人脸识别

[KGitHub - ageitgey/face\_recognition: The world's simplest facial recognition api for Python and the command line](https://kkgithub.com/ageitgey/face_recognition)

GitHub上有不少最新的人脸识别项目，以下为你介绍几个：

- SeetaFace6：由中科视拓宣布开放，包含人脸检测、关键点定位、人脸识别等基本能力，还增加了活体检测、口罩检测等功能。项目开放三个版本模型，分别是通用人脸识别、带口罩人脸识别、通用人脸识别（小），以应对不同应用需求。项目地址为https://github.com/SeetaFace6Open/index。

- CompreFace：一个免费、开源的人脸识别系统，后端采用Java编写，基于FaceNet和InsightFace实现。提供人脸识别、检测、验证、头部姿势检测等REST API服务，支持Docker部署。具有灵活性部署、数据安全自主、零机器学习门槛等优势，只需一条Docker命令即可快速启动服务。

- FaceRecognitionDotNet：一个C#面部识别API项目，在GitHub上有1.2k星标。它是face\_recognition的C#移植版本，使用了OpenCVSharp和face\_recognition开源库，提供了NuGet包，方便集成到项目中，支持Windows、Linux和macOS跨平台使用。

此外，还有一些基于Python的项目，如利用deepface库和opencv实现实时面部情绪检测的项目facial - emotion - recognition等。

CompreFace有多个相关网址，具体如下：

- 官方网址：https://exadel.com/solutions/compreface/

- GitHub地址：https://github.com/exadel-inc/CompreFace

- 下载地址：https://github.com/exadel-inc/CompreFace/releases/download/v1.2.0/CompreFace\_1.2.0.zip（以v1.2.0版本为例，可前往GitHub releases页面获取更多版本）

- 本地访问地址：http://localhost:8000/login（部署后在本地浏览器访问的地址）

物体识别

GitHub上有许多优秀的物体识别项目，以下是一些常见的介绍：

- Object\_Detection\_Tracking：由JunweiLiang开发，是一个专注于物体检测与跟踪的开源项目。它结合深度学习技术，支持多种模型和算法，适用于视频监控、自动驾驶等场景。项目依赖TensorFlow、OpenCV等，可通过 git clone https://github.com/JunweiLiang/Object\_Detection\_Tracking.git 克隆到本地。

- YOLO-Object-Detection：由Garima13a开源，提供了YOLO算法的基本实现和应用示例。具有实时性强、精度高的特点，在COCO数据集上mAP可达57.9%。项目还提供了Jupyter Notebook示例，方便用户快速上手，可用于安防监控、工业检测等领域，地址为<https://github.com/Garima13a/YOLO-Object-Detection。>

git clone <https://github.com/Garima13a/YOLO-Object-Detection.git>

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

或者

git clone [https://hub.fastgit.xyz/pjreddie/darknet.git](https://github.com/pjreddie/darknet.git)

cd darknet

make

./darknet

pip install opencv-python matplotlib

git clone <https://kkgithub.com/llSourcell/YOLO_Object_Detection.git>

cd YOLO\_Object\_Detection

pip install cython

pip install -e .

[KGitHub - ultralytics/yolov5：YOLOv5 🚀 在 PyTorch > ONNX > CoreML > TFLite](https://kkgithub.com/ultralytics/yolov5)

[从PyTorch Hub 加载YOLOv5 -Ultralytics YOLO 文档](https://docs.ultralytics.com/zh/yolov5/tutorials/pytorch_hub_model_loading/)

pip install -r <https://raw.githubusercontent.com/ultralytics/yolov5/master/requirements.txt>

import torch

# Model

model = torch.hub.load("ultralytics/yolov5", "yolov5s")

# Image

im = "https://ultralytics.com/images/zidane.jpg"

# Inference

results = model(im)

results.pandas().xyxy[0]

# xmin ymin xmax ymax confidence class name

# 0 749.50 43.50 1148.0 704.5 0.874023 0 person

# 1 433.50 433.50 517.5 714.5 0.687988 27 tie

# 2 114.75 195.75 1095.0 708.0 0.624512 0 person

# 3 986.00 304.00 1028.0 420.0 0.286865 27 tie

result.show()

