

Simulate YOLOv5 Segmentation Inference



TIP

This document aims to demonstrate how to use `rknn-toolkit2` on an x86 PC to simulate inference of the YOLOv5 object segmentation model without the need for a development board. For the required environment setup, please refer to [RKNN Installation](#).

Prepare the Model

In this example, we will use a pre-trained ONNX format model from the `rknn_model_zoo` as an example, convert the model, and perform simulated inference on the PC.

- If you are using conda, please activate the rknn conda environment first.

```
conda activate rknn
```

- Download the `yolov5s-seg.onnx` model

```
cd rknn_model_zoo/examples/yolov5_seg/model
# Download the pre-trained yolov5s-seg.onnx model
bash download_model.sh
```

If you encounter network issues, you can visit [this page](#) to download the corresponding model into the appropriate folder.

- Use `rknn-toolkit2` to convert it into `yolov5s-seg.rknn`

```
cd rknn_model_zoo/examples/yolov5_seg/python
python3 convert.py ../model/yolov5s-seg.onnx <TARGET_PLATFORM>
```

Parameter explanations:

`<onnx_model>`: Specify the path to the ONNX model

`<TARGET_PLATFORM>`: Specify the name of the NPU platform. Supported platforms can be found [here](#)

`<dtype>(optional)`: Specify `i8` for int8 quantization or `fp` for fp16 quantization. The default is `i8`.

`<output_rknn_path>(optional)`: Specify the save path for the RKNN model. By default, it is saved in the same directory as the ONNX model with the filename `yo1ov5s-seg.rknn`.

Run the YOLOv5 Segmentation Simulated Inference Python Demo

- Install the required dependencies via pip3

```
pip3 install torchvision==0.11.2 pycocotools
```

- Run the simulated inference program
 - Modify `rknn_model_zoo/py_utils/rknn_executor.py` to the following code, **and make sure to back up the original code**

```
from rknn.api import RKNN

class RKNN_model_container():
    def __init__(self, model_path, target=None, device_id=None) -> None:
        rknn = RKNN()
        print('--> Init runtime environment')
        if target==None:
            DATASET_PATH = '../ ../ ../datasets/COCO/coco_subset_20.txt'
            onnx_model = model_path[:-4] + 'onnx'
            print('--> Config model')
            rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255,
255]], target_platform='rk3588')
            print('done')
            # Load model
            print('--> Loading model')
            ret = rknn.load_onnx(model=onnx_model)
```

```

if ret != 0:
    print('Load model failed!')
    exit(ret)
print('done')
# Build model
print('--> Building model')
ret = rknn.build(do_quantization=True, dataset=DATASET_PATH)
if ret != 0:
    print('Build model failed!')
    exit(ret)
print('done')
ret = rknn.init_runtime()
else:
    ret = rknn.init_runtime(target=target, device_id=device_id)
if ret != 0:
    print('Init runtime environment failed')
    exit(ret)
print('done')

self.rknn = rknn

def run(self, inputs):
    if isinstance(inputs, list) or isinstance(inputs, tuple):
        pass
    else:
        inputs = [inputs]

    result = self.rknn.inference(inputs=inputs)
    return result

```

- Modify `rknn_model_zoo/examples/yolov5_seg/python/yolov5_seg.py` line 260 to set the default value of `target` to None

```

# parser.add_argument('--target', type=str, default='rk3566', help='target
RKNPU platform')
parser.add_argument('--target', type=str, default=None, help='target RKNPU
platform')

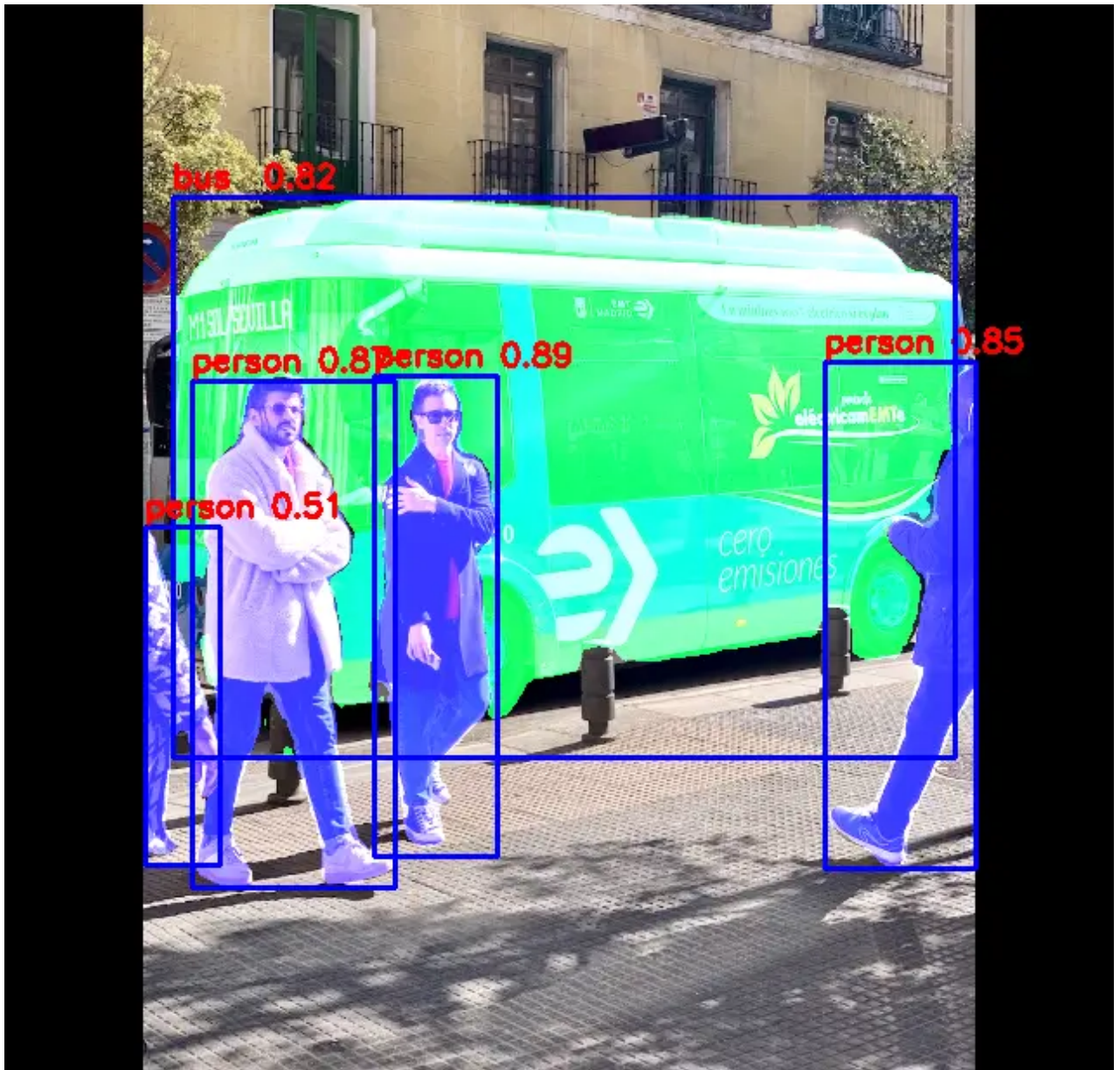
```

- Run the simulated inference program


```
python3 yolov5_seg.py --model_path ../model/yolov5s-seg.rknn --img_show
```

- Simulated inference results (the simulator only simulates NPU computation results; actual effects and accuracy are based on board inference)





 Report Issue

 Edit this page