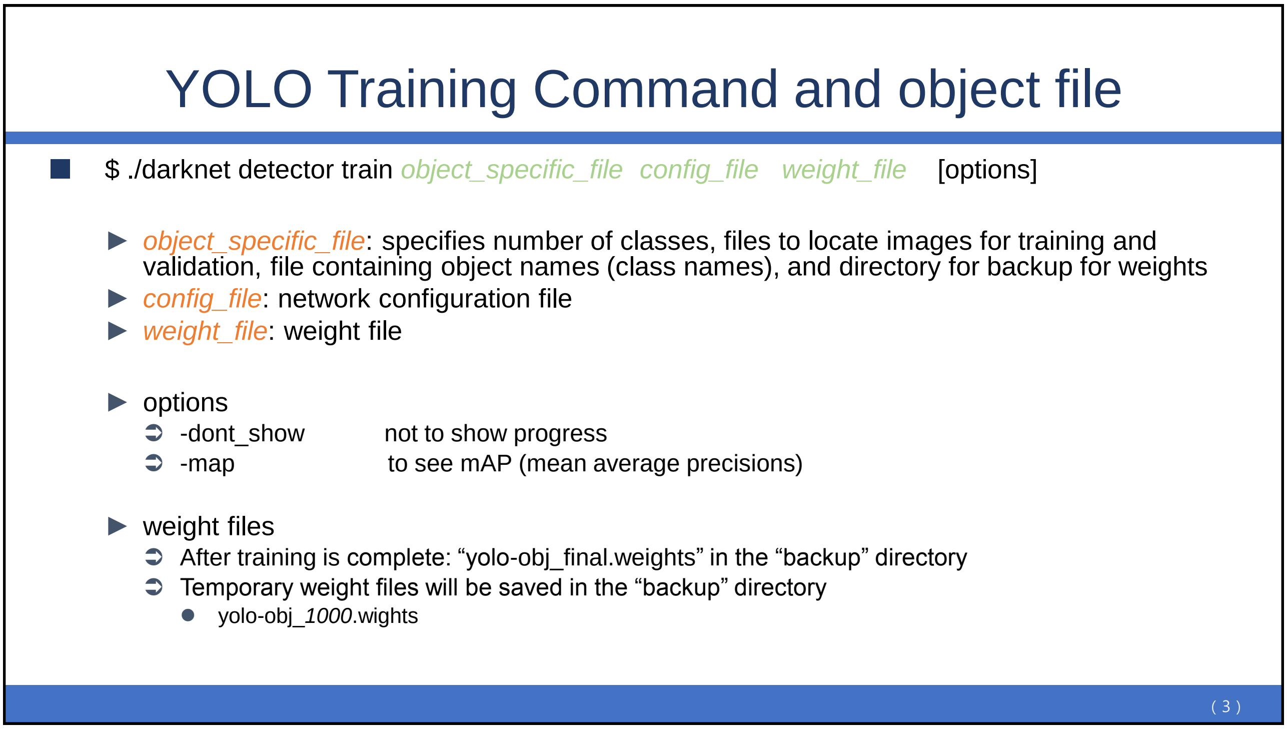
Darknet Training

Aug. 2019

Ando Ki, Ph.D.
adki@future-ds.com

Contents

- YOLO Training Command and object file
- Object file and train/valid file
- Configuration file
- Labeling object with bouncing box
- Labeling image files
- YOLO-Annotation-Tool case
- Prepare train and test file



