This example shows how to implement YOLO object detection with ggml using pretrained model.

- 这个例子展示如何使用预训练模型实现YOLO 目标检测
- 预训练模型使用ggml进行推理

YOLOv3-tiny

Download the model weights（下载模型权重）:

```
$ wget https://pjreddie.com/media/files/yolov3-tiny.weights
$ sha1sum yolov3-tiny.weights
40f3c11883bef62fd850213bc14266632ed4414f  yolov3-tiny.weights
```

Convert the weights to GGUF format（将模型权重转化为GGUF格式）:

```
$ ./convert-yolov3-tiny.py yolov3-tiny.weights
yolov3-tiny.weights converted to yolov3-tiny.gguf
```

Object detection（目标检测）:

```
$ wget https://raw.githubusercontent.com/pjreddie/darknet/master/data/dog.jpg
$ ./yolov3-tiny -m yolov3-tiny.gguf -i dog.jpg
Layer  0 output shape:  416 x 416 x   16 x   1
Layer  1 output shape:  208 x 208 x   16 x   1
Layer  2 output shape:  208 x 208 x   32 x   1
Layer  3 output shape:  104 x 104 x   32 x   1
Layer  4 output shape:  104 x 104 x   64 x   1
Layer  5 output shape:   52 x  52 x   64 x   1
Layer  6 output shape:   52 x  52 x  128 x   1
Layer  7 output shape:   26 x  26 x  128 x   1
Layer  8 output shape:   26 x  26 x  256 x   1
Layer  9 output shape:   13 x  13 x  256 x   1
Layer 10 output shape:   13 x  13 x  512 x   1
Layer 11 output shape:   13 x  13 x  512 x   1
Layer 12 output shape:   13 x  13 x 1024 x   1
Layer 13 output shape:   13 x  13 x  256 x   1
Layer 14 output shape:   13 x  13 x  512 x   1
Layer 15 output shape:   13 x  13 x  255 x   1
Layer 18 output shape:   13 x  13 x  128 x   1
Layer 19 output shape:   26 x  26 x  128 x   1
Layer 20 output shape:   26 x  26 x  384 x   1
Layer 21 output shape:   26 x  26 x  256 x   1
Layer 22 output shape:   26 x  26 x  255 x   1
dog: 57%
car: 52%
truck: 56%
```

```
car: 62%
bicycle: 59%
Detected objects saved in 'predictions.jpg' (time: 0.357000 sec.)
```