
PFLD-pytorch

Implementation of PFLD A Practical Facial Landmark Detector by pytorch.

1. install requirements

```
1 pip3 install -r requirements.txt
```

2. Datasets

• WFLW Dataset Download

Wider Facial Landmarks in-the-wild (WFLW) is a new proposed face dataset. It contains 10000 faces (7500 for training and 2500 for testing) with 98 fully manual annotated landmarks.

1. WFLW Training and Testing images [Google Drive] [Baidu Drive]
2. WFLW Face Annotations
3. Unzip above two packages and put them on `./data/WFLW/`
4. move `Mirror98.txt` to `WFLW/WFLW_annotations`

```
1 $ cd data
2 $ python3 SetPreparation.py
```

3. training & testing training :

```
1 $ python3 train.py
```

use tensorboard, open a new terminal `~~~ $ tensorboard --logdir=./checkpoint/tensorboard/ ~~~` testing:

```
1 $ python3 test.py
```



4. results:

5. pytorch -> onnx -> ncnn Pytorch -> onnx

```
1 python3 pytorch2onnx.py
```

onnx -> ncnn

how to build :<https://github.com/Tencent/ncnn/wiki/how-to-build>

```
1 cd ncnn/build/tools/onnx
2 ./onnx2ncnn pfld-sim.onnx pfld-sim.param pfld-sim.bin
```

Now you can use **pfld-sim.param** and **pfld-sim.bin** in ncnn:

```
1 ncnn::Net pfld;
2 pfld.load_param("path/to/pfld-sim.param");
3 pfld.load_model("path/to/pfld-sim.bin");
4
5 cv::Mat img = cv::imread(imagepath, 1);
6 ncnn::Mat in = ncnn::Mat::from_pixels_resize(img.data, ncnn::Mat::
    PIXEL_BGR, img.cols, img.rows, 112, 112);
7 const float norm_vals[3] = {1/255.f, 1/255.f, 1/255.f};
8 in.substract_mean_normalize(0, norm_vals);
9
10 ncnn::Extractor ex = pfld.create_extractor();
11 ex.input("input_1", in);
12 ncnn::Mat out;
13 ex.extract("415", out);
```

6. reference: PFLD: A Practical Facial Landmark Detector <https://arxiv.org/pdf/1902.10859.pdf>

Tensorflow Implementation: <https://github.com/guoqiangqi/PFLD>