数据准备：

1、由脚本dataset.py从annotations文件夹中读取xml文件，生成对应文件名的.txt文件放到label文件夹中，并将xml文件名对应的jpg文件也放到label文件中

2、由脚本label\_image\_to\_txt.py将label文件夹中的jpg文件读出放到pupil.txt中；

3、由脚本txt\_to\_txt.py将pupil.txt数据按比例分为两份，分别存放到label\_train.txt和label\_val.txt中

训练步骤：

1. 更改pupil.yaml中label\_train.txt和label\_val.txt文件存放路径；
2. 使用python3 train.py --data ./pupil.yaml --cfg ./models/yolov5s-pupil-256.yaml

--batch-size 64 --weights ./pre\_train/last.pt --hyp ./data/hyps/hyp.finetune.yaml指令开始训练

测试：

使用python3 detect.py --source test.mp4 --weights ./runs/train/exp15/weights/last.pt指令进行测试

Nvidia：131服务器构建docker

sudo docker run -it --name zwhu -p 9999:9999 -v /docker-images/:/mnt --runtime nvidia --network host f9e19935ef6a /bin/bash

2022.05.05在131开发板构建docker

sudo docker run -it --name trt\_test -p 9997:9997 -v /ssd/docker-images/zwhu:/mnt --runtime nvidia --network host 96818a623ae3 /bin/bash

查看文件数量

ls JPEGImages/ | wc -w

V5平台Opencv交叉编译

1. 将CrossCompilation.cmake拷贝至opencv源文件目录下
2. 更改源文件目录下CMakeLists.txt，在文件头加include(CrossCompilation.cmake)
3. 新建build文件夹，在build下执行如下指令：

CROSS\_COMPILE=$HOME/DevelopTools/v5/toolchain/gcc-linaro-5.3.1-2016.05-x86\_64\_arm-linux-gnueabi/bin/arm-linux-gnueabi- SYSROOT=$HOME/DevelopTools/v5/toolchain/sysroot-glibc-linaro-2.21-2016.05-arm-linux-gnueabi cmake -DBUILD\_opencv\_highgui=off -DWITH\_1394=off ..