

# 【AI技術活用入門】 はじめての物体検出 ～ YOLOに触れてみよう ～

第2回 2022/6/18(土)

# 全体スケジュール

■ 第1回：5/28(土) 14:00 – 15:00  
「YOLO の紹介」

■ 第2回：6/18(土) 14:00 – 15:00  
「環境構築とモデルの作成」

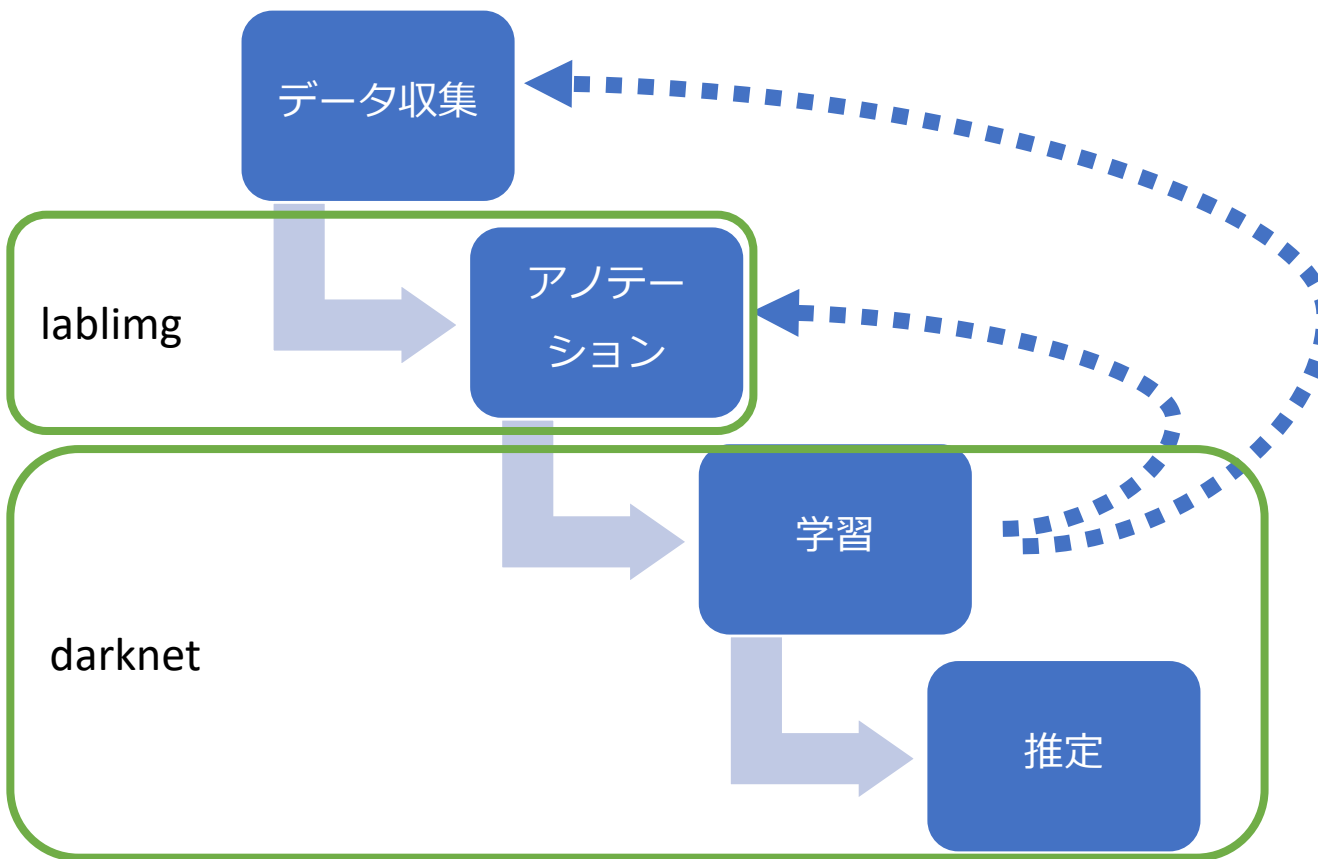
■ 第3回：7/23(土) 14:00 – 15:00  
「Ras Pi で物体検出」

