SubsetRandomSampler

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] = 1

        if self.transforms is not None:
            img, mask = self.transforms(img, mask)
```

```
        return img, mask

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

```
%%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
```

```
class MyCompose(object):
    def __init__(self, transforms):
        self.transforms = transforms

    def __call__(self, img, tar):
        for t in self.transforms:
            img = t(img)
            tar = t(tar)
        return img, tar
```

```
import torchvision.transforms as T

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

UNet Architecture

```
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

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

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

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))

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

```
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)

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()

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()

from torchsummary import summary
model = UNet(3)
# summary(model, (3, 128, 128))
model

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))
  )
)

optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
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)
```

```
54241
```

```
# def calculate_accuracy(y_pred, y):
#     top_pred = y_pred.argmax(1, keepdim = True)
#     correct = top_pred.eq(y.view_as(top_pred)).sum()
#     acc = correct.float() / y.shape[0]
#     return acc

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)

        # acc = calculate_accuracy(y_pred, y)
        loss.backward()
        optimizer.step()
        # loss_list +=[loss.item()]*x.shape[0]
        epoch_loss += loss.item()
        # epoch_acc += acc.item()
        # return np.mean(loss_list)
        return epoch_loss / len(iterator)

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)
            # loss_dice = criterion_dice(y_pred, y)
            # acc = calculate_accuracy(y_pred, y)
            epoch_loss += loss.item()
            # epoch_dice_loss += loss_dice()
            epoch_dice_score += DiceScore(y_pred, y)
            # epoch_acc += acc.item()

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

EPOCHS = 200

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

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)

  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=0.034812
Epoch: 1 \ Validation Loss=0.644727
Epoch: 2 \ Training Loss=0.032634
Epoch: 2 \ Validation Loss=0.636636
Epoch: 3 \ Training Loss=0.030859
Epoch: 3 \ Validation Loss=0.628255
Epoch: 4 \ Training Loss=0.029233
Epoch: 4 \ Validation Loss=0.620202
Epoch: 5 \ Training Loss=0.027491
Epoch: 5 \ Validation Loss=0.610151
Epoch: 6 \ Training Loss=0.026064
Epoch: 6 \ Validation Loss=0.598370
Epoch: 7 \ Training Loss=0.024575
```

```
Epoch: 7 \ Validation Loss=0.584096
Epoch: 8 \ Training Loss=0.023218
Epoch: 8 \ Validation Loss=0.566676
Epoch: 9 \ Training Loss=0.021847
Epoch: 9 \ Validation Loss=0.543139
Epoch: 10 \ Training Loss=0.020679
Epoch: 10 \ Validation Loss=0.518506
Epoch: 11 \ Training Loss=0.019344
Epoch: 11 \ Validation Loss=0.491105
Epoch: 12 \ Training Loss=0.018301
Epoch: 12 \ Validation Loss=0.457025
Epoch: 13 \ Training Loss=0.017170
Epoch: 13 \ Validation Loss=0.424316
Epoch: 14 \ Training Loss=0.016140
Epoch: 14 \ Validation Loss=0.397160
Epoch: 15 \ Training Loss=0.015342
Epoch: 15 \ Validation Loss=0.363041
Epoch: 16 \ Training Loss=0.014314
Epoch: 16 \ Validation Loss=0.342151
Epoch: 17 \ Training Loss=0.013589
Epoch: 17 \ Validation Loss=0.317549
Epoch: 18 \ Training Loss=0.012715
Epoch: 18 \ Validation Loss=0.294087
Epoch: 19 \ Training Loss=0.012076
Epoch: 19 \ Validation Loss=0.283461
Epoch: 20 \ Training Loss=0.011200
Epoch: 20 \ Validation Loss=0.264809
Epoch: 21 \ Training Loss=0.011027
Epoch: 21 \ Validation Loss=0.249231
Epoch: 22 \ Training Loss=0.010072
Epoch: 22 \ Validation Loss=0.225641
Epoch: 23 \ Training Loss=0.009511
Epoch: 23 \ Validation Loss=0.212930
Epoch: 24 \ Training Loss=0.009005
Epoch: 24 \ Validation Loss=0.196999
Epoch: 25 \ Training Loss=0.008518
Epoch: 25 \ Validation Loss=0.180598
Epoch: 26 \ Training Loss=0.008117
Epoch: 26 \ Validation Loss=0.167588
Epoch: 27 \ Training Loss=0.007643
Epoch: 27 \ Validation Loss=0.159394
Epoch: 28 \ Training Loss=0.007215
Epoch: 28 \ Validation Loss=0.147545
Epoch: 29 \ Training Loss=0.006878
Epoch: 29 \ Validation Loss=0.139376
Epoch: 30 \ Training Loss=0.006671
Epoch: 30 \ Validation Loss=0.135019
Epoch: 31 \ Training Loss=0.006332
Epoch: 31 \ Validation Loss=0.131503
Epoch: 32 \ Training Loss=0.005926
```

```
Epoch: 32 \ Validation Loss=0.126877
Epoch: 33 \ Training Loss=0.005582
Epoch: 33 \ Validation Loss=0.124851
Epoch: 34 \ Training Loss=0.005384
Epoch: 34 \ Validation Loss=0.118319
Epoch: 35 \ Training Loss=0.005083
Epoch: 35 \ Validation Loss=0.112428
Epoch: 36 \ Training Loss=0.004951
Epoch: 36 \ Validation Loss=0.111191
Epoch: 37 \ Training Loss=0.004643
Epoch: 37 \ Validation Loss=0.101755
Epoch: 38 \ Training Loss=0.004538
Epoch: 38 \ Validation Loss=0.095836
Epoch: 39 \ Training Loss=0.004303
Epoch: 39 \ Validation Loss=0.093144
Epoch: 40 \ Training Loss=0.004199
Epoch: 40 \ Validation Loss=0.087719
Epoch: 41 \ Training Loss=0.003997
Epoch: 41 \ Validation Loss=0.078405
Epoch: 42 \ Training Loss=0.003795
Epoch: 42 \ Validation Loss=0.075598
Epoch: 43 \ Training Loss=0.003682
Epoch: 43 \ Validation Loss=0.069502
Epoch: 44 \ Training Loss=0.003561
Epoch: 44 \ Validation Loss=0.067191
Epoch: 45 \ Training Loss=0.003388
Epoch: 45 \ Validation Loss=0.063446
Epoch: 46 \ Training Loss=0.003360
Epoch: 46 \ Validation Loss=0.061438
Epoch: 47 \ Training Loss=0.003197
Epoch: 47 \ Validation Loss=0.058249
Epoch: 48 \ Training Loss=0.003149
Epoch: 48 \ Validation Loss=0.057979
Epoch: 49 \ Training Loss=0.002997
Epoch: 49 \ Validation Loss=0.056218
Epoch: 50 \ Training Loss=0.002860
Epoch: 50 \ Validation Loss=0.055171
Epoch: 51 \ Training Loss=0.002821
Epoch: 51 \ Validation Loss=0.053053
Epoch: 52 \ Training Loss=0.002702
Epoch: 52 \ Validation Loss=0.051732
Epoch: 53 \ Training Loss=0.002687
Epoch: 53 \ Validation Loss=0.048332
Epoch: 54 \ Training Loss=0.002583
Epoch: 54 \ Validation Loss=0.048013
Epoch: 55 \ Training Loss=0.002493
Epoch: 55 \ Validation Loss=0.044454
Epoch: 56 \ Training Loss=0.002412
Epoch: 56 \ Validation Loss=0.044362
Epoch: 57 \ Training Loss=0.002316
```

```
Epoch: 57 \ Validation Loss=0.040801
Epoch: 58 \ Training Loss=0.002297
Epoch: 58 \ Validation Loss=0.039941
Epoch: 59 \ Training Loss=0.002258
Epoch: 59 \ Validation Loss=0.038009
Epoch: 60 \ Training Loss=0.002220
Epoch: 60 \ Validation Loss=0.035876
Epoch: 61 \ Training Loss=0.002218
Epoch: 61 \ Validation Loss=0.035552
Epoch: 62 \ Training Loss=0.002062
Epoch: 62 \ Validation Loss=0.034864
Epoch: 63 \ Training Loss=0.002012
Epoch: 63 \ Validation Loss=0.033991
Epoch: 64 \ Training Loss=0.002011
Epoch: 64 \ Validation Loss=0.033592
Epoch: 65 \ Training Loss=0.001931
Epoch: 65 \ Validation Loss=0.033257
Epoch: 66 \ Training Loss=0.001873
Epoch: 66 \ Validation Loss=0.032894
Epoch: 67 \ Training Loss=0.001826
Epoch: 67 \ Validation Loss=0.032820
Epoch: 68 \ Training Loss=0.001842
Epoch: 68 \ Validation Loss=0.032096
Epoch: 69 \ Training Loss=0.001785
Epoch: 69 \ Validation Loss=0.031727
Epoch: 70 \ Training Loss=0.001731
Epoch: 70 \ Validation Loss=0.030886
Epoch: 71 \ Training Loss=0.001715
Epoch: 71 \ Validation Loss=0.029826
Epoch: 72 \ Training Loss=0.001751
Epoch: 72 \ Validation Loss=0.028978
Epoch: 73 \ Training Loss=0.001659
Epoch: 73 \ Validation Loss=0.027349
Epoch: 74 \ Training Loss=0.001618
Epoch: 74 \ Validation Loss=0.026959
Epoch: 75 \ Training Loss=0.001548
Epoch: 75 \ Validation Loss=0.026244
Epoch: 76 \ Training Loss=0.001639
Epoch: 76 \ Validation Loss=0.025774
Epoch: 77 \ Training Loss=0.001520
Epoch: 77 \ Validation Loss=0.025292
Epoch: 78 \ Training Loss=0.001515
Epoch: 78 \ Validation Loss=0.025191
Epoch: 79 \ Training Loss=0.001486
Epoch: 79 \ Validation Loss=0.024513
Epoch: 80 \ Training Loss=0.001488
Epoch: 80 \ Validation Loss=0.023875
Epoch: 81 \ Training Loss=0.001448
Epoch: 81 \ Validation Loss=0.024577
Epoch: 82 \ Training Loss=0.001420
```

```
Epoch: 82 \ Validation Loss=0.024049
Epoch: 83 \ Training Loss=0.001426
Epoch: 83 \ Validation Loss=0.024098
Epoch: 84 \ Training Loss=0.001406
Epoch: 84 \ Validation Loss=0.023481
Epoch: 85 \ Training Loss=0.001345
Epoch: 85 \ Validation Loss=0.023041
Epoch: 86 \ Training Loss=0.001322
Epoch: 86 \ Validation Loss=0.022528
Epoch: 87 \ Training Loss=0.001278
Epoch: 87 \ Validation Loss=0.022176
Epoch: 88 \ Training Loss=0.001308
Epoch: 88 \ Validation Loss=0.021365
Epoch: 89 \ Training Loss=0.001264
Epoch: 89 \ Validation Loss=0.020964
Epoch: 90 \ Training Loss=0.001271
Epoch: 90 \ Validation Loss=0.019916
Epoch: 91 \ Training Loss=0.001240
Epoch: 91 \ Validation Loss=0.020184
Epoch: 92 \ Training Loss=0.001258
Epoch: 92 \ Validation Loss=0.019599
Epoch: 93 \ Training Loss=0.001249
Epoch: 93 \ Validation Loss=0.020379
Epoch: 94 \ Training Loss=0.001196
Epoch: 94 \ Validation Loss=0.019960
Epoch: 95 \ Training Loss=0.001136
Epoch: 95 \ Validation Loss=0.019509
Epoch: 96 \ Training Loss=0.001179
Epoch: 96 \ Validation Loss=0.019683
Epoch: 97 \ Training Loss=0.001180
Epoch: 97 \ Validation Loss=0.018950
Epoch: 98 \ Training Loss=0.001141
Epoch: 98 \ Validation Loss=0.019325
Epoch: 99 \ Training Loss=0.001092
Epoch: 99 \ Validation Loss=0.018806
Epoch: 100 \ Training Loss=0.001109
Epoch: 100 \ Validation Loss=0.018540
Epoch: 101 \ Training Loss=0.001060
Epoch: 101 \ Validation Loss=0.018825
Epoch: 102 \ Training Loss=0.001092
Epoch: 102 \ Validation Loss=0.018831
Epoch: 103 \ Training Loss=0.001018
Epoch: 103 \ Validation Loss=0.017874
Epoch: 104 \ Training Loss=0.001051
Epoch: 104 \ Validation Loss=0.017975
Epoch: 105 \ Training Loss=0.001015
Epoch: 105 \ Validation Loss=0.017298
Epoch: 106 \ Training Loss=0.001033
Epoch: 106 \ Validation Loss=0.016969
Epoch: 107 \ Training Loss=0.001067
```

```
Epoch: 107 \ Validation Loss=0.017531
Epoch: 108 \ Training Loss=0.001034
Epoch: 108 \ Validation Loss=0.016925
Epoch: 109 \ Training Loss=0.001037
Epoch: 109 \ Validation Loss=0.016420
Epoch: 110 \ Training Loss=0.000987
Epoch: 110 \ Validation Loss=0.017011
Epoch: 111 \ Training Loss=0.000988
Epoch: 111 \ Validation Loss=0.016840
Epoch: 112 \ Training Loss=0.000998
Epoch: 112 \ Validation Loss=0.015994
Epoch: 113 \ Training Loss=0.001018
Epoch: 113 \ Validation Loss=0.015797
Epoch: 114 \ Training Loss=0.000918
Epoch: 114 \ Validation Loss=0.015712
Epoch: 115 \ Training Loss=0.000948
Epoch: 115 \ Validation Loss=0.015925
Epoch: 116 \ Training Loss=0.000946
Epoch: 116 \ Validation Loss=0.015335
Epoch: 117 \ Training Loss=0.000889
Epoch: 117 \ Validation Loss=0.015366
Epoch: 118 \ Training Loss=0.000874
Epoch: 118 \ Validation Loss=0.014864
Epoch: 119 \ Training Loss=0.000945
Epoch: 119 \ Validation Loss=0.015185
Epoch: 120 \ Training Loss=0.000906
Epoch: 120 \ Validation Loss=0.014717
Epoch: 121 \ Training Loss=0.000826
Epoch: 121 \ Validation Loss=0.015074
Epoch: 122 \ Training Loss=0.000928
Epoch: 122 \ Validation Loss=0.015021
Epoch: 123 \ Training Loss=0.000887
Epoch: 123 \ Validation Loss=0.014209
Epoch: 124 \ Training Loss=0.000849
Epoch: 124 \ Validation Loss=0.014376
Epoch: 125 \ Training Loss=0.000847
Epoch: 125 \ Validation Loss=0.014799
Epoch: 126 \ Training Loss=0.000870
Epoch: 126 \ Validation Loss=0.014653
Epoch: 127 \ Training Loss=0.000877
Epoch: 127 \ Validation Loss=0.013982
Epoch: 128 \ Training Loss=0.000836
Epoch: 128 \ Validation Loss=0.014160
Epoch: 129 \ Training Loss=0.000871
Epoch: 129 \ Validation Loss=0.014073
Epoch: 130 \ Training Loss=0.000832
Epoch: 130 \ Validation Loss=0.013765
Epoch: 131 \ Training Loss=0.000786
Epoch: 131 \ Validation Loss=0.014330
Epoch: 132 \ Training Loss=0.000816
```

