



Implementasi

# You Only Look Once

By:  
William Sean C14180031  
Abraham Imanuel C14180066



# 1. Framework

- Darknet
- <https://github.com/AlexeyAB/darknet.git>

The screenshot displays the GitHub interface for the repository **AlexeyAB / darknet**, which is a fork of **pjreddie / darknet**. The repository has 15.9k stars and 16.9k forks. A 'Clone' dropdown menu is open, showing options for cloning via HTTPS, GitHub CLI, or SVN, and options to open with GitHub Desktop or download a ZIP file. The file list on the left includes `.dirded`, `.github`, `3rdparty`, `build/darknet`, `cfg`, `cmake/Modules`, `data`, and `include`. The right sidebar shows the 'About' section with a description of YOLOv4 and links to related projects.

**Clone**

- HTTPS GitHub CLI
- `https://github.com/AlexeyAB/darknet.git`
- Use Git or checkout with SVN using the web URL.
- Open with GitHub Desktop
- Download ZIP

**About**

YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet)

[pjreddie.com/darknet/](#)

[deep-neural-networks](#) [computer-vision](#) [deep-learning](#) [neural-network](#) [dnn](#) [yolo](#) [object-detection](#) [deep-learning-tutorial](#) [yolov3](#) [yolov4](#) [scaledyolov4](#) [scaled-yolov4](#)

[Readme](#) [View license](#)

**Releases** 6

# 1. Framework

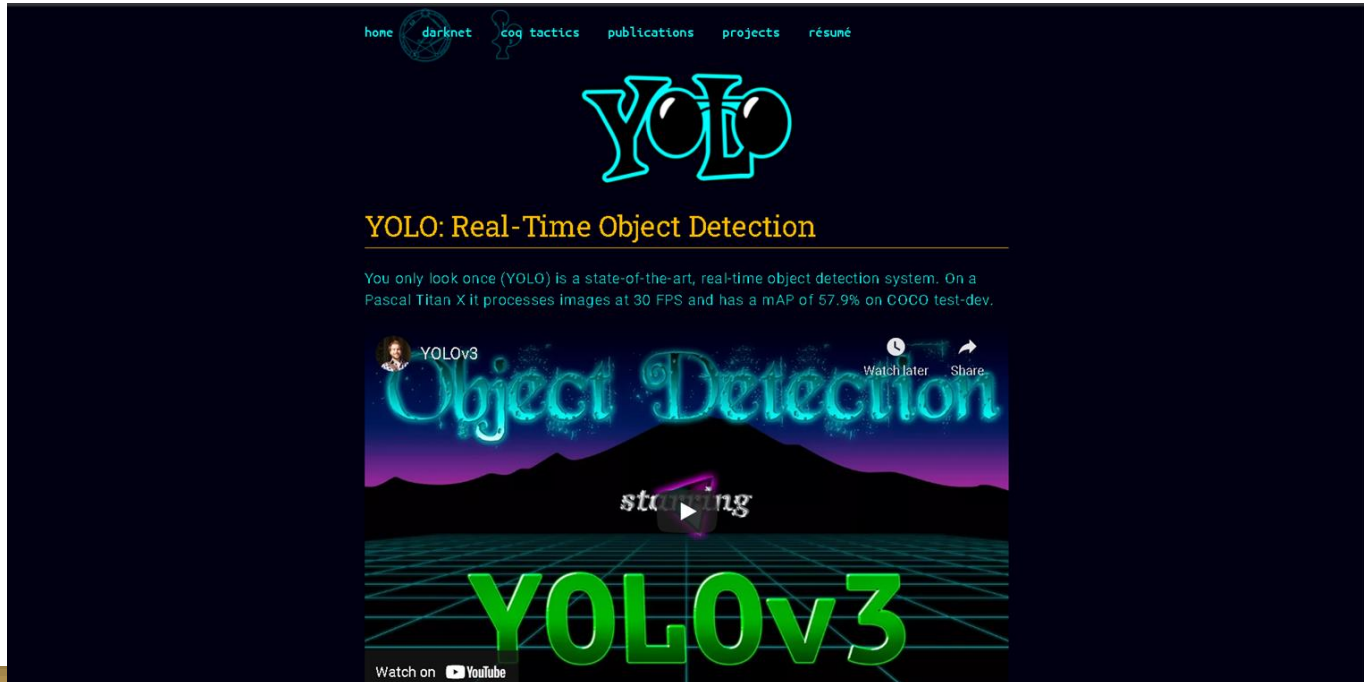
The screenshot shows the GitHub repository page for `pjreddie/darknet`. The repository is a Convolutional Neural Networks framework. It has 1.7k issues, 159 pull requests, 20.6k stars, and 16.9k forks. The repository is on the `master` branch with 3 branches and 0 tags. The file list includes:

File	Description	Last Commit
<code>cfg</code>	guys one of my beehives died :-( 🐝 🐝	3 years ago
<code>data</code>	SELU activation and yolo openimages	3 years ago
<code>examples</code>	GUYS I THINK MAYBE IT WAS BROKEN ON OPENCV IDX	3 years ago
<code>include</code>	GUYS I THINK MAYBE IT WAS BROKEN ON OPENCV IDX	3 years ago
<code>python</code>	faster nms and stuff	3 years ago
<code>scripts</code>	coco dataset script	4 years ago
<code>src</code>	GUYS I THINK MAYBE IT WAS BROKEN ON OPENCV IDX	3 years ago
<code>.gitignore</code>		4 years ago
<code>LICENSE</code>	Update LICENSE	5 years ago
<code>LICENSE</code>	YO DAWG, I HEARD YOU LIKE LICENSES	4 years ago
<code>LICENSE.gen</code>	generate own license, totally legal-verified	4 years ago
<code>LICENSE.gpl</code>	YO DAWG, I HEARD YOU LIKE LICENSES	4 years ago

The right sidebar shows the repository's description: "Convolutional Neural Networks". It also includes links to the README, license, releases, packages, and contributors.


# Source Code YOLO

- <http://pjreddie.com/yolo/> (Redmod, J. et al., 2016)



# Paper dari YOLO

- <https://arxiv.org/abs/1506.02640>

 Cornell University

We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv.org > cs > arXiv:1506.02640

Search... All fields Search

Help | Advanced Search

Computer Science > Computer Vision and Pattern Recognition

[Submitted on 8 Jun 2015 (v1), last revised 9 May 2016 (this version, v5)]

**You Only Look Once: Unified, Real-Time Object Detection**

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is far less likely to predict false detections where nothing exists. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and R-CNN, by a wide margin when generalizing from natural images to artwork on both the Picasso Dataset and the People-Art Dataset.

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

Cite as: [arXiv:1506.02640](#) [[cs.CV](#)]  
(or [arXiv:1506.02640v5](#) [[cs.CV](#)] for this version)

**Submission history**

From: Joseph Redmon [[view email](#)]

[v1] Mon, 8 Jun 2015 19:52:52 UTC (5,267 KB)

[v2] Tue, 9 Jun 2015 07:51:14 UTC (5,267 KB)

[v3] Thu, 11 Jun 2015 19:21:47 UTC (5,267 KB)

[v4] Thu, 12 Nov 2015 22:53:44 UTC (4,483 KB)

[v5] Mon, 9 May 2016 22:22:11 UTC (14,934 KB)

**Download:**

- PDF
- Other formats

([license](#))

Current browse context:

[cs.CV](#)

< prev | next >

[new](#) | [recent](#) | [1506](#)

Change to browse by:

[CS](#)

**References & Citations**

- NASA ADS
- Google Scholar
- Semantic Scholar

**21 blog links** ([external to this?](#))

**DBLP - CS Bibliography**


listing | bibtex

Joseph Redmon  
Santosh Kumar Divvala  
Ross B. Girshick  
Ali Farhadi

**Export BibTeX Citation**

**You Only Look Once:  
Unified, Real-Time O...**

- Joseph Redmon
- S. Divvala
- Ross B. Girshick
- A. Farhadi



# Requirements

- Windows or Linux
- CMake  $\geq 3.8$
- CUDA 10.0 or newer
- OpenCV (Pakai opencv-python boleh)
- cuDNN  $\geq 7.0$  for CUDA
- GPU (Untuk video)
- MSVC 2015/2017/2019

# Download YOLOV3 Weights

scale predictions, a better backbone classifier, and more. The full details are in our [paper](#)!

