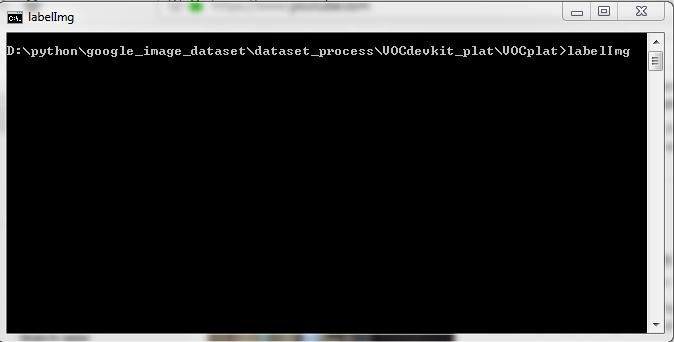
**Panduan Penggunaan YOLO untuk deteksi**

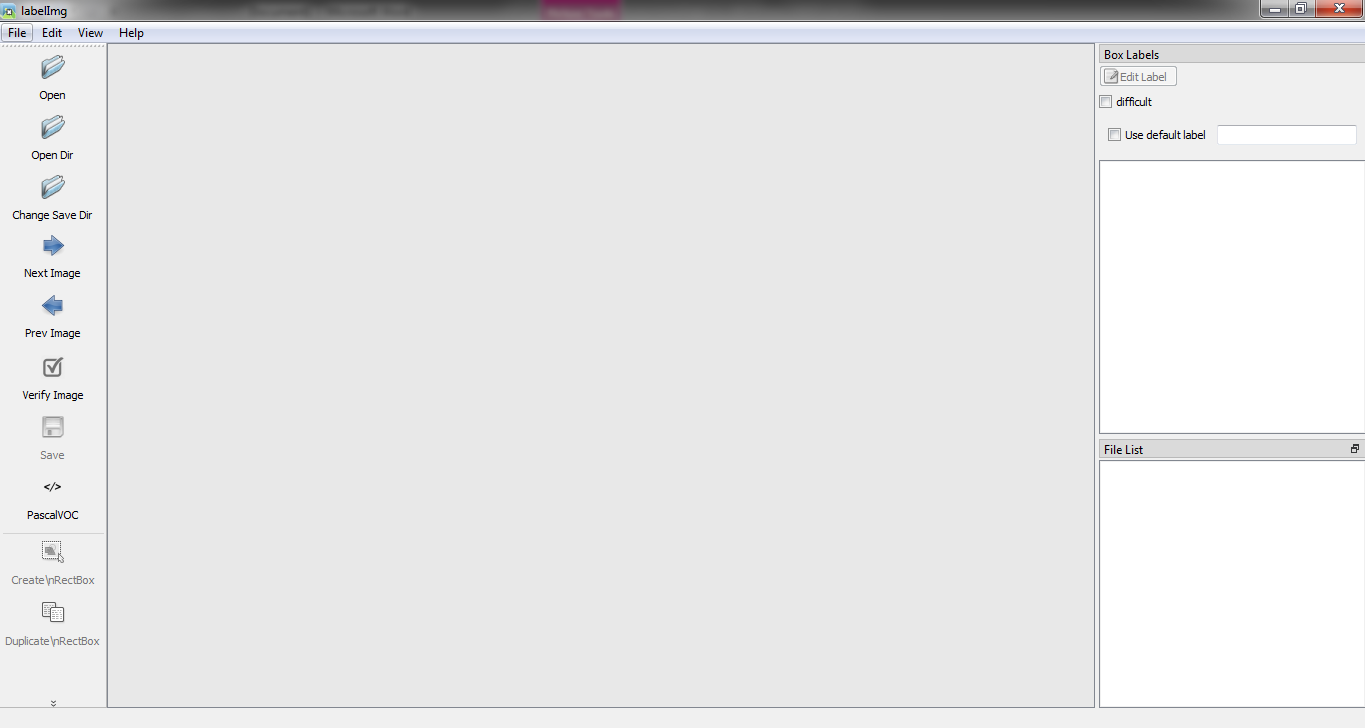
Pada panduan ini akan dijelaskan cara menggunakan YOLO untuk deteksi objek mulai dari proses pembuatan dataset, training dan inference.