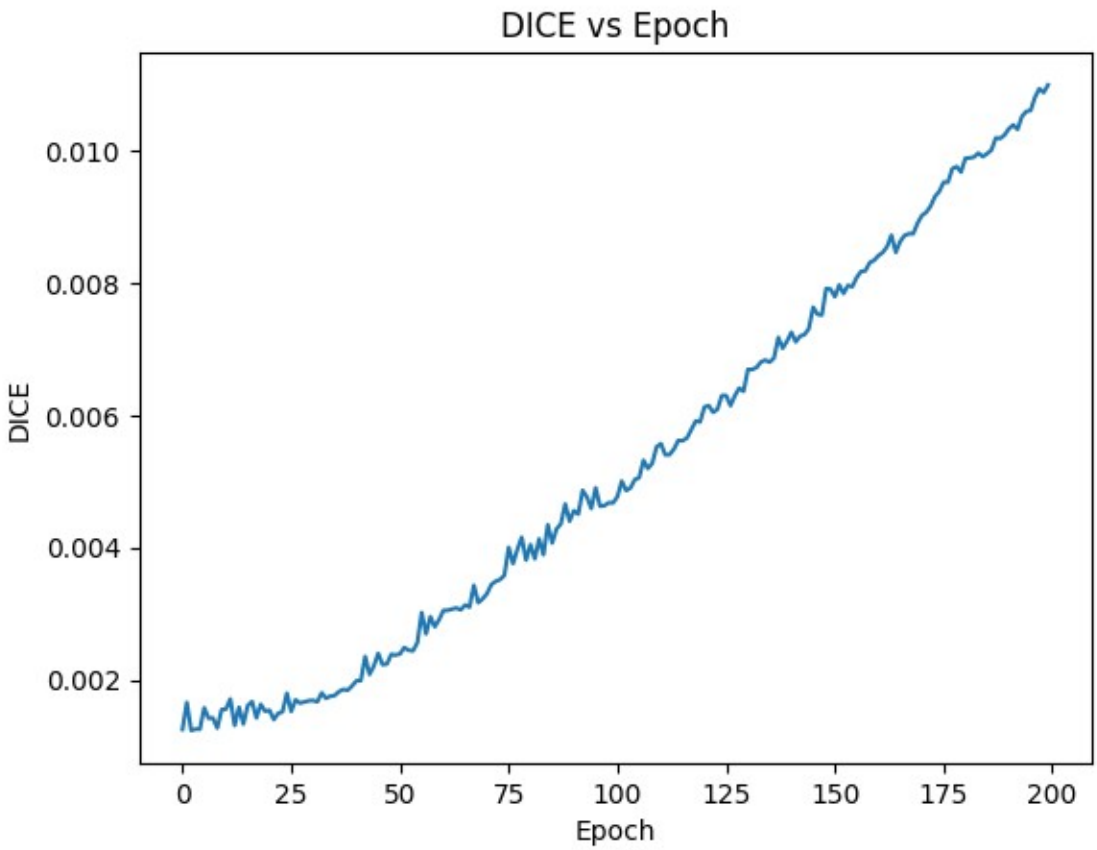
```
Epoch: 132 \ Validation Loss=0.013944
Epoch: 133 \ Training Loss=0.000766
Epoch: 133 \ Validation Loss=0.013746
Epoch: 134 \ Training Loss=0.000736
Epoch: 134 \ Validation Loss=0.013265
Epoch: 135 \ Training Loss=0.000787
Epoch: 135 \ Validation Loss=0.013058
Epoch: 136 \ Training Loss=0.000820
Epoch: 136 \ Validation Loss=0.012634
Epoch: 137 \ Training Loss=0.000736
Epoch: 137 \ Validation Loss=0.012525
Epoch: 138 \ Training Loss=0.000772
Epoch: 138 \ Validation Loss=0.013382
Epoch: 139 \ Training Loss=0.000756
Epoch: 139 \ Validation Loss=0.012163
Epoch: 140 \ Training Loss=0.000814
Epoch: 140 \ Validation Loss=0.012563
Epoch: 141 \ Training Loss=0.000728
Epoch: 141 \ Validation Loss=0.012915
Epoch: 142 \ Training Loss=0.000777
Epoch: 142 \ Validation Loss=0.012269
Epoch: 143 \ Training Loss=0.000790
Epoch: 143 \ Validation Loss=0.012240
Epoch: 144 \ Training Loss=0.000723
Epoch: 144 \ Validation Loss=0.012122
Epoch: 145 \ Training Loss=0.000767
Epoch: 145 \ Validation Loss=0.011993
Epoch: 146 \ Training Loss=0.000691
Epoch: 146 \ Validation Loss=0.012659
Epoch: 147 \ Training Loss=0.000749
Epoch: 147 \ Validation Loss=0.011519
Epoch: 148 \ Training Loss=0.000715
Epoch: 148 \ Validation Loss=0.011538
Epoch: 149 \ Training Loss=0.000696
Epoch: 149 \ Validation Loss=0.012014
Epoch: 150 \ Training Loss=0.000667
Epoch: 150 \ Validation Loss=0.012283
Epoch: 151 \ Training Loss=0.000716
Epoch: 151 \ Validation Loss=0.011673
Epoch: 152 \ Training Loss=0.000774
Epoch: 152 \ Validation Loss=0.011752
Epoch: 153 \ Training Loss=0.000716
Epoch: 153 \ Validation Loss=0.011329
Epoch: 154 \ Training Loss=0.000623
Epoch: 154 \ Validation Loss=0.011444
Epoch: 155 \ Training Loss=0.000697
Epoch: 155 \ Validation Loss=0.010962
Epoch: 156 \ Training Loss=0.000736
Epoch: 156 \ Validation Loss=0.011286
Epoch: 157 \ Training Loss=0.000666
```

```
Epoch: 157 \ Validation Loss=0.010652
Epoch: 158 \ Training Loss=0.000653
Epoch: 158 \ Validation Loss=0.010676
Epoch: 159 \ Training Loss=0.000691
Epoch: 159 \ Validation Loss=0.010442
Epoch: 160 \ Training Loss=0.000745
Epoch: 160 \ Validation Loss=0.010720
Epoch: 161 \ Training Loss=0.000659
Epoch: 161 \ Validation Loss=0.010330
Epoch: 162 \ Training Loss=0.000708
Epoch: 162 \ Validation Loss=0.010851
Epoch: 163 \ Training Loss=0.000666
Epoch: 163 \ Validation Loss=0.010691
Epoch: 164 \ Training Loss=0.000721
Epoch: 164 \ Validation Loss=0.011061
Epoch: 165 \ Training Loss=0.000614
Epoch: 165 \ Validation Loss=0.010524
Epoch: 166 \ Training Loss=0.000620
Epoch: 166 \ Validation Loss=0.010124
Epoch: 167 \ Training Loss=0.000638
Epoch: 167 \ Validation Loss=0.010097
Epoch: 168 \ Training Loss=0.000661
Epoch: 168 \ Validation Loss=0.010501
Epoch: 169 \ Training Loss=0.000605
Epoch: 169 \ Validation Loss=0.010572
Epoch: 170 \ Training Loss=0.000690
Epoch: 170 \ Validation Loss=0.010453
Epoch: 171 \ Training Loss=0.000618
Epoch: 171 \ Validation Loss=0.010597
Epoch: 172 \ Training Loss=0.000622
Epoch: 172 \ Validation Loss=0.010688
Epoch: 173 \ Training Loss=0.000625
Epoch: 173 \ Validation Loss=0.009687
Epoch: 174 \ Training Loss=0.000628
Epoch: 174 \ Validation Loss=0.010730
Epoch: 175 \ Training Loss=0.000640
Epoch: 175 \ Validation Loss=0.009539
Epoch: 176 \ Training Loss=0.000629
Epoch: 176 \ Validation Loss=0.009904
Epoch: 177 \ Training Loss=0.000613
Epoch: 177 \ Validation Loss=0.009870
Epoch: 178 \ Training Loss=0.000610
Epoch: 178 \ Validation Loss=0.010132
Epoch: 179 \ Training Loss=0.000595
Epoch: 179 \ Validation Loss=0.009249
Epoch: 180 \ Training Loss=0.000523
Epoch: 180 \ Validation Loss=0.009440
Epoch: 181 \ Training Loss=0.000594
Epoch: 181 \ Validation Loss=0.009874
Epoch: 182 \ Training Loss=0.000590
```

```
Epoch: 182 \ Validation Loss=0.009619
Epoch: 183 \ Training Loss=0.000654
Epoch: 183 \ Validation Loss=0.009511
Epoch: 184 \ Training Loss=0.000566
Epoch: 184 \ Validation Loss=0.009058
Epoch: 185 \ Training Loss=0.000574
Epoch: 185 \ Validation Loss=0.009134
Epoch: 186 \ Training Loss=0.000585
Epoch: 186 \ Validation Loss=0.009096
Epoch: 187 \ Training Loss=0.000555
Epoch: 187 \ Validation Loss=0.009490
Epoch: 188 \ Training Loss=0.000580
Epoch: 188 \ Validation Loss=0.008896
Epoch: 189 \ Training Loss=0.000540
Epoch: 189 \ Validation Loss=0.008948
Epoch: 190 \ Training Loss=0.000568
Epoch: 190 \ Validation Loss=0.009121
Epoch: 191 \ Training Loss=0.000576
Epoch: 191 \ Validation Loss=0.009554
Epoch: 192 \ Training Loss=0.000630
Epoch: 192 \ Validation Loss=0.008715
Epoch: 193 \ Training Loss=0.000574
Epoch: 193 \ Validation Loss=0.009064
Epoch: 194 \ Training Loss=0.000561
Epoch: 194 \ Validation Loss=0.008686
Epoch: 195 \ Training Loss=0.000599
Epoch: 195 \ Validation Loss=0.008640
Epoch: 196 \ Training Loss=0.000486
Epoch: 196 \ Validation Loss=0.009125
Epoch: 197 \ Training Loss=0.000547
Epoch: 197 \ Validation Loss=0.008513
Epoch: 198 \ Training Loss=0.000563
Epoch: 198 \ Validation Loss=0.009334
Epoch: 199 \ Training Loss=0.000567
Epoch: 199 \ Validation Loss=0.008639
Epoch: 200 \ Training Loss=0.000520
Epoch: 200 \ Validation Loss=0.008655
```