■ 第4回：8/27(土) 14:00 – 15:00  
「最終発表会」

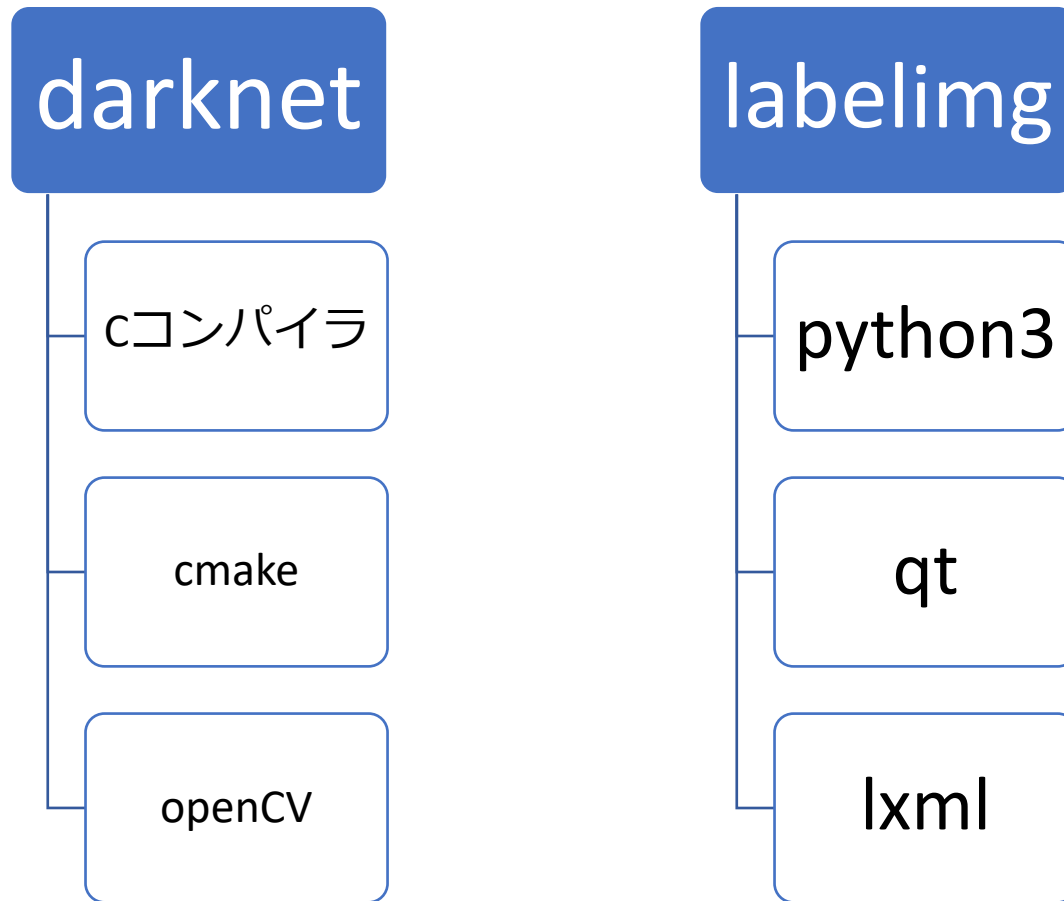
# 本日のスケジュール

- 作業内容の説明 ~ 14:05
- 環境構築 ~ 14:30
- アノテーション ~ 14:45
- 学習 ~ 14:50

# YOLOで独自モデルを作成するには



# darknetとlabelimgのビルド・実行に必要なソフト



GPU版を使う場合はCUDA等のインストールが必要だが本日は省略

# インストールするソフト一覧

## window

- MS build tools
- opencv
- python3

## mac

- xcode  
commandline  
tools
- homebrew
- opencv
- python3

## linux

- git