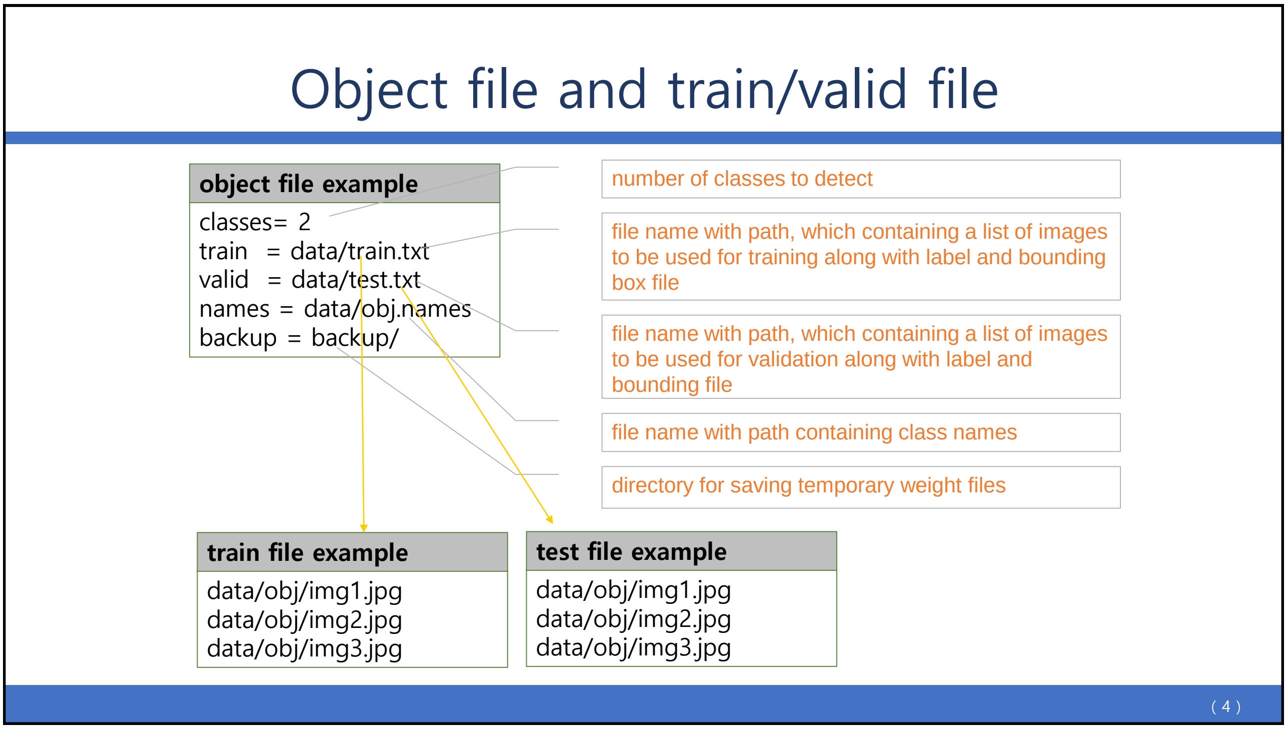
YOLO Training Command and object file

- $ ./darknet detector train *object_specific_file config_file weight_file* [options]

  - *object_specific_file*: specifies number of classes, files to locate images for training and validation, file containing object names (class names), and directory for backup for weights
  - *config_file*: network configuration file
  - *weight_file*: weight file

  - options
    - -dont_show not to show progress
    - -map to see mAP (mean average precisions)

  - weight files
    - After training is complete: “yolo-obj_final.weights” in the “backup” directory
    - Temporary weight files will be saved in the “backup” directory
      - yolo-obj_*1000*.wights



Object file and train/valid file

object file example

```
classes= 2
train  = data/train.txt
valid  = data/test.txt
names = data/obj.names
backup = backup/
```

number of classes to detect

file name with path, which containing a list of images to be used for training along with label and bounding box file

file name with path, which containing a list of images to be used for validation along with label and bounding file

file name with path containing class names

directory for saving temporary weight files

train file example

```
data/obj/img1.jpg
data/obj/img2.jpg
data/obj/img3.jpg
```

test file example

```
data/obj/img1.jpg
data/obj/img2.jpg
data/obj/img3.jpg
```

Configuration file

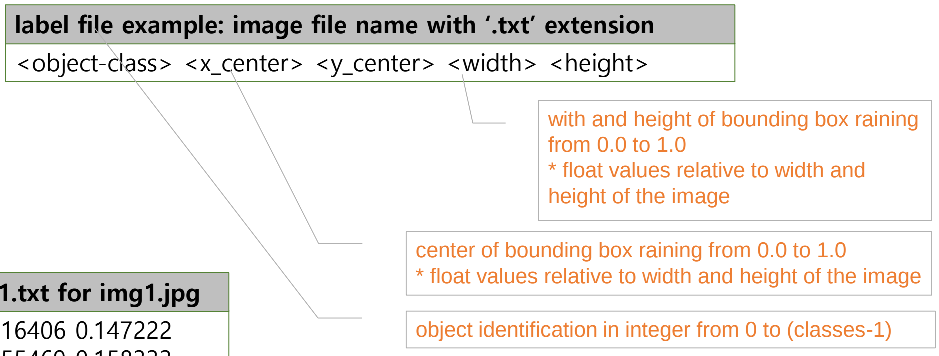
- Modify YOLO configuration file for your own training
  - Set the number of classes (i.e., objects)
  - Set the value of filter
    - For Yolov2 (classes + 5)*5
    - For Yolov3 (classes + 5)*3

Configuration file example

```
…
[convolutional]
…
filters=35
…
[region]
…
classes=2
…
```

Labeling object with bouncing box

- Darknet requires text file for each image file
  - E.g., ‘my_image.txt’ for ‘my_image.jpg’



label file example: image file name with ‘.txt’ extension

```
<object-class> <x_center> <y_center> <width> <height>
```

with and height of bounding box raining from 0.0 to 1.0
* float values relative to width and height of the image

center of bounding box raining from 0.0 to 1.0
* float values relative to width and height of the image

object identification in integer from 0 to (classes-1)

label file example: img1.txt for img1.jpg

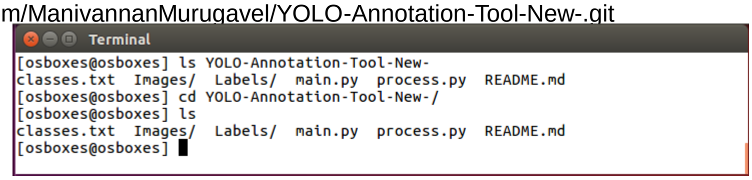
```
1 0.716797 0.395833 0.216406 0.147222
0 0.687109 0.379167 0.255469 0.158333
1 0.420312 0.395833 0.140625 0.166667
```

