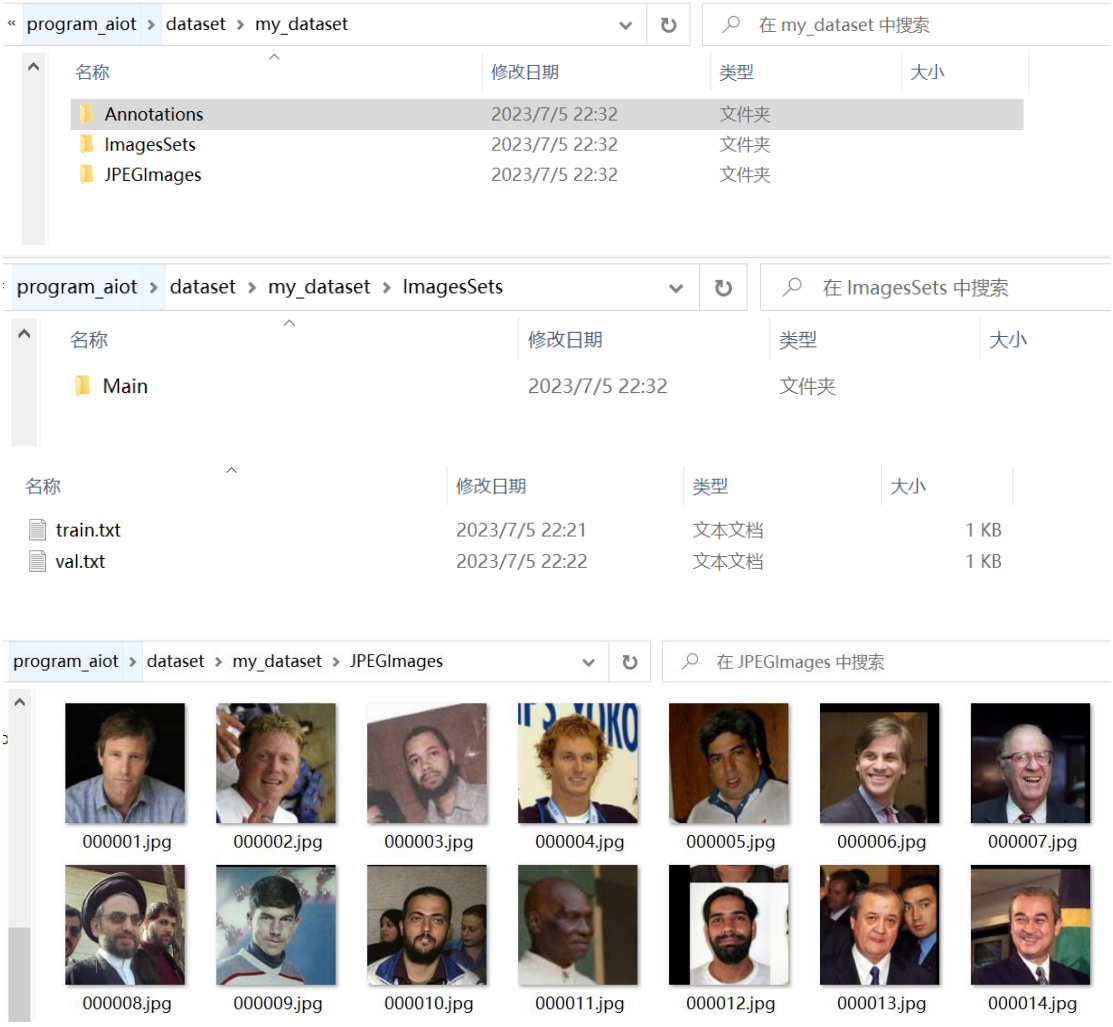
在自己构建的小数据集上训练。
首先，下载公开数据集 Labeled Faces in the Wild Home (LFW)，

官方下载链接：http://vis-www.cs.umass.edu/lfw/
然后创建自己的数据集文件夹，文件夹结构如下图



这些图片都是从上面的公开数据集中的图片中自己随机挑选的，也包括几张笔者自己拍摄的人脸图像。
通过 labelImg 对文件夹中的图像进行手动标注，生成 50 个.xml 格式的文件，存放在 Annotations 文件夹中，ImagesSets 文件夹中还有一个 Main 文件夹，其中存放训练集的图像序号和验证集的图像序号。

然后我们把自己的数据集放到 dataset 文件夹中，在 data_progress 文件夹中新建一个 data1.txt 文件，用来存放新的数据集的.xml 文件转换得到的.txt 文件。
然后按照下面的图片修改 configuration.py 里的代码。

```
PASCAL_VOC_CLASSES = {"person": 1}
'''PASCAL_VOC_CLASSES = {"person": 1, "bird": 2, "cat": 3, "cow": 4, "dog": 5,
                         "horse": 6, "sheep": 7, "aeroplane": 8, "bicycle": 9,
                         "boat": 10, "bus": 11, "car": 12, "motorbike": 13,
                         "train": 14, "bottle": 15, "chair": 16, "diningtable": 17,
                         "pottedplant": 18, "sofa": 19, "tvmonitor": 20, "face": 21}'''
```

```
TXT_DIR = "./data_process/data1.txt"
```

```
PASCAL_VOC_DIR = "./dataset/my_dataset/"
PASCAL_VOC_ANNOTATION = PASCAL_VOC_DIR + "Annotations"
PASCAL_VOC_IMAGE = PASCAL_VOC_DIR + "JPEGImages"
```

