Abirami Sivakumar

as16288

```
!wget https://www.cis.upenn.edu/~jshi/ped_html/PennFudanPed.zip
!unzip /content/PennFudanPed.zip -d /content/
```

```
--2023-11-18 04:43:59--
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  30.0MB/s    in
1.7s

2023-11-18 04:44:01 (30.0 MB/s) - ‘PennFudanPed.zip’ saved
[53723336/53723336]

Archive:  /content/PennFudanPed.zip
   creating: /content/PennFudanPed/
  inflating: /content/PennFudanPed/added-object-list.txt
   creating: /content/PennFudanPed/Annotation/
  inflating: /content/PennFudanPed/Annotation/FudanPed00001.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00002.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00003.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00004.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00005.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00006.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00007.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00008.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00009.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00010.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00011.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00012.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00013.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00014.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00015.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00016.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00017.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00018.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00019.txt
```

```
  inflating: /content/PennFudanPed/Annotation/FudanPed00020.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00021.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00022.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00023.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00024.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00025.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00026.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00027.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00028.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00029.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00030.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00031.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00032.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00033.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00034.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00035.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00036.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00037.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00038.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00039.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00040.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00041.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00042.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00043.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00044.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00045.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00046.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00047.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00048.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00049.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00050.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00051.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00052.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00053.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00054.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00055.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00056.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00057.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00058.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00059.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00060.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00061.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00062.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00063.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00064.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00065.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00066.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00067.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00068.txt
```

```
  inflating: /content/PennFudanPed/Annotation/FudanPed00069.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00070.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00071.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00072.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00073.txt
  inflating: /content/PennFudanPed/Annotation/FudanPed00074.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00001.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00002.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00003.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00004.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00005.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00006.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00007.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00008.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00009.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00010.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00011.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00012.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00013.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00014.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00015.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00016.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00017.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00018.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00019.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00020.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00021.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00022.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00023.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00024.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00025.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00026.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00027.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00028.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00029.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00030.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00031.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00032.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00033.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00034.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00035.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00036.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00037.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00038.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00039.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00040.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00041.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00042.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00043.txt
```

```
  inflating: /content/PennFudanPed/Annotation/PennPed00044.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00045.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00046.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00047.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00048.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00049.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00050.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00051.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00052.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00053.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00054.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00055.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00056.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00057.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00058.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00059.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00060.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00061.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00062.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00063.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00064.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00065.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00066.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00067.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00068.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00069.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00070.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00071.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00072.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00073.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00074.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00075.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00076.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00077.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00078.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00079.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00080.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00081.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00082.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00083.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00084.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00085.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00086.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00087.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00088.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00089.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00090.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00091.txt
  inflating: /content/PennFudanPed/Annotation/PennPed00092.txt
```

```
  inflating: /content/PennFudanPed/Annotation/PennPed00093.txt  
  inflating: /content/PennFudanPed/Annotation/PennPed00094.txt  
  inflating: /content/PennFudanPed/Annotation/PennPed00095.txt  
  inflating: /content/PennFudanPed/Annotation/PennPed00096.txt  
   creating: /content/PennFudanPed/PedMasks/
  inflating: /content/PennFudanPed/PedMasks/FudanPed00001_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00002_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00003_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00004_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00005_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00006_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00007_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00008_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00009_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00010_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00011_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00012_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00013_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00014_mask.png  
 extracting: /content/PennFudanPed/PedMasks/FudanPed00015_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00016_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00017_mask.png  
 extracting: /content/PennFudanPed/PedMasks/FudanPed00018_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00019_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00020_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00021_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00022_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00023_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00024_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00025_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00026_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00027_mask.png  
 extracting: /content/PennFudanPed/PedMasks/FudanPed00028_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00029_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00030_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00031_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00032_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00033_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00034_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00035_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00036_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00037_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00038_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00039_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00040_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00041_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00042_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00043_mask.png  
  inflating: /content/PennFudanPed/PedMasks/FudanPed00044_mask.png  
```

```
  inflating: /content/PennFudanPed/PedMasks/FudanPed00045_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00046_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00047_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00048_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00049_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00050_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00051_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00052_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00053_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00054_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00055_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00056_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00057_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00058_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00059_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00060_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00061_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00062_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00063_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00064_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00065_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00066_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00067_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00068_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00069_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00070_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00071_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00072_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00073_mask.png
  inflating: /content/PennFudanPed/PedMasks/FudanPed00074_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00001_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00002_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00003_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00004_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00005_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00006_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00007_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00008_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00009_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00010_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00011_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00012_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00013_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00014_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00015_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00016_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00017_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00018_mask.png
  inflating: /content/PennFudanPed/PedMasks/PennPed00019_mask.png
```

```
  inflating: /content/PennFudanPed/PedMasks/PennPed00020_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00021_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00022_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00023_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00024_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00025_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00026_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00027_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00028_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00029_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00030_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00031_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00032_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00033_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00034_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00035_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00036_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00037_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00038_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00039_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00040_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00041_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00042_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00043_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00044_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00045_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00046_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00047_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00048_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00049_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00050_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00051_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00052_mask.png  
 extracting: /content/PennFudanPed/PedMasks/PennPed00053_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00054_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00055_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00056_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00057_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00058_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00059_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00060_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00061_mask.png  
 extracting: /content/PennFudanPed/PedMasks/PennPed00062_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00063_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00064_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00065_mask.png  
 extracting: /content/PennFudanPed/PedMasks/PennPed00066_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00067_mask.png  
  inflating: /content/PennFudanPed/PedMasks/PennPed00068_mask.png  
```

```
extracting: /content/PennFudanPed/PedMasks/PennPed00069_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00070_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00071_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00072_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00073_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00074_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00075_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00076_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00077_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00078_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00079_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00080_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00081_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00082_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00083_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00084_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00085_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00086_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00087_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00088_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00089_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00090_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00091_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00092_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00093_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00094_mask.png
 inflating: /content/PennFudanPed/PedMasks/PennPed00095_mask.png
extracting: /content/PennFudanPed/PedMasks/PennPed00096_mask.png
   creating: /content/PennFudanPed/PNGImages/
 inflating: /content/PennFudanPed/PNGImages/FudanPed00001.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00002.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00003.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00004.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00005.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00006.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00007.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00008.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00009.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00010.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00011.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00012.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00013.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00014.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00015.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00016.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00017.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00018.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00019.png
 inflating: /content/PennFudanPed/PNGImages/FudanPed00020.png
```

```
  inflating: /content/PennFudanPed/PNGImages/FudanPed00021.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00022.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00023.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00024.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00025.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00026.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00027.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00028.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00029.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00030.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00031.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00032.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00033.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00034.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00035.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00036.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00037.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00038.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00039.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00040.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00041.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00042.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00043.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00044.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00045.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00046.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00047.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00048.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00049.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00050.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00051.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00052.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00053.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00054.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00055.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00056.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00057.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00058.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00059.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00060.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00061.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00062.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00063.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00064.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00065.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00066.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00067.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00068.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00069.png
```

```
  inflating: /content/PennFudanPed/PNGImages/FudanPed00070.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00071.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00072.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00073.png
  inflating: /content/PennFudanPed/PNGImages/FudanPed00074.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00001.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00002.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00003.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00004.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00005.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00006.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00007.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00008.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00009.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00010.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00011.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00012.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00013.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00014.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00015.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00016.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00017.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00018.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00019.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00020.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00021.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00022.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00023.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00024.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00025.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00026.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00027.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00028.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00029.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00030.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00031.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00032.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00033.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00034.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00035.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00036.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00037.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00038.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00039.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00040.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00041.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00042.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00043.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00044.png
```

```
  inflating: /content/PennFudanPed/PNGImages/PennPed00045.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00046.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00047.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00048.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00049.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00050.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00051.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00052.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00053.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00054.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00055.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00056.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00057.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00058.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00059.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00060.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00061.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00062.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00063.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00064.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00065.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00066.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00067.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00068.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00069.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00070.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00071.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00072.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00073.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00074.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00075.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00076.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00077.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00078.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00079.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00080.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00081.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00082.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00083.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00084.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00085.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00086.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00087.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00088.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00089.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00090.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00091.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00092.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00093.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00094.png
```

