





Home - OpenCV

opencv.org


OpenCV


ReleasesForumCoursesStorePartnershipResourcesLicenseAbout







OpenCV.AI Store is LIVE!


Purchase OAK from the OpenCV Store


Purchase OAK


Github


Online documentation


Tutorials


User Q&A forum


Courses

OpenCV Tutorials

- [Introduction to OpenCV](#) - build and install OpenCV on your computer
- [The Core Functionality \(core module\)](#) - basic building blocks of the library
- [Image Processing \(imgproc module\)](#) - image processing functions
- [Application utils \(highgui, imgcodecs, videoio modules\)](#) - application utils (GUI, image/video input/output)
- [Camera calibration and 3D reconstruction \(calib3d module\)](#) - extract 3D world information from 2D images
- [2D Features framework \(feature2d module\)](#) - feature detectors, descriptors and matching framework
- [Deep Neural Networks \(dnn module\)](#) - infer neural networks using built-in *dnn* module
- [Graph API \(gapi module\)](#) - graph-based approach to computer vision algorithms building
- [Other tutorials \(ml, objdetect, photo, stitching, video\)](#) - other modules (ml, objdetect, stitching, video, photo)
- [OpenCV iOS](#) - running OpenCV on an iDevice
- [GPU-Accelerated Computer Vision \(cuda module\)](#) - utilizing power of video card to run CV algorithms



Deep Neural Networks (dnn module)

- [Load Caffe framework models](#)
- [How to enable Halide backend for improve efficiency](#)
- [How to schedule your network for Halide backend](#)
- [How to run deep networks on Android device](#)
- [YOLO DNNs](#)
- [How to run deep networks in browser](#)
- [Custom deep learning layers support](#)
- [How to run custom OCR model](#)
- [High Level API: TextDetectionModel and TextRecognitionModel](#)



yolo homepage



전체

이미지

뉴스

동영상

지도

더보기

설정

도구

검색결과 약 5,400,000개 (0.42초)

[pjreddie.com](#) > yolo ▾

YOLO: Real-Time Object Detection - Joseph Redmon

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% ...

Darknet

YOLO - Installing Darknet -
Nightmare - Coq Tactic - ...

YOLO: Real-Time Object ...

You only look once (YOLO) is a
state-of-the-art, real-time object ...

[pjreddie.com](#) 검색결과 더보기 »

Publications

My publications.

Pascal VOC Dataset Mirror

Here is a mirror for the Pascal
VOC files in case, you know, you ...



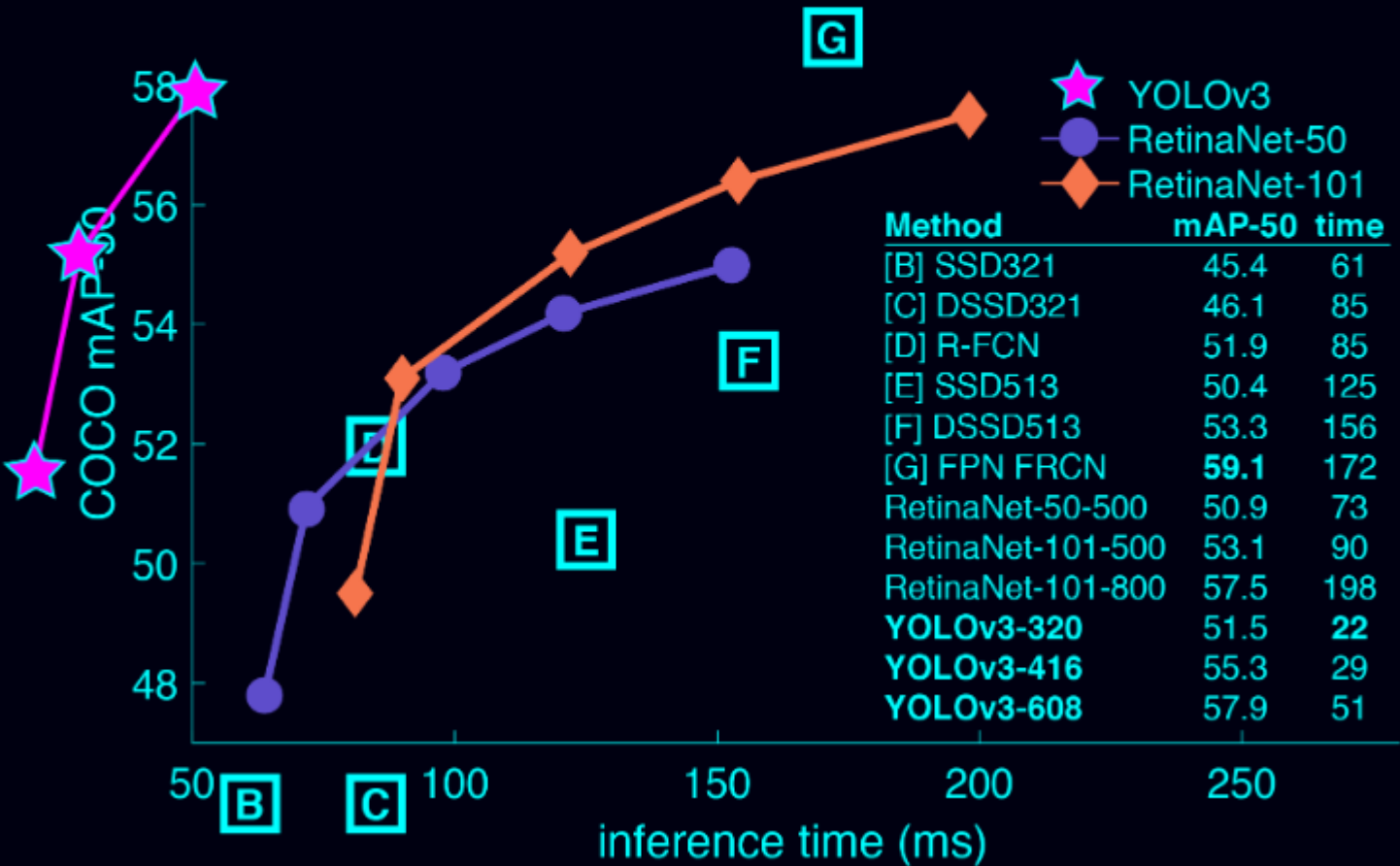
YOLO: Real-Time Object Detection

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.



Comparison to Other Detectors

YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!



Single Shot Detector(SSD)

YOLO(You only Look Once)



info@cocodataset.org

Home

People

Dataset

Tasks

Evaluate

News

- We are pleased to announce the COCO 2020 [Detection](#), [Keypoint](#), [Panoptic](#), and [DensePose](#) Challenges.
- The new rules and awards for this year challenges encourage innovative methods.
- Results to be announced at the [Joint COCO and LVIS Recognition ECCV](#) workshop.

What is COCO?



COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Collaborators

Tsung-Yi Lin Google Brain

Genevieve Patterson MSR, Trash TV

Matteo R. Ronchi Caltech

Yin Cui Google

Michael Maire TTI-Chicago

Serge Belongie Cornell Tech

Lubomir Bourdev WaveOne, Inc.