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

```
import matplotlib.pyplot as plt
import numpy as np

epoch_x = np.arange(0, 200, 1)
plt.plot(epoch_x, val_dice)
plt.title("DICE vs Epoch")
plt.xlabel('Epoch')
# naming the y axis
plt.ylabel('DICE')
plt.show()
```



Avg DICE socre of test data => 0.010767950986822447

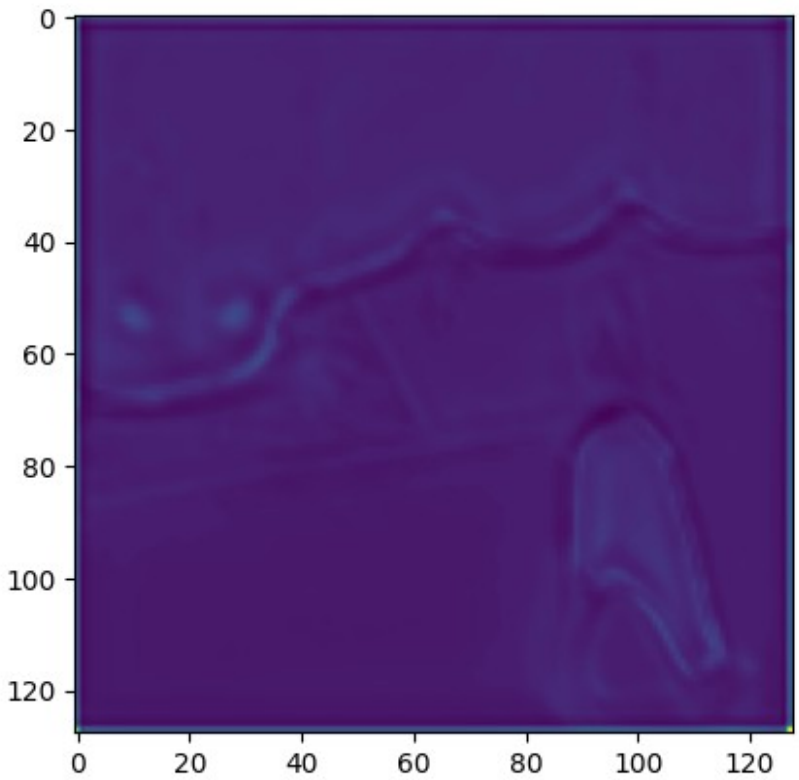
```
epoch_dice_score = 0
model = torch.load('best_model.pt')
model.eval()
with torch.no_grad():
  for (x, y) in test_loader:

      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))
```