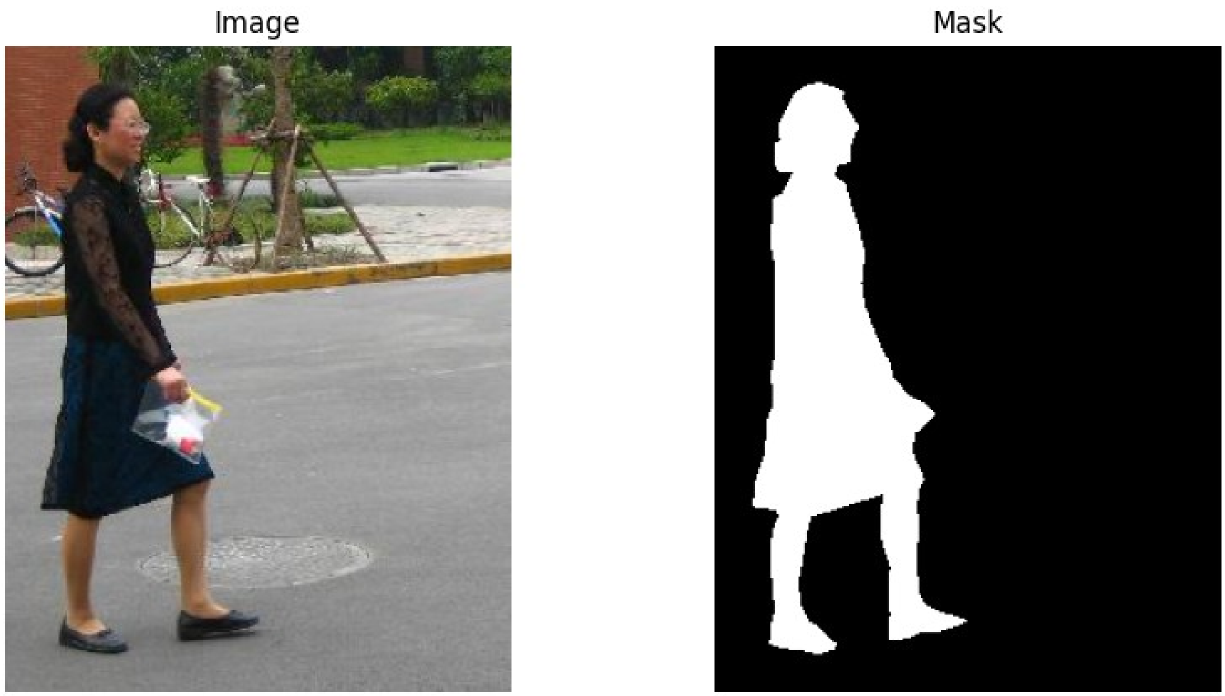
```
  inflating: /content/PennFudanPed/PNGImages/PennPed00095.png
  inflating: /content/PennFudanPed/PNGImages/PennPed00096.png
  inflating: /content/PennFudanPed/readme.txt
```

```
import numpy as np
import random
import shutil
import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from torch.utils.data import Dataset, DataLoader
from torchvision import transforms, utils
from PIL import Image
import torchvision.transforms.functional as TF
from torch import nn
import torch
import torch.optim as optim
from tqdm import tqdm
```