Ross Girshick FAIR

James Hays Georgia Tech

Pietro Perona Caltech

Deva Ramanan CMU

Larry Zitnick FAIR

Piotr Dollár FAIR

Sponsors



CVDF



Microsoft



Mighty Ai



4.5.1-pre

Open Source Computer Vision

[Main Page](#)

Related Pages

Modules

Namespaces ▾

Classes ▼

Files ▼

Examples

Java documentation

Search

OpenCV Tutorials

- Deep Neural Networks (dnn module)

YOLO DNNs

Prev Tutorial: [How to run deep networks on Android device](#)

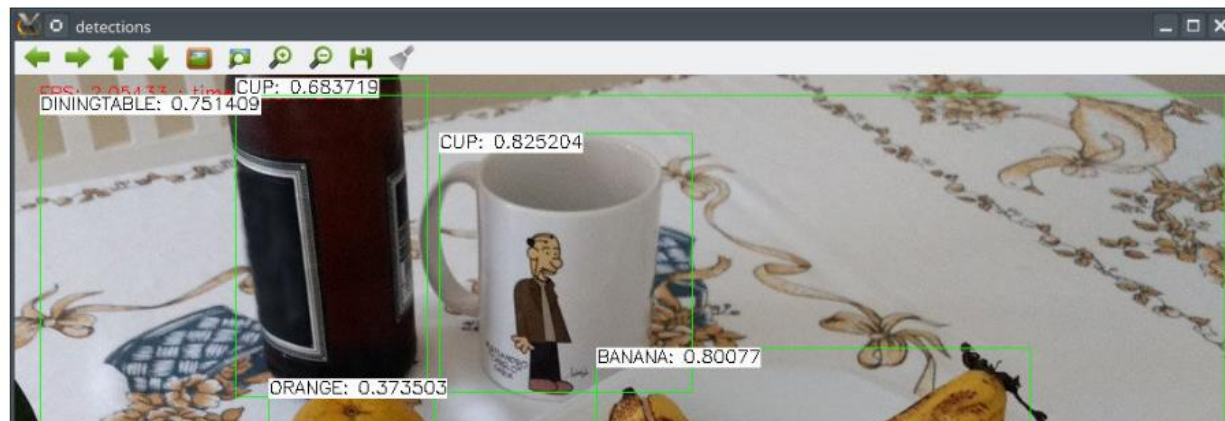
Next Tutorial: [How to run deep networks in browser](#)

Original author	Alessandro de Oliveira Faria
Compatibility	OpenCV >= 3.3.1

Introduction

In this text you will learn how to use `opencv_dnn` module using `yolo_object_detection` (Sample of using OpenCV dnn module in real time with device capture, video and image).

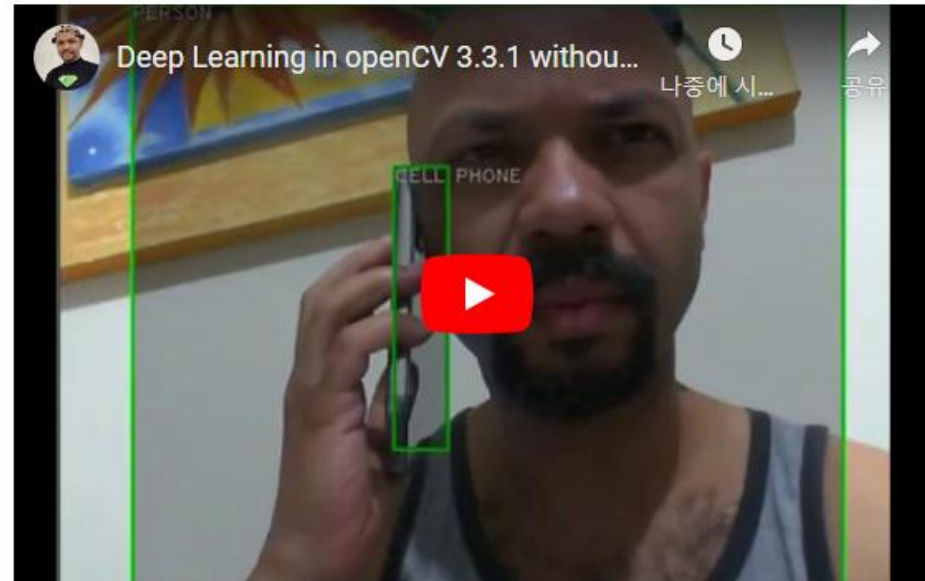
We will demonstrate results of this example on the following picture.





Examples

VIDEO DEMO:



Source Code

Use a universal sample for object detection models written in C++ and in Python languages

Usage examples

Execute in webcam:

```
$ example_dnn_object_detection --config=[PATH-TO-DARKNET]/cfg/yolo.cfg --model=[PATH-TO-DARKNET]/yolo.weights --  
classes=object_detection_classes_pascal_voc.txt --width=416 --height=416 --scale=0.00392 --rgb
```

Execute with image or video file:

```
$ example_dnn_object_detection --config=[PATH-TO-DARKNET]/cfg/yolo.cfg --model=[PATH-TO-DARKNET]/yolo.weights --  
classes=object_detection_classes_pascal_voc.txt --width=416 --height=416 --scale=0.00392 --input=[PATH-TO-IMAGE-OR-VIDEO-FILE]  
--rgb
```

Questions and suggestions email to: Alessandro de Oliveira Faria cabelo@opensuse.org or OpenCV Team.

윈도우에서 LabelImg 설치하기

Windows

Install [Python](#), [PyQt5](#) and [install lxml](#).

Open cmd and go to the [labelImg](#) directory

```
pyrcc4 -o lib/resources.py resources.qrc
For pyqt5, pyrcc5 -o libs/resources.py resources.qrc

python labelImg.py
python labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

Windows + Anaconda

Download and install [Anaconda](#) (Python 3+)

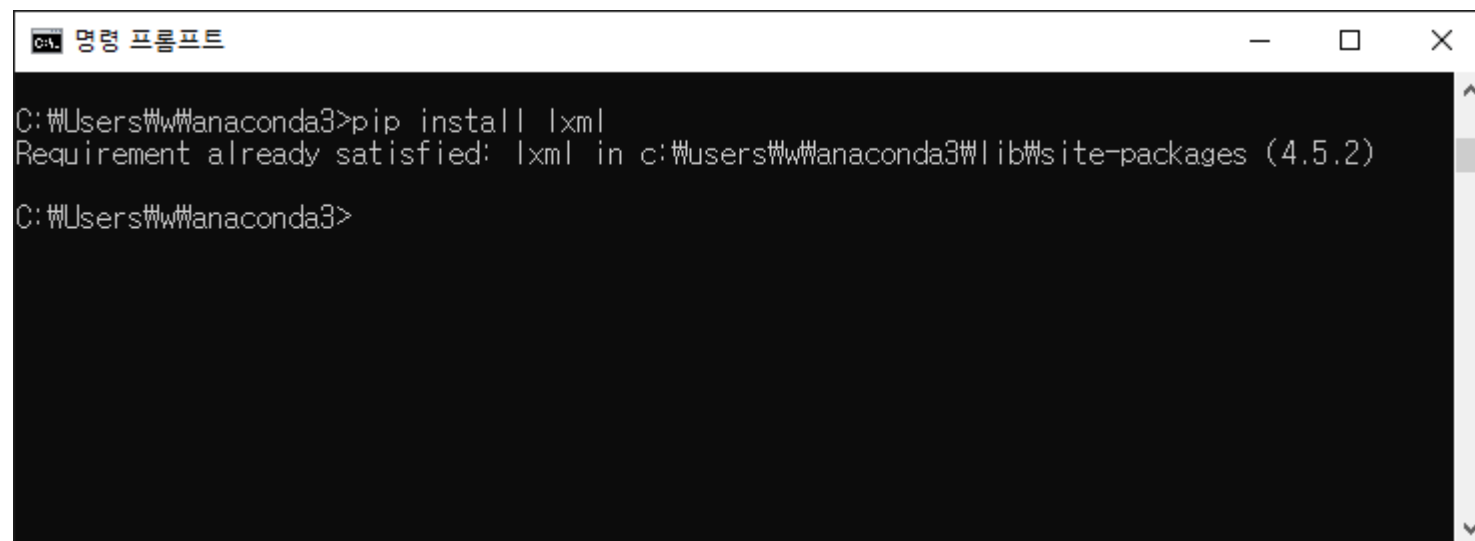
Open the Anaconda Prompt and go to the [labelImg](#) directory

```
conda install pyqt=5
conda install -c anaconda lxml
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py
python labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

윈도우에서 LabelImg 설치하기

1. Python 설치
 - pip install --upgrade pip
2. Opencv-Python 설치
 - python -m pip install opencv-python
3. PyQt5 설치
 - pip install pyqt5-tools
 - pip install pyqt5
4. lxml 설치
 - pip install lxml
5. Labelimg 다운로드
 - <https://github.com/tzutalin/labelImg#labelimg>
6. OSGeo4W 설치
 - pyrc5 을 사용하기 위해 필요
 - <https://trac.osgeo.org/osgeo4w/>