**Pembuatan Dataset**

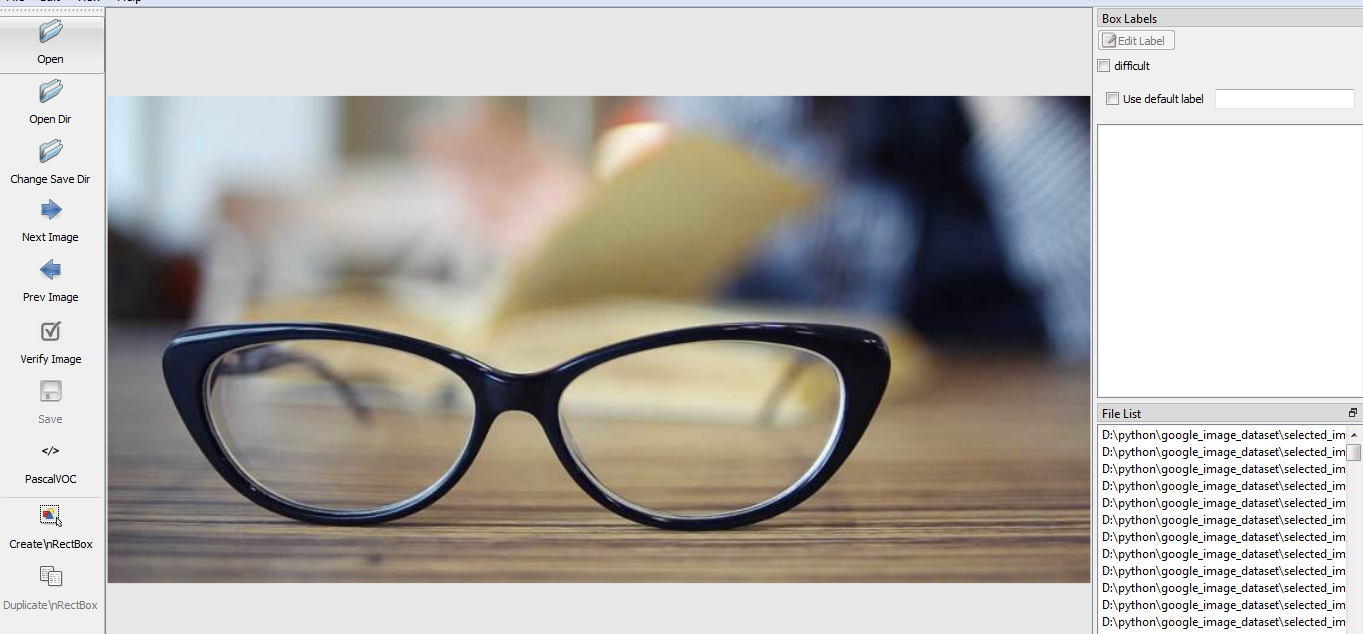
Format yang digunakan adalah dataset VOC , software yang akan digunakan untuk mempermudah proses pelabelan adalah labelImg.

1. Download dan install labelImg dari <https://github.com/tzutalin/labelImg>
2. Jalankan lewat cmd (Ikuti github untuk cara penggunaan yang lebih sesuai dengan perangkat)