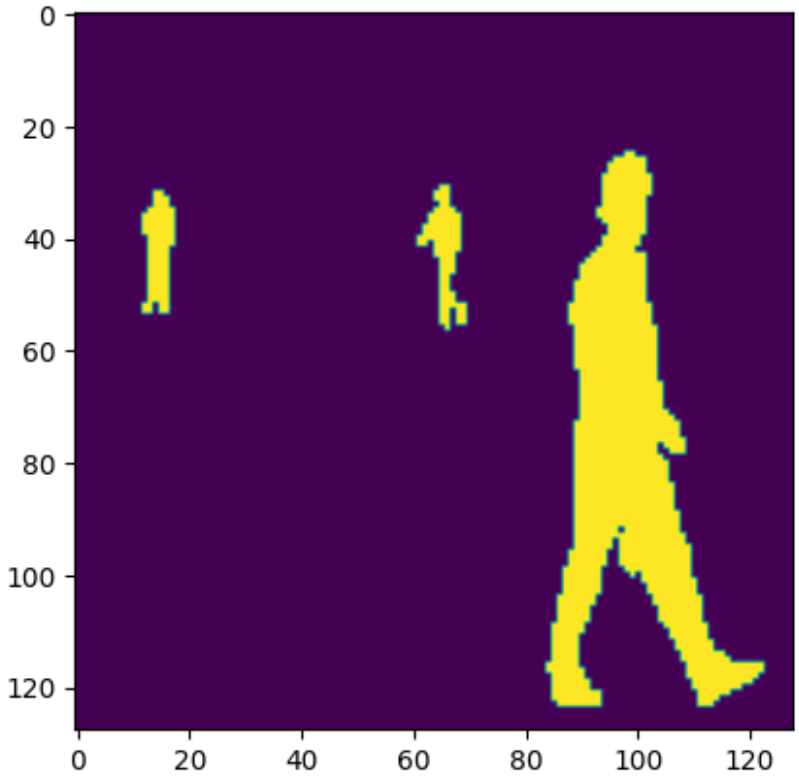
```
0.010767950986822447
```

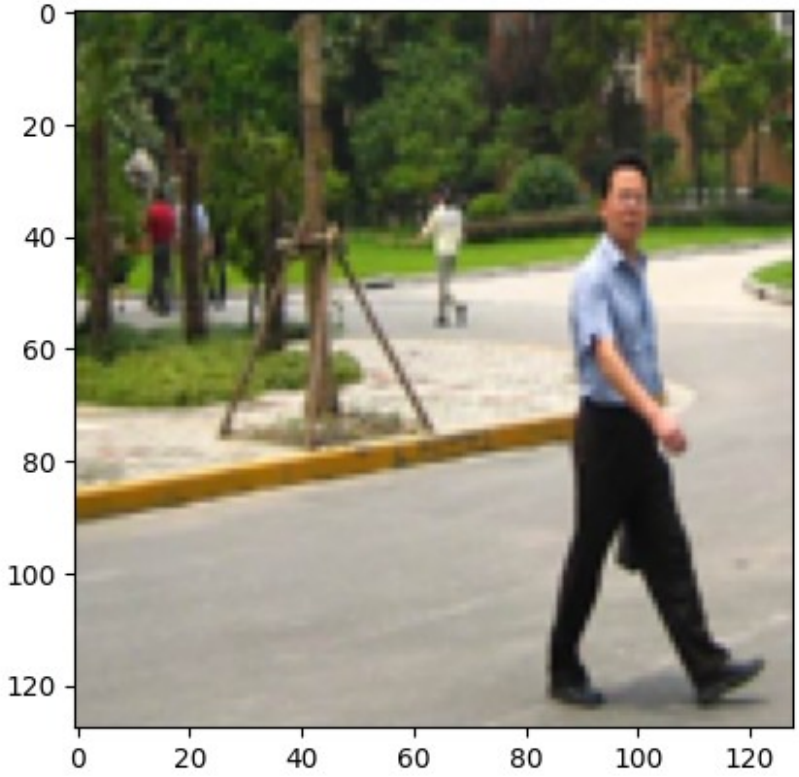
Image, Mask and predicted mask of test data

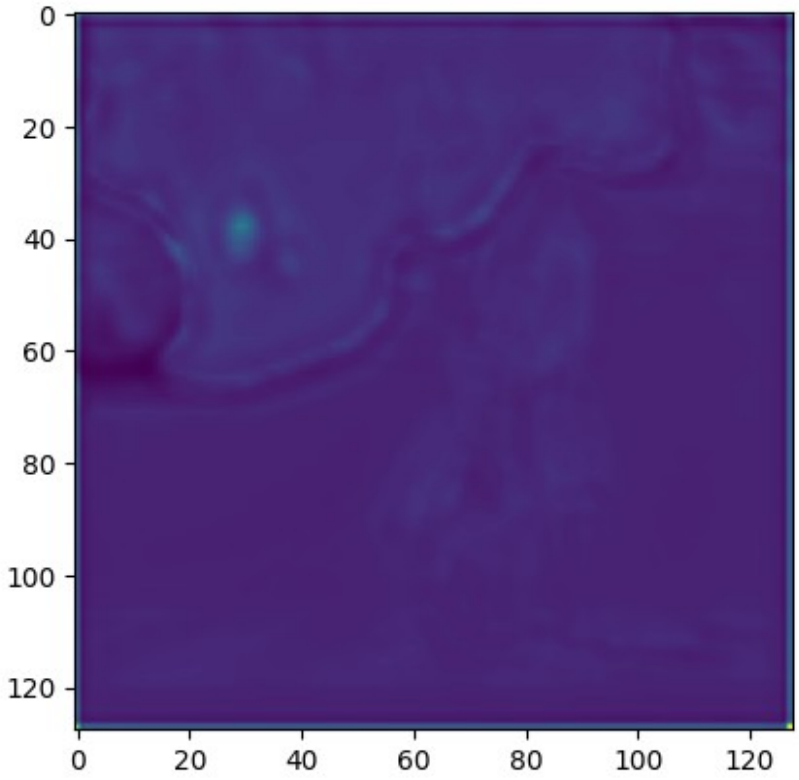
```
model = torch.load('best_model.pt')
model.eval()
num = 0
```

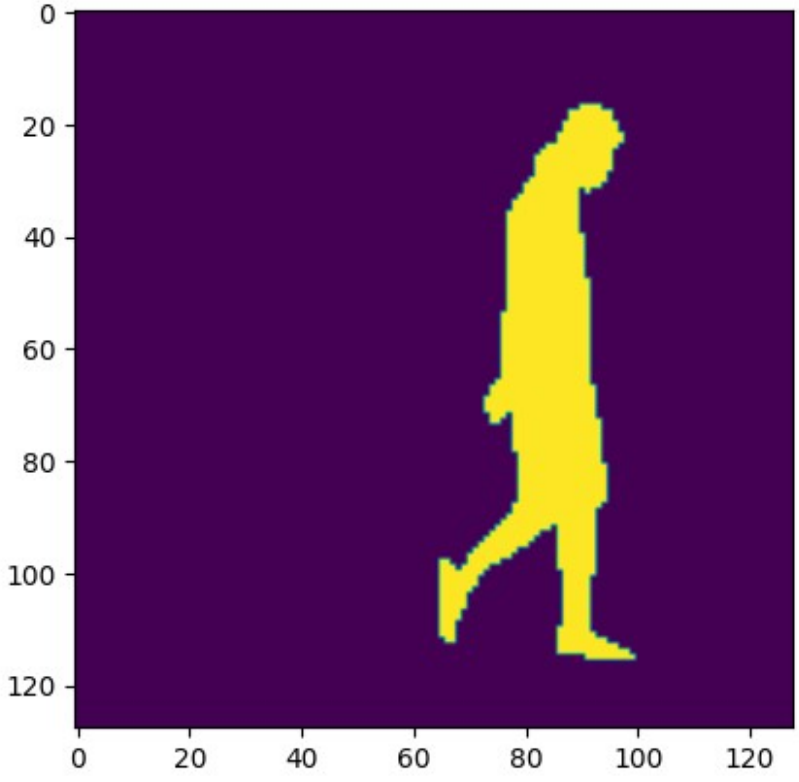
```python
with torch.no_grad():
  for (x, y) in test_loader:
    num = num + 1
    x = x.to(device)
    y = y.to(device)

    y_pred = model(x)

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

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

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

    if num == 3:
      break
```