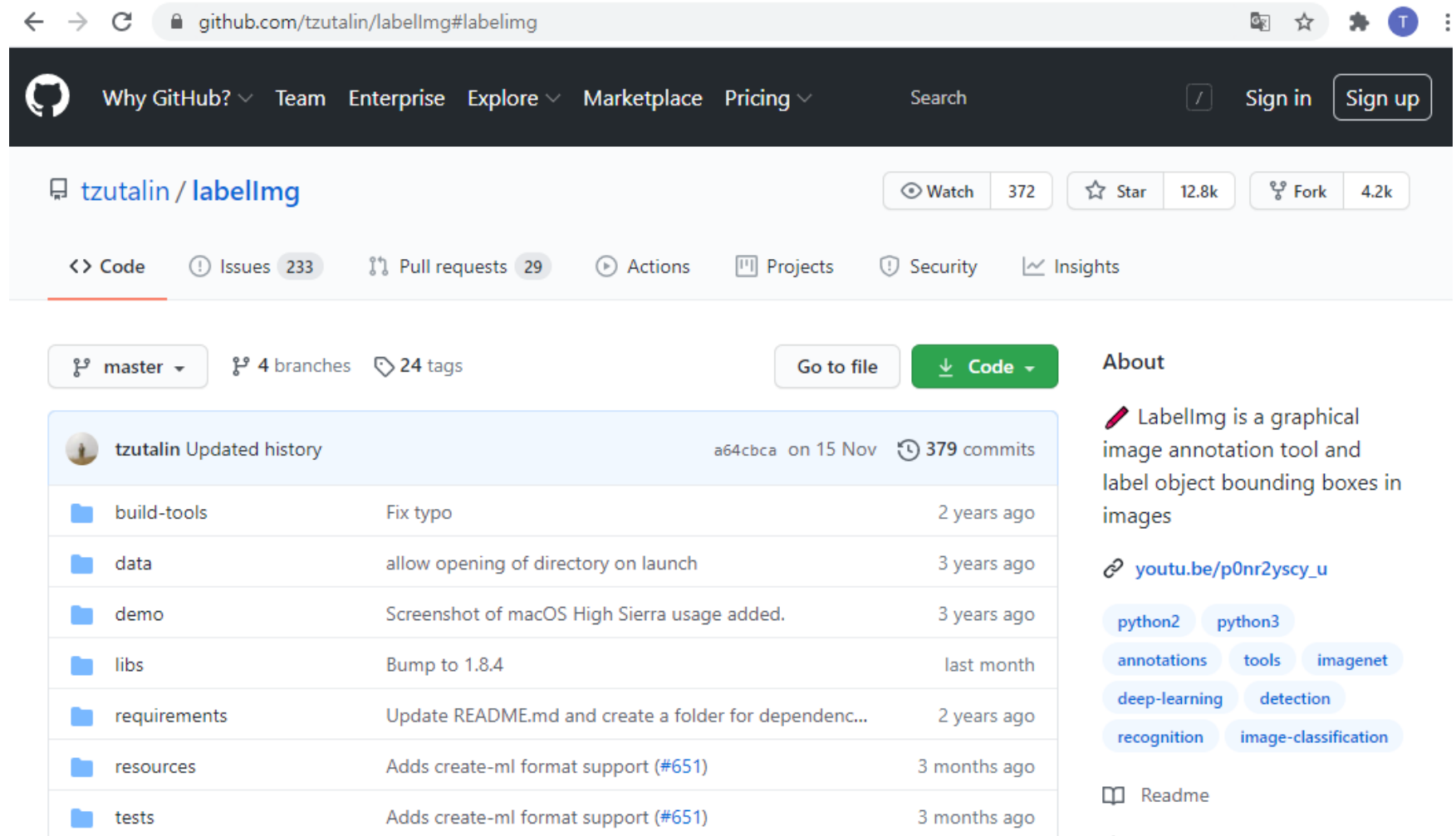
lxml 설치



```
CA. 명령 프롬프트
C:\Users\w\anaconda3>pip install lxml
Requirement already satisfied: lxml in c:\users\w\anaconda3\lib\site-packages (4.5.2)
C:\Users\w\anaconda3>
```

Labelimg 다운로드

1. 코드를 눌러 다운로드 (labellmg-master.zip)
2. 다운로드한 파일을 압축 풀기
3. 압축을 푼 폴더를 C:\labellmg-master 로 이동



← → ↻ 🔒 github.com/tzutalin/labellmg#labellmg

Why GitHub? Team Enterprise Explore Marketplace Pricing Search / Sign in Sign up

tzutalin / labellmg

Watch 372 Star 12.8k Fork 4.2k

Code Issues 233 Pull requests 29 Actions Projects Security Insights

master 4 branches 24 tags Go to file Code

tzutalin Updated history a64cbca on 15 Nov 379 commits

build-tools	Fix typo	2 years ago
data	allow opening of directory on launch	3 years ago
demo	Screenshot of macOS High Sierra usage added.	3 years ago
libs	Bump to 1.8.4	last month
requirements	Update README.md and create a folder for dependenc...	2 years ago
resources	Adds create-ml format support (#651)	3 months ago
tests	Adds create-ml format support (#651)	3 months ago

About

Labellmg is a graphical image annotation tool and label object bounding boxes in images

youtu.be/p0nr2yscy_u

python2 python3

annotations tools imagenet

deep-learning detection

recognition image-classification

Readme

OSGeo4W 설치

1. 32비트 또는 64비트를 다운로드

OSGeo4W

https://trac.osgeo.org/osgeo4w/

Internet Explorer에...

OSGeo4W
FOSSGIS for Windows

로그인 | 도움말 | Trac에 대해서 | 설정

위키 | 시간이력 | 로드맵 | 소스 둘러보기 | 티켓 보기 | 검색

위키: WikiStart

시작 페이지 | 페이지 색인 | 이력

OSGeo4W

[Japanese](#) [French](#) [Polish](#) [Deutsch](#)

This is the web site, wiki and issue tracking database for the **OSGeo4W project**. OSGeo4W is a binary distribution of a broad set of open source geospatial software for Windows environments (Windows 10 down to XP). OSGeo4W includes [QGIS](#), [GDAL/OGR](#), [GRASS](#) as well as [many other packages](#) (over 150).

This project is under the umbrella of the [Open Source Geospatial Foundation](#)

