

1. **Compiling**

To compile the YOLOv3 code type
"make"
in the darknet directory.

To compile the .callpython.c code type
"g++ -o callpython callpython.c -lpython2.7 -lm -L/usr/lib/python2.7/config"
in the darknet directory.

To compile the ROS node code see
["http://wiki.ros.org/ROS/Tutorials"](http://wiki.ros.org/ROS/Tutorials)

2. **Running**

To run the YOLOv3 algorithm, use the command
"/darknet detector demo cfg/coco.data cfg/yolov3-tiny.cfg yolov3-tiny.weights /dev/videox -thresh y"
Replace x with the correct number for the USB cam, usually 0 or 1.
Replace y with a threshold value, ex 0.1 for 10%.

To run the ROS node see
["http://wiki.ros.org/ROS/Tutorials"](http://wiki.ros.org/ROS/Tutorials)

Start the ROS node before the YOLOv3 algorithm to avoid complications.

3. **File structure**

In \Darknet\src\image.c and the function draw_detections is where the modifications of the algorithm is done to only detect other trucks. This is also where the algorithm calls our .callpython code to create ROS node messages.

In \Darknet\callpython.c is the code that makes communication between ROS and YOLO possible.

\Darknet\ROS_signaler.py is the code called by .callpython.c, the code communicates with the ROS node code.

The ROS code is located in \src\yolo\scripts. trucktalker.py is the node for communication with the main system, trucklistener.py is for testing the ROS node.

4. **Other**

"auto_master_objdet.py" is the version of the main system code that includes the object detection subsystem. This code needs further testing to make sure everything works as it should. Please compare the code of this file to the normal main system code "auto_master.py".

See these websites for more information:
<https://pjreddie.com/darknet/yolo/> and <http://wiki.ros.org/ROS/Tutorials>