然后运行 write_voc_to_txt.ipynb，最后运行 train_from_scrach.ipynb 开始训练！
训练结果如下;

```
Cause: mangled names are not yet supported
To silence this warning, decorate the function with @tf.autograph.experimental.do_not_convert
Epoch: 1/10, step: 1/25.0, loss: 22.13652
Epoch: 1/10, step: 2/25.0, loss: 17.87612
Epoch: 1/10, step: 3/25.0, loss: 17.77104
Epoch: 1/10, step: 4/25.0, loss: 16.97855
Epoch: 1/10, step: 5/25.0, loss: 15.89847
Epoch: 1/10, step: 6/25.0, loss: 15.32655
Epoch: 1/10, step: 7/25.0, loss: 14.02430
Epoch: 1/10, step: 8/25.0, loss: 14.65469
Epoch: 1/10, step: 9/25.0, loss: 13.47237
Epoch: 1/10, step: 10/25.0, loss: 12.66751
Epoch: 1/10, step: 11/25.0, loss: 11.70530
Epoch: 1/10, step: 12/25.0, loss: 11.32792
Epoch: 1/10, step: 13/25.0, loss: 10.88313
Epoch: 1/10, step: 14/25.0, loss: 10.40961
Epoch: 1/10, step: 15/25.0, loss: 10.16978
Epoch: 1/10, step: 16/25.0, loss: 10.00631
Epoch: 1/10, step: 17/25.0, loss: 9.75460
Epoch: 1/10, step: 18/25.0, loss: 9.30929
```

```
Epoch: 1/10, step: 14/25.0, loss: 10.40961
Epoch: 1/10, step: 15/25.0, loss: 10.16978
Epoch: 1/10, step: 16/25.0, loss: 10.00631
Epoch: 1/10, step: 17/25.0, loss: 9.75460
Epoch: 1/10, step: 18/25.0, loss: 9.30929
Epoch: 1/10, step: 19/25.0, loss: 8.97344
Epoch: 1/10, step: 20/25.0, loss: 8.79533
Epoch: 1/10, step: 21/25.0, loss: 8.46958
Epoch: 1/10, step: 22/25.0, loss: 8.25396
Epoch: 1/10, step: 23/25.0, loss: 8.11276
Epoch: 1/10, step: 24/25.0, loss: 7.86405
Epoch: 1/10, step: 25/25.0, loss: 7.63584
Epoch: 2/10, step: 1/25.0, loss: 3.16929
Epoch: 2/10, step: 2/25.0, loss: 3.86165
Epoch: 2/10, step: 3/25.0, loss: 3.02337
Epoch: 2/10, step: 4/25.0, loss: 2.60890
Epoch: 2/10, step: 5/25.0, loss: 2.35519
Epoch: 2/10, step: 6/25.0, loss: 2.18449
Epoch: 2/10, step: 7/25.0, loss: 1.97902
Epoch: 2/10, step: 8/25.0, loss: 1.90611
```

```
Epoch: 2/10, step: 7/25.0, loss: 1.97902
Epoch: 2/10, step: 8/25.0, loss: 1.90611
Epoch: 2/10, step: 9/25.0, loss: 1.82738
Epoch: 2/10, step: 10/25.0, loss: 1.76581
Epoch: 2/10, step: 11/25.0, loss: 1.81190
Epoch: 2/10, step: 12/25.0, loss: 1.78537
Epoch: 2/10, step: 13/25.0, loss: 1.75409
Epoch: 2/10, step: 14/25.0, loss: 1.69916
Epoch: 2/10, step: 15/25.0, loss: 1.64391
Epoch: 2/10, step: 16/25.0, loss: 1.60727
Epoch: 2/10, step: 17/25.0, loss: 1.57189
Epoch: 2/10, step: 18/25.0, loss: 1.54338
Epoch: 2/10, step: 19/25.0, loss: 1.52414
Epoch: 2/10, step: 20/25.0, loss: 1.51145
Epoch: 2/10, step: 21/25.0, loss: 1.48612
Epoch: 2/10, step: 22/25.0, loss: 1.45667
Epoch: 2/10, step: 23/25.0, loss: 1.45394
Epoch: 2/10, step: 24/25.0, loss: 1.43705
Epoch: 2/10, step: 25/25.0, loss: 1.42224
Epoch: 3/10, step: 1/25.0, loss: 1.07708
Epoch: 9/10, step: 7/25.0, loss: 0.84845
Epoch: 9/10, step: 8/25.0, loss: 0.86551
Epoch: 9/10, step: 9/25.0, loss: 0.88401
Epoch: 9/10, step: 10/25.0, loss: 0.89467
Epoch: 9/10, step: 11/25.0, loss: 0.89928
Epoch: 9/10, step: 12/25.0, loss: 0.92071
Epoch: 9/10, step: 13/25.0, loss: 0.91403
Epoch: 9/10, step: 14/25.0, loss: 0.91161
Epoch: 9/10, step: 15/25.0, loss: 0.89645
Epoch: 9/10, step: 16/25.0, loss: 0.89991
Epoch: 9/10, step: 17/25.0, loss: 0.89482
Epoch: 9/10, step: 18/25.0, loss: 0.89303
Epoch: 9/10, step: 19/25.0, loss: 0.90167
Epoch: 9/10, step: 20/25.0, loss: 0.90146
Epoch: 9/10, step: 21/25.0, loss: 0.90074
Epoch: 9/10, step: 22/25.0, loss: 0.89688
Epoch: 9/10, step: 23/25.0, loss: 0.89649
Epoch: 9/10, step: 24/25.0, loss: 0.89649
Epoch: 9/10, step: 25/25.0, loss: 0.90267
```

显然，我们可以看到，因为数据集过小，所以在训练过程中，loss 的收敛速度很快，而且最后的结果 loss 过低，出现了非常严重的过拟合现象，并且在最后的测试集中识别效果不太理想，这是以后需要改进的!