

YOLOv4 on Ubuntu

Step 1 - Installing Pre-requisites

- **CMake >= 3.8**

Open the terminal and run the following command:

```
sudo apt install cmake
```

- **CUDA 10.0 Toolkit**

Go to the following link and download **CUDA Toolkit 10.0**. After downloading install the CUDA Toolkit in your Operating System.

<https://developer.nvidia.com/cuda-toolkit-archive>

- **OpenCV >= 2.4**

Open the terminal and run the following command:

```
sudo apt install libopencv-dev python3-opencv
```

- **CuDNN >= 7.0 for CUDA 10.0**

In order to download cuDNN on your operating system, you have to register for the NVIDIA Developer Program. It is simple and easy:

Go to the following link

<https://developer.nvidia.com/developer-program>

and signup using your email.

After signing up go to the following link to download cuDNN

<https://developer.nvidia.com/cudnn>

There, you will have to find the cuDNN version 7.0 (atleast) and click download and choose the installation method that meets



your environment needs preferred is **tar** file as it applies for all Linux platforms.

Keypoints:

- Your CUDA directory path is referred to as **/usr/local/cuda/**
- Your cuDNN download path is referred to as **<cuda_path>**

Step 2 - Installing cuDNN:

1. Navigate to your **<cuda_path>** directory containing the **cuDNN Tar file**.
2. Unzip the cuDNN package
3. Copy the following files into the CUDA Toolkit directory, and change the file permissions.

```
sudo cp cuda/include/cudnn.h /usr/local/cuda/include
```

```
sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
```

```
sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

For further help, you can visit the following link:

https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn_741/cudnn-install/index.html

- **OpenMP**

Run the following command in the terminal:

```
sudo apt install libomp-dev
```

- **Some other dependencies**

Open the terminal and run the following command:

```
sudo apt install make git g++
```

Step 3 - Downloading YOLO v4

Now create a new folder (any where you prefer) open the terminal in that folder and run the following command.

```
git clone https://github.com/AlexeyAB/darknet
```

This will download the Yolo v4 Github Repo in your System that will be used later in the building and multi-purpose usages. You can visit the Github repo at this link: <https://github.com/AlexeyAB/darknet>

Step 4 - Building YOLOv4

You can build YOLO v4 using either Make or CMake command.

Building using Make

Switch to darknet folder after cloning GitHub repo. Open the **Makefile** file in the darknet folder. Here you can see few of the variable at the start.

If you are building for **CPU** just the do the following.

Set **AVX=1** and **OPENMP=1** (if an error occurs then set **AVX=0**)

Set **LIBSO=1**

Set **ZED_CAMERA=1** (if you are working with ZED camera)

They will look like this:

```
GPU=0
CUDNN=0
CUDNN_HALF=0
OPENCV=1
AVX=1
OPEMMP=1
LIBSO=1
ZED_CAMERA=0
ZED_CAMERA_v2_8=0
```

However, if you building for **GPU** then do the following.

Set **GPU=1** and **CUDNN=1**

Set **CUDNN_HALF=1** (if you have GPU: Volta, Xavier, Turing or higher)

They will look like this:

```
GPU=1
CUDNN=1
CUDNN_HALF=1
OPENCV=1
AVX=0
OPENMP=0
LIBSO=1
ZED_CAMERA=0
ZED_CAMERA_v2_8=0
```

After doing these changes, just execute the following command in the darknet folder on the terminal:

make

If everything works fine, you will not get an error and process will complete smoothly. After build, you will be able to find **darknet** and **libdarknet.so** in the build path in the darknet folder.

Step 5 - Building Yolo v4 using CMake

To build YOLO using CMAKE, just open the terminal inside the **darknet** folder and execute the following commands in order as written.

```
mkdir build_release
cd build_release
cmake ..
make
```

If the execution runs succesfull it will create the following files inside the **build_release** folder.

darknet
libdark.so

After building, copy the **darknet** and **libdark.so** from the **build_release** folder to the **darknet** folder and rename the **libdark.so** to **libdarknet.so**

Step 6 - Testing Yolo v4

After building YOLO, lets download the pre-trained model of YOLO. Go to the following and download the weights file.

https://drive.google.com/file/d/1cewMfusmPjYWbrnuJRuKhPMwRe_b9PaT/view

After downloading the **yolov4.weights** copy them to the **darknet** folder. Now make sure that you have the following files in the **darknet** folder.

Darknet
Libdarknet.so
Yolov4.weights

Now open the terminal from the darknet folder and execute the following command.

```
./darknet detect cfg/yolov4.cfg yolov4.weights data/person.jpg
```

Hurray, we have our YOLO v4 up and running.

However, if you would like to run YOLO on a video, execute the following command:

```
./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights  
test50.mp4 -i 0 -thresh 0.25
```

Step 7 - Testing YOLO v4 using Webcam.

Run the Following command:

```
./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights -c 0
```