

Deploy YOLOv5 Object Detection on the Board



TIP

This document aims to demonstrate how to run on-board inference of the YOLOv5 object detection model on Rockchip RK3588/3566 series chips. For the required environment setup, please refer to [RKNN Installation](#).

This example uses a pre-trained ONNX format model from the [rknn_model_zoo](#) as an example to convert the model for on-board inference, providing a complete demonstration.

Deploying YOLOv5 with RKNN requires two steps:

- On the PC, use `rknn-toolkit2` to convert models from different frameworks into RKNN format.
- On the board, use the Python API of `rknn-toolkit2-lite` for on-board model inference.

PC Model Conversion



TIP

Radxa provides a pre-converted `yo1ov5s_rk35XX.rknn` model, and users can directly refer to [YOLOv5 On-Board Inference](#) to skip the PC model conversion section.

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

```
conda activate rknn
```

- Download the `yo1ov5s_re1u.onnx` model

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

- ```

cd rknn_model_zoo/examples/yolov5/model
Download the pre-trained yolov5s_relu.onnx model

cd rknn_model_zoo/examples/yolov5/python
python3 convert.py ../model/yolov5s_relu.onnx <TARGET_PLATFORM> <dtype>
<output_rknn_path>

```

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 `yolov5.rknn`.

- Copy the `yolov5.rknn` model to the board.

## YOLOv5 On-Board Inference



### TIP

For RK3566/3568 chip users, NPU must be enabled in rsetup overlays before use. Please refer to [rsetup](#) for details.

- (Optional) Download the Radxa-provided YOLOv5s RKNN model.

| Platform | Download Link                       |
|----------|-------------------------------------|
| rk3566   | <a href="#">yolov5s_rk3566.rknn</a> |
| rk3568   | <a href="#">yolov5s_rk3568.rknn</a> |
| rk3588   | <a href="#">yolov5s_rk3588.rknn</a> |

- Modify the `rknn_model_zoo/py_utils/rknn_executor.py` code Please configure the

RKNN Model Zoo code repository according to [Install RKNN Model Zoo on the Board](#).

```
1 # from rknn.api import RKNN
2 try:
3 from rknn.api import RKNN
4 except:
5 from rknnlite.api import RKNNLite as RKNN
...
...
18 ret = rknn.init_runtime()
```

- Modify the `rknn_model_zoo/examples/yolov5/python/yolov5.py` code

```
262 outputs = model.run([np.expand_dims(input_data, 0)])
```

- Install the required environment

```
pip3 install opencv-python-headless
```

- Run the YOLOv5 example code

```
cd rknn_model_zoo/examples/yolov5/python
python3 yolov5.py --model_path <your model path> --img_save
```

If you are using a self-converted model, copy it from the PC to the board, and specify the model path with the `--model_path` parameter.

```
rock@radxa-zero3:~/rknn_model_zoo/examples/yolov5/python$ python3 yolov5.py
--model_path ./yolov5s_rk3566.rknn --target rk3566 --img_save
use anchors from './model/anchors_yolov5.txt', which is [[[10.0, 13.0],
[16.0, 30.0], [33.0, 23.0]], [[30.0, 61.0], [62.0, 45.0], [59.0, 119.0]],
[[116.0, 90.0], [156.0, 198.0], [373.0, 326.0]]]
--> Init runtime environment
I RKNN: [09:28:50.071] RKNN Runtime Information, librknrt version: 1.6.0
(9a7b5d24c@2023-12-13T17:31:11)
I RKNN: [09:28:50.071] RKNN Driver Information, version: 0.8.8
I RKNN: [09:28:50.073] RKNN Model Information, version: 6, toolkit version:
2.1.0+708089d1(compiler version: 2.1.0 (967d001cc8@2024-08-07T11:32:45)),
target: RKNPU lite, target platform: rk3566, framework name: ONNX, framework
layout: NCHW, model inference type: static_shape
```

```

W RKNN: [09:28:50.073] RKNN Model version: 2.1.0 not match with rknn runtime
version: 1.6.0
W RKNN: [09:28:50.141] query RKNN_QUERY_INPUT_DYNAMIC_RANGE error, rknn
model is static shape type, please export rknn with dynamic_shapes
W Query dynamic range failed. Ret code: RKNN_ERR_MODEL_INVALID. (If it is a
static shape RKNN model, please ignore the above warning message.)
done
Model-./yolov5s_rk3566.rknn is rknn model, starting val
infer 1/1

IMG: bus.jpg
person @ (209 243 286 510) 0.880
person @ (479 238 560 526) 0.871
person @ (109 238 231 534) 0.840
person @ (79 353 121 517) 0.301
bus @ (91 129 555 464) 0.692
Detection result save to ./result/bus.jpg

```

Parameter explanations:

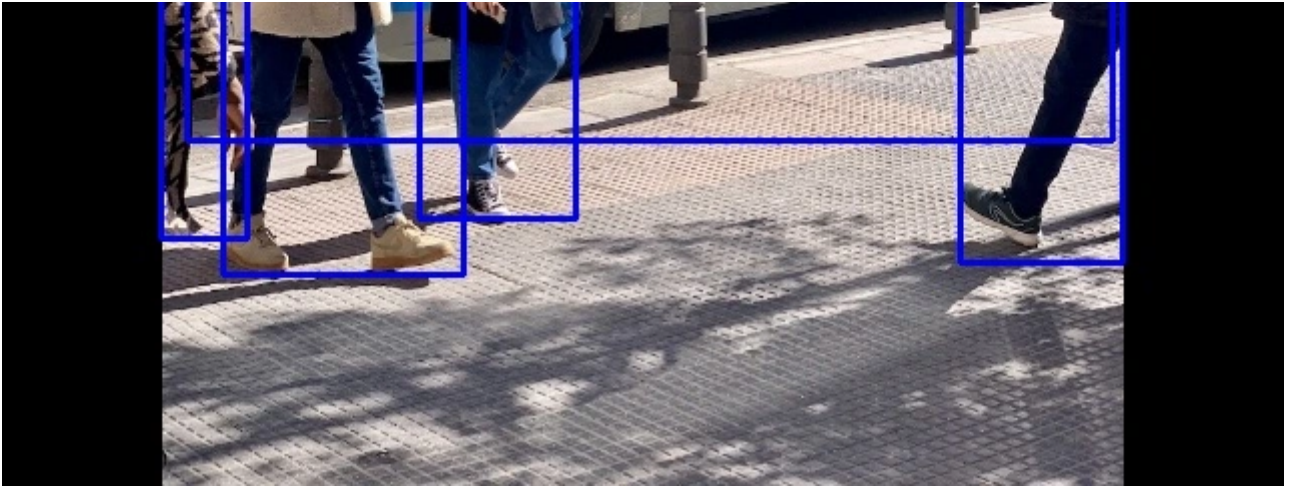
`--model_path`: Specify the path to the RKNN model

`--img_folder`: Specify the image library for inference, default is `../model`

`--img_save`: Whether to save the inference result image to `./result`, default is `False`

- All inference results are saved in `./result`.





 [Report Issue](#)

 [Edit this page](#)