## Detection Using A Pre-Trained Model

This post will guide you through detecting objects with the YOLO system using a pre-trained model. If you don't already have Darknet installed, you should [do that first](#). Or instead of reading all that just run:

```
git clone https://github.com/pjreddie/darknet
cd darknet
make
```

Easy!

You already have the config file for YOLO in the `cfg/` subdirectory. You will have to download the pre-trained weight file [here \(237 MB\)](#). Or just run this:

```
wget https://pjreddie.com/media/files/yolov3.weights
```

Then run the detector!

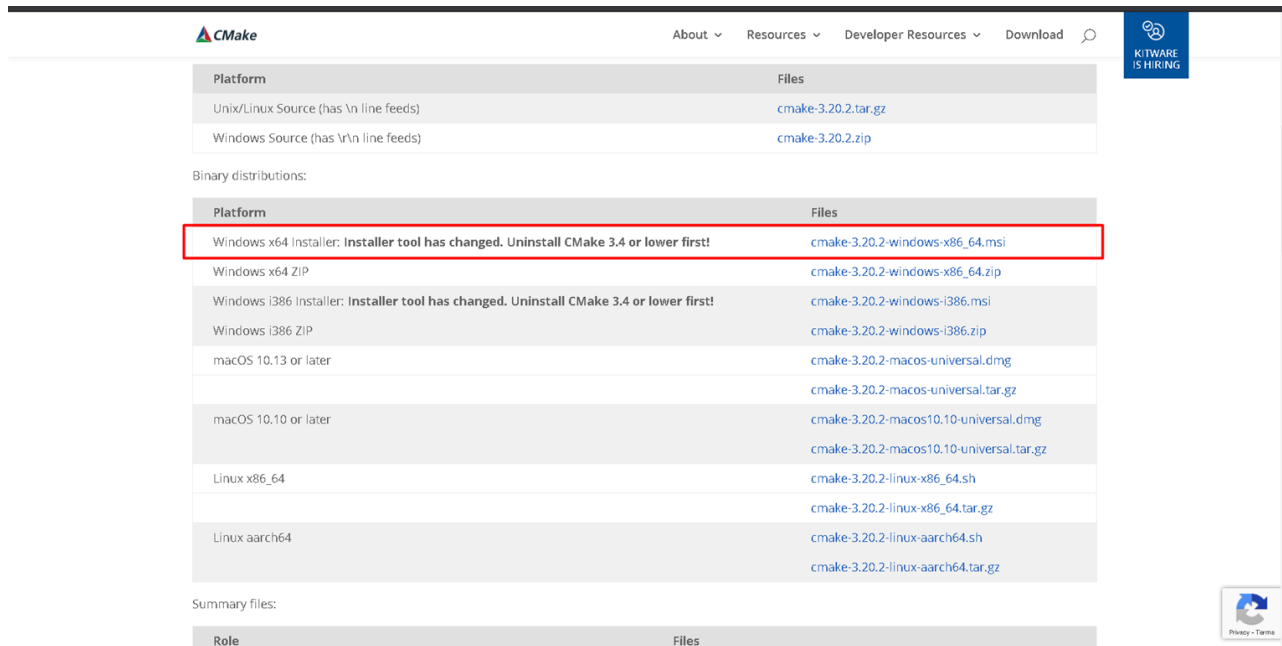
```
./darknet detect cfg/yolov3.cfg yolov3.weights data/dog.jpg
```

You will see some output like this:

layer	filters	size	input	output
0 conv	32	3 x 3 / 1	416 x 416 x 3	-> 416 x 416 x 32 0.299 BFLOPs

# Download CMake

- <https://cmake.org/download/>



The screenshot shows the CMake website's download page. At the top, there is a navigation bar with links for 'About', 'Resources', 'Developer Resources', and 'Download'. A 'KITWARE IS HIRING' badge is visible in the top right corner. Below the navigation bar, there are two tables. The first table, titled 'Platform' and 'Files', lists source code downloads for Unix/Linux and Windows. The second table, titled 'Binary distributions:', lists pre-built binaries for various platforms. The first row of this second table is highlighted with a red rectangle. Below the binary distributions table, there is a section for 'Summary files:' which is partially visible at the bottom of the page.

Platform	Files
Unix/Linux Source (has \n line feeds)	<a href="#">cmake-3.20.2.tar.gz</a>
Windows Source (has \r\n line feeds)	<a href="#">cmake-3.20.2.zip</a>

Binary distributions:

Platform	Files
Windows x64 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	<a href="#">cmake-3.20.2-windows-x86_64.msi</a>
Windows x64 ZIP	<a href="#">cmake-3.20.2-windows-x86_64.zip</a>
Windows i386 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	<a href="#">cmake-3.20.2-windows-i386.msi</a>
Windows i386 ZIP	<a href="#">cmake-3.20.2-windows-i386.zip</a>
macOS 10.13 or later	<a href="#">cmake-3.20.2-macos-universal.dmg</a>
	<a href="#">cmake-3.20.2-macos-universal.tar.gz</a>
macOS 10.10 or later	<a href="#">cmake-3.20.2-macos10.10-universal.dmg</a>
	<a href="#">cmake-3.20.2-macos10.10-universal.tar.gz</a>
Linux x86_64	<a href="#">cmake-3.20.2-linux-x86_64.sh</a>
	<a href="#">cmake-3.20.2-linux-x86_64.tar.gz</a>
Linux aarch64	<a href="#">cmake-3.20.2-linux-aarch64.sh</a>
	<a href="#">cmake-3.20.2-linux-aarch64.tar.gz</a>

Summary files:

Role	Files
------	-------



# CUDA & cuDNN

- <https://developer.nvidia.com/cuda-downloads> (untuk CUDA)
- <https://developer.nvidia.com/rdp/cudnn-download> (Untuk cuDNN)