(a) Cut the FudanPed dataset into an 80-10-10 train-val-test split.

```
image_directory = '/content/PennFudanPed/PNGImages'
mask_directory = '/content/PennFudanPed/PedMasks'

train_dir = '/content/data/train'
valid_dir = '/content/data/valid'
test_dir = '/content/data/test'

os.makedirs(os.path.join(train_dir, 'images'), exist_ok=True)
os.makedirs(os.path.join(train_dir, 'masks'), exist_ok=True)
os.makedirs(os.path.join(valid_dir, 'images'), exist_ok=True)
os.makedirs(os.path.join(valid_dir, 'masks'), exist_ok=True)
os.makedirs(os.path.join(test_dir, 'images'), exist_ok=True)
os.makedirs(os.path.join(test_dir, 'masks'), exist_ok=True)

image_filenames = os.listdir(image_directory)
random.shuffle(image_filenames)

train_size = int(len(image_filenames) * 0.8)
valid_size = int(len(image_filenames) * 0.1)
test_size = len(image_filenames) - train_size - valid_size

train_filenames = image_filenames[:train_size]
valid_filenames = image_filenames[train_size:train_size + valid_size]
test_filenames = image_filenames[train_size + valid_size:]

def copy_files(filenames, source_dir, mask_dir, dest_image_dir,
dest_mask_dir):
    for filename in filenames:
        image_path = os.path.join(source_dir, filename)
```

```
        mask_path = os.path.join(mask_dir, filename.split('.')[0] +
'_mask.png')

        shutil.copy(image_path, os.path.join(dest_image_dir,
filename))
        shutil.copy(mask_path, os.path.join(dest_mask_dir,
filename.split('.')[0] + '_mask.png'))

copy_files(train_filenames, image_directory, mask_directory,
os.path.join(train_dir, 'images'), os.path.join(train_dir, 'masks'))
copy_files(valid_filenames, image_directory, mask_directory,
os.path.join(valid_dir, 'images'), os.path.join(valid_dir, 'masks'))
copy_files(test_filenames, image_directory, mask_directory,
os.path.join(test_dir, 'images'), os.path.join(test_dir, 'masks'))

train_images_path = '/content/data/train/images'
train_masks_path = '/content/data/train/masks'

train_image_filenames = os.listdir(train_images_path)

image_index = 20

image_file_path = os.path.join(train_images_path,
train_image_filenames[image_index])
mask_file_path = os.path.join(train_masks_path,
train_image_filenames[image_index].split('.')[0] + '_mask.png')

img = mpimg.imread(image_file_path)
mask = mpimg.imread(mask_file_path)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img)
plt.title("Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(mask, cmap='gray')
plt.title("Mask")
plt.axis('off')

plt.show()
```




```python
class ImageTransforms:
    def __init__(self):
        self.image_resizer = transforms.Resize((128, 128), 
Image.BICUBIC)
        self.mask_resizer = transforms.Resize((128, 128), 
Image.BICUBIC)

    def transform(self, image, mask, convert_to_tensor=True):
        rotation_angle = random.randint(-30, 30)

        image = TF.rotate(image, rotation_angle)
        mask = TF.rotate(mask, rotation_angle)

        if convert_to_tensor:
            image = self.image_resizer(image)
            mask = self.mask_resizer(mask)

            image = TF.to_tensor(image)
            mask = TF.to_tensor(mask)
            mask = (mask > 0).float()

            normalizer = transforms.Normalize(mean=[0.485, 0.456, 
0.406], std=[0.229, 0.224, 0.225])
            image = normalizer(image)

        return image, mask
```

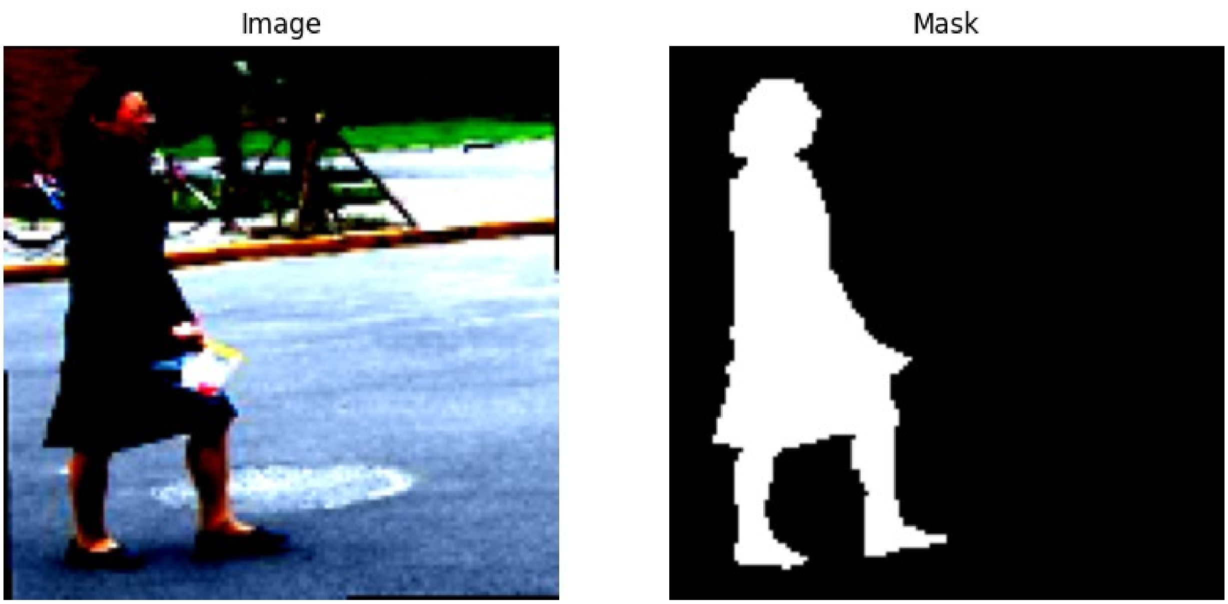
```
    def __call__(self, image, mask):
        return self.transform(image, mask)

class PennFudanDataset(Dataset):
    def __init__(self, images_directory, masks_directory,
transform=None):
        self.images_directory = images_directory
        self.masks_directory = masks_directory
        self.transform = transform
        self.image_files = os.listdir(self.images_directory)

    def __len__(self):
        return len(self.image_files)

    def __getitem__(self, idx):
        image_path = os.path.join(self.images_directory,
self.image_files[idx])
        mask_file = self.image_files[idx].split('.')[0] + '_mask.png'
        mask_path = os.path.join(self.masks_directory, mask_file)

        image = Image.open(image_path).convert("RGB")
        mask = Image.open(mask_path).convert("L")

        if self.transform:
            image, mask = self.transform(image, mask)

        return image, mask
```

(b) Apply data augmentation to your dataset during training and show an example of your data augmentation in your report.

```
imageTransforms = ImageTransforms()
train = PennFudanDataset('/content/data/train/images/',
'/content/data/train/masks/', imageTransforms)
valid = PennFudanDataset('/content/data/valid/images/',
'/content/data/valid/masks/', imageTransforms)
test = PennFudanDataset('/content/data/test/images/',
'/content/data/test/masks/', imageTransforms)

img, mask = train[20]

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img.permute(1, 2, 0))
plt.title("Image")
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(mask.squeeze(), cmap='gray')
```

```
plt.title("Mask")
plt.axis('off')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



Above image is after augmentation which is rotated and reversed.

```
trainLoader = DataLoader(train, batch_size=4, shuffle=True)
validLoader = DataLoader(valid, batch_size=4, shuffle=False)
testLoader = DataLoader(test, batch_size=4, shuffle=False)
```

(c) Implement and train a CNN for binary segmentation on your train split. Describe your network architecture2 , loss function, and any training hyper-parameters. You may implement any architecture you'd like, but the implementation must be your own code.

Using the UNet architecture provided. Resizing images to 128x128 before passing. I used Dice Loss. I used Adam optimizer. I used a learning rate scheduler with the from 0.0001 and decreases by a factor of 0.1 every 20 epochs.