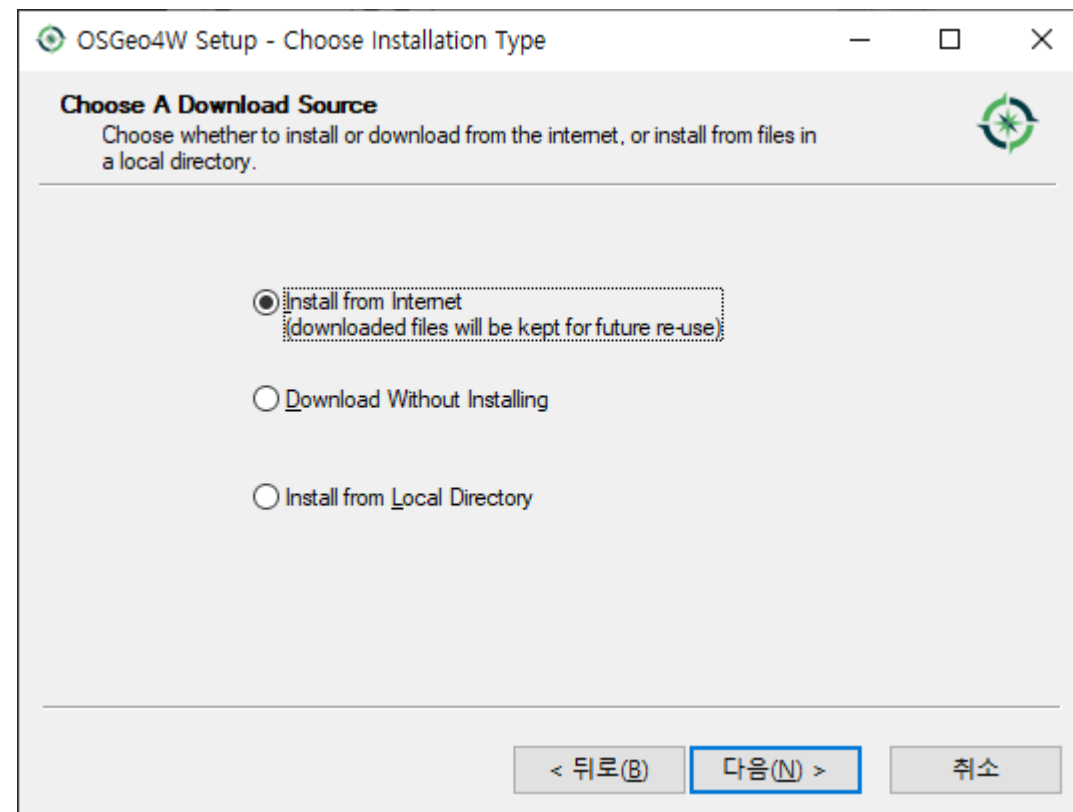
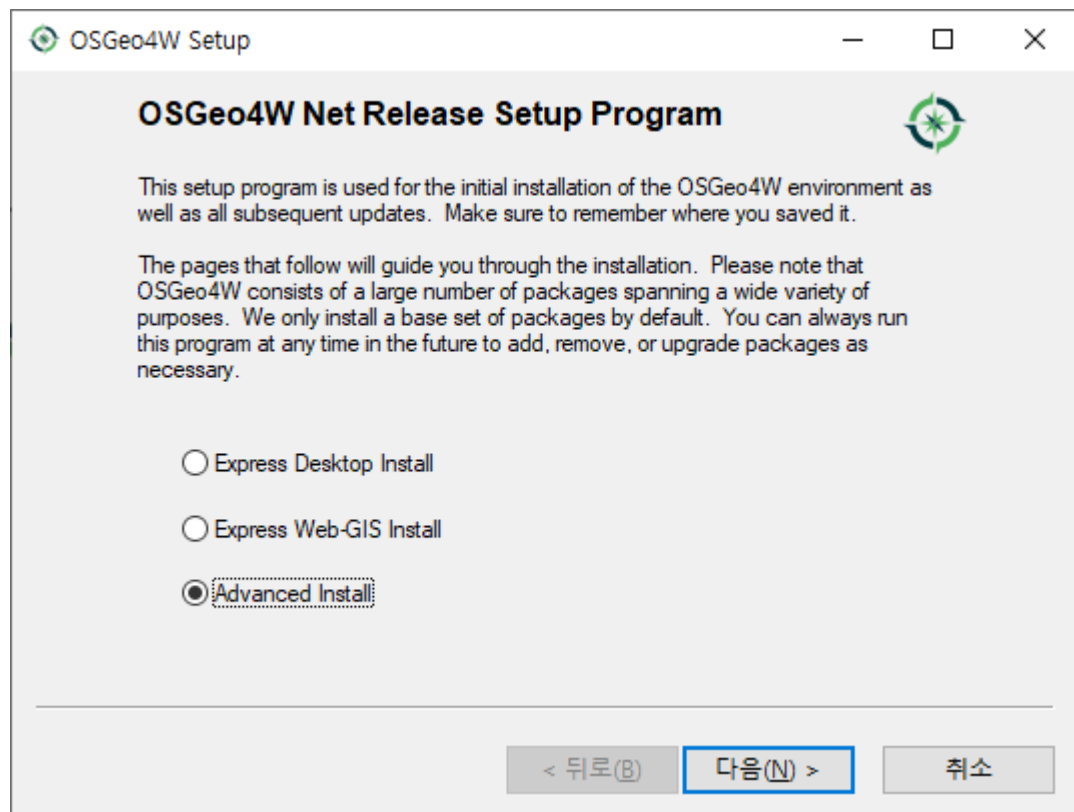
Only authenticated (logged in) users can submit and edit tickets, and modify the wiki. Use your [OSGeo userid/password](#) to login.

[About licenses](#)

Quick Start for OSGeo4W Users

1. Download the [32bit](#) or [64bit](#) OSGeo4W network installer
2. Run the installer.
3. Select *Express Install*, and *Next*.
4. Pick one or more packages to install, and *Next*.
5. The selected packages and their required sub packages will be downloaded and installed automatically.

OSGeo4W 설치



OSGeo4W 설치

OSGeo4W Setup - Choose Installation Directory

Select Root Install Directory

Select the directory where you want to install OSGeo4W. Also choose a few installation parameters.

Root Directory

C:\OSGeo4W64

Browse...

Install For

☒ All Users (RECOMMENDED)

OSGeo4W will be available to all users of the system.

☐ Just Me

OSGeo4W will only be available to the current user. Only select this if you lack Admin. privileges or you have specific needs.

☐ Create icon on Desktop

☒ Add icon to Start Menu

< 뒤로(B)

다음(N) >

취소

OSGeo4W Setup - Select Local Package Directory

Select Local Package Directory

Select a directory where you want Setup to store the installation files it downloads. The directory will be created if it does not already exist.

Local Package Directory

C:\Users\www\AppData\Local\Temp

Browse...

Start menu name

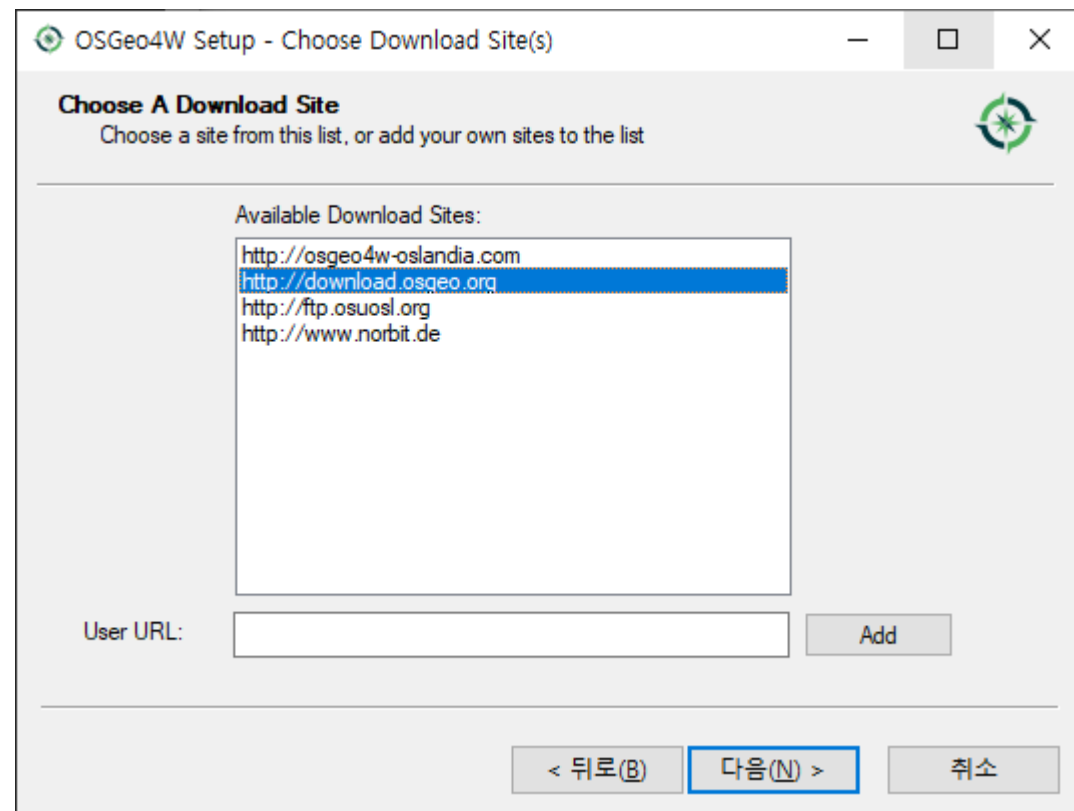
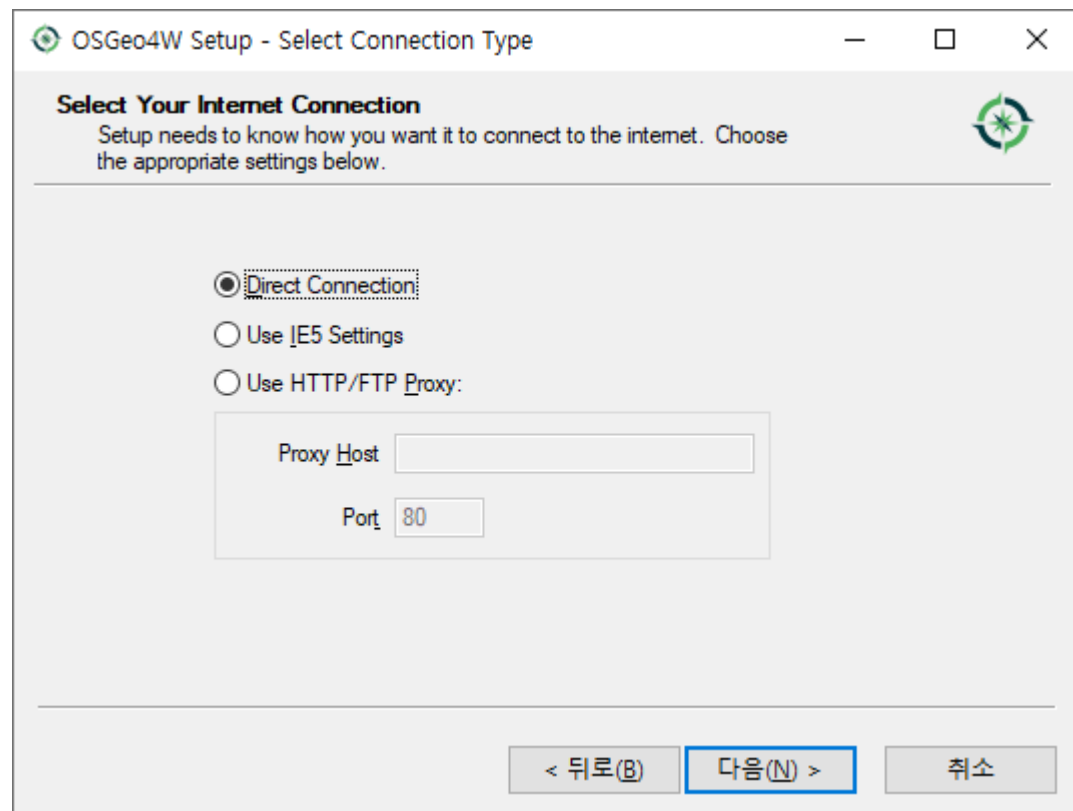
OSGeo4W