# How to Check Your CUDA

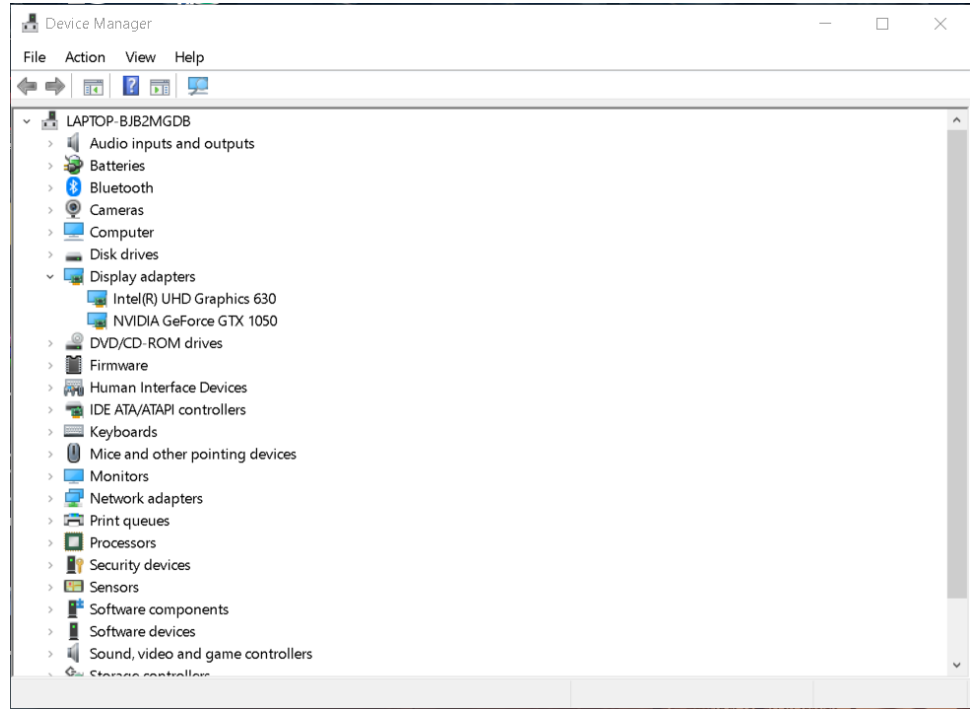
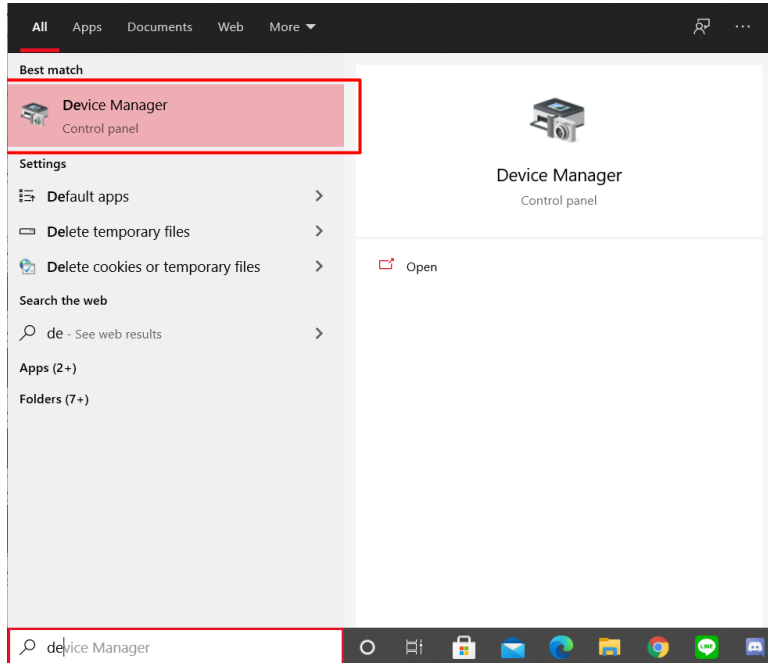
Compute capability (version)	Micro-architecture	GPUs	GeForce	Quadro, NVS	Tesla/Datacenter GPU	Tegra, Jetson, DRIVE
1.0	Tesla	G80	GeForce 8800 Ultra, GeForce 8800 GTX, GeForce 8800 GTS(G80)	Quadro FX 5600, Quadro FX 4600, Quadro Plex 2100 S4	Tesla C870, Tesla D870, Tesla S870	
1.1		G92, G94, G96, G98, G84, G86	GeForce GTS 250, GeForce 9800 GX2, GeForce 9800 GTX, GeForce 9800 GT, GeForce 8800 GTS(G92), GeForce 8800 GT, GeForce 9600 GT, GeForce 9500 GT, GeForce 9400 GT, GeForce 8600 GTS, GeForce 8600 GT, GeForce 8500 GT, GeForce G110M, GeForce 9300M GS, GeForce 9200M GS, GeForce 9100M G, GeForce 8400M GT, GeForce G105M	Quadro FX 4700 X2, Quadro FX 3700, Quadro FX 1800, Quadro FX 1700, Quadro FX 580, Quadro FX 570, Quadro FX 470, Quadro FX 380, Quadro FX 370, Quadro FX 370 Low Profile, Quadro NVS 450, Quadro NVS 420, Quadro NVS 290, Quadro NVS 295, Quadro Plex 2100 D4, Quadro FX 3800M, Quadro FX 3700M, Quadro FX 3600M, Quadro FX 2800M, Quadro FX 2700M, Quadro FX 1700M, Quadro FX 1600M, Quadro FX 770M, Quadro FX 570M, Quadro FX 370M, Quadro FX 360M, Quadro NVS 320M, Quadro NVS 160M, Quadro NVS 150M, Quadro NVS 140M, Quadro NVS 135M, Quadro NVS 130M, Quadro NVS 450, Quadro NVS 420, <sup>[40]</sup> Quadro NVS 295		
1.2		GT218, GT216, GT215	GeForce GT 340*, GeForce GT 330*, GeForce GT 320*, GeForce 315*, GeForce 310*, GeForce GT 240, GeForce GT 220, GeForce 210, GeForce GTS 360M, GeForce GTS 350M, GeForce GT 335M, GeForce GT 330M, GeForce GT 325M, GeForce GT 240M, GeForce G210M, GeForce 310M, GeForce 305M	Quadro FX 380 Low Profile, Quadro FX 1800M, Quadro FX 880M, Quadro FX 380M, Nvidia NVS 300, NVS 5100M, NVS 3100M, NVS 2100M, ION		
1.3		GT200, GT200b	GeForce GTX 285, GTX 285, GTX 280, GeForce GTX 275, GeForce GTX 260	Quadro FX 5800, Quadro FX 4800, Quadro FX 4800 for Mac, Quadro FX 3800, Quadro CX, Quadro Plex 2200 D2	Tesla C1060, Tesla S1070, Tesla M1060	
2.0	Fermi	GF100, GF110	GeForce GTX 590, GeForce GTX 580, GeForce GTX 570, GeForce GTX 480, GeForce GTX 470, GeForce GTX 465, GeForce GTX 480M	Quadro 6000, Quadro 5000, Quadro 4000, Quadro 4000 for Mac, Quadro Plex 7000, Quadro 5010M, Quadro 5000M	Tesla C2075, Tesla C2050/C2070, Tesla M2050/M2070/M2075/M2090	
2.1		GF104, GF106, GF108, GF114, GF116, GF117, GF119	GeForce GTX 580 Ti, GeForce GTX 560 Ti, GeForce GTX 480, GeForce GTS 450, GeForce GTS 450*, GeForce GT 840 (GDDR3), GeForce GT 830, GeForce GT 620, GeForce GT 610, GeForce GT 520, GeForce GT 440, GeForce GT 440*, GeForce GT 430, GeForce GT 430*, GeForce GT 420*, GeForce GTX 675M, GeForce GTX 670M, GeForce GT 635M, GeForce GT 630M, GeForce GT 625M, GeForce GT 720M, GeForce GT 620M, GeForce 710M, GeForce 610M, GeForce 820M, GeForce GTX 580M, GeForce GTX 570M, GeForce GTX 560M, GeForce GT 555M, GeForce GT 550M, GeForce GT 540M, GeForce GT 525M, GeForce GT 520MX, GeForce GT 520M, GeForce GTX 485M, GeForce GTX 470M, GeForce GTX 460M, GeForce GT 445M, GeForce GT 435M, GeForce GT 420M, GeForce GT 415M, GeForce 710M, GeForce 410M	Quadro 2000, Quadro 2000D, Quadro 600, Quadro 4000M, Quadro 3000M, Quadro 2000M, Quadro 1000M, NVS 310, NVS 315, NVS 5400M, NVS 5200M, NVS 4200M		

# How to Check Your CUDA

					DRIVE PX
6.0	Pascal	GP100		Quadro GP100	Tesla P100
6.1		GP102, GP104, GP106, GP107, GP108	Nvidia TITAN Xp, Titan X, GeForce GTX 1080 Ti, GTX 1080, GTX 1070 Ti, GTX 1070, GTX 1060, GTX 1050 Ti, GTX 1050, GT 1030, GT 1010, MX350, MX330, MX250, MX230, MX150, MX130, MX110	Quadro P6000, Quadro P5000, Quadro P4000, Quadro P2200, Quadro P2000, Quadro P1000, Quadro P400, Quadro P500, Quadro P520, Quadro P600, Quadro P5000(Mobile), Quadro P4000(Mobile), Quadro P3000(Mobile)	Tesla P40, Tesla P6, Tesla P4
6.2		GP10B <sup>[41]</sup>			Tegra X2, Jetson TX2, DRIVE PX 2