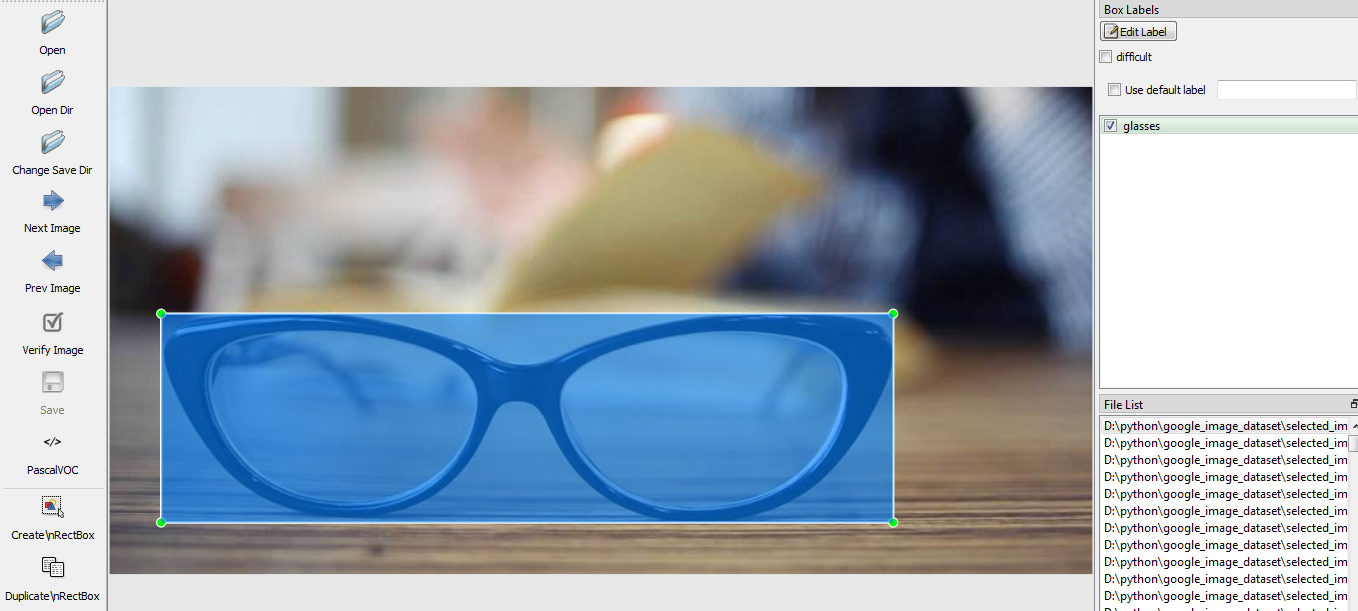




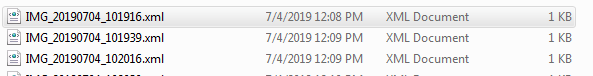
1. Pilih **Open Dir** dan pilih folder yang berisikan gambar-gambar yang akan diberikan label



1. Pilih **Change Save Dir** dan pilih folder untuk tempat annotasi
2. Untuk memberikan label pilih **Create\nRectBox**, tandai objek yang ingin dilabeli dengan dengan memberi kotak pada objek. Lalu **Save** (View > Auto Save Mode bisa digunakan jika ingin save secara otomatis)



1. Jika berhasil maka akan muncul file xml di folder khusus anotasi



Contoh isi file xml:

<annotation>

    <folder>JPEGImages</folder>

    <filename>IMG\_20190704\_101916.jpg</filename>

    <path>D:\python\google\_image\_dataset\dataset\_process\VOCdevkit\_plat\VOCplat\JPEGImages\IMG\_20190704\_101916.jpg</path>

    <source>

        <database>Unknown</database>

    </source>

    <size>

        <width>4160</width>

        <height>3120</height>

        <depth>3</depth>

    </size>

    <segmented>0</segmented>

    <object>

        <name>plat</name>

        <pose>Unspecified</pose>

        <truncated>0</truncated>

        <difficult>0</difficult>

        <bndbox>

            <xmin>1418</xmin>

            <ymin>2062</ymin>

            <xmax>2822</xmax>

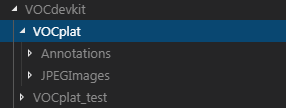
            <ymax>2481</ymax>

        </bndbox>

    </object>

</annotation>

1. Masukkan semua dataset ke folder VOCdevkit dan susun folder seperti ini:



Folder plat berisi dataset untuk training dan folder plat\_test berisi dataset untuk test perbandingan yang disarankan adalah 80:20

**Proses Training**

Untuk tambahan informasi dapat diperoleh di <https://github.com/qqwweee/keras-yolo3>

Ada beberapa persiapan yang perlu dilakukan sebelum dilakukan training.

1. Program python yang akan digunakan bisa diproleh dari <https://github.com/BillyGun27/license_plate_YOLO>
2. Di github juga contoh jupyter notebook **plat\_train\_custom\_mobilenet\_yolo.ipynb** yang dapat dijalankan pada google colab. Untuk dijalankan pada perangkat sendiri bisa dihilangkan bagian yang tidak diperlukan.
3. Copy folder dataset (VOCdevkit) ke dalam folder program.