```
class ConvolutionalBlock(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(ConvolutionalBlock, self).__init__()
        self.relu = nn.ReLU(inplace=True)
        self.conv1 = nn.Conv2d(in_channels, out_channels,
kernel_size=3, padding=1)
```

```python
        self.conv2 = nn.Conv2d(out_channels, out_channels,
kernel_size=3, padding=1)
        self.batch_norm1 = nn.BatchNorm2d(out_channels)
        self.batch_norm2 = nn.BatchNorm2d(out_channels)

    def forward(self, x):
        x = self.conv1(x)
        x = self.batch_norm1(x)
        x = self.relu(x)
        x = self.conv2(x)
        x = self.batch_norm2(x)
        x = self.relu(x)
        return x

class UNet(nn.Module):
    def __init__(self):
        super(UNet, self).__init__()
        self.max_pool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.final_conv = nn.Conv2d(16, 1, 1)
        self.down_block1 = ConvolutionalBlock(3, 16)
        self.down_block2 = ConvolutionalBlock(16, 32)
        self.final_block = ConvolutionalBlock(32, 32)
        self.up_sample1 = nn.Upsample(scale_factor=2, mode='bilinear',
align_corners=True)
        self.up_block1 = ConvolutionalBlock(64, 16)
        self.up_sample2 = nn.Upsample(scale_factor=2, mode='bilinear',
align_corners=True)
        self.up_block2 = ConvolutionalBlock(32, 16)
        self.sigmoid_activation = nn.Sigmoid()

    def forward(self, x):
        down1 = self.down_block1(x)
        pooled1 = self.max_pool(down1)
        down2 = self.down_block2(pooled1)
        pooled2 = self.max_pool(down2)
        bridge = self.final_block(pooled2)
        up1 = self.up_sample1(bridge)
        merge1 = torch.cat([up1, down2], dim=1)
        up_block1 = self.up_block1(merge1)
        up2 = self.up_sample2(up_block1)
        merge2 = torch.cat([up2, down1], dim=1)
        up_block2 = self.up_block2(merge2)
        final_conv = self.final_conv(up_block2)
        output = self.sigmoid_activation(final_conv)
        return output

device = torch.device("cuda:0" if torch.cuda.is_available() else
"cpu")
```

```
model = UNet()
model = model.to(device)

class DiceLoss(nn.Module):
    def forward(self, inputs, targets):
        inputs = inputs.sigmoid()
        intersection = (inputs * targets).sum()
        dice = (2.*intersection) / (inputs.sum() + targets.sum())
        return 1 - dice

loss_function = DiceLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)
scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=30,
gamma=0.1)

def dice_coefficient(outputs, labels):
    outputs = (outputs > 0.5).float()
    labels = labels.float()
    intersect = (outputs * labels).sum()
    return (2. * intersect) / (outputs.sum() + labels.sum())

training_losses = []
validation_losses = []
average_dice_scores = []

for epoch in range(40):
    model.train()
    total_train_loss = 0.0
    for data in tqdm(trainLoader, desc=f"Training Epoch {epoch+1}"):
        images, true_masks = data[0].to(device), data[1].to(device)
        optimizer.zero_grad()
        predicted_masks = model(images)
        loss = loss_function(predicted_masks, true_masks)
        loss.backward()
        optimizer.step()
        total_train_loss += loss.item()

    mean_train_loss = total_train_loss / len(trainLoader)
    print(f"Epoch {epoch+1}: Average Training Loss:
{mean_train_loss:.4f}")

    model.eval()
    total_val_loss = 0.0
    epoch_dice_scores = []
    with torch.no_grad():
        for data in tqdm(validLoader, desc=f"Validating Epoch
{epoch+1}"):
            images, true_masks = data[0].to(device),
data[1].to(device)
            predicted_masks = model(images)
```

```
            loss = loss_function(predicted_masks, true_masks)
            total_val_loss += loss.item()

            dice_score = dice_coefficient(predicted_masks, true_masks)
            epoch_dice_scores.append(dice_score.item())

    scheduler.step()
    mean_val_loss = total_val_loss / len(validLoader)
    mean_dice_score = sum(epoch_dice_scores) / len(epoch_dice_scores)
    current_learning_rate = scheduler.get_last_lr()[0]

    training_losses.append(mean_train_loss)
    validation_losses.append(mean_val_loss)
    average_dice_scores.append(mean_dice_score)

    print(f"Epoch {epoch+1}: Val Loss: {mean_val_loss:.4f}, Dice:
{mean_dice_score:.4f}")
```

```
Training Epoch 1: 100%|██████████| 34/34 [00:02<00:00, 13.80it/s]

Epoch 1: Average Training Loss: 0.7078

Validating Epoch 1: 100%|██████████| 5/5 [00:00<00:00, 21.24it/s]

Epoch 1: Val Loss: 0.6505, Dice: 0.5942

Training Epoch 2: 100%|██████████| 34/34 [00:02<00:00, 13.08it/s]

Epoch 2: Average Training Loss: 0.7044

Validating Epoch 2: 100%|██████████| 5/5 [00:00<00:00, 14.23it/s]

Epoch 2: Val Loss: 0.6472, Dice: 0.6080

Training Epoch 3: 100%|██████████| 34/34 [00:04<00:00,  7.62it/s]

Epoch 3: Average Training Loss: 0.7019

Validating Epoch 3: 100%|██████████| 5/5 [00:00<00:00, 21.52it/s]

Epoch 3: Val Loss: 0.6469, Dice: 0.5780

Training Epoch 4: 100%|██████████| 34/34 [00:02<00:00, 13.76it/s]

Epoch 4: Average Training Loss: 0.7012

Validating Epoch 4: 100%|██████████| 5/5 [00:00<00:00, 20.21it/s]

Epoch 4: Val Loss: 0.6418, Dice: 0.6235

Training Epoch 5: 100%|██████████| 34/34 [00:02<00:00, 13.79it/s]