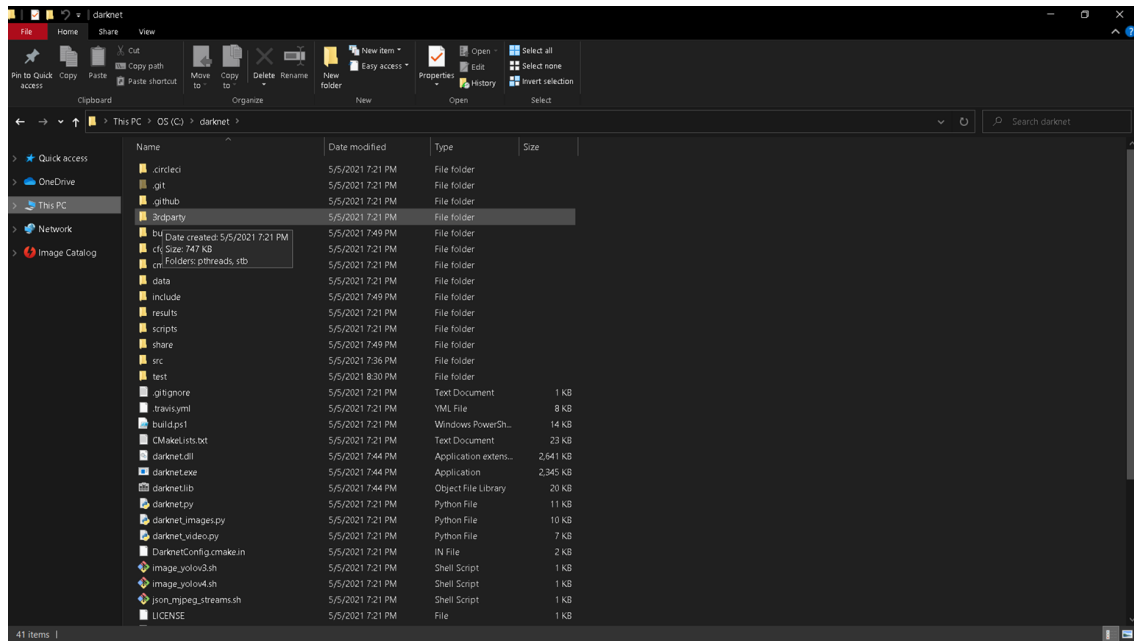
- CUDA SDK 8.0 support for compute capability 2.0 – 6.x (Fermi, Kepler, Maxwell, Pascal). Last version with support for compute capability 2.x (Fermi) (Pascal GTX 1070Ti Not Supported).
- CUDA SDK 9.0 – 9.2 support for compute capability 3.0 – 7.2 (Kepler, Maxwell, Pascal, Volta) (Pascal GTX 1070Ti Not Supported. CUDA SDK 9.0 and support CUDA SDK 9.2).
- CUDA SDK 10.0 – 10.2 support for compute capability 3.0 – 7.5 (Kepler, Maxwell, Pascal, Volta, Turing). Last version with support for compute capability 3.x (Kepler). 10.2 is the last official release for macOS, as support will not be available for macOS in newer releases.
- CUDA SDK 11.0 support for compute capability 3.5 – 8.0 (Kepler (in part), Maxwell, Pascal, Volta, Turing, Ampere (in part)).<sup>[37]</sup> New data types: **Bfloat16** and TF32 on third-generations Tensor Cores.<sup>[38]</sup>
- CUDA SDK 11.1 – 11.3 support for compute capability 3.5 – 8.6 (Kepler (in part), Maxwell, Pascal, Volta, Turing, Ampere).<sup>[39]</sup>

# How to Check GPU

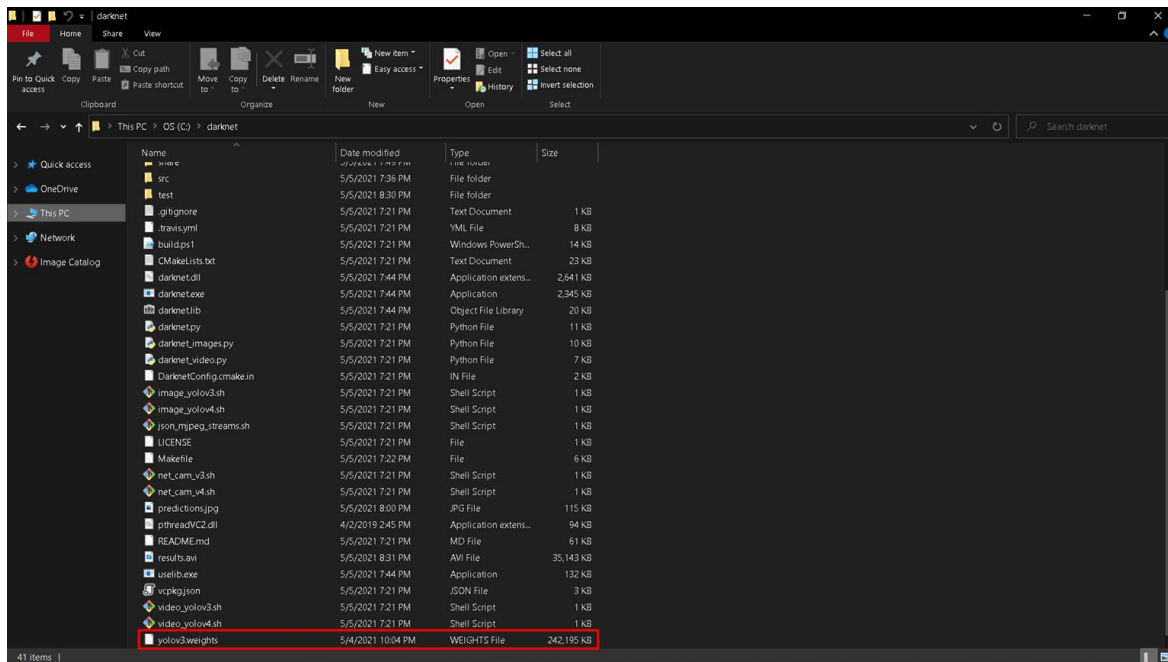


# Step 1 - Clone the Framework Repos

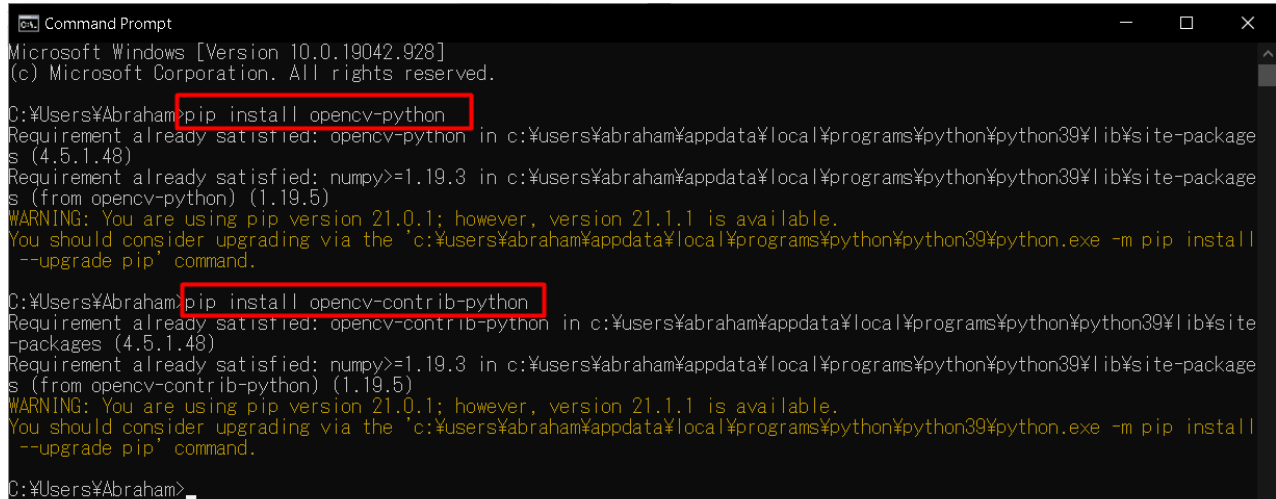
- git clone <https://github.com/AlexeyAB/darknet.git>