Buat file list data gambar dan label:

Dataset perlu dibuat dalam bentuk list dalam file txt agar lebih mudah diolah

1. Buka **voc\_custom\_annotation.py**
2. Ubah variable sesuai dengan nama folder dan kelas objek.

sets=['plat','plat\_test']

classes = ["plat"]

1. Jalankan di cmd:

python voc\_custom\_annotation.py

jika berhasil akan muncul file **plat.txt** dan **plat\_test.txt**

Buat file nilai anchor:

Anchor beris nilai awal untuk referensi berbagai ukuran w , h objek yang akan dideteksi. Nilai diperoleh dengan menggunakan algoritma k-means pada dataset.

1. Buka **kmeans.py**
2. Ubah variable sesuai dengan file txt dan jumlah cluster

cluster\_number = 9 #9 utk YOLO dan 6 untuk Tiny YOLO

filename = "plat.txt" #plat.txt , plat\_test.txt

1. Jalankan di cmd:

python kmeans.py

Jika berhasil maka akan muncul **yolo\_anchors.txt**

1. Pindahkan file **yolo\_anchors.txt** ke folder **anchors** agar lebih rapi
2. Jika pada hasil deteksi objek ukuran box terlalu kecil atau terlalu besar, pada saat proses pelabelan kotak dapat diubah ukurannya atau bisa memodifikasi nilai anchor pada file sesuai dengan keperluan.

Buat file kelas:

File untuk menyimpan nama kelas objek yang akan dideteksi.

1. Pada folder **class,** **buat file txt** dengan nama group kelas yang akan dideteksi. Contoh: **plat\_classes.txt**
2. Masukkan nama-nama kelas yang akan dideteksi (urutan kelas harus sama dengan pengaturan di variable **classes** pada **voc\_custom\_annotation.py**)

Training :

(panduan pretrain dan training yang lebih jelas dapat dibaca di <https://github.com/qqwweee/keras-yolo3#training> )

1. Atur Konfigurasi di train.py

Ada tiga model yang dapat digunakan YOLO, Mobilenet YOLO dan YOLO. Model dapat dilihat folder model

yolo3.py (yolo dan tiny yolo)

small\_mobilenets2.py (mobilenet yolo)

##Pilih model yang akan digunakan (pilih salah satu)

#from model.yolo3 import yolo\_body

from model.small\_mobilenets2 import yolo\_body

#from model.yolo3 import tiny\_yolo\_body

def \_main():

epoch\_end\_first = 50#75

epoch\_end\_final = 100#200

model\_name = 'plat\_yolo' # nama model

log\_dir = 'logs/000/'

## pretrain

## contoh model\_data/yolo\_weights.h5, model\_data/tiny\_yolo\_weights.h5

## mobilenet '' kosong karena pretrain otomatis dari keras

model\_path = ''

train\_path = 'plat.txt' #dari voc\_custom\_annotation.py

val\_path = 'plat\_test.txt' #dari voc\_custom\_annotation.py

#test\_path = '2007\_test.txt'

classes\_path = 'class/plat\_classes.txt' #dari folder class

anchors\_path = 'anchors/plat\_yolo\_anchors.txt' #dari kmeans.py

1. Jalankan di cmd:

python train.py

1. Hasil Training dapat diperoleh di folder **logs/000/**

**Proses Deteksi (Inference)**

Pengaturan untuk deteksi objek dan model yang digunakan di yolo.py

##Pilih model yang akan digunakan (pilih salah satu)

#from model.yolo3 import yolo\_body

from model.small\_mobilenets2 import yolo\_body

#from model.yolo3 import tiny\_yolo\_body

model\_name = 'plat\_yolo\_trained\_weights\_final.h5'

class YOLO(object):

\_defaults = {

"model\_path": 'model\_data/'+model\_name,#yolo.h5,trained\_weights\_final.h5

"classes\_path": 'class/plat\_classes.txt',#voc\_classes.txt,coco\_classes.txt

"anchors\_path": 'anchors/plat\_yolo\_anchors\_test.txt',#yolo\_anchors.txt

"score" : 0.1,

"iou" : 0.45,

"model\_image\_size" : (416, 416),#416,288,224,160,96 kelipatan 32

"gpu\_num" : 1,

}

Test Gambar

1. Ubah direktori gambar di yolo\_image.py

def detect\_img(yolo):

img = "test\_data/IMG\_20190704\_112805.jpg"

2. Jalankan cmd

python yolo\_image.py

Test Video

1. Jalankan cmd

python yolo\_video.py --input test\_data/car.mp4