< 뒤로(B)

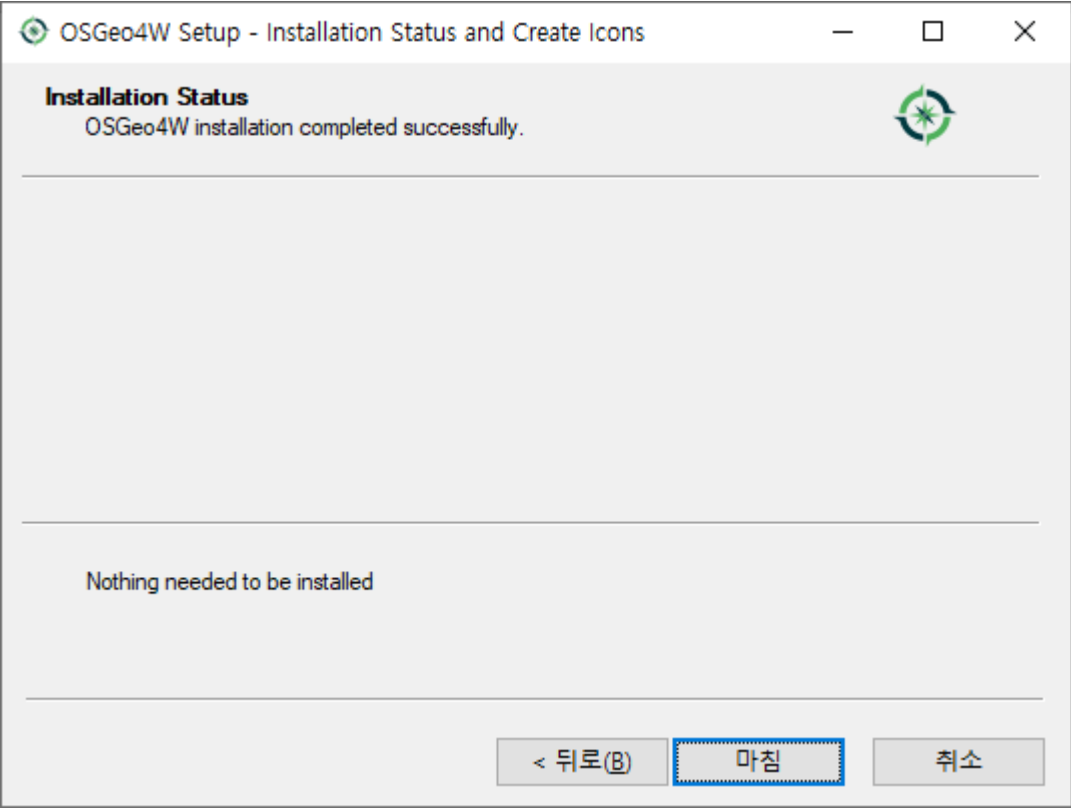
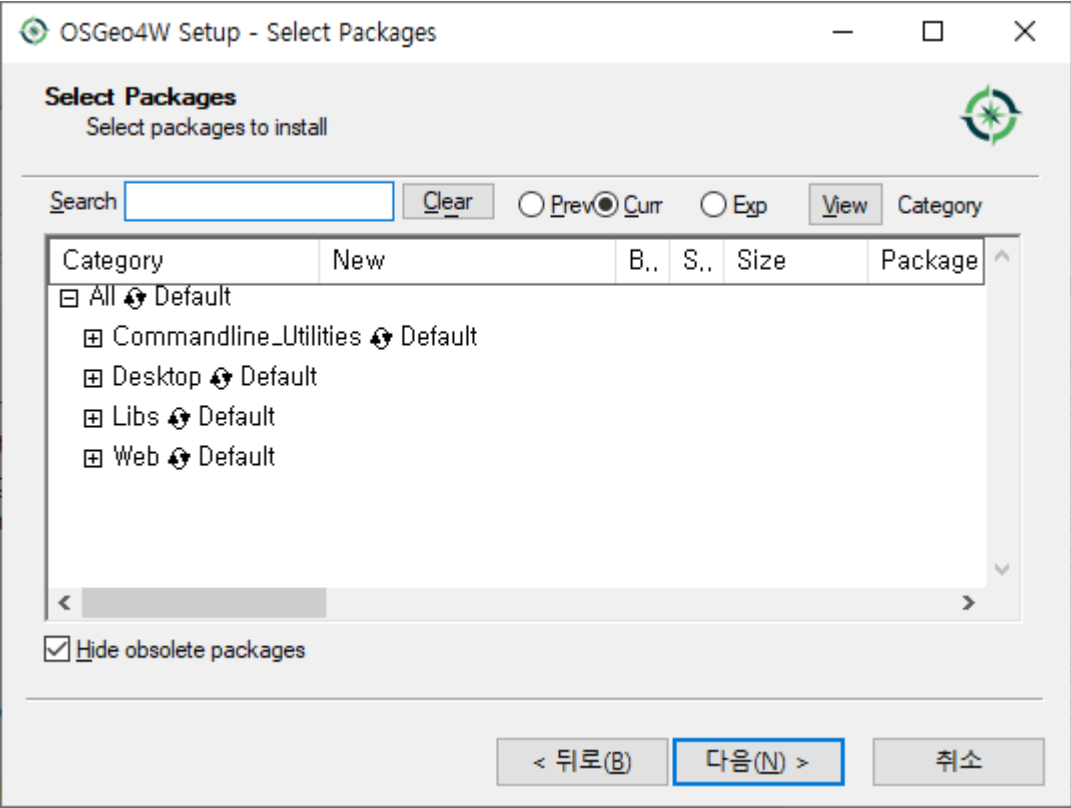
다음(N) >