# Step 2 - Put Weights File in Repos



# Step 3 - Install OpenCV



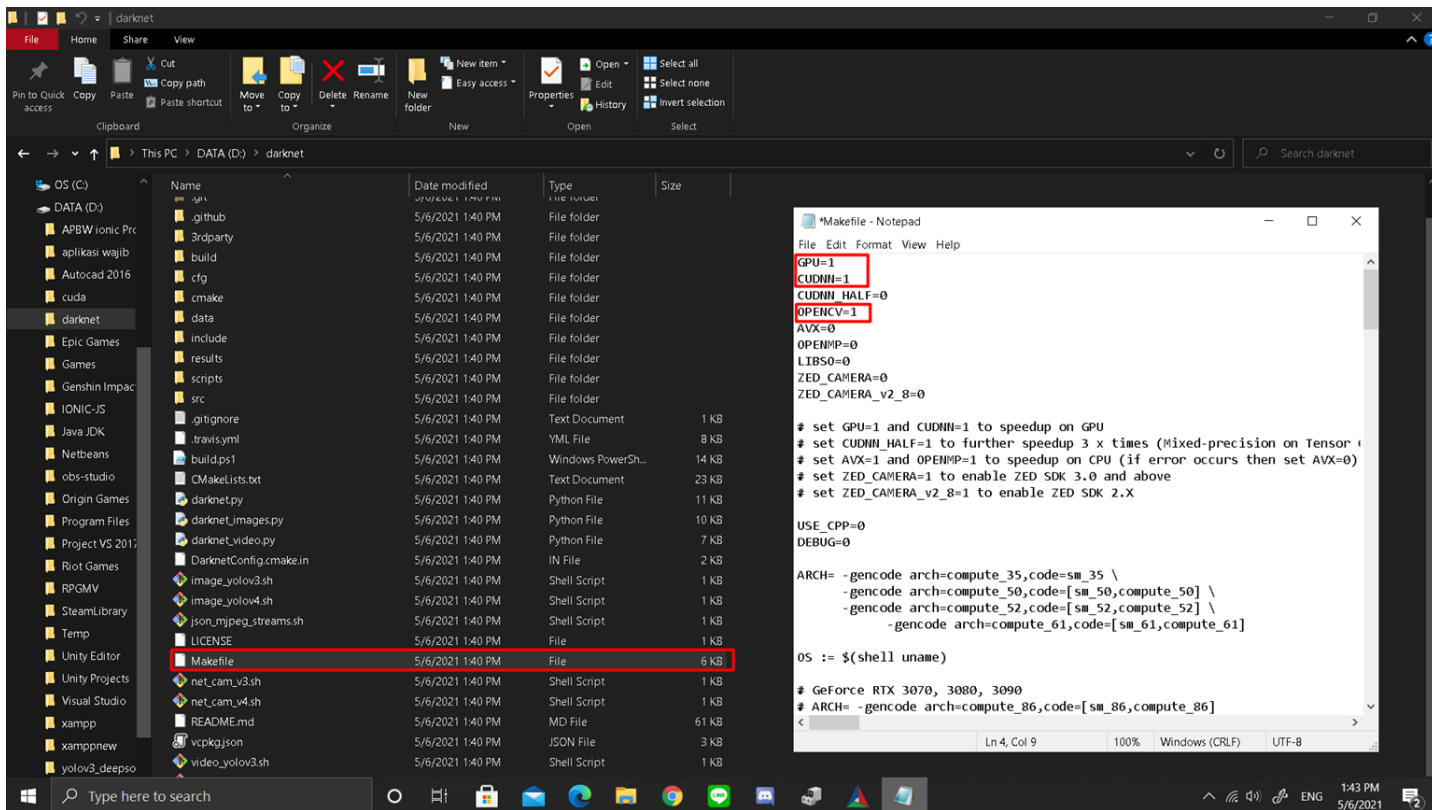
```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abraham>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\abraham\appdata\local\programs\python\python39\lib\site-packages (4.5.1.48)
Requirement already satisfied: numpy>=1.19.3 in c:\users\abraham\appdata\local\programs\python\python39\lib\site-packages (from opencv-python) (1.19.5)
WARNING: You are using pip version 21.0.1; however, version 21.1.1 is available.
You should consider upgrading via the 'c:\users\abraham\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\Abraham>pip install opencv-contrib-python
Requirement already satisfied: opencv-contrib-python in c:\users\abraham\appdata\local\programs\python\python39\lib\site-packages (4.5.1.48)
Requirement already satisfied: numpy>=1.19.3 in c:\users\abraham\appdata\local\programs\python\python39\lib\site-packages (from opencv-contrib-python) (1.19.5)
WARNING: You are using pip version 21.0.1; however, version 21.1.1 is available.
You should consider upgrading via the 'c:\users\abraham\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\Abraham>_
```

# Step 4 - Change "Makefile" file



The screenshot shows a Windows File Explorer window titled 'darknet' with the address bar displaying 'This PC > DATA (D:) > darknet'. The left sidebar shows the 'darknet' folder selected. The main pane displays a list of files and folders. The 'Makefile' file is highlighted with a red box. An inset window titled '\*Makefile - Notepad' shows the contents of the file, with 'GPU=1', 'CUDNN=1', and 'OPENCL=1' highlighted with red boxes.