Epoch 5: Average Training Loss: 0.6993
```

```
Validating Epoch 5: 100%|██████████| 5/5 [00:00<00:00, 21.60it/s]
Epoch 5: Val Loss: 0.6454, Dice: 0.6530
Training Epoch 6: 100%|██████████| 34/34 [00:02<00:00, 13.71it/s]
Epoch 6: Average Training Loss: 0.6967
Validating Epoch 6: 100%|██████████| 5/5 [00:00<00:00, 21.86it/s]
Epoch 6: Val Loss: 0.6404, Dice: 0.6351
Training Epoch 7: 100%|██████████| 34/34 [00:03<00:00, 10.10it/s]
Epoch 7: Average Training Loss: 0.6959
Validating Epoch 7: 100%|██████████| 5/5 [00:00<00:00, 14.72it/s]
Epoch 7: Val Loss: 0.6402, Dice: 0.6259
Training Epoch 8: 100%|██████████| 34/34 [00:02<00:00, 12.96it/s]
Epoch 8: Average Training Loss: 0.6965
Validating Epoch 8: 100%|██████████| 5/5 [00:00<00:00, 21.48it/s]
Epoch 8: Val Loss: 0.6409, Dice: 0.6532
Training Epoch 9: 100%|██████████| 34/34 [00:02<00:00, 13.73it/s]
Epoch 9: Average Training Loss: 0.6944
Validating Epoch 9: 100%|██████████| 5/5 [00:00<00:00, 21.18it/s]
Epoch 9: Val Loss: 0.6402, Dice: 0.6601
Training Epoch 10: 100%|██████████| 34/34 [00:03<00:00,  9.87it/s]
Epoch 10: Average Training Loss: 0.6929
Validating Epoch 10: 100%|██████████| 5/5 [00:00<00:00, 20.76it/s]
Epoch 10: Val Loss: 0.6382, Dice: 0.6614
Training Epoch 11: 100%|██████████| 34/34 [00:02<00:00, 11.39it/s]
Epoch 11: Average Training Loss: 0.6926
Validating Epoch 11: 100%|██████████| 5/5 [00:00<00:00, 13.79it/s]
Epoch 11: Val Loss: 0.6396, Dice: 0.6718
Training Epoch 12: 100%|██████████| 34/34 [00:03<00:00, 10.93it/s]
Epoch 12: Average Training Loss: 0.6926
```

```
Validating Epoch 12: 100%|██████████| 5/5 [00:00<00:00, 21.03it/s]
Epoch 12: Val Loss: 0.6349, Dice: 0.6483
Training Epoch 13: 100%|██████████| 34/34 [00:02<00:00, 13.38it/s]
Epoch 13: Average Training Loss: 0.6905
Validating Epoch 13: 100%|██████████| 5/5 [00:00<00:00, 21.21it/s]
Epoch 13: Val Loss: 0.6334, Dice: 0.6677
Training Epoch 14: 100%|██████████| 34/34 [00:02<00:00, 13.68it/s]
Epoch 14: Average Training Loss: 0.6889
Validating Epoch 14: 100%|██████████| 5/5 [00:00<00:00, 20.31it/s]
Epoch 14: Val Loss: 0.6359, Dice: 0.6407
Training Epoch 15: 100%|██████████| 34/34 [00:02<00:00, 13.66it/s]
Epoch 15: Average Training Loss: 0.6908
Validating Epoch 15: 100%|██████████| 5/5 [00:00<00:00, 19.69it/s]
Epoch 15: Val Loss: 0.6341, Dice: 0.6453
Training Epoch 16: 100%|██████████| 34/34 [00:03<00:00,  9.66it/s]
Epoch 16: Average Training Loss: 0.6898
Validating Epoch 16: 100%|██████████| 5/5 [00:00<00:00, 16.50it/s]
Epoch 16: Val Loss: 0.6333, Dice: 0.6789
Training Epoch 17: 100%|██████████| 34/34 [00:02<00:00, 13.74it/s]
Epoch 17: Average Training Loss: 0.6882
Validating Epoch 17: 100%|██████████| 5/5 [00:00<00:00, 20.79it/s]
Epoch 17: Val Loss: 0.6319, Dice: 0.6624
Training Epoch 18: 100%|██████████| 34/34 [00:02<00:00, 13.64it/s]
Epoch 18: Average Training Loss: 0.6885
Validating Epoch 18: 100%|██████████| 5/5 [00:00<00:00, 21.51it/s]
Epoch 18: Val Loss: 0.6303, Dice: 0.6639
Training Epoch 19: 100%|██████████| 34/34 [00:02<00:00, 13.43it/s]
Epoch 19: Average Training Loss: 0.6866
```

