

# picodet 部署文档

## picodet 版本信息

由于 GPU 环境搭建复杂，这里使用 aistudio 作为导出环境，这样每个人的配置都能一样。这个项目所使用的 picodet 为最新版本的 picodet（2022.03.20），这里我放在 aistudio 上做转换。

## 使用指南

### 下载可使用的 onnx 模型

运行 aistudio 项目的第一部分，在 PaddleDetection 目录下下载 picodet\_xs\_416\_coco\_lcnet\_sim.onnx 模型放到 weights/onnx 目录下

### 转换模型

运行以下代码，在 weights/rknn 目录下生成 rknn 模型

```
1 cd python
2 python picodet_infer_rknn_PC.py
```

纯文本

## 推理

在 images/after 目录下生成推理后的图片

### PC 端

```
1 cd python
2 python picodet_infer_rknn_PC.py
```

纯文本

### Board 端

```
1 cd python
2 python picodet_infer_rknn_board.py
```

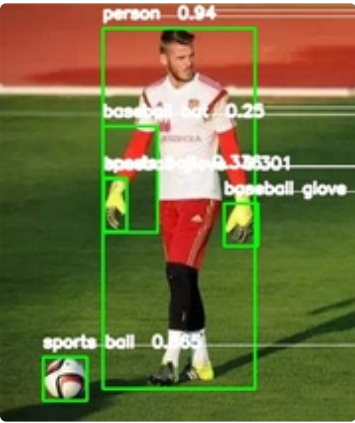
纯文本

## 结果展示

推理前



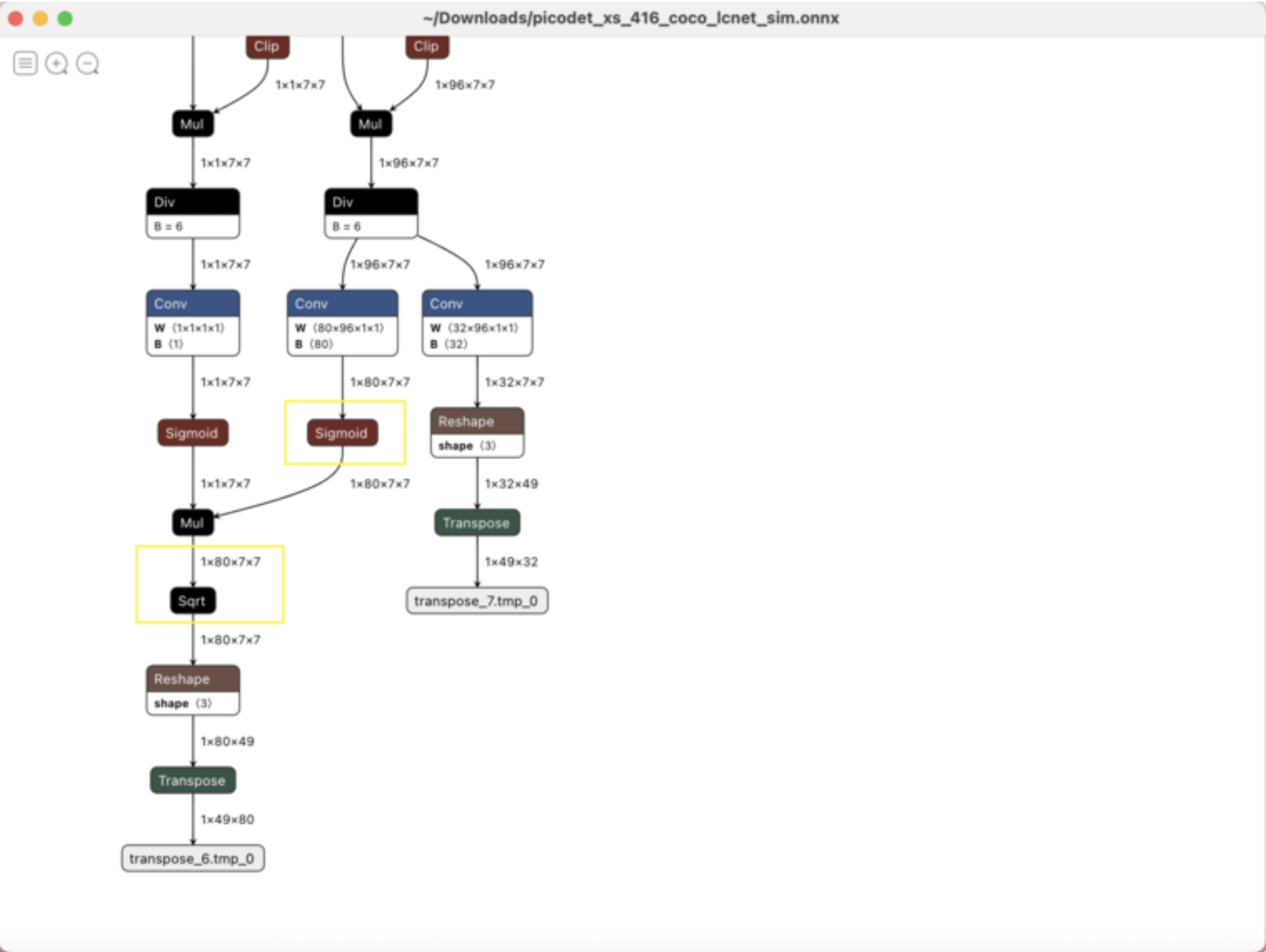
推理后



## 转换模型过程中踩的坑

### 查看模型

onnx 模型支持在线查看，可以使用 Netron 做可视化，这里可视化后直接给出坑的位置！



### 转换模型

- 3588 暂时不支持 sqrt 算子，因此模型最多只能运行到最下面黄色方框框出来的上面的 Mul 层，因此一开始我选择 Mul 代替 transpose 作为输出层
- 接下来发现，PC 端可以完美推理了，放到板子上跑不通。一定是某一个层出现了问题，这里教大家一个方法看哪个层有问题。rknn 支持指定输出层，只需要指定你想查看的输出层，板子和 PC 同时运行代码，print 查看结果即可。
- 通过以上方法，我找到黄色方框框出来的 sigmoid 层，板载和 PC 模拟推理差 10 倍的数值，因此跑不通。所以自定义了 sigmoid 函数，将黄色方框上从左往右平行的两个 Conv 层作为输出层，自己补齐了后处理的代码，最后跑通了代码