취소

OSGeo4W 설치



OSGeo4W 설치



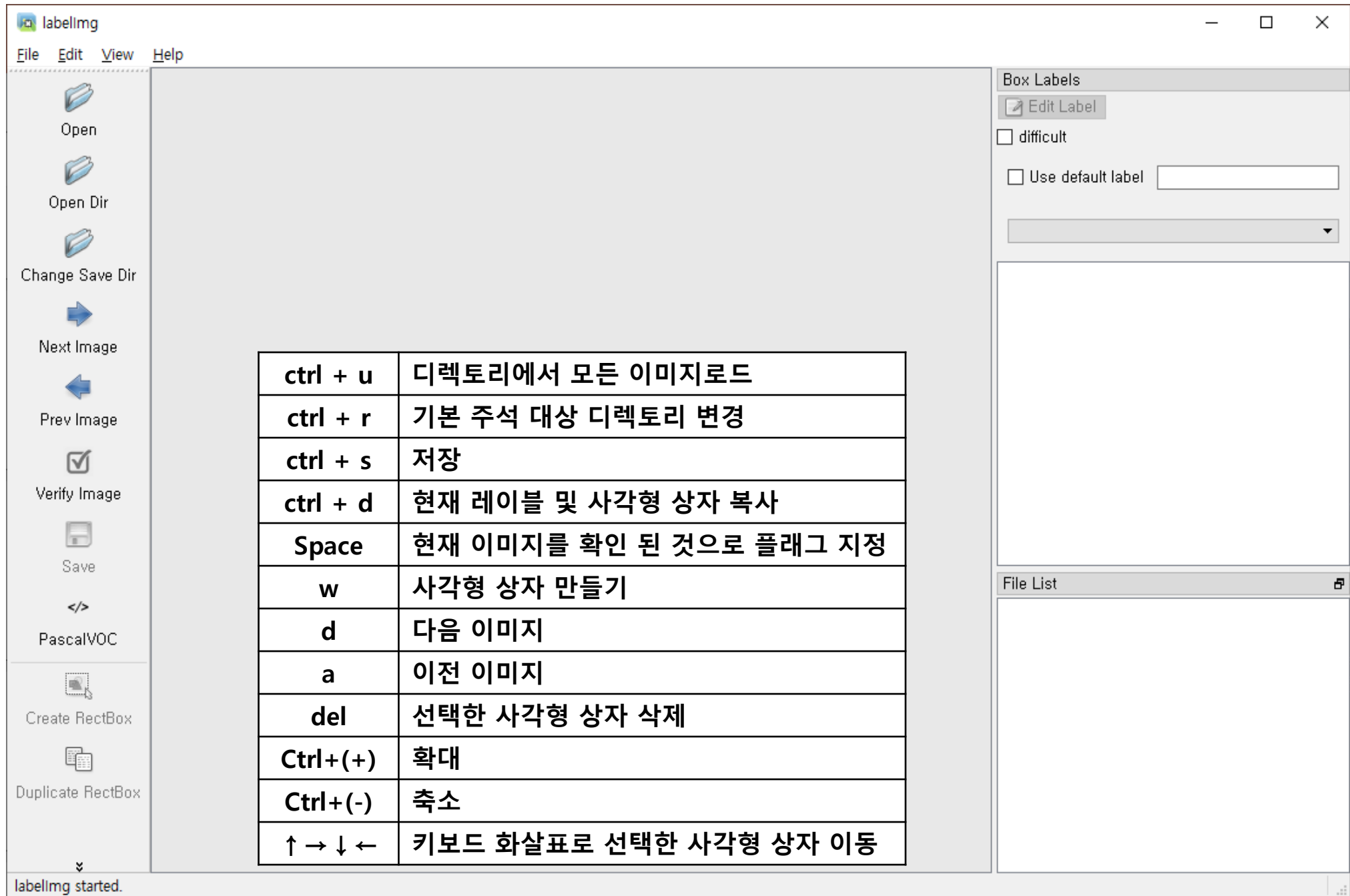
labelImg 실행

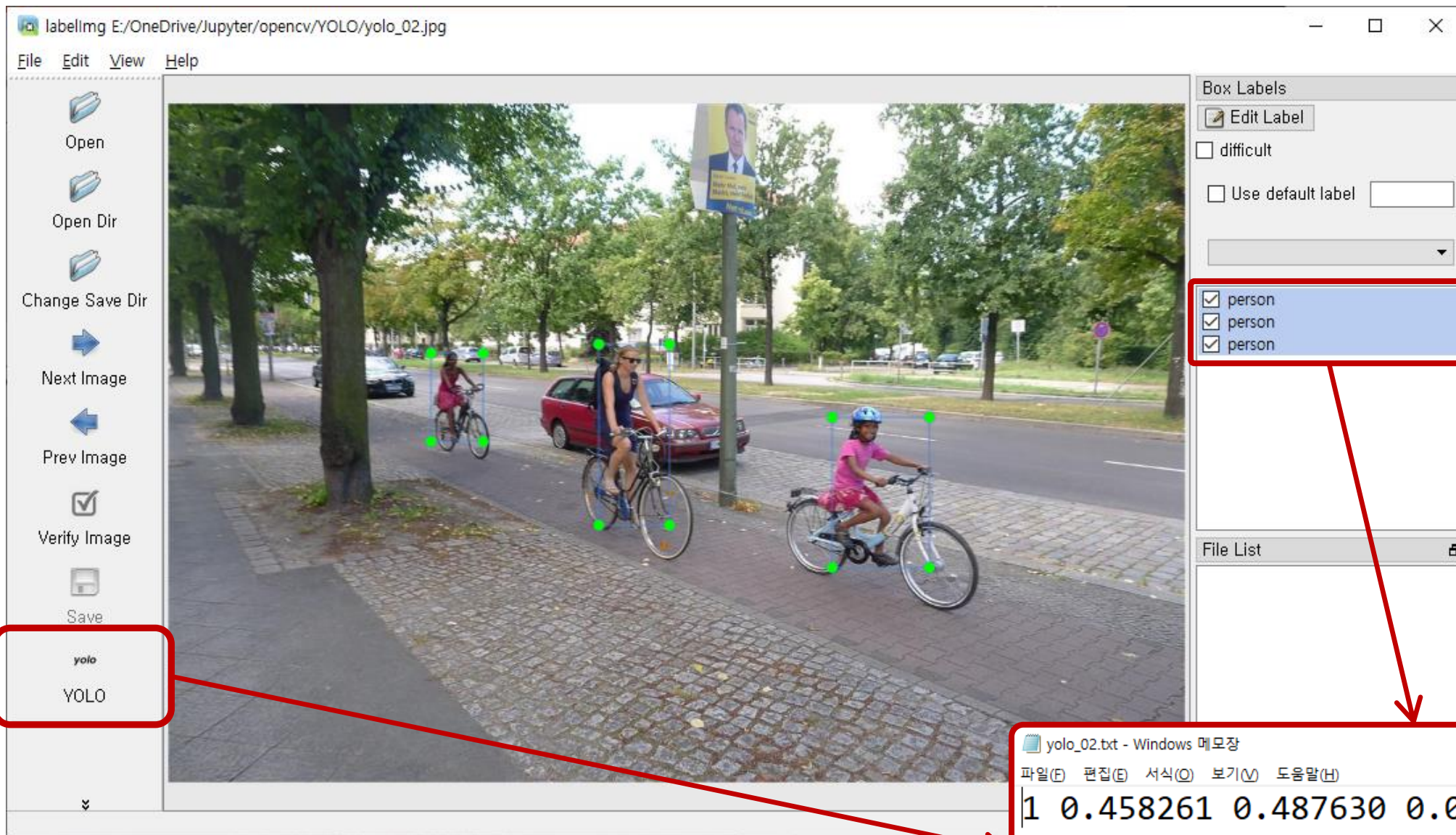
```
C:\>cd labelImg-master  
C:\labelImg-master>pyrcc5 -o libs/resources.py resources.qrc  
C:\labelImg-master>python labelImg.py  
labelImg.py:203: DeprecationWarning: an integer is required (got type DockWidgetFeatures). Implicit conversion to integers using __int__ is deprecated, and may be removed in a future version of Python.  
    self.dock.setFeatures(self.dock.features() ^ self.dockFeatures)  
C:\labelImg-master>
```

```
C:\> cd labelImg-master
```

```
C:\labelImg-master> pyrcc5 -o libs/resources.py resources.qrc
```

```
C:\labelImg-master> python labelImg.py
```