- gcc
- cmake
- libopencv-  
core
- python3

# インストール作業

手順書はgithubの以下のページから、お使いの環境にあったものを選んでください

<https://github.com/so5/YOLO-seminar/tree/main/install>

作業中に謎のエラーが出た、何もしていないのにPCが壊れたなどのトラブルが発生した方は挙手してください

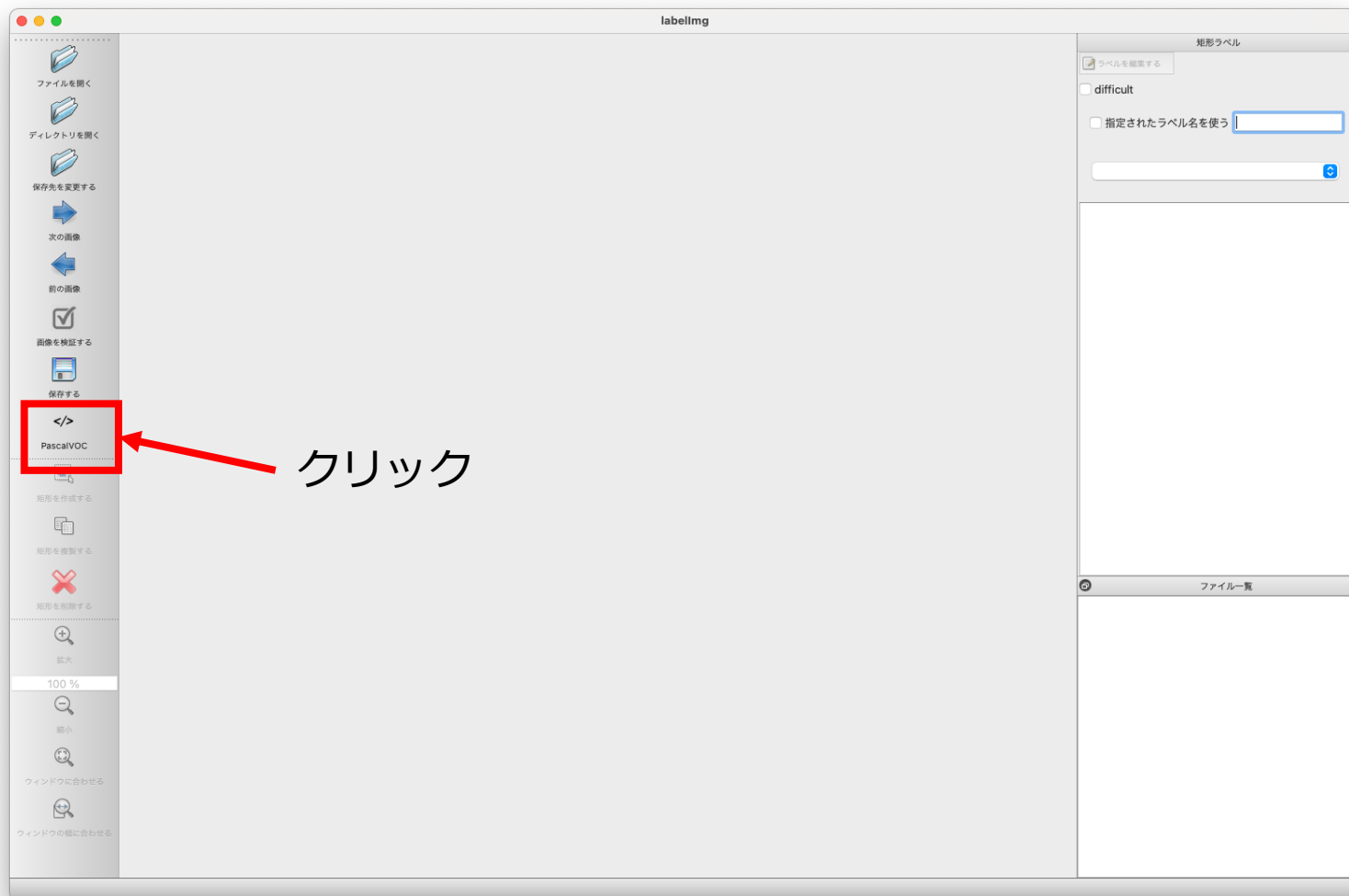
# アノテーション作業の準備

- アノテーションの対象にするjpeg画像を1つのディレクトリに集めてください
- predefined\_classes.txtを編集して使う予定のラベル名を1行に1つずつ記入してください(アノテーション作業中に追加することもできます)