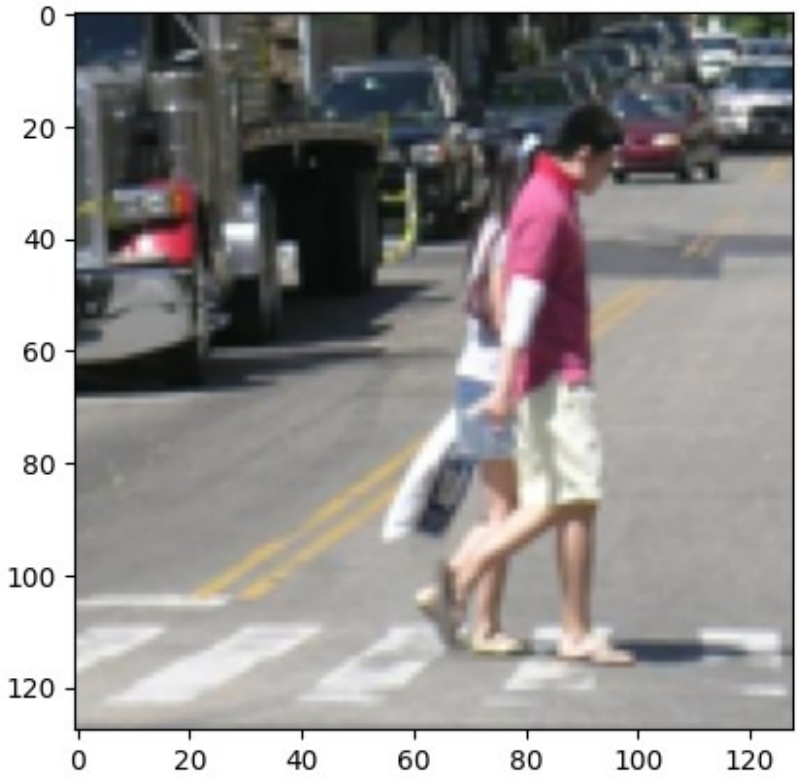


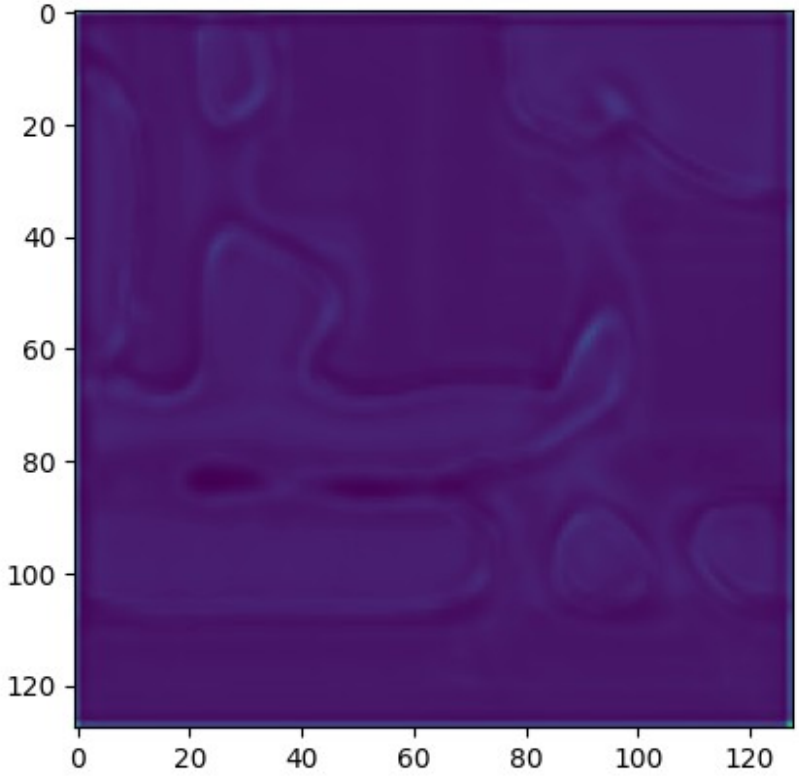

0
20
40
60
80
100
120
0
20
40
60
80
100
120

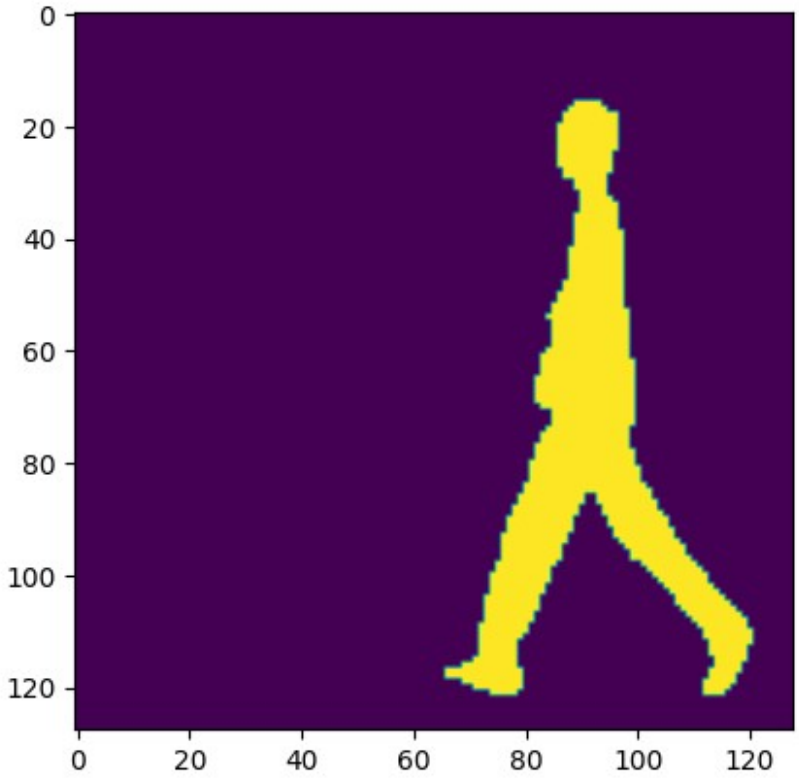

0
20
40
60
80
100
120
0
20
40
60
80
100
120

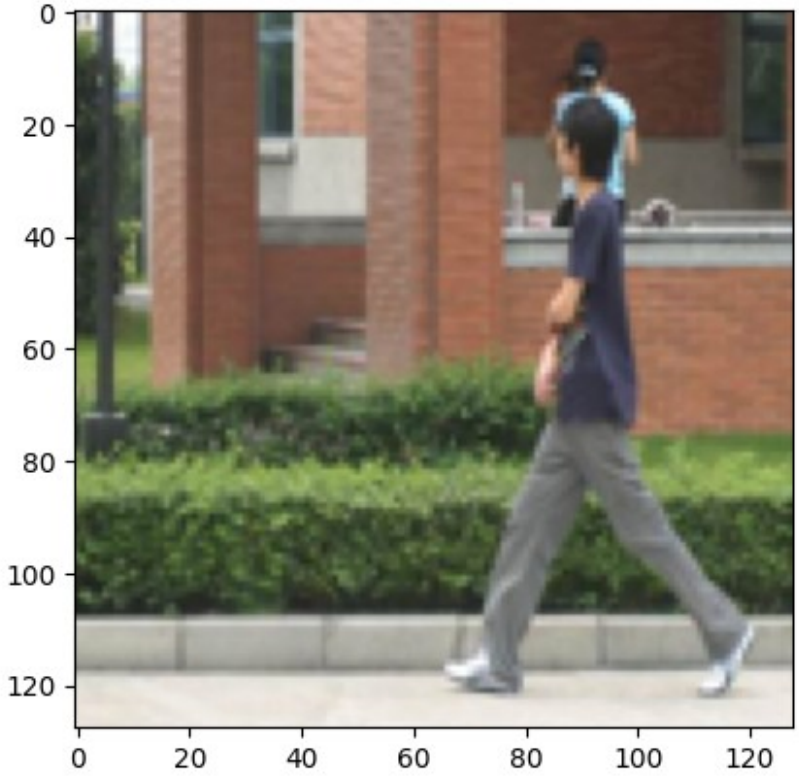

0
20
40
60
80
100
120
0
20
40
60
80
100
120


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

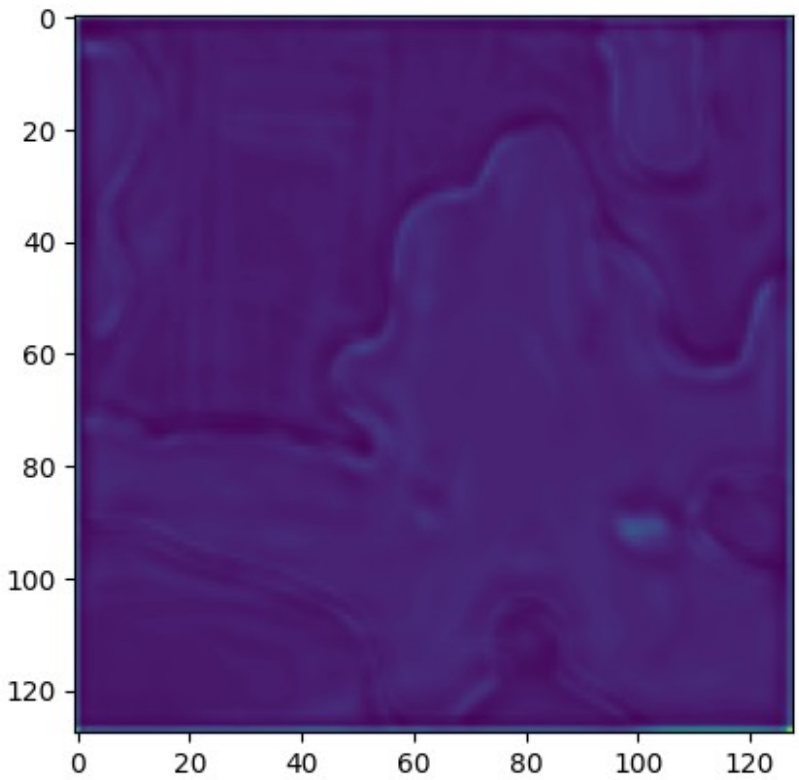

0
20
40
60
80
100
120
0
20
40
60
80
100
120

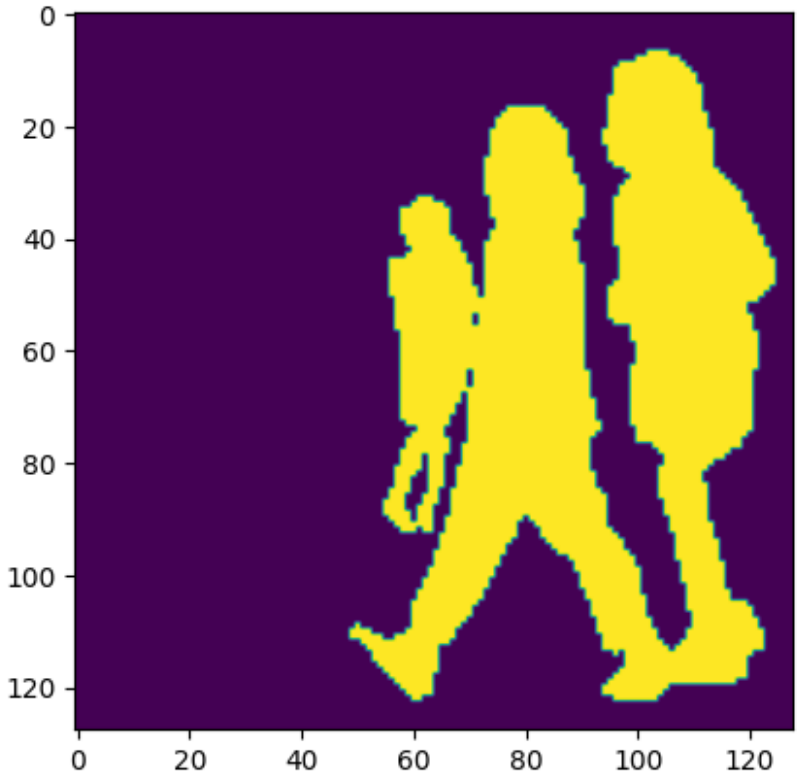

0
20
40
60
80
100
120
0
20
40
60
80
100
120

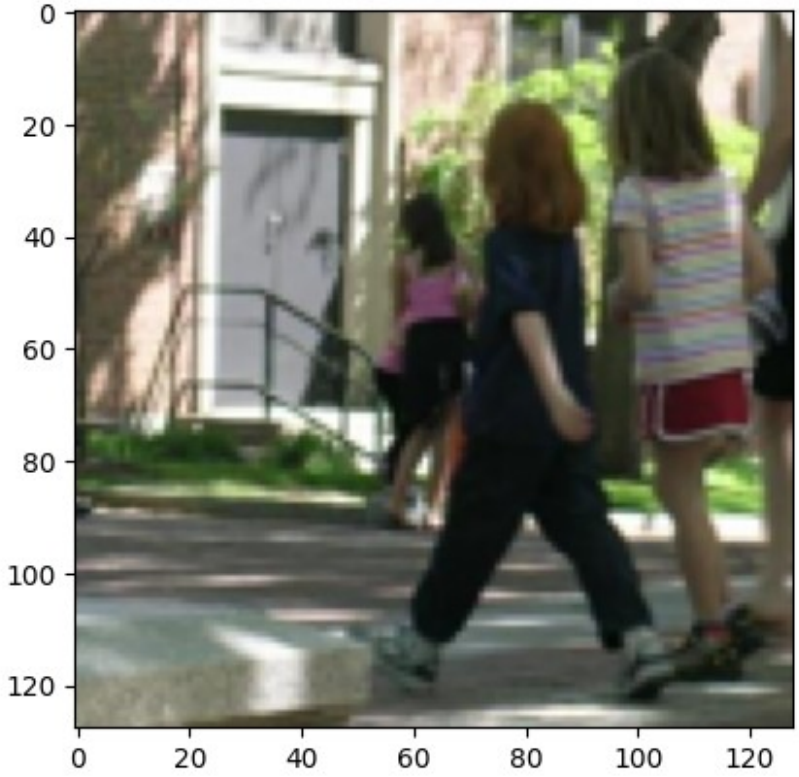

0
20
40
60
