YOLO notes

Setting up CUDA

https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#ubuntu-installation

Donwload CUDA toolkit

Disable noveau:
Create a file at /etc/modprobe.d/blacklist-nouveau.conf with the following contents:

```
      blacklist nouveau
      options nouveau modeset=0Regenerate the kernel initramfs:

sudo update-initramfs -u

lsmod | grep nouveau
```
- //This should not print anything

Restart and temporarily disable 'Secure boot' in BIOS and add number '3' to system's kernel boot parameters and add "nomodeset"

Install the downloaded cuda runfile
`sudo sh cuda_<version>_linux.run --silent --override --no-opengl-libs`
Restart after done.

After Installation
`sudo nvidia-xconfig` //to write the xorg config
If the display is blank after rebooting, remove /etc/X11/xorg.conf and retry.

Remove all boot parameters. use "nvidia-drm.modeset=1" for bootparameters if screen turns blank.
Boot parameters in /etc/default/grub.
`sudo update-grub`

`glxinfo | grep OpenGL`
`inxi -G` //rendered should be Nvidia GPU

If renderer is different, use sudo nvidia-installer --uninstall to uninstall the driver
and install appropriate version with "Softwares and updates" application

`cat /proc/driver/nvidia/version` will print actual nvidia driver version
`nvidia-smi`
`nvcc --version`

Copy the following to .bashrc or run in the terminal to setup environment.

```
      export PATH=/usr/local/cuda-10.2/bin:/usr/local/cuda-10.2/NsightCompute-2019.1${PATH:+:${PATH}}
      export LD_LIBRARY_PATH=/usr/local/cuda-10.2/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}


sudo update-alternatives --config gcc #to check for gcc priorities
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 100
```

Test the installation

cd to ~/NVIDIA_CUDA-10.2_Samples and delete NVIDIA_CUDA-10.2_Samples/0_Simple/cudaNvSci

```
sudo make -k
./bin/x86_64/linux/release/deviceQuery

cd ~/NVIDIA_CUDA-10.2_Samples/5_Simulations/nbody
sudo make
./nbody
```

To remove:

```
sudo apt-get --purge remove "*cublas*" "cuda"
sudo apt-get --purge remove "nvidia*"
sudo apt remove --autoremove nvidia-*
```

Removes all traces of nvidia.

Testing Webcame

`rosrun usb_cam usb_cam_node ~video_device "/dev/video0"` to launch the webcam. Publishes /usb_cam/image_raw as topic. Leave vide_device parameter for laptop's webcam.
`rosrun image_view image_view image:=/image_raw`
`ls /dev/video*` to check for the usb_camera name

YOLO

Install OpenCV

Install Boost C++

```
sudo apt install libboost-dev
sudo apt install libboost-all-dev
dpkg -s libboost-dev | grep Version
```

Clone YOLO with ssh: https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh

```
cd selfDrive_ws/src
git clone git@github.com:leggedrobotics/darknet_ros.git
cd ../
sudo apt-get install python-catkin-tools
catkin build darknet_ros -DCMAKE_BUILD_TYPE=Release
catkin build darknet_ros --no-deps --verbose --catkin-make-args run_tests
```

On the Car

Main subscriber in *darknet_ros/config/ros.yaml or YoloObjectDetector.cpp* . Find cameraTopicName at imageSubscriber.

ls -la #to display permissions

Change image topic in .yaml, launch files.

launch yolov3.launch and view detection_image topic