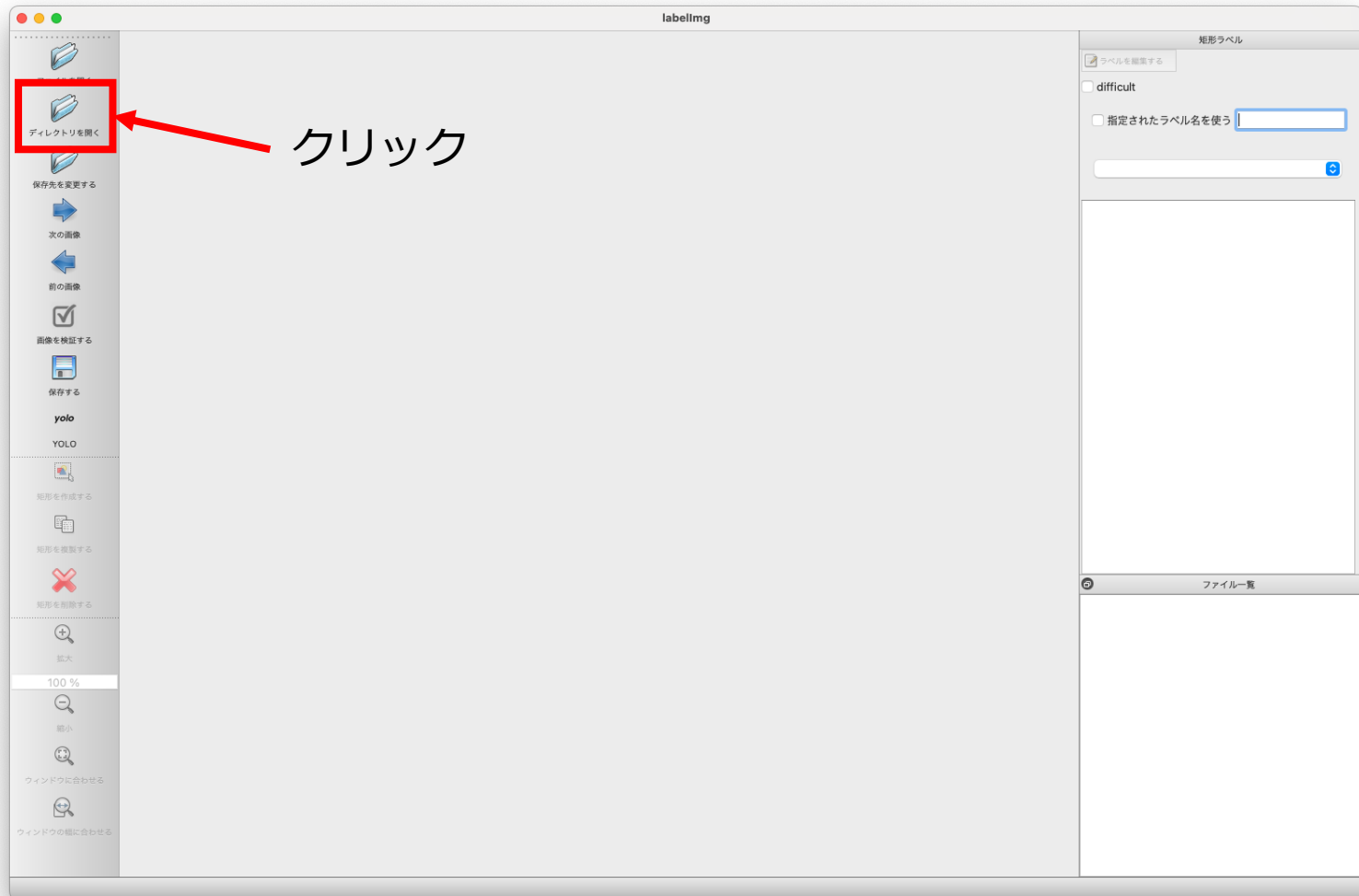
# アノテーション1

labellmgを起動して、データ形式をPascalVOCからYOLOに変更してください



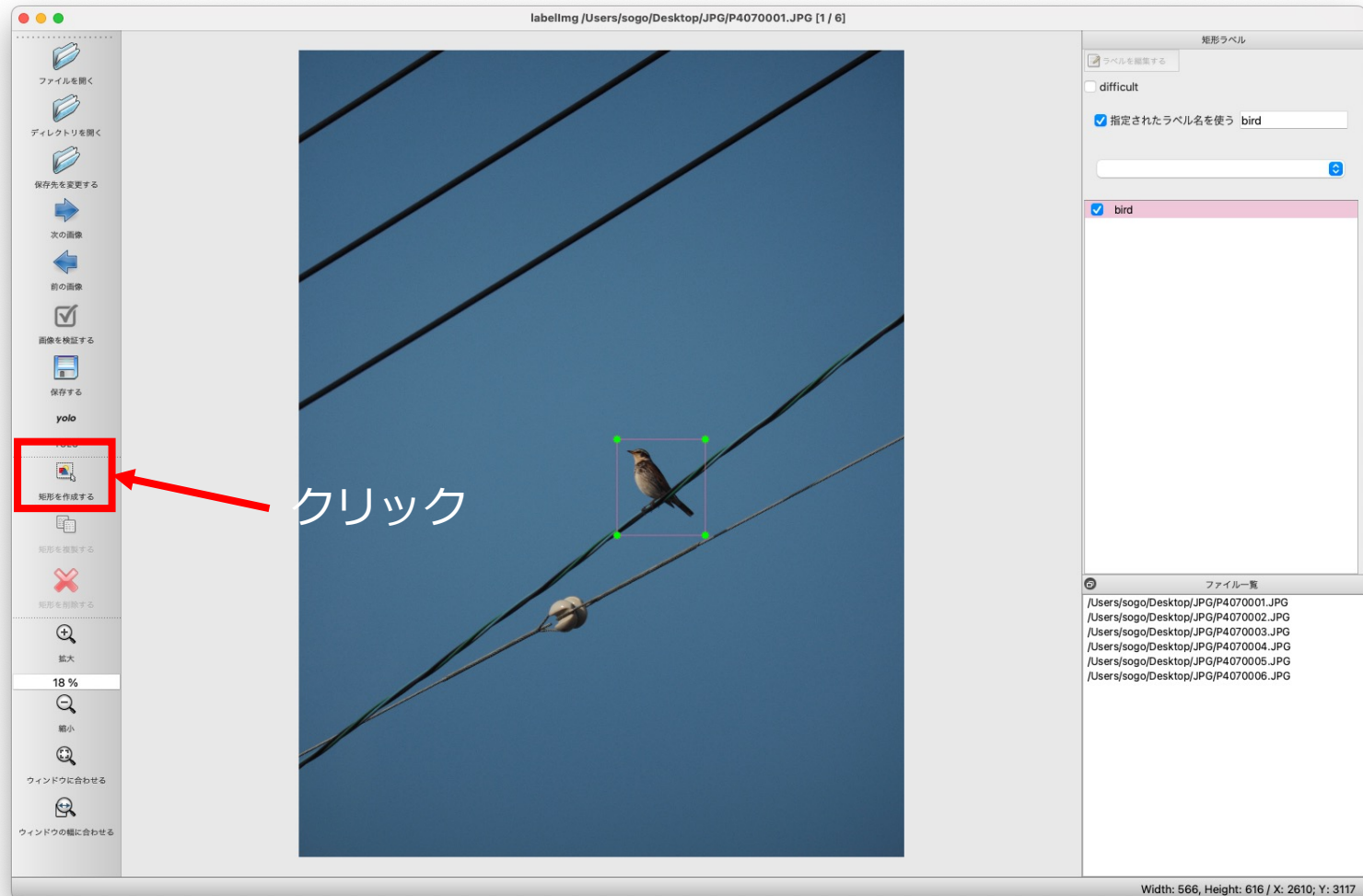
# アノテーション2

ディレクトリを開くをクリックして用意したディレクトリを開いてください。



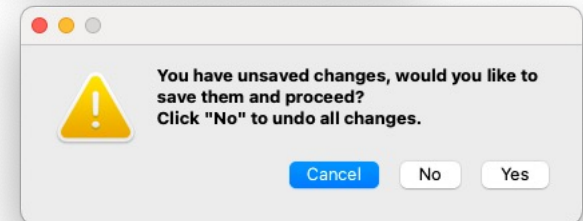
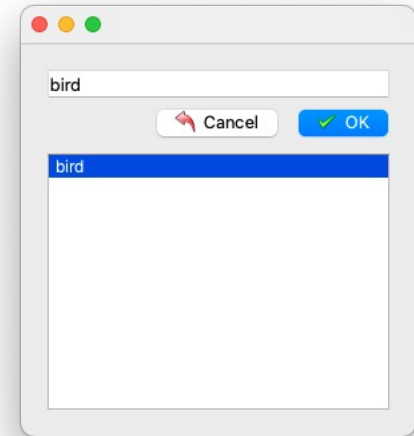
# アノテーション3

矩形を作成するをクリックし、対象領域の1頂点から対角線上の点までドラッグして領域を作成してください



# アノテーション4

- 領域を設定するとラベルを聞かれるので、事前に設定したラベルから選ぶか、新規のラベルを入力してください。
- 次の画像に進めようとするときsaveするか確認されるのでsaveしてください。
- この時、txtファイルではなくxmlやjsonファイルになっていた時は、YOLO形式が選択されていないので、キャンセルしてYOLOに変更してから保存してください



# アノテーション(補足)

- labellmgのショートカット
  - W 矩形領域を作る
  - Ctrl+S 保存
  - D 次の画像に進む
- 目安として1クラスあたり300ヶ所程度必要と言われています(諸説あり)
- 実際に物体検出のモデルを学習する時には、公開されているアノテーション済のデータセットや、アノテーションサービスの活用も検討してください。

# 学習前の準備

1. weightファイル(yolov4-tiny.conv.29)をダウンロードしてください  