Name	Date modified	Type	Size
.git	5/6/2021 1:40 PM	File folder	
.github	5/6/2021 1:40 PM	File folder	
3rdparty	5/6/2021 1:40 PM	File folder	
build	5/6/2021 1:40 PM	File folder	
cfg	5/6/2021 1:40 PM	File folder	
cmake	5/6/2021 1:40 PM	File folder	
data	5/6/2021 1:40 PM	File folder	
include	5/6/2021 1:40 PM	File folder	
results	5/6/2021 1:40 PM	File folder	
scripts	5/6/2021 1:40 PM	File folder	
src	5/6/2021 1:40 PM	File folder	
.gitignore	5/6/2021 1:40 PM	Text Document	1 KB
.travis.yml	5/6/2021 1:40 PM	YML File	8 KB
build.ps1	5/6/2021 1:40 PM	Windows PowerShell...	14 KB
CMakeLists.txt	5/6/2021 1:40 PM	Text Document	23 KB
darknet.py	5/6/2021 1:40 PM	Python File	11 KB
darknet_images.py	5/6/2021 1:40 PM	Python File	10 KB
darknet_video.py	5/6/2021 1:40 PM	Python File	7 KB
DarknetConfig.cmake.in	5/6/2021 1:40 PM	IN File	2 KB
image_yolov3.sh	5/6/2021 1:40 PM	Shell Script	1 KB
image_yolov4.sh	5/6/2021 1:40 PM	Shell Script	1 KB
json_mjpeg_streams.sh	5/6/2021 1:40 PM	Shell Script	1 KB
LICENSE	5/6/2021 1:40 PM	File	1 KB
Makefile	5/6/2021 1:40 PM	File	6 KB
net_cam_v3.sh	5/6/2021 1:40 PM	Shell Script	1 KB
net_cam_v4.sh	5/6/2021 1:40 PM	Shell Script	1 KB
README.md	5/6/2021 1:40 PM	MD File	61 KB
vcpkg.json	5/6/2021 1:40 PM	JSON File	3 KB
video_yolov3.sh	5/6/2021 1:40 PM	Shell Script	1 KB

```
*Makefile - Notepad
File Edit Format View Help

GPU=1
CUDNN=1
CUDNN_HALF=0
OPENCL=1
AVX=0
OPENMP=0
LTO=0
ZED_CAMERA=0
ZED_CAMERA_v2_8=0

# set GPU=1 and CUDNN=1 to speedup on GPU
# set CUDNN_HALF=1 to further speedup 3 x times (Mixed-precision on TensorRT)
# set AVX=1 and OPENMP=1 to speedup on CPU (if error occurs then set AVX=0)
# set ZED_CAMERA=1 to enable ZED SDK 3.0 and above
# set ZED_CAMERA_v2_8=1 to enable ZED SDK 2.X

USE_CPP=0
DEBUG=0

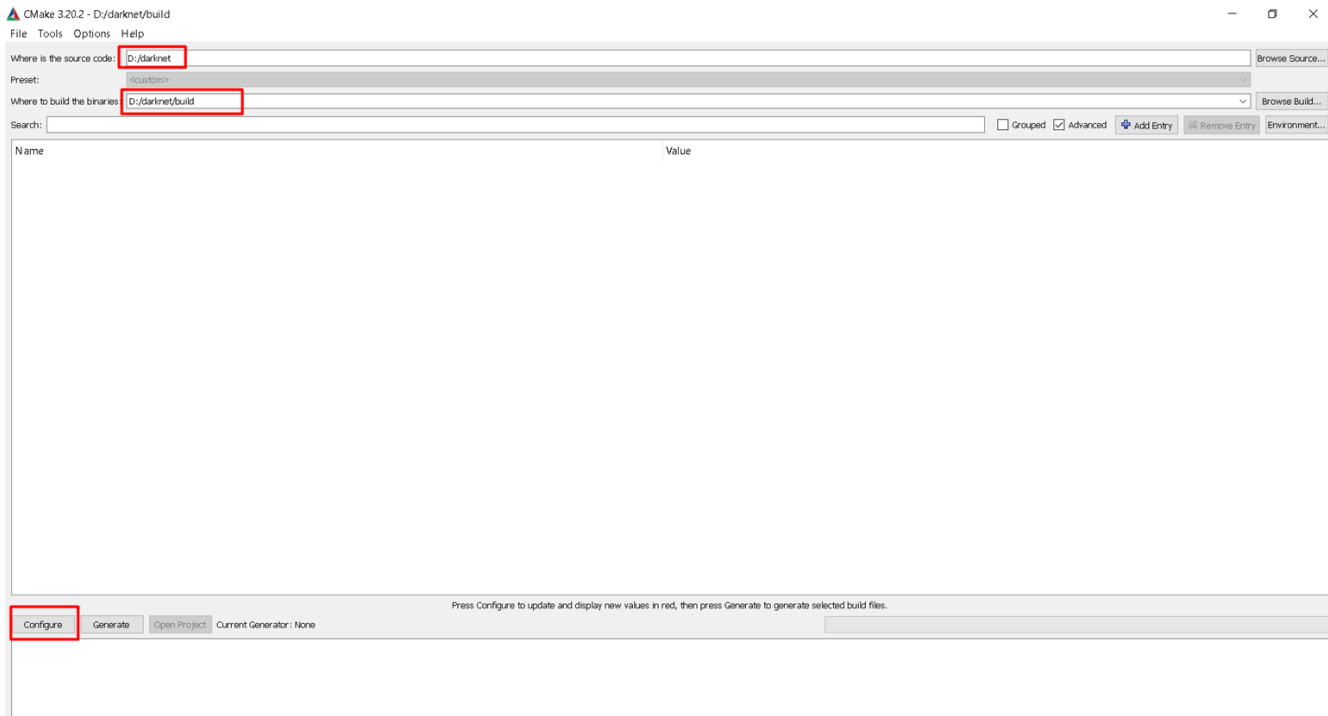
ARCH= -gencode arch=compute_35,code=sm_35 \
      -gencode arch=compute_50,code=[sm_50,compute_50] \
      -gencode arch=compute_52,code=[sm_52,compute_52] \
      -gencode arch=compute_61,code=[sm_61,compute_61]

OS := $(shell uname)

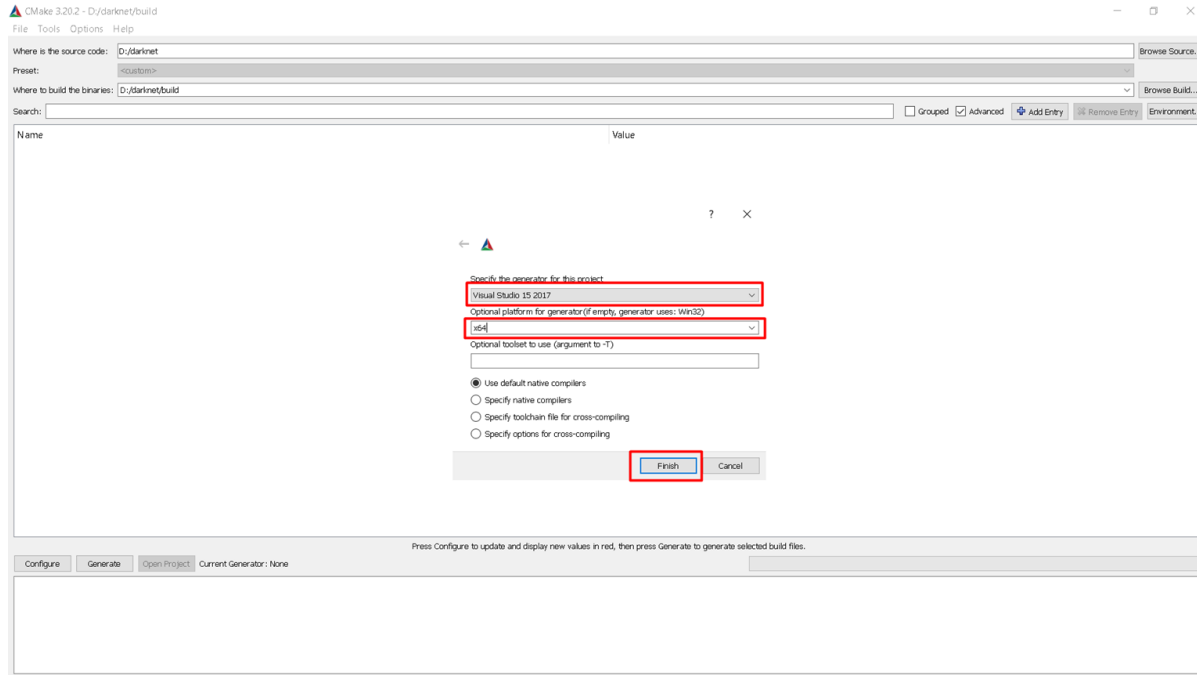
# GeForce RTX 3070, 3080, 3090
# ARCH= -gencode arch=compute_86,code=[sm_86,compute_86]
```



