

:# Retinaface Python Examples

In the Python directory, a series of Python examples is provided as follows:

Index	Example File	Description
1	retinaface_opencv.py	Preprocessing using OpenCV, Inference using SAIL
2	retinaface_bmcv.py	Preprocessing using BMCV, Inference using SAIL

1. x86 PCIe Platform

1.1 Environment Setup

If you have installed a PCIe acceleration card on an x86 platform and are using it to test this example, you need to install libsophon ($\geq 0.3.0$), sophon-opencv ($\geq 0.2.4$), sophon-ffmpeg ($\geq 0.2.4$), and sophon-sail ($\geq 3.1.0$). Please refer to the development and runtime environment setup for the x86-pcie platform. Additionally, you may need to install other third-party libraries:

```
``bash
$ cd python
$ pip3 install -r requirements.txt
```

Note: If you encounter the error "ImportError: libGL.so.1: cannot open shared object file: No such file or directory," you need to execute the following command:

```
$ sudo apt-get install libgl1
```

1.2 Testing Command

The command parameters for retinaface_opencv.py and retinaface_bmcv.py are the same. Taking the inference of retinaface_opencv.py as an example, the parameter explanation is as follows:

```
usage: retinaface_opencv.py [--bmodel BMODEL] [--network NETWORK] [--input_path INPUT]
[--tpu_id TPU] [--conf CONF] [--nms NMS] [--use_np_file_as_input False]
```

--bmodel: Path to the bmodel used for inference. By default, it uses the network from stage 0 for inference.

--network: Backbone selection, choose between mobile0.25 or resnet50, default is mobile0.25.

--input_path: Path to the test image. You can input a single image or video, or a folder containing images.

--tpu_id: TPU device ID used for inference.

--conf: Confidence threshold, default is 0.02.

--nms: NMS threshold, default is 0.3.

--use_np_file_as_input: Whether to use other data as input. Default is False.

Here is an example of testing:

```
# Example: Testing on the WIDERVAL dataset
# Using 1 batch bmodel
$ python3 retinaface_opencv.py --bmodel
../data/models/BM1684X/retinaface_mobilenet0.25_fp32_1b.bmodel --network mobile0.25 --
input_path ../data/images/WIDERVAL --tpu_id 0 --conf 0.02 --nms 0.4 --use_np_file_as_input
False
```

After execution, the prediction results will be saved in the file `results/retinaface_mobilenet0.25_fp32_1b.bmodel_opencv_WIDERVAL_python_result.txt`, and the predicted images will be saved in the folder `results/retinaface_mobilenet0.25_fp32_1b.bmodel_opencv_WIDERVAL_python_result/`. Additionally, the prediction results, inference time, function runtime, and other information will be printed.

Example Output:

```
- face 1: x, y, w, h, conf = 130 133 67 97 0.99906695
- face 2: x, y, w, h, conf = 685 165 63 82 0.9955299
- face 3: x, y, w, h, conf = 365 154 60 80 0.9952643
- face 4: x, y, w, h, conf = 418 418 50 54 0.027065556
- face 5: x, y, w, h, conf = 996 717 16 21 0.020527808
- face 6: x, y, w, h, conf = 737 429 84 121 0.020261558
- ----- Inference Time Info -----
- inference_time(ms): 6.50
- total_time(ms): 68396.19, img_num: 3226
- average latency time(ms): 6.50, QPS: 153.963054
=====
+-----+
| Running Time Cost Summary |
+-----+-----+-----+-----+-----+
| Function | Run Count| Avg Time (s) | Max Time (s) | Min Time (s) |
+-----+-----+-----+-----+-----+
| preprocess_with_opencv | 3226 | 0.007066955982641045 | 0.012 | 0.007 |
| infer_numpy | 3226 | 0.007012089274643524 | 0.01 | 0.007 |
| postprocess_batch | 3226 | 0.006522628642281463 | 0.074 | 0.004 |
+-----+-----+-----+-----+-----+
```

You can perform batch testing with `batch_size=4` by changing the model.

For testing on face images and videos, use the following commands:

```
# Example: Testing with fp32 1 batch bmodel, different bmodels with the same parameters
$ python3 retinaface_opencv.py --bmodel
../data/models/BM1684X/retinaface_mobilenet0.25_fp32_1b.bmodel --network mobile0.25 --
input_path ../data/images/face --tpu_id 0 --conf 0.02 --nms 0.5 --use_np_file_as_input False
```

The predicted images will be saved in the folder `results/retinaface_mobilenet0.25_fp32_1b.bmodel_opencv_face_python_result/`, and the prediction results, inference time, function runtime, and other information will be printed.

```
# Example: Testing with fp32 1 batch bmodel, different bmodels with the same parameters
$ python3 retinaface_opencv.py --bmodel
../data/models/BM1684X/retinaface_mobilenet0.25_fp32_1b.bmodel --network mobile0.25 --
input_path ../data/videos/station.avi --tpu_id 0 --conf 0.02 --nms 0.5 --use_np_file_as_input False
```

The predicted images will be saved in the folder `results/`, and the prediction results, inference time, function runtime, and other information will be printed.

2. ARM SoC Platform

2.1 Environment Setup

If you are testing this example on an SoC platform, you need to cross-compile and install sophon-sail ($\geq 3.1.0$). Please refer to the cross-compilation and installation guide for sophon-sail. Additionally, you may need to install other third-party libraries:

```
```bash
$ cd python
$ pip3 install -r requirements.txt
```

### **2.2 Testing Command**

The testing method for the SoC platform is the same as the x86 PCIe platform. Please refer to section 1.2 Testing Command for details.