yolo_02.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
1 0.458261 0.487630 0.069565 0.266927
1 0.700870 0.572917 0.095652 0.221354
1 0.283913 0.432292 0.051304 0.130208
```

Ln 1, Col 1 30% Unix (LF) UTF-8



cat01.jpg

Class

Center x

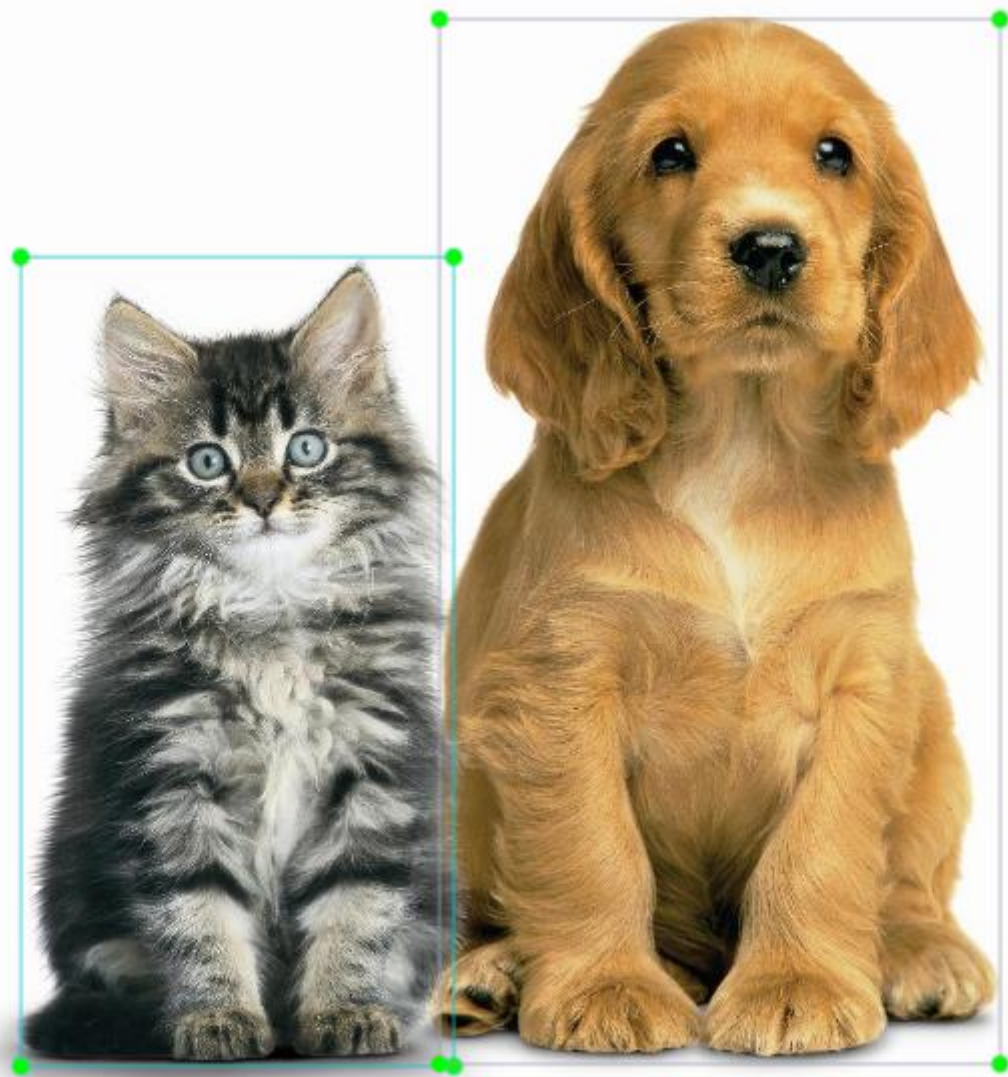
Center y

Width

Height

cat01.txt

검출 해야 할 물체의 종류

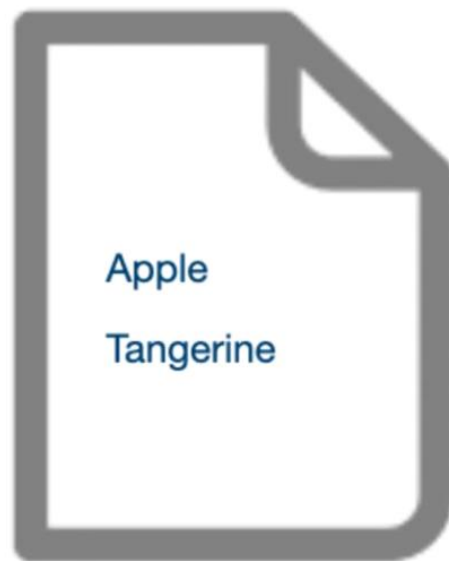


Cat-Dog.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
2 0.289837 0.593981 0.319512 0.680556
0 0.645935 0.493056 0.413821 0.878704
|
```

Ln 3, Col 1 100% Unix (LF) UTF-8



classes.names



custom_data.data

```
classes.names - Windows 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
Apple
Tangerine
Ln 1, Col 1
```

```
custom_data.data - Windows 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
classes = 2
train = /content/gdrive/My Drive/darknet/custom/train.txt
valid = /content/gdrive/My Drive/darknet/custom/test.txt
names = /content/gdrive/My Drive/darknet/custom/classes.names
backup = backup
Ln 1, Col 1  100%  Unix (LF)  UTF-8
```