- YOLOv10: Real - Time End - to - End Object Detection：是YOLO系列的最新一代，旨在推进YOLO的性能效率边界。该项目提供了无需非极大值抑制（NMS）的一致双重分配，以实现竞争性能和低推理延迟。YOLO系列模型轻量化、结构简洁，YOLOv10适合在校学生和初学者学习计算机视觉算法。

- DarkNet\_ROS：这是一个用于ROS的实时目标检测项目，可在GPU和CPU上使用YOLO（V3）。该项目已在ROS Melodic和Ubuntu 18.04下进行了测试，能够检测包括VOC和COCO数据集中的预训练类别，也可使用自己的检测对象创建网络，适合有ROS开发需求的用户。

- YoloDotNet v2.1：一个基于C#和.NET 8的实时物体检测框架，集成了Yolov8~Yolov11模型，通过ML.NET和ONNX运行时实现高效物体检测，支持GPU加速。它不仅支持传统物体检测，还涵盖分类、OBB检测等多种功能，提供简洁API和丰富示例代码，跨平台性好，适用于智能监控、医疗影像分析等场景，项目地址为https://github.com/NickSwardh/YoloDotNet。

工衣识别

GitHub上有工衣识别的项目，以下是一些相关项目介绍：

- reflective - clothes - detect：<https://github.com/gengyanlei/reflective->clothes-detect，开源了工作服（反光衣）检测数据集和基于YOLOv5s的预训练模型，可用于施工区域或危险区域的人员检测，数据集包含不同颜色、多种场景下的1083张图像及标注。

1.下载

<https://github.com/gengyanlei/reflective-clothes-detect-yolov5/blob/master/yolov5/best.pt>

也可在里面的百度云下载

[reflective.tar.gz\_免费高速下载|百度网盘-分享无限制](https://pan.baidu.com/s/1_Ei9bYmUpa-8q-hXZk1u8w)

将best.pt改成yolov5reflectiveclothes.pt放在yolov5文件夹下

git clone <https://kkgithub.com/ultralytics/yolov5.git>

2.改hubconf.py

加以下代码

def yolov5reflectiveclothes(pretrained=True, channels=3, classes=80, autoshape=True, \_verbose=True, device=None):

"""

Creates the YOLOv5-xlarge-P6 model with options for pretraining, number of input channels, class count, autoshaping,

verbosity, and device selection.

Args:

pretrained (bool): If True, loads pretrained weights into the model. Default is True.

channels (int): Number of input channels. Default is 3.

classes (int): Number of model classes. Default is 80.

autoshape (bool): If True, applies YOLOv5 .autoshape() wrapper to the model. Default is True.

\_verbose (bool): If True, prints all information to the screen. Default is True.

device (str | torch.device | None): Device to use for model parameters, can be a string, torch.device object, or

None for default device selection. Default is None.

Returns:

torch.nn.Module: The instantiated YOLOv5-xlarge-P6 model.

Example:

```python

import torch

model = torch.hub.load('ultralytics/yolov5', 'yolov5x6') # load the YOLOv5-xlarge-P6 model