80
100
120
0
20
40
60
80
100
120

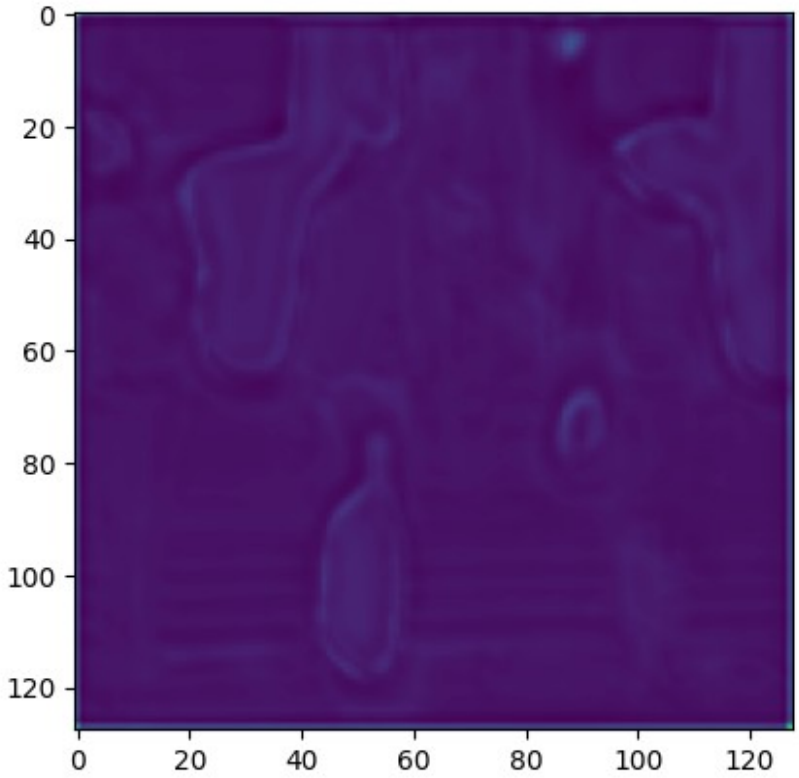
Image, Mask and predicted mask of Training data

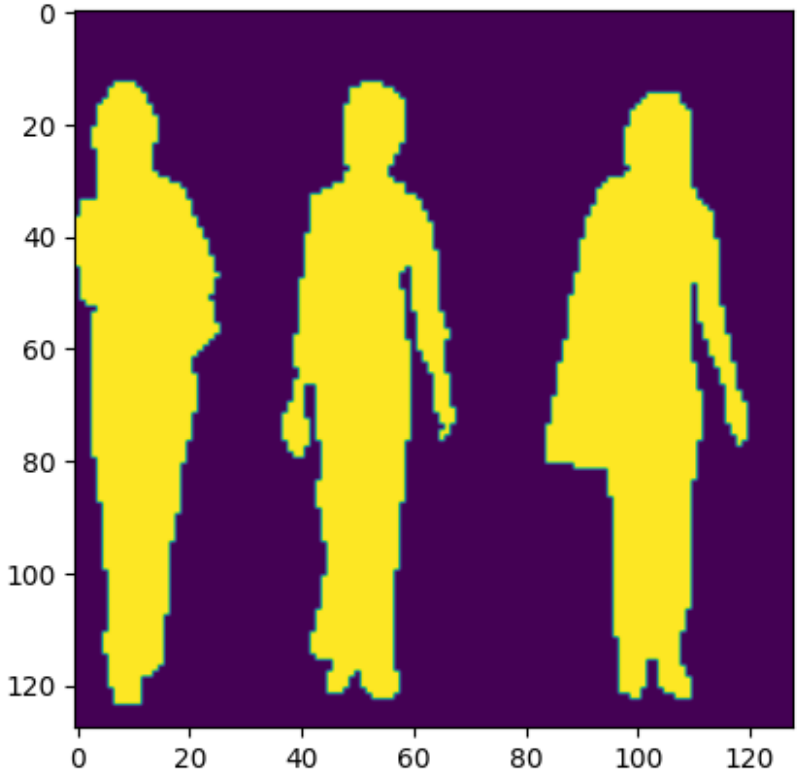
```
model = torch.load('best_model.pt')
model.eval()
num = 0
with torch.no_grad():
  for (x, y) in train_loader:
    num = num + 1
    x = x.to(device)
    y = y.to(device)

    y_pred = model(x)

    plt.imshow(y_pred[0].mul(255).permute(1,2,0).detach().numpy())
    plt.show()

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

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

    if num == 3:
      break
```