train.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
/content/gdrive/My Drive/darknet/custom/fruit05.jpg  
/content/gdrive/My Drive/darknet/custom/fruit07.jpg  
/content/gdrive/My Drive/darknet/custom/fruit06.jpg  
/content/gdrive/My Drive/darknet/custom/fruit02.jpg  
/content/gdrive/My Drive/darknet/custom/fruit03.jpg  
/content/gdrive/My Drive/darknet/custom/fruit01.jpg  
/content/gdrive/My Drive/darknet/custom/fruit08.jpg  
/content/gdrive/My Drive/darknet/custom/fruit09.jpg
```

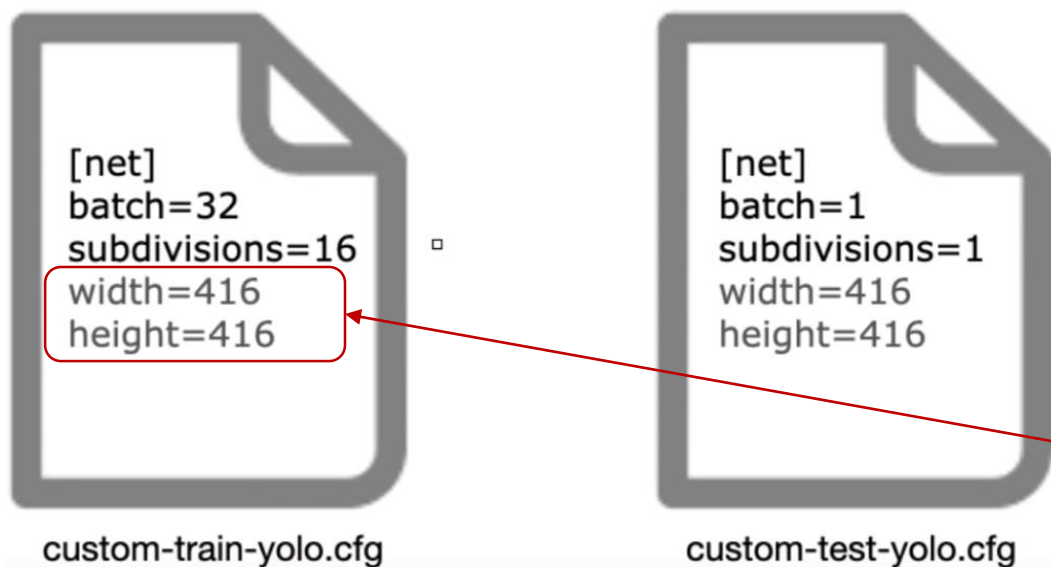
Ln 1, Col 1 100% Unix (LF) UTF-8

test.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

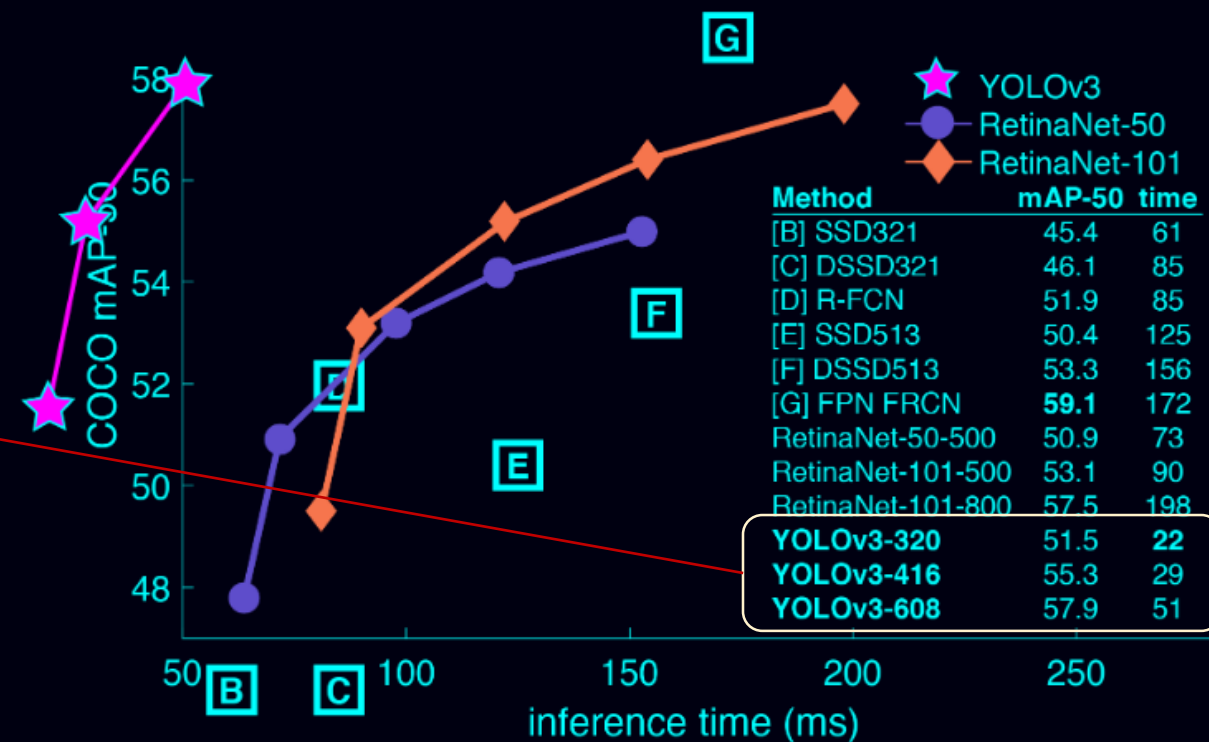
```
/content/gdrive/My Drive/darknet/custom/fruit04.jpg  
/content/gdrive/My Drive/darknet/custom/fruit10.jpg
```

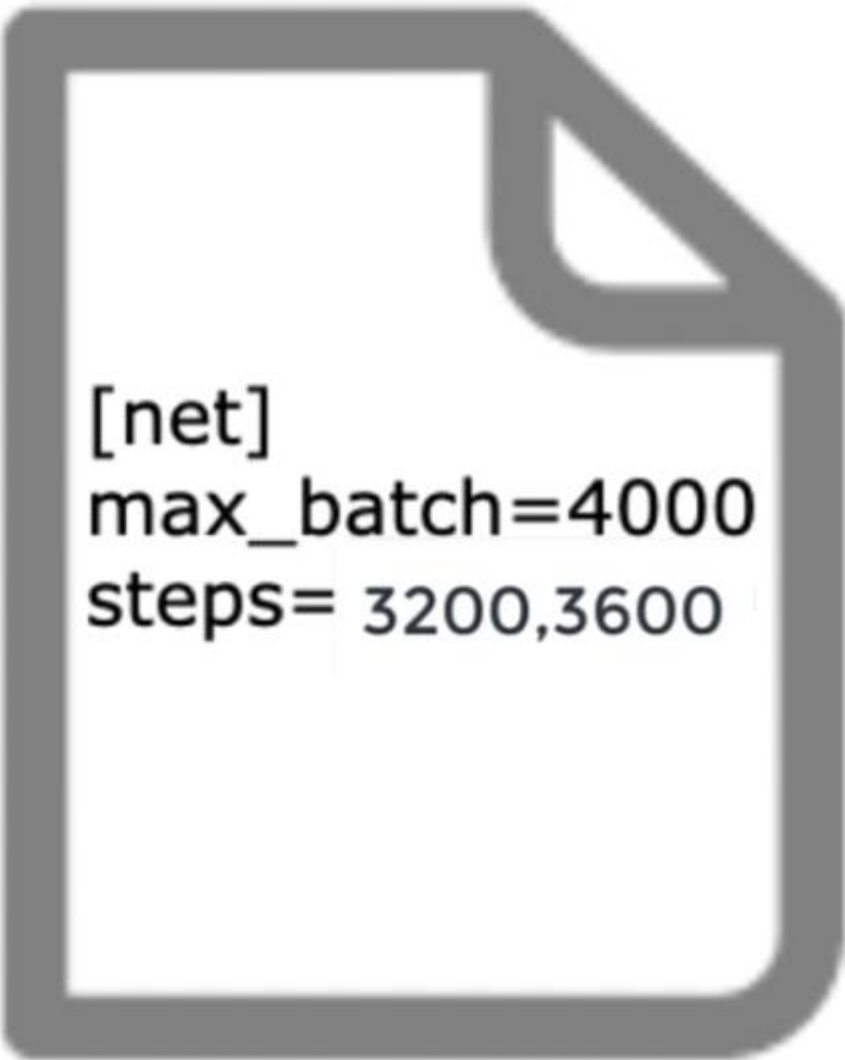
Ln 1, Col 1 100% Unix (LF) UTF-8



Comparison to Other Detectors

YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!





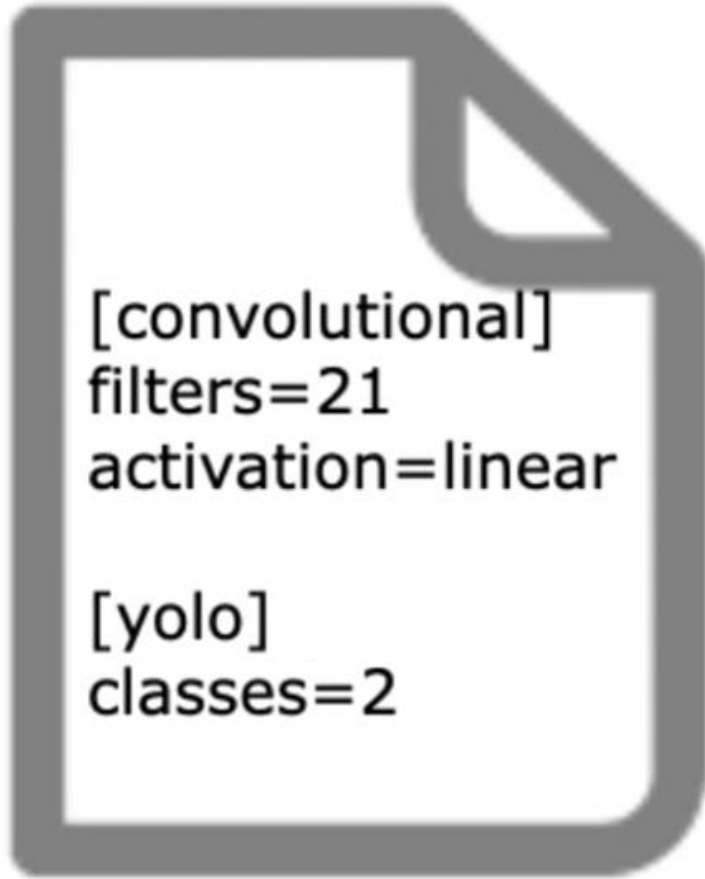
```
[net]
max_batch=4000
steps= 3200,3600
```

custom-train-yolo.cfg
custom-test-yolo.cfg

```
max_batches = classes * 2000
(>= 4000)
```

```
steps = max_batches * 0.8,
        max_batches * 0.9
```

Line 603, 689, 776



custom-train-yolo.cfg
custom-test-yolo.cfg

$$\text{filters} = (\text{classes} + \text{coordinates} + 1) * \text{masks}$$


$$\text{filters} = (\text{classes} + 5) * 3$$

$$\begin{aligned} \text{filters} &= (2 + 5) * 3 \\ &= 21 \end{aligned}$$

classes : 2 (apple, tangerine)
coordinates : 4 (center x, center y, width, height)
mask : 3 (color : r, g, b)

← → ↻ 🏠 pjreddie.com/darknet/ 📄 ⭐ 🖼️ 🔍 🛠️ ⚙️ 📁 📄

[home](#) [darknet](#) [cog](#) [tactics](#) [publications](#) [projects](#) [résumé](#)



Darknet: Open Source Neural Networks in C

Darknet is an open source neural network framework written in **C and CUDA**. It is fast, easy to install, and supports CPU and GPU computation. You can find the source on [GitHub](#) or you can read more about what Darknet can do right here:

Installing Darknet

Darknet is easy to install and run. This post will guide you through it.

YOLO: Real-Time Object Detection

You only look once (YOLO) is a state-of-the-art, real-time object detection system.

ImageNet Classification

Classify images with popular models like ResNet and ResNeXt.