```
Validating Epoch 19: 100%|██████████| 5/5 [00:00<00:00, 20.90it/s]
Epoch 19: Val Loss: 0.6316, Dice: 0.7014
Training Epoch 20: 100%|██████████| 34/34 [00:06<00:00,  5.63it/s]
Epoch 20: Average Training Loss: 0.6861
Validating Epoch 20: 100%|██████████| 5/5 [00:00<00:00, 21.30it/s]
Epoch 20: Val Loss: 0.6318, Dice: 0.6901
Training Epoch 21: 100%|██████████| 34/34 [00:02<00:00, 13.73it/s]
Epoch 21: Average Training Loss: 0.6855
Validating Epoch 21: 100%|██████████| 5/5 [00:00<00:00, 20.82it/s]
Epoch 21: Val Loss: 0.6301, Dice: 0.6869
Training Epoch 22: 100%|██████████| 34/34 [00:03<00:00,  9.98it/s]
Epoch 22: Average Training Loss: 0.6845
Validating Epoch 22: 100%|██████████| 5/5 [00:00<00:00, 21.03it/s]
Epoch 22: Val Loss: 0.6305, Dice: 0.7013
Training Epoch 23: 100%|██████████| 34/34 [00:02<00:00, 13.77it/s]
Epoch 23: Average Training Loss: 0.6841
Validating Epoch 23: 100%|██████████| 5/5 [00:00<00:00, 21.63it/s]
Epoch 23: Val Loss: 0.6277, Dice: 0.7112
Training Epoch 24: 100%|██████████| 34/34 [00:03<00:00, 10.19it/s]
Epoch 24: Average Training Loss: 0.6837
Validating Epoch 24: 100%|██████████| 5/5 [00:00<00:00, 14.78it/s]
Epoch 24: Val Loss: 0.6292, Dice: 0.6620
Training Epoch 25: 100%|██████████| 34/34 [00:02<00:00, 12.98it/s]
Epoch 25: Average Training Loss: 0.6823
Validating Epoch 25: 100%|██████████| 5/5 [00:00<00:00, 21.76it/s]
Epoch 25: Val Loss: 0.6276, Dice: 0.6946
Training Epoch 26: 100%|██████████| 34/34 [00:02<00:00, 13.73it/s]
Epoch 26: Average Training Loss: 0.6846
```

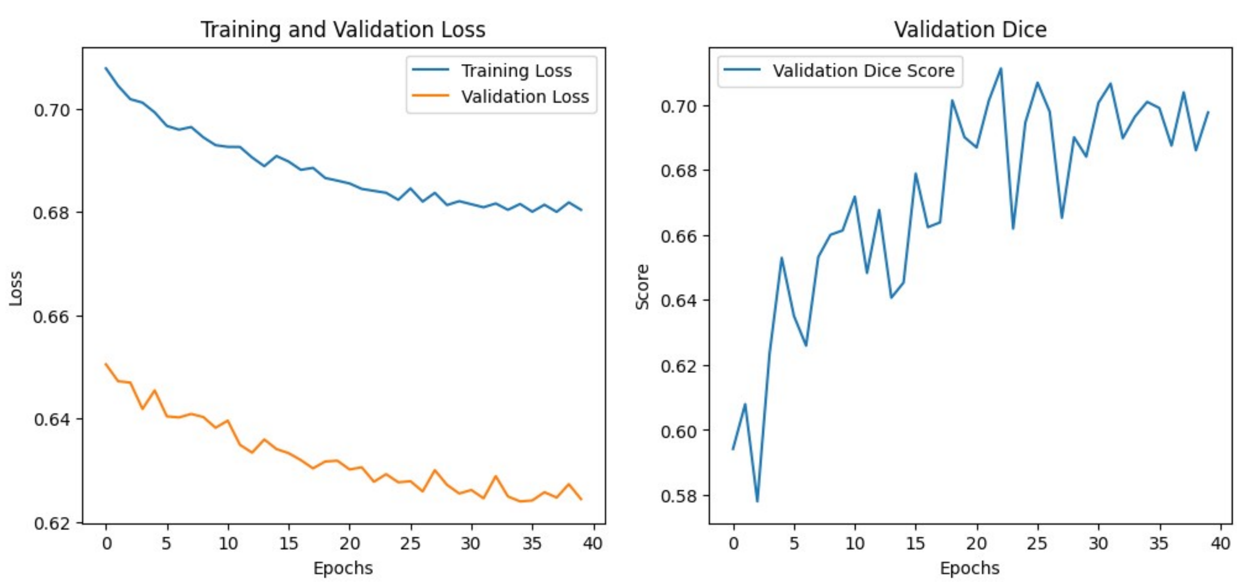
```
Validating Epoch 26: 100%|██████████| 5/5 [00:00<00:00, 21.31it/s]
Epoch 26: Val Loss: 0.6278, Dice: 0.7069
Training Epoch 27: 100%|██████████| 34/34 [00:02<00:00, 13.90it/s]
Epoch 27: Average Training Loss: 0.6820
Validating Epoch 27: 100%|██████████| 5/5 [00:00<00:00, 20.02it/s]
Epoch 27: Val Loss: 0.6258, Dice: 0.6979
Training Epoch 28: 100%|██████████| 34/34 [00:02<00:00, 13.01it/s]
Epoch 28: Average Training Loss: 0.6837
Validating Epoch 28: 100%|██████████| 5/5 [00:00<00:00, 14.44it/s]
Epoch 28: Val Loss: 0.6300, Dice: 0.6653
Training Epoch 29: 100%|██████████| 34/34 [00:03<00:00,  9.79it/s]
Epoch 29: Average Training Loss: 0.6813
Validating Epoch 29: 100%|██████████| 5/5 [00:00<00:00, 21.83it/s]
Epoch 29: Val Loss: 0.6271, Dice: 0.6901
Training Epoch 30: 100%|██████████| 34/34 [00:02<00:00, 11.90it/s]
Epoch 30: Average Training Loss: 0.6821
Validating Epoch 30: 100%|██████████| 5/5 [00:00<00:00, 21.17it/s]
Epoch 30: Val Loss: 0.6254, Dice: 0.6841
Training Epoch 31: 100%|██████████| 34/34 [00:02<00:00, 12.00it/s]
Epoch 31: Average Training Loss: 0.6815
Validating Epoch 31: 100%|██████████| 5/5 [00:00<00:00, 20.23it/s]
Epoch 31: Val Loss: 0.6261, Dice: 0.7007
Training Epoch 32: 100%|██████████| 34/34 [00:02<00:00, 13.60it/s]
Epoch 32: Average Training Loss: 0.6809
Validating Epoch 32: 100%|██████████| 5/5 [00:00<00:00, 20.89it/s]
Epoch 32: Val Loss: 0.6245, Dice: 0.7066
Training Epoch 33: 100%|██████████| 34/34 [00:03<00:00, 10.02it/s]
Epoch 33: Average Training Loss: 0.6816
```

```
Validating Epoch 33: 100%|██████████| 5/5 [00:00<00:00, 14.40it/s]
Epoch 33: Val Loss: 0.6288, Dice: 0.6898
Training Epoch 34: 100%|██████████| 34/34 [00:02<00:00, 13.13it/s]
Epoch 34: Average Training Loss: 0.6804
Validating Epoch 34: 100%|██████████| 5/5 [00:00<00:00, 21.21it/s]
Epoch 34: Val Loss: 0.6249, Dice: 0.6964
Training Epoch 35: 100%|██████████| 34/34 [00:02<00:00, 14.01it/s]
Epoch 35: Average Training Loss: 0.6815
Validating Epoch 35: 100%|██████████| 5/5 [00:00<00:00, 21.31it/s]
Epoch 35: Val Loss: 0.6239, Dice: 0.7010
Training Epoch 36: 100%|██████████| 34/34 [00:02<00:00, 13.89it/s]
Epoch 36: Average Training Loss: 0.6800
Validating Epoch 36: 100%|██████████| 5/5 [00:00<00:00, 20.72it/s]
Epoch 36: Val Loss: 0.6241, Dice: 0.6991
Training Epoch 37: 100%|██████████| 34/34 [00:03<00:00,  9.72it/s]
Epoch 37: Average Training Loss: 0.6814
Validating Epoch 37: 100%|██████████| 5/5 [00:01<00:00,  2.65it/s]
Epoch 37: Val Loss: 0.6257, Dice: 0.6875
Training Epoch 38: 100%|██████████| 34/34 [00:05<00:00,  6.28it/s]
Epoch 38: Average Training Loss: 0.6800
Validating Epoch 38: 100%|██████████| 5/5 [00:00<00:00, 21.89it/s]
Epoch 38: Val Loss: 0.6247, Dice: 0.7039
Training Epoch 39: 100%|██████████| 34/34 [00:02<00:00, 12.54it/s]
Epoch 39: Average Training Loss: 0.6818
Validating Epoch 39: 100%|██████████| 5/5 [00:00<00:00,  8.67it/s]
Epoch 39: Val Loss: 0.6273, Dice: 0.6860
Training Epoch 40: 100%|██████████| 34/34 [00:03<00:00,  9.84it/s]
Epoch 40: Average Training Loss: 0.6804
```

```
Validating Epoch 40: 100%|██████████| 5/5 [00:00<00:00, 14.28it/s]

Epoch 40: Val Loss: 0.6244, Dice: 0.6977
```

(d) Report training loss, validation loss, and validation DICE curves. Comment on any overfitting or underfitting observed.