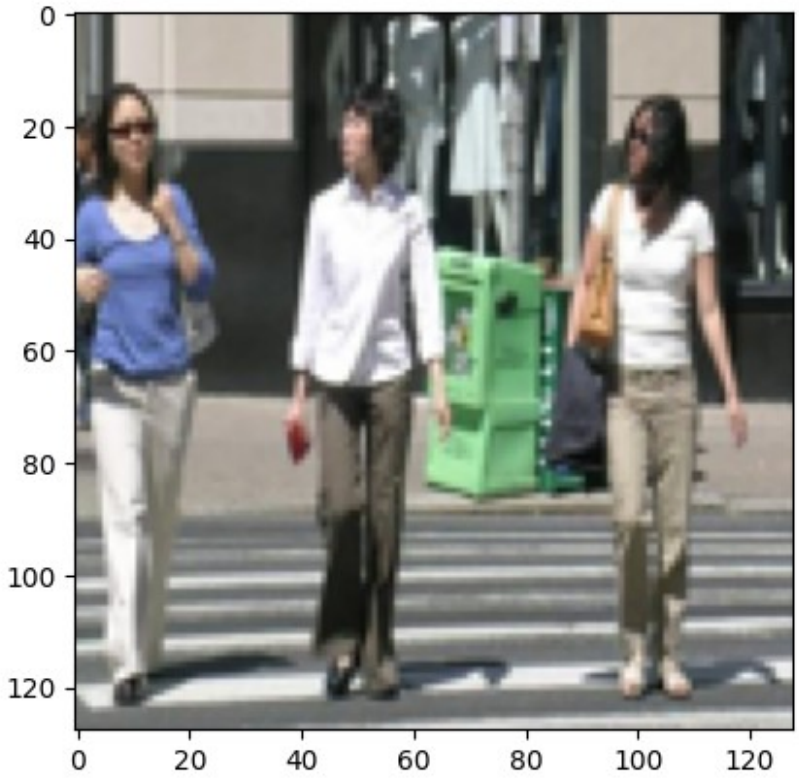


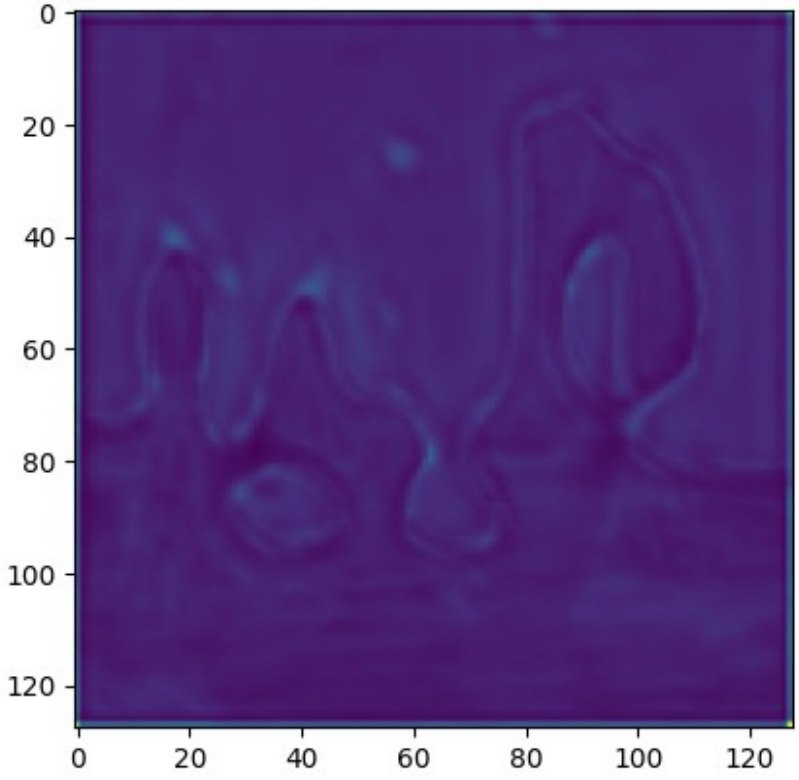

0
20
40
60
80
100
120
0
20
40
60
80
100
120

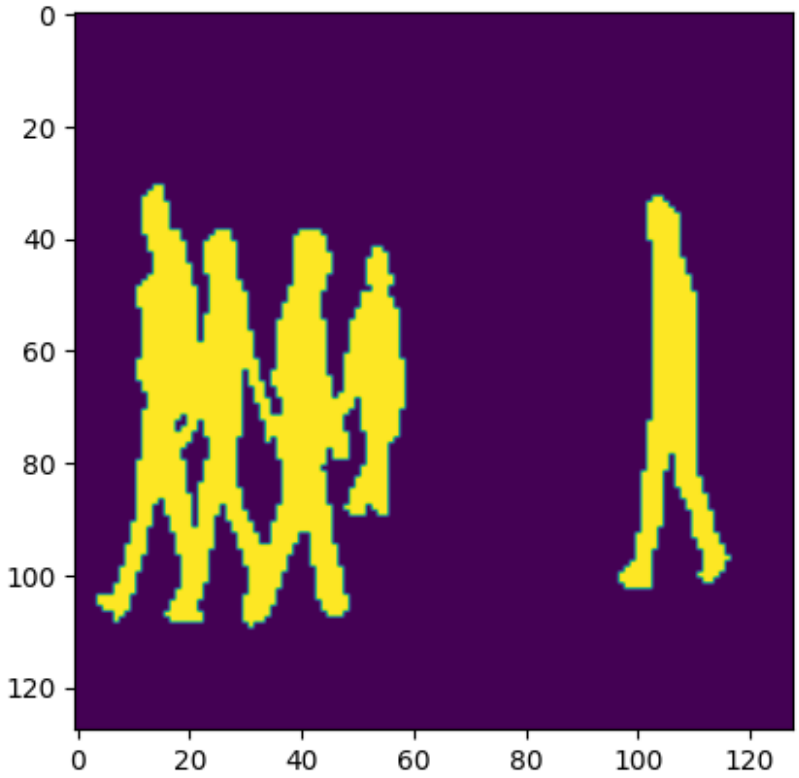

0
20
40
60
80
100
120
0
20
40
60
80
100
120

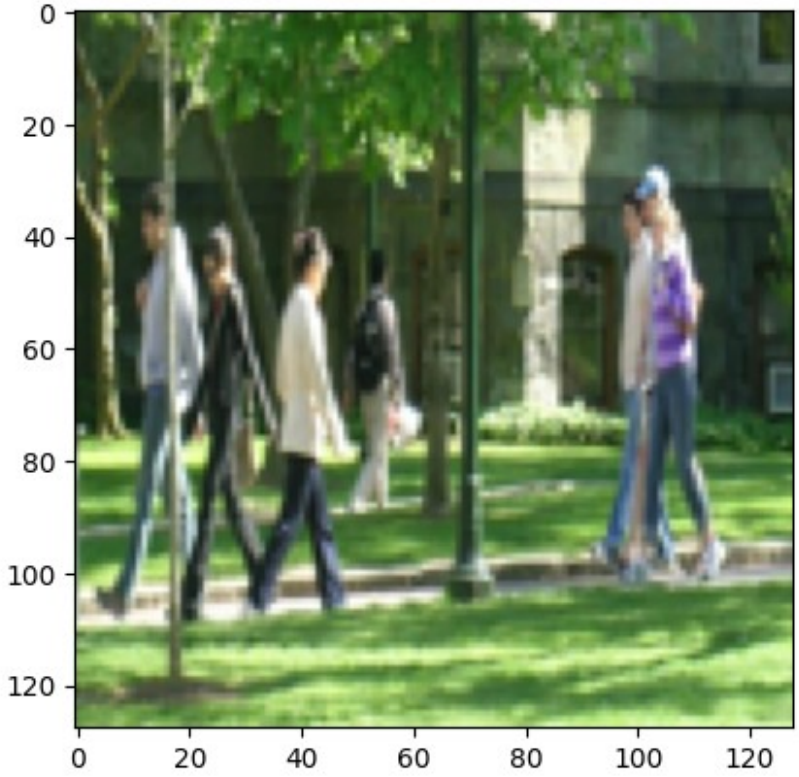

0
20
40
60
80
100
120
0
20
40
60
80
100
120


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


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

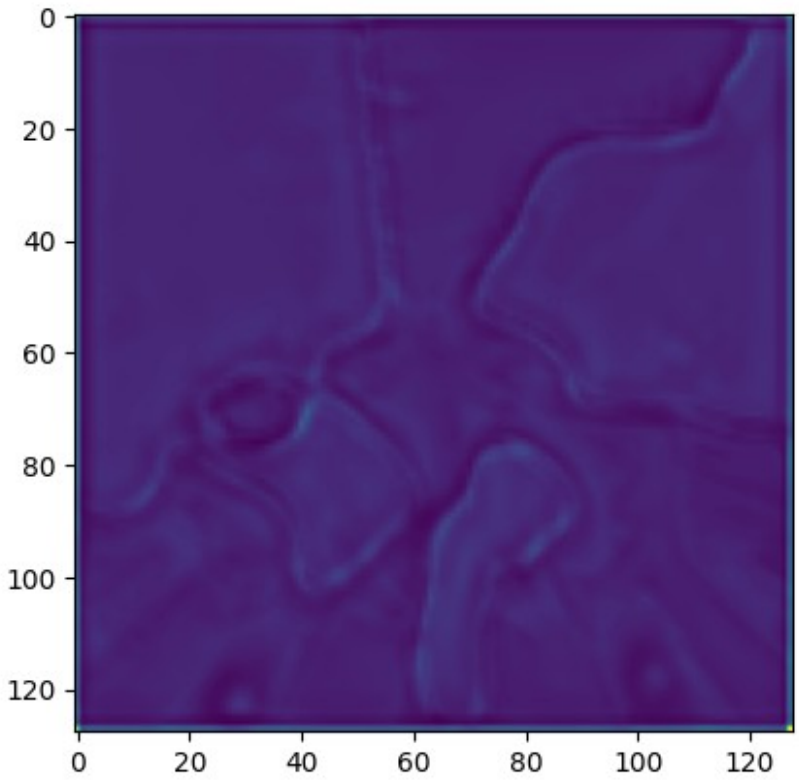

0
20
40
60
80
100
120
0
20
40
60
80
100
120

Model output on different image other than pennfundan dataset

```
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())
```