[https://github.com/AlexeyAB/darknet/releases/download/darknet\\_yolo\\_v4\\_pre/yolov4-tiny.conv.29](https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29)
2. darknet/cfg/yolov4-tiny-custom.cfg を コ ピ ー し て yolo-obj.cfgを作ってください
3. yolov-obj.cfgを編集して次の行を変更してください
  - max\_batches: クラス数x2000ただし、クラス数が3以下の時は6000また、学習データの画像ファイル数が6000を超える場合は画像数に設定する
  - classes: アノテーションを設定したクラスの数 (2ヶ所)
  - filters: (クラス数+5)x3  
[yolo]と書かれた行の直前のブロックにあるものだけを変更すること(2ヶ所)

## 学習前の準備その2

4. classes.txtをコピーしてobj.namesを作成

5. 次の内容でobj.dataファイルを作成

```
classes = 2
```

```
train = train.txt
```

```
valid = test.txt
```

```
names = obj.names
```

```
backup = backup
```

6. train.txtファイルを作成し学習に使うデータのファイル名を、darknetの実行ファイルからの相対パスで記載する

7. test.txtファイルを作成し、train.txtに記載しなかったデータのファイル名を記載する

8. backupディレクトリを作成

# 独自モデルを使った学習

次のコマンドを入力します

```
../darknet/darknet detector train obj.data yolo-obj.cfg yolov4-tiny.conv.29
```

windowsの場合はdarknet.exe

データ不足、計算機性能の不足といった理由によりたぶんまともに学習できないので、適当なところでCtrl-Cで止めてください。



# 学習結果を利用した物体検出

- 学習が終了すると weights ファイル (backup/yolo-obj\_final.weights) が生成されます
- これを用いて物体検出を行なうためには次のコマンドを入力してください

```
../darknet/darknet detector test obj.data yolo-obj.cfg backup/yolo-obj_final.weights ファイル名
```

# Appendix1 既存のデータセット

- こちらのURLを参照のこと

<https://github.com/arXivTimes/arXivTimes/tree/master/datasets>

# Appendix 2 FiftyOne

著名な画像データセットのMS COCOや、Google Open Imageをダウンロードしようとする、巨大すぎて取扱いが非常に難しくなります。

これらのデータセット(画像+アノテーション情報+ $\alpha$ )を簡単に扱うためのツールとして、FiftyOneというものがあります。

<https://voxel51.com/docs/fiftyone/index.html>

日本語の詳しい解説はこちら

<https://qiita.com/RyoWakabayashi/items/ffcf21558855f6d5a9be>