```
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(training_losses, label='Training Loss')
plt.plot(validation_losses, label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(average_dice_scores, label='Validation Dice Score')
plt.title('Validation Dice')
plt.xlabel('Epochs')
plt.ylabel('Score')
plt.legend()

plt.show()
```



(e) Report the average dice score over your test-set. You should be able to achieve a score of around 0.7 or better.

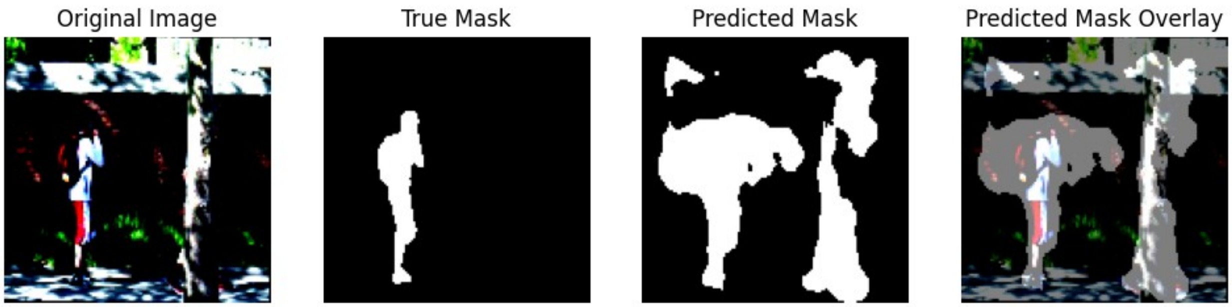
```
model.eval()
test_dice_scores = []

with torch.no_grad():
    for batch in testLoader:
        test_images, test_masks = batch[0].to(device), 
batch[1].to(device)
        predicted_masks = model(test_images)
        dice_val = dice_coefficient(predicted_masks, test_masks)
        test_dice_scores.append(dice_val.item())

mean_dice_score = sum(test_dice_scores) / len(test_dice_scores)
print(f"Average Dice Score on Test Set: {mean_dice_score:.4f}")

Average Dice Score on Test Set: 0.6363
```

(f) Show at least 3 example segmentations (i.e. show the RGB image, mask, and RGB image × mask for 3 samples) from your training data and 3 from your testing data. Comment on the generalization capabilities of your trained network.

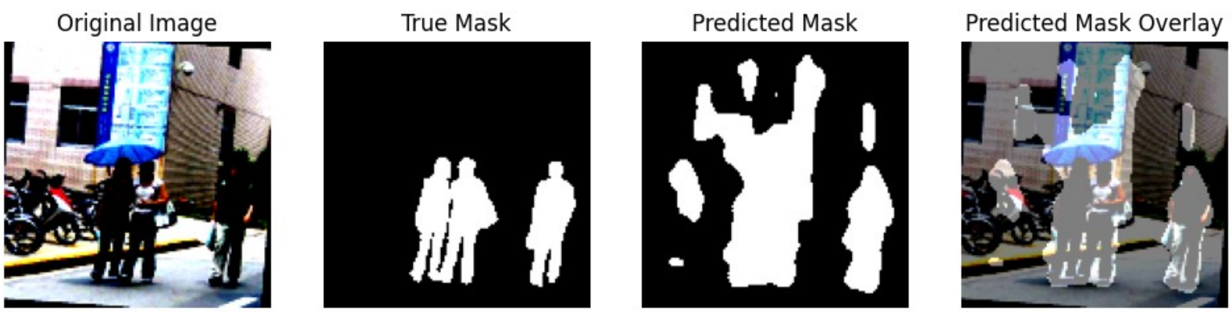
```
model.eval()
training_samples = []
testing_samples = []

with torch.no_grad():
    for i, (input_images, true_masks) in enumerate(trainLoader):
        if i >= 3:
            break
        input_images, true_masks = input_images.to(device), 
true_masks.to(device)
        outputs = model(input_images)
        model_predictions = (outputs > 0.5).float()
        training_samples.append((input_images.cpu(), true_masks.cpu(), 
model_predictions.cpu()))

    for i, (input_images, true_masks) in enumerate(testLoader):
        if i >= 3:
            break
        input_images, true_masks = input_images.to(device), 
true_masks.to(device)
        outputs = model(input_images)
        model_predictions = (outputs > 0.5).float()
        testing_samples.append((input_images.cpu(), true_masks.cpu(), 
model_predictions.cpu()))
```

```python
for i, (input_images, true_masks, model_predictions) in
enumerate(training_samples):
    fig, ax = plt.subplots(1, 4, figsize=(12, 4))
    image = input_images[0].permute(1, 2, 0)
    true_mask = true_masks[0][0]
    predicted_mask = model_predictions[0][0]
    ax[0].imshow(image)
    ax[0].set_title("Original Image")
    ax[0].axis('off')

    ax[1].imshow(true_mask, cmap='gray')
    ax[1].set_title("True Mask")
    ax[1].axis('off')

    ax[2].imshow(image)
    ax[2].imshow(predicted_mask, cmap='gray', alpha=1)
    ax[2].set_title("Predicted Mask")
    ax[2].axis('off')

    ax[3].imshow(image)
    ax[3].imshow(predicted_mask, cmap='gray', alpha=0.5)
    ax[3].set_title("Predicted Mask Overlay")
    ax[3].axis('off')
    plt.show()

for i, (input_images, true_masks, model_predictions) in
enumerate(testing_samples):
    fig, ax = plt.subplots(1, 4, figsize=(12, 4))
    image = input_images[0].permute(1, 2, 0)
    true_mask = true_masks[0][0]
    predicted_mask = model_predictions[0][0]
    ax[0].imshow(image)
    ax[0].set_title("Original Image")
    ax[0].axis('off')

    ax[1].imshow(true_mask, cmap='gray')
    ax[1].set_title("True Mask")
    ax[1].axis('off')

    ax[2].imshow(image)
    ax[2].imshow(predicted_mask, cmap='gray', alpha=1)
    ax[2].set_title("Predicted Mask")
    ax[2].axis('off')