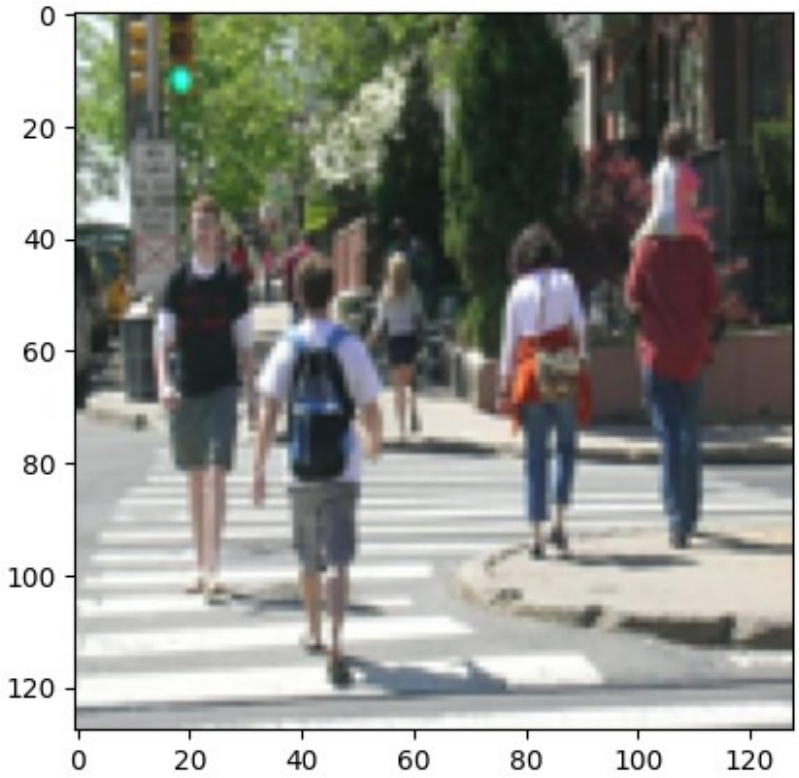
```
torch.Size([1, 128, 128])
```

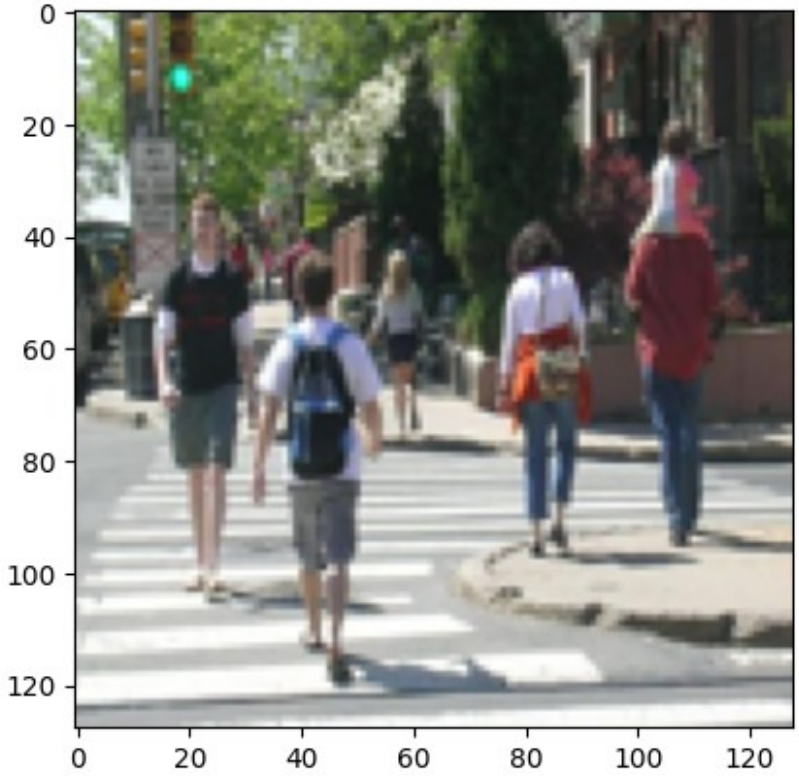
```
plt.imshow(prediction[0].permute(1,2,0).detach().numpy())
plt.show()
```

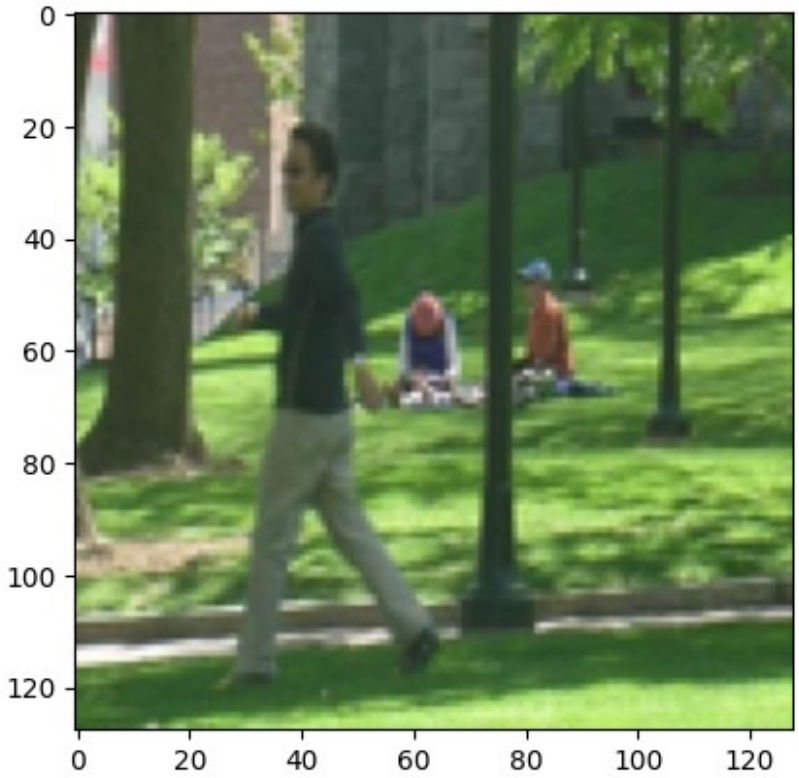


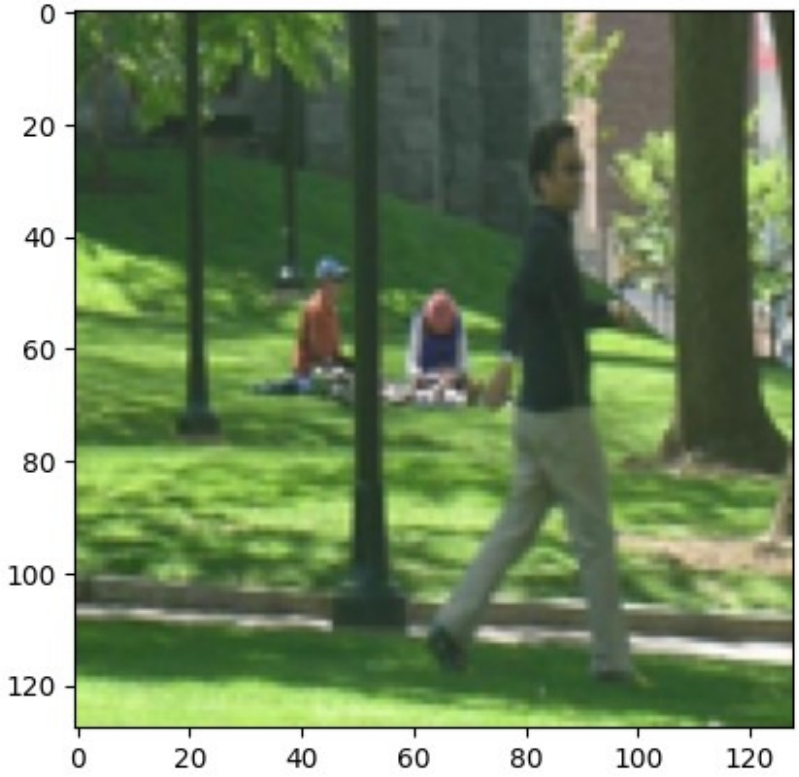
Example of Data Augmentation

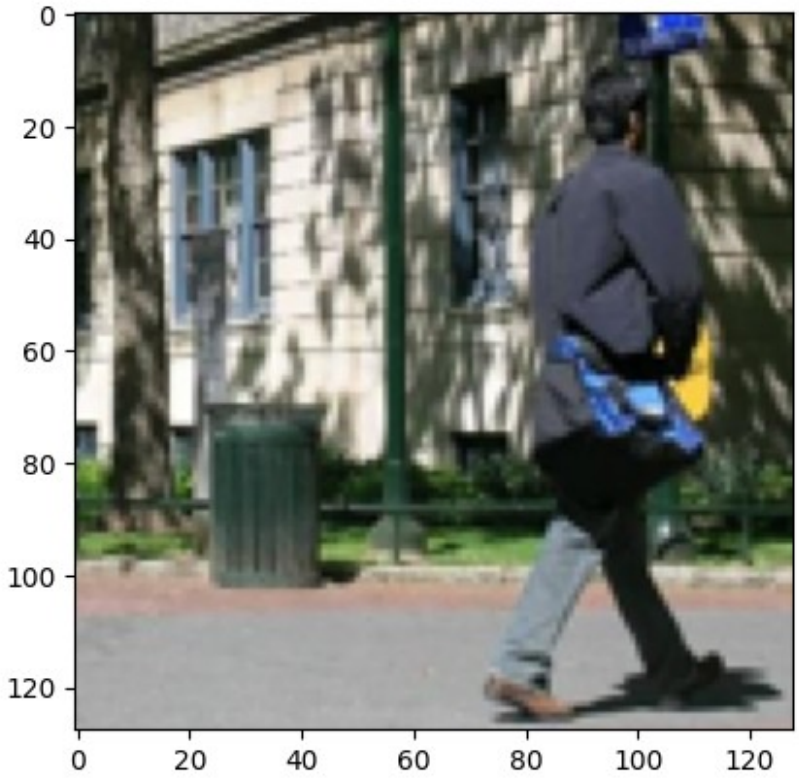
```
num = 0
for (x, y) in train_loader:
  num = num + 1
  x = x.to(device)
  y = y.to(device)

  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
```

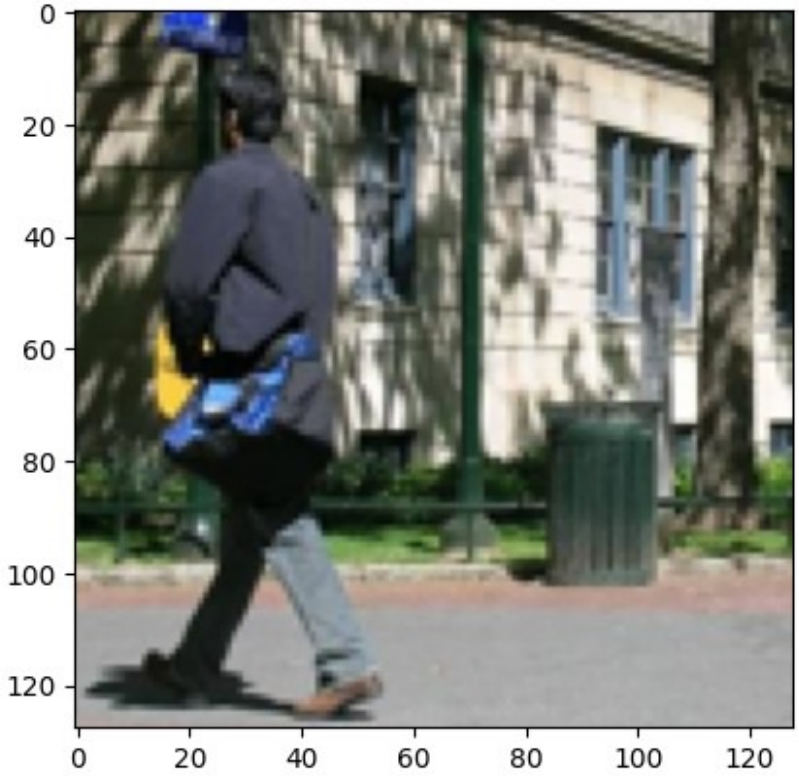

0
20
40
60
80
100
120
0
20
40
60
80
100
120


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


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