Labeling image files

- Yolo_mark
  - https://github.com/AlexeyAB/Yolo_mark
  - GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2
  - C++
  - It requires OpenCV.
- Yolo-Annotation-Tool
  - https://github.com/ManivannanMurugavel/YOLO-Annotation-Tool
  - https://github.com/ManivannanMurugavel/Yolo-Annotation-Tool-New-
  - Pyton
  - It requires conversion for Darknet format of labeling.
- BBox_Label
  - https://github.com/puzzledqs/BBox-Label-Tool
  - A simple tool for labeling object bounding boxes in images, implemented with Python Tkinter.
  - Pyton
  - It requires conversion for Darknet format of labeling.

YOLO-Annotation-Tool case (1/3)

- Download the tool
  - $ git clone https://github.com/ManivannanMurugavel/YOLO-Annotation-Tool-New-.git

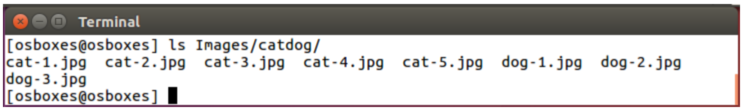


```
[osboxes@osboxes] ls YOLO-Annotation-Tool-New-
classes.txt  Images/  Labels/  main.py  process.py  README.md
[osboxes@osboxes] cd YOLO-Annotation-Tool-New-/
[osboxes@osboxes] ls
classes.txt  Images/  Labels/  main.py  process.py  README.md
[osboxes@osboxes] 
```

- Modify 'classes.txt' for your classes

classes.txt
cat
dog

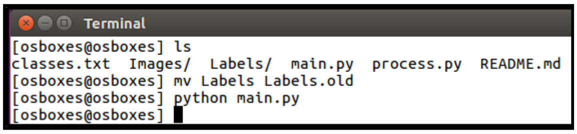
- Make a directory under "Images" directory and then put image files with ".jpg" extension in the new directory, say 'Images/catdog'.



```
[osboxes@osboxes] ls Images/catdog/
cat-1.jpg  cat-2.jpg  cat-3.jpg  cat-4.jpg  cat-5.jpg  dog-1.jpg  dog-2.jpg
dog-3.jpg
[osboxes@osboxes] 
```

YOLO-Annotation-Tool case (2/3)

- Rename ‘Labels’ directory if you need it, since it will be updated.
  - $ mv Labels Labels.old
- Run ‘main.py’ with Python
  - $ python main.py
- Enter the directory name
  - Directory under the ‘Images’
- Click ‘Load’
  - 1. choose class
  - 2. draw bounding box on the image
  - 3. click ‘Next’ button
  - 4. repeat from 1 for each image
- Label files will be created
  - See ‘Labels/catdog’



```
[osboxes@osboxes] ls
classes.txt  Images/  Labels/  main.py  process.py  README.md
[osboxes@osboxes] mv Labels Labels.old
[osboxes@osboxes] python main.py
[osboxes@osboxes] 
```

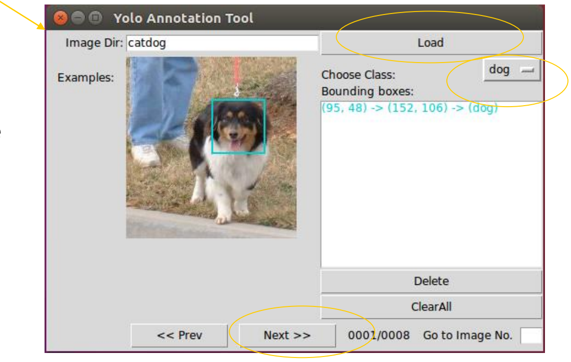


Image Dir: catdog

Choose Class: dog

Bounding boxes:

(95, 48) -> (152, 106) -> (dog)

0001/0008

Prepare train and test file

- You need to prepare two files say ‘tain.txt’ and ‘test.txt’, which contains file names to be used for train and test (verification).
- Modify ‘process.py’
  - ‘current_dir= ....’ ➔ “current_dir = “Images/catdog”
  - ‘percentage_test = 10’ ➔ portion to be used for verification if you want to change.
- Run ‘process.py’
  - $ python process.py
- Then ‘train.txt’ and ‘test.txt’ file will be generated.
  - ‘test.txt’ may not contain any file name when there are not sufficient image files.

㈜퓨쳐디자인시스템
34051 대전광역시 유성구 문지로 193, KAIST 문지캠퍼스, F723호
(042) 864-0211~0212 / contact@future-ds.com / www.future-ds.com

Future Design Systems, Inc.
Faculty Wing F723, KAIST Munji Campus, 193 Munji-ro, Yuseong-gu, Daejeon 34051, Korea
+82-042-864-0211~0212 / contact@future-ds.com / www.future-ds.com

FUTURE
Design Systems