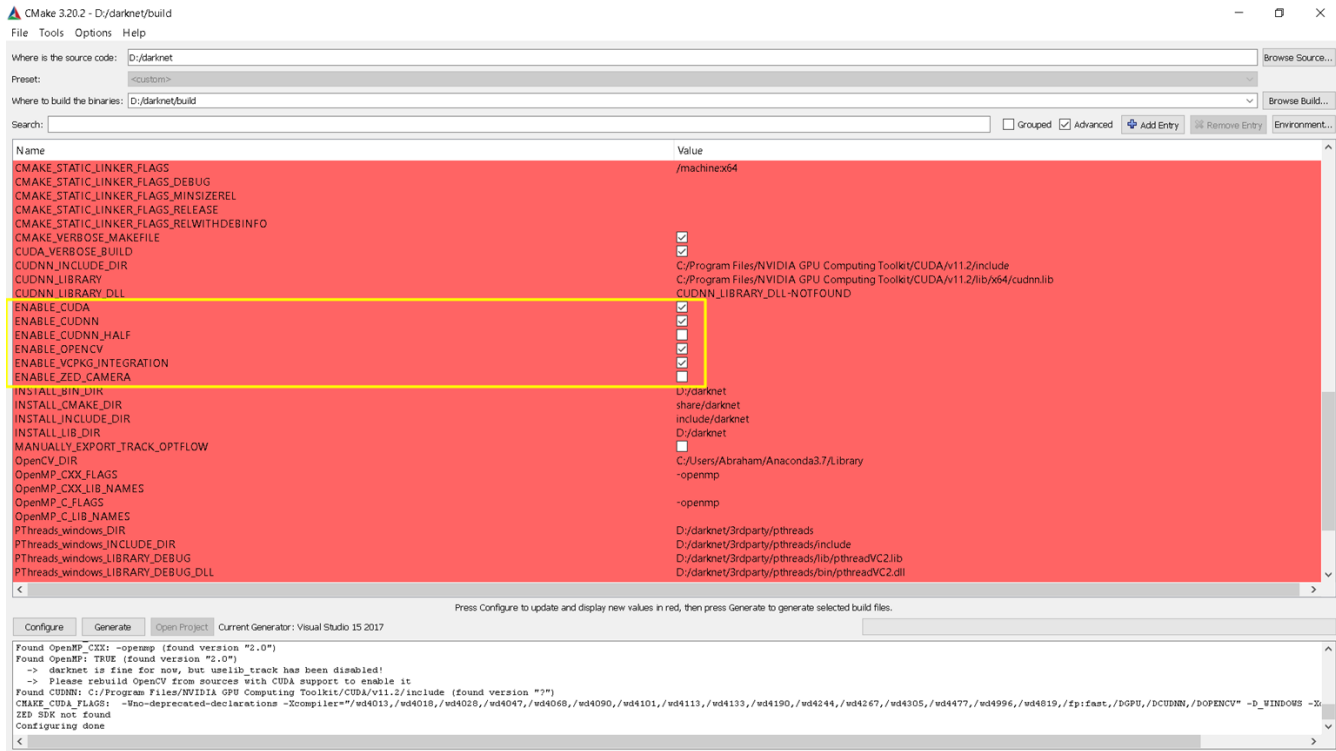
# Step 5 - Open CMake, Choose Source Repos and Build repos, and Configure



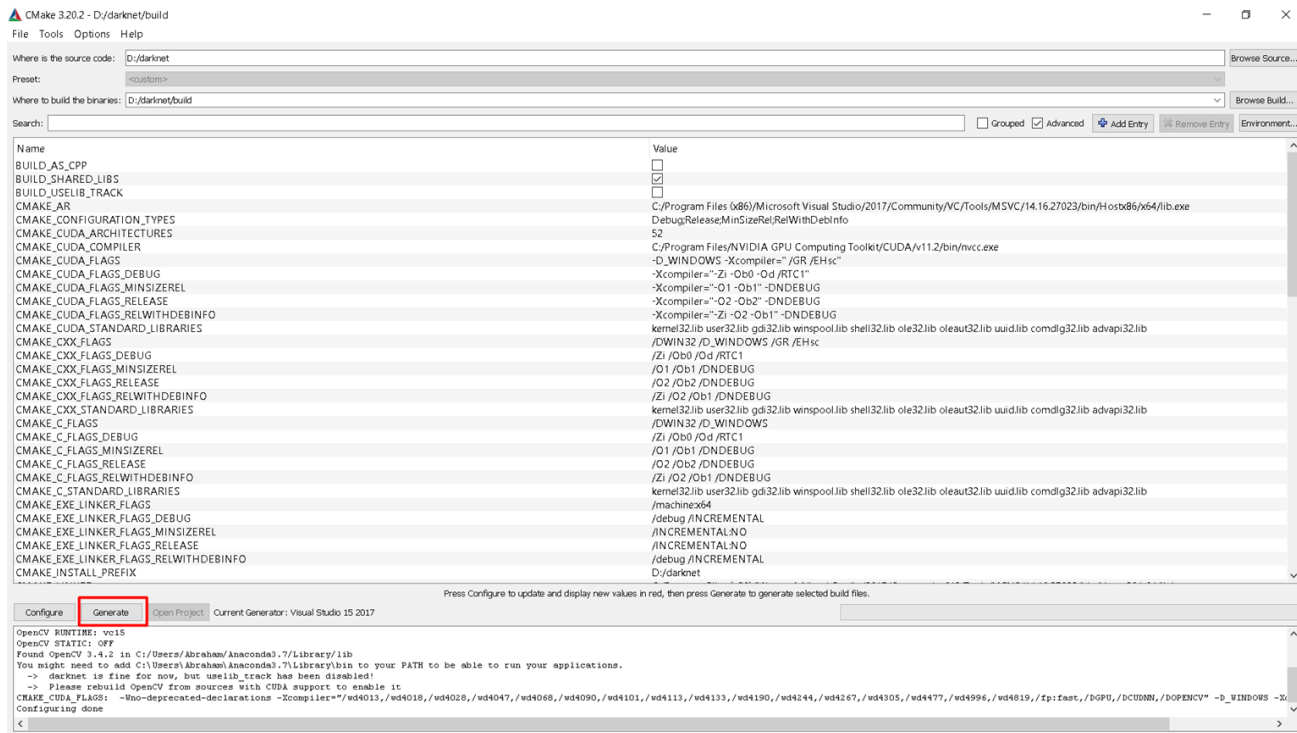
# Step 6 - Use Cmake to Configure and Generate Build Code



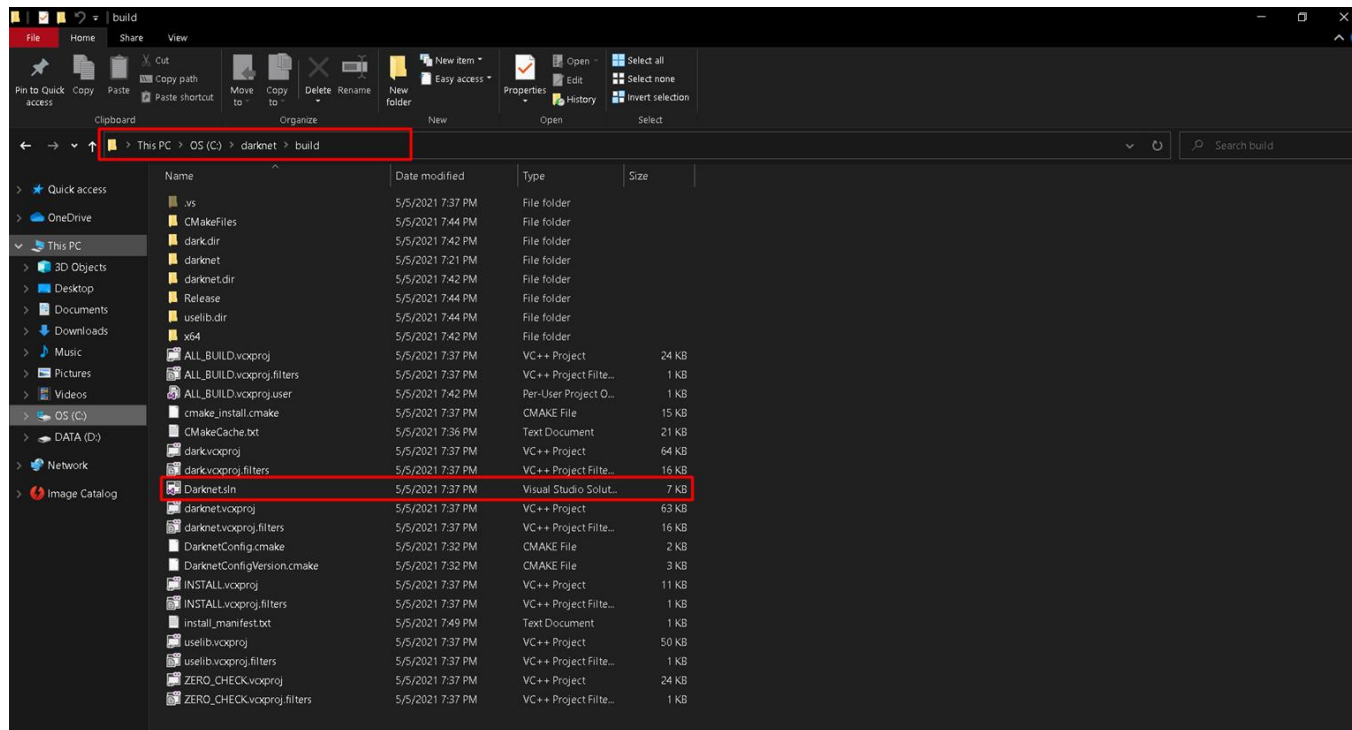
# Step 7 - Change some value and Configure again