    ax[3].imshow(image)
    ax[3].imshow(predicted_mask, cmap='gray', alpha=0.5)
    ax[3].set_title("Predicted Mask Overlay")
    ax[3].axis('off')
    plt.show()
```

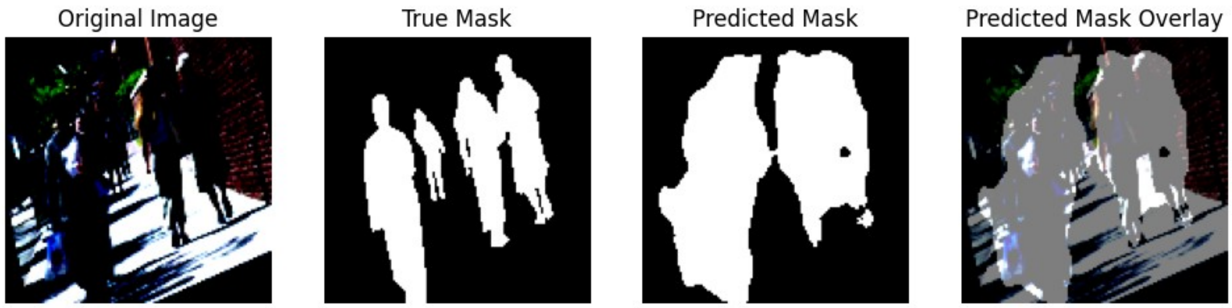
```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



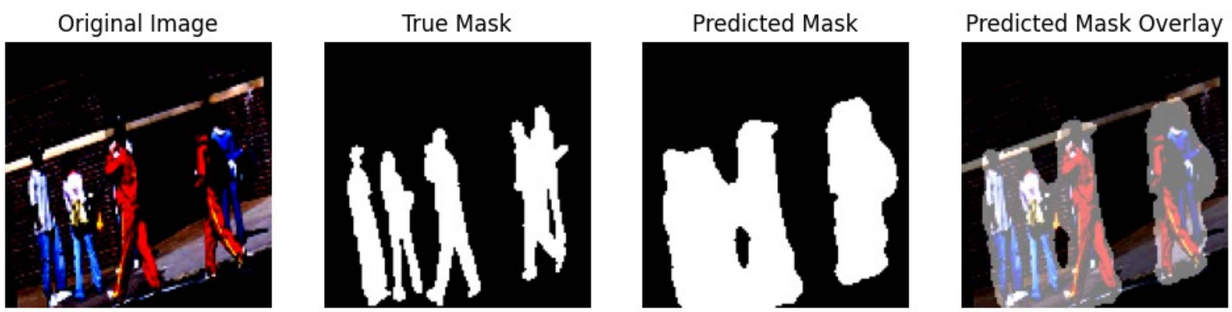

```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



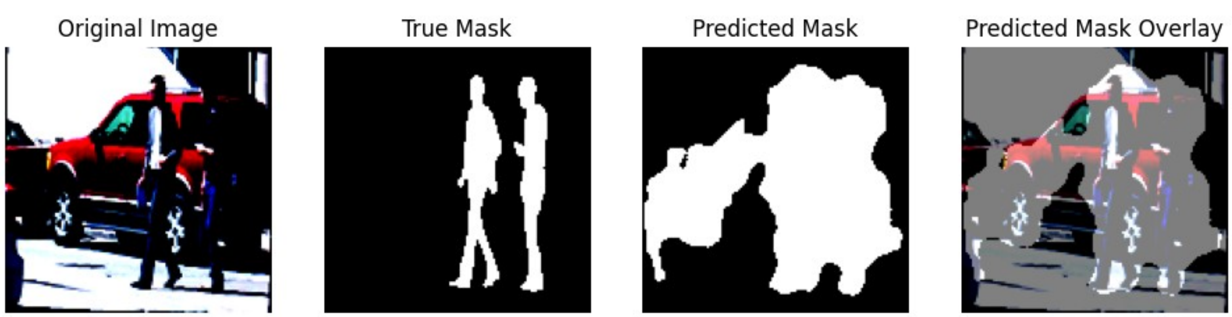

```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```




```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

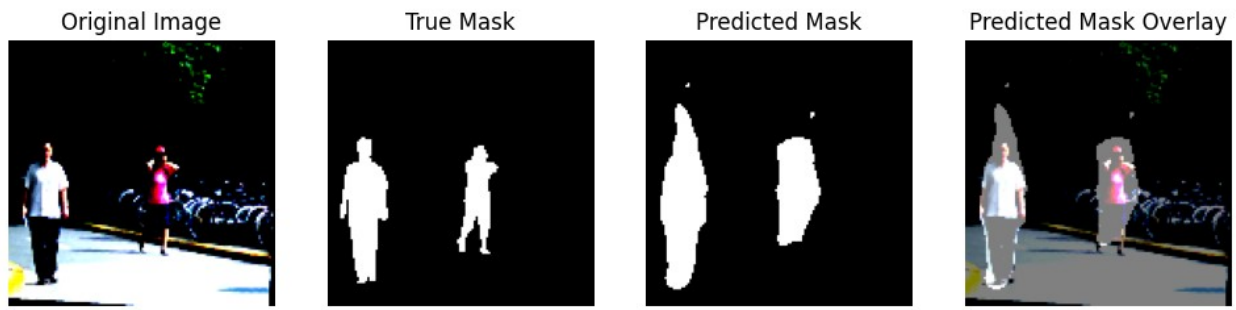



```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```




```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for
imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



(g) Show at least 1 example segmentation on an input image not from the FudanPed dataset. Again, comment on the generalization capabilities of your network with respect to this “out-of-distribution” image.

```
image_path = '/content/testimg.png'
image = Image.open(image_path).convert('RGB')
resize_image = transforms.Resize((128, 128), Image.BICUBIC)
image = resize_image(image)
image = TF.to_tensor(image)
image = image.unsqueeze(0)

model.eval()
with torch.no_grad():
    image = image.to(device)
    output = model(image)
    predicted_mask = (output > 0.5).float().cpu()[0][0]

image_cpu = image.cpu().squeeze(0)
predicted_mask_cpu = predicted_mask.cpu().squeeze(0)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
if image_cpu.dim() == 3:
    image_cpu = image_cpu.permute(1, 2, 0)
plt.imshow(image_cpu)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
if predicted_mask_cpu.dim() > 2:
    predicted_mask_cpu = predicted_mask_cpu.squeeze(0)
plt.imshow(predicted_mask_cpu, cmap='gray')
plt.title('Predicted Mask')
```

```
plt.axis('off')

plt.show()
```