

yolo

[yolov4源码复现](#)

[windows](#)

[linux](#)

[yolov4使用自己的数据集训练](#)

[环境搭建](#)

[数据集准备](#)

[训练](#)

[YOLOv5 for ROS](#)

[下载三个包](#)

[detection_msgs](#)

[yolov5](#)

[yolov5_ros](#)

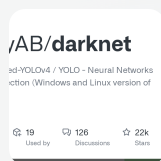
[发布tf](#)

[主程序](#)

[EKF滤波](#)

yolov4源码复现

源码和权重文件下载：

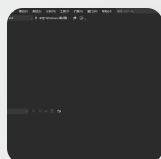


GitHub – AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO – Neural ...

YOLOv4 / Scaled-YOLOv4 / YOLO – Neural Networks for Object Detection (Windows and Linux...

GitHub

运行：



YOLOV4在windows和linux系统中的使用（非GPU版）_yolo安装到linux一定要...

文章浏览阅读1.2k次。文章目录前言一、Windows环境下运行YOLOV41.运行环境2.下载源代码3.将工...

https://blog.csdn.net/feifei890128/article/details/113407467?ops_request_misc=&request_id=&...

windows

visual studio编译，命令行运行darknet_no_gpu.exe detect cfg/yolov4.cfg yolov4.weights 1.jpg。
darknet_no_gpu.exe由visual studio生成；yolov4.cfg在源码中可以直接下载；yolov4.weights在https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.weights中下载；1.jpg自己找网上的照片。

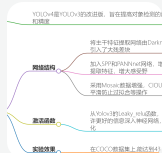
linux

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

darknet文件夹内，make

使用：./darknet detect cfg/yolov4.cfg yolov4.weights 1.jpg

yolov4使用自己的数据集训练



使用Yolov4训练自己的数据集(20240415版) –CSDN博客

文章浏览阅读2.3k次，点赞34次，收藏47次。本实验使用了Yolov4和PyTorch训练了水果数据集，实...

https://blog.csdn.net/Gltu_java/article/details/137782694

环境搭建

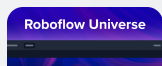
conda create -n yolov4 python=3.6

pip install -r requirements.txt

数据集准备

(需要xml格式)

下载开源数据集：

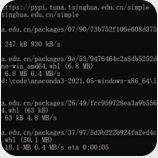


Roboflow Universe: Open Source Computer Vision Community

Download free, open source datasets and pre-trained computer vision machine learning models.

Roboflow

labelimg自制数据集：



目标检测---利用labelimg制作自己的深度学习目标检测数据集-CSDN博客

文章浏览阅读10w+次，点赞528次，收藏3k次。看了网上的xml转txt的博客很多上来就给代码，关于怎...
https://blog.csdn.net/didiaopao/article/details/119808973?ops_request_misc=%257B%2522req...

用一个python脚本，拍摄不同光照的视频，自动截取，手工标注。

训练



GitHub – bubbliiiing/yolov4-pytorch: 这是一个YoloV4-pytorch的源码，可以...

这是一个YoloV4-pytorch的源码，可以用于训练自己的模型。 . Contribute to bubbliiiing/yolov4-pyt...
GitHub

修改cls_class.txt

运行voc_annotation.py

修改train.py中预训练文件model_path=""; UnFreeze_Epoch=300

运行train.py

从log文件夹中复制best weights到model_data中

修改yolo.py中model_path和onnx_path为自己训练的权重文件

运行predict.py

YOLOv5 for ROS

下载三个包

下载这三个包：

detection_msgs

```
1 cd ~/mavros_ws/src
2 git clone https://github.com/mats-robotics/detection_msgs.git
3 cd ~/mavros_ws
4 catkin_make_isolated --source src/detection_msgs --build src/detection_msgs/build
```

yolov5

```
1 cd ~/mavros_ws/src
2 git clone https://github.com/ultralytics/yolov5.git
3 cd ~/mavros_ws/src/yolov5
4 pip install -r requirements.txt
```

yolov5_ros

```
1 cd ~/mavros_ws/src
2 git clone https://github.com/HuaYuXiao/yolov5_ros.git
3 cd ~/mavros_ws
4 catkin_make_isolated --source src/yolov5_ros --build src/yolov5_ros/build
```

替换这几个文件：

 [detect.py](#)

 [cuadc.yaml](#)

 [yolov5.xml](#)

在合适的位置加上模型文件，并修改文件名，记得修改launch文件里的参数：

```
1 <arg name="weights" default="$(find yolov5_ros)/yolov5s-cuadc-2-196.pt" />
2 <arg name="data" default="$(find yolov5_ros)/config/cuadc.yaml" />
3 <arg name="confidence_threshold" default="0.75" />
4 <arg name="input_image_topic" default="/monocular/image_raw" />
```

下载模型文件到yolov5_ros文件夹下：

发布tf

从base_link到下视相机

x, y, z, yaw, pitch, roll

改成自己测出来的（里面的不能直接用，要改）

```
1 <node pkg="tf" type="static_transform_publisher" name="base2monocular"
2   args="x y z yaw pitch roll base_link monocular_link 100"/>
```

主程序

25、26、27、30行改成精确的内参：

```
1 # Updated intrinsic matrix and distortion coefficients
2 self.K = np.array([[fx, 0, u0],
3                    [0, fy, v0],
4                    [0, 0, 1]])
5
6 # Distortion coefficients
7 self.dist_coeffs = np.array([k1, k2, k3, p1, p2])
```

96、98、100行修改成实际测得的半径。

```
1 if 27.5 <= radius <= 32.5:
2     self.world_coords_pub1.publish(point_msg)
3 elif 22.5 <= radius <= 27.5:
4     self.world_coords_pub2.publish(point_msg)
5 elif 17.5 <= radius <= 22.5:
6     self.world_coords_pub3.publish(point_msg)
```

这个节点会直接将圆筒位置发布成geometry_msgs::PoseStamped，1最大，2次之，3最小。

```
1 self.world_coords_pub1 = rospy.Publisher("/yolov5/world_coords1", PointStamped, queue_size=10)
2 self.world_coords_pub2 = rospy.Publisher("/yolov5/world_coords2", PointStamped, queue_size=10)
3 self.world_coords_pub3 = rospy.Publisher("/yolov5/world_coords3", PointStamped, queue_size=10)
```

如果圆筒从画面中消失，会保留之前的数据。

EKF滤波

 pos_filter.py

这个节点会分别针对3个圆筒位置进行滤波，并依次发布到以下话题：

```
1 self.pub1 = rospy.Publisher('/yolov5/filtered_coords1', PointStamped, queue_size=10)
2 self.pub2 = rospy.Publisher('/yolov5/filtered_coords2', PointStamped, queue_size=10)
3 self.pub3 = rospy.Publisher('/yolov5/filtered_coords1', PointStamped, queue_size=10)
```