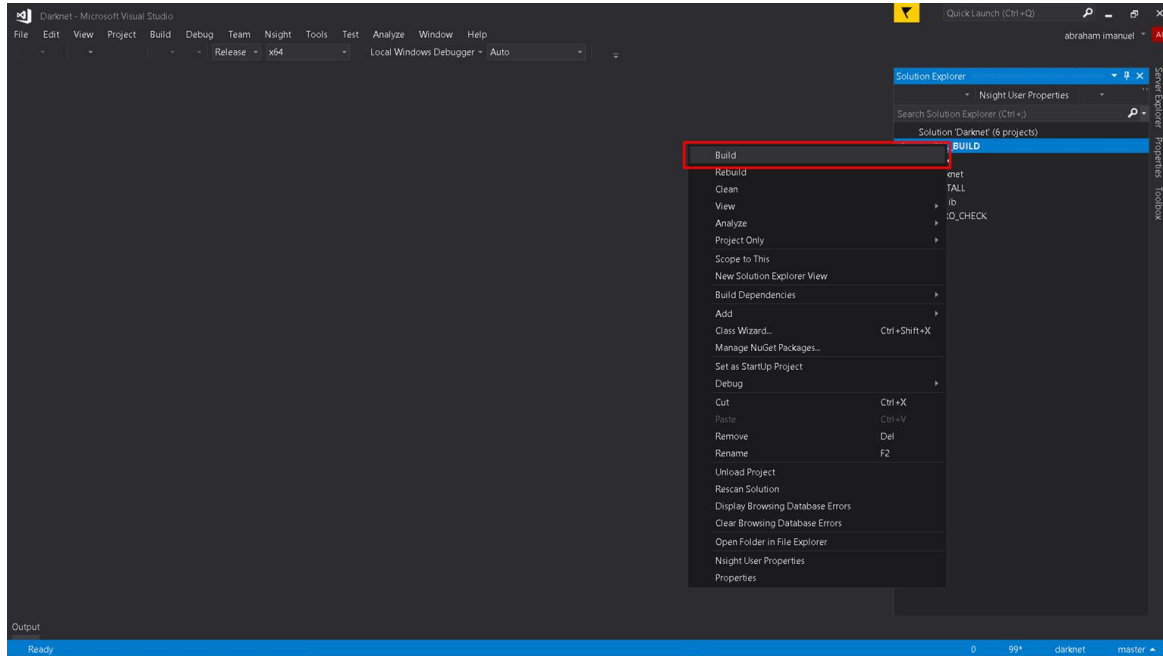
# Step 8 - Make sure there aren't any reds and Generate Code



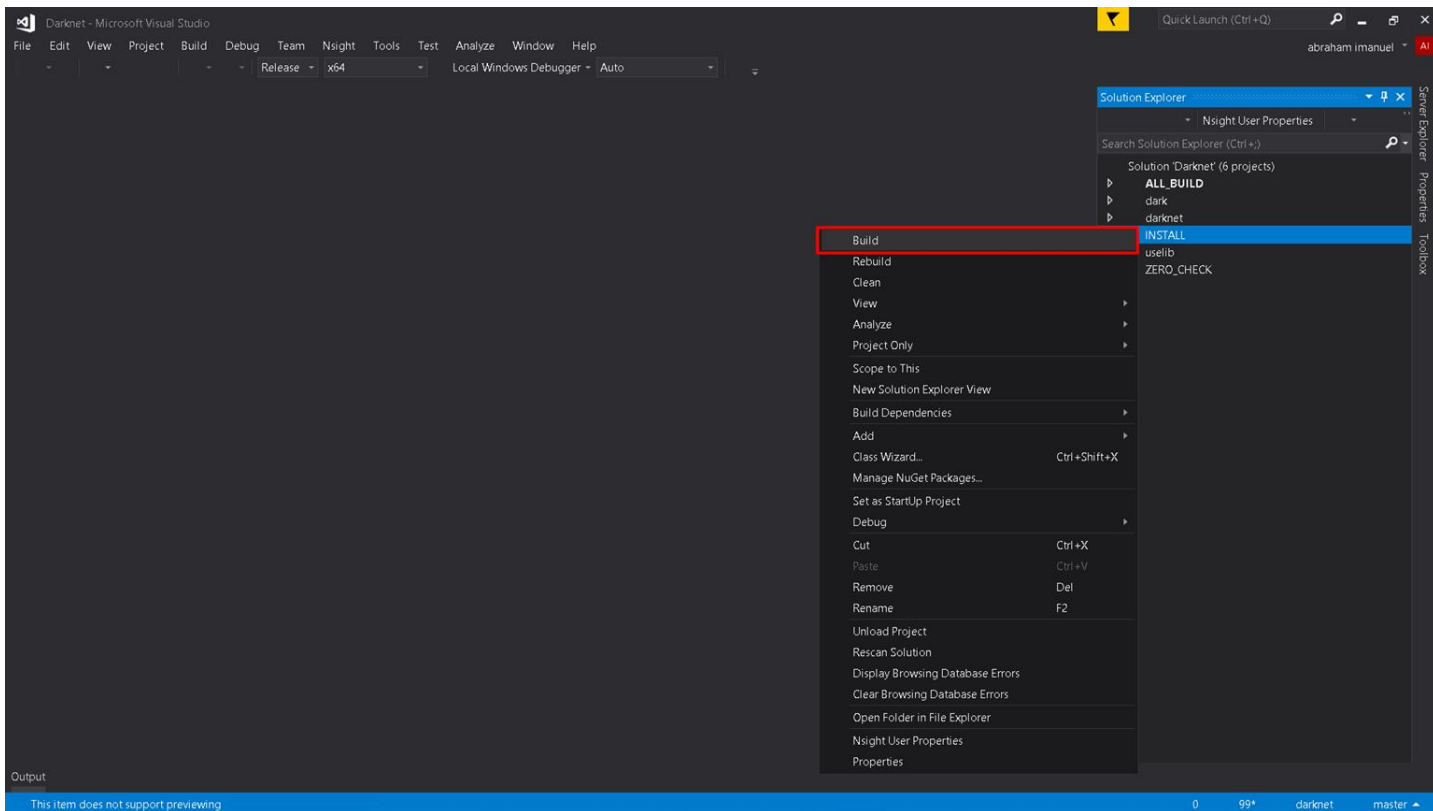
# Step 9 - Go to Build Folder and open Darknet.sln



# Step 10 - Build “All\_Build”



# Step 11 - Build “INSTALL”



## Step 12 - Demo