```

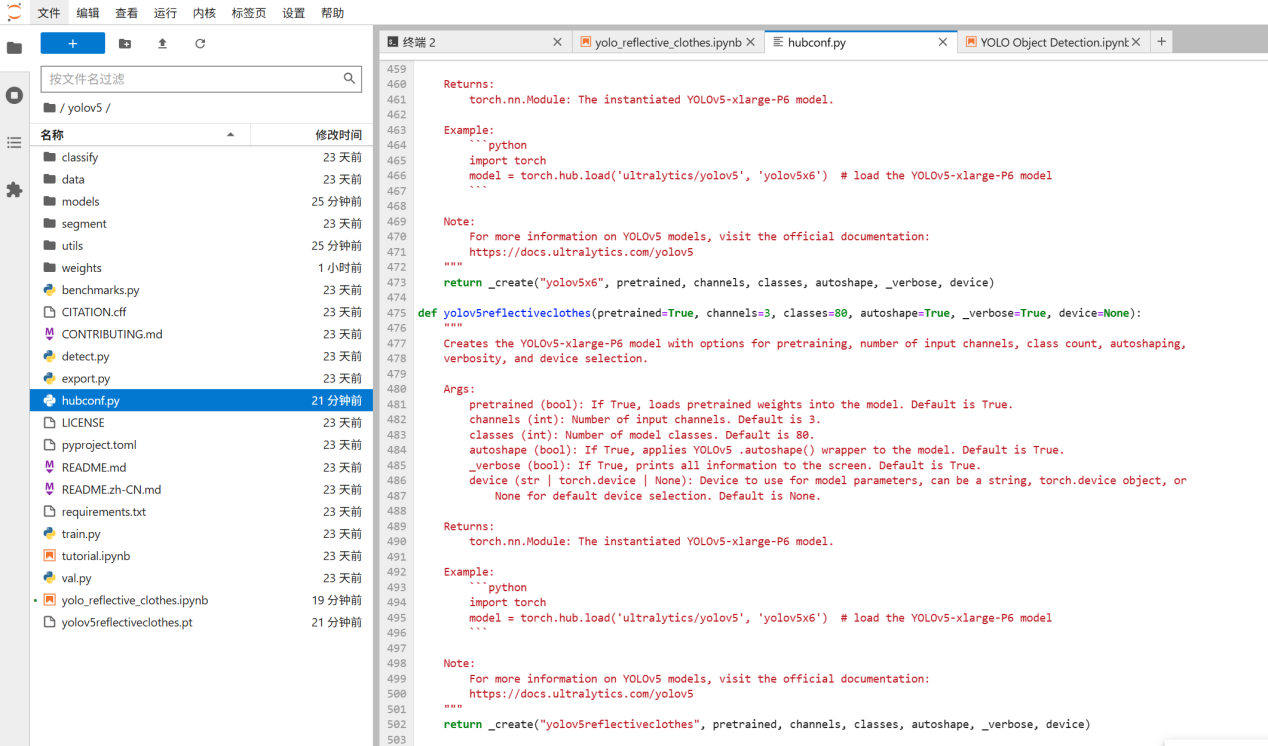
Note:

For more information on YOLOv5 models, visit the official documentation:

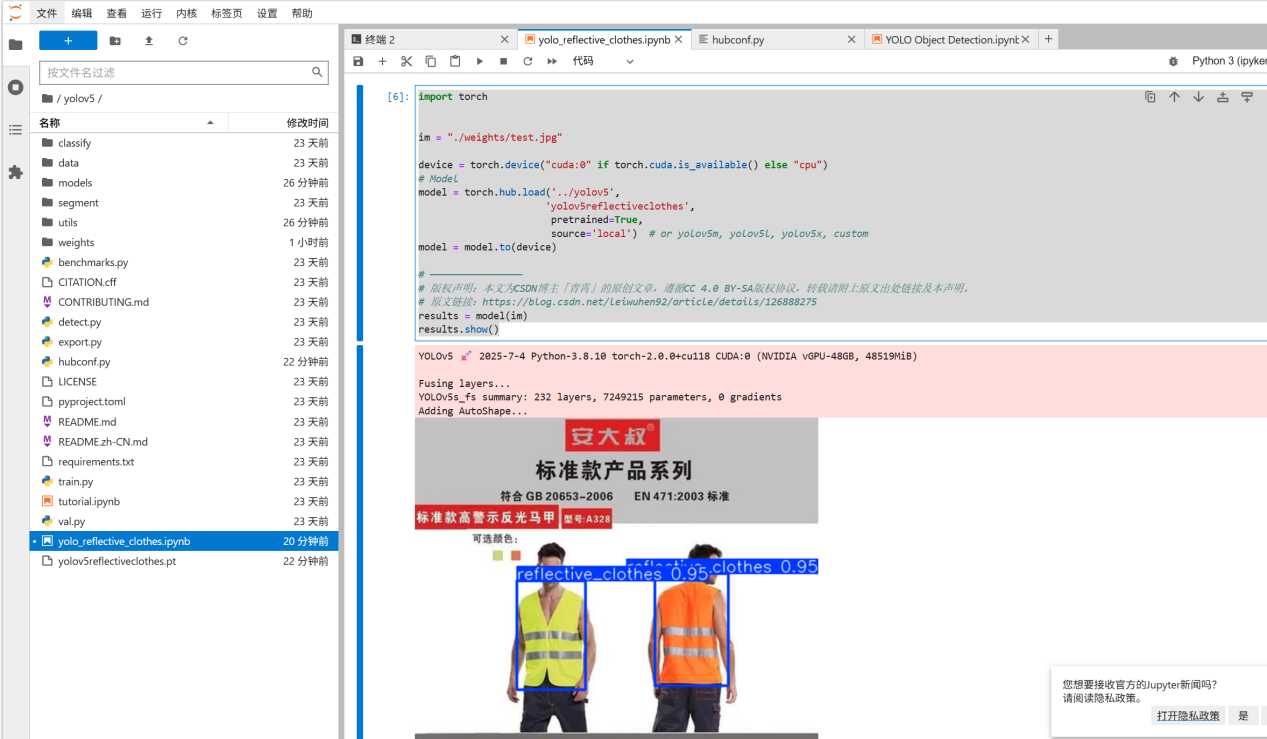
https://docs.ultralytics.com/yolov5

"""

return \_create("yolov5reflectiveclothes", pretrained, channels, classes, autoshape, \_verbose, device)



3.检测



import torch

im = "./weights/test.jpg"

device = torch.device("cuda:0" if torch.cuda.is\_available() else "cpu")

# Model

model = torch.hub.load('../yolov5',

'yolov5reflectiveclothes',

pretrained=True,

source='local') # or yolov5m, yolov5l, yolov5x, custom

model = model.to(device)

results = model(im)

results.show()



- CompreFace：https://github.com/exadel - inc/CompreFace，虽然主要是人脸识别系统，但也可用于一些包含工服识别等相关场景的人员检测等应用，后端采用Java编写，提供REST API服务，支持Docker部署。

- RZV2L\_AiLibrary：https://github.com/Ignitarium - Renesas/RZV2L\_AiLibrary，瑞萨基于RZ/V2L AI MPU开发的AI应用程序库，其中包含工服工帽检测功能，可识别工作时是否穿戴了工作背心和安全帽。

- 基于YOLOv5的反光衣与头盔检测项目：未明确给出仓库主链接，但根据CSDN博客介绍，其GitHub仓库提供了基于YOLOv5的反光衣与头盔检测完整解决方案，包括数据集准备、模型训练及推理测试等环节，针对NVIDIA Jetson Xavier NX和Jetson Nano等嵌入式平台进行了优化，能在Jetson Xavier NX上达到33帧每秒的速度。

区域入侵识别

[uttej2001/Image-based-Human-Fall-Detection: Here I integrated the YOLOv5 object detection algorithm with my own created dataset which consists of human activity images to achieve low cost, high accuracy, and real-time computing requirements](https://kkgithub.com/uttej2001/Image-based-Human-Fall-Detection)

git clone <https://kkgithub.com/uttej2001/Image-based-Human-Fall-Detection.git>

GitHub上有不少区域入侵识别项目，以下是一些典型的项目介绍：

- object-tracking-line-crossing-area-intrusion：https://github.com/yas-sim/object-tracking-line-crossing-area-intrusion，是一个基于深度学习的对象跟踪和区域入侵检测项目。结合了对象检测模型和再识别模型，可在视频中找到并跟踪对象，检测其是否跨越预设线或进入特定区域。项目使用OpenCV进行图像处理和显示，对象检测和再识别模型基于TensorFlow实现。

- yolov5+bytrack入侵检测相关项目：基于YOLOv5和ByteTrack实现，可实现行人跟踪并计算入侵人数，或保存入侵者图像。通过准备与检测图像同样大小的属性板，指定禁入区域和安全区的值，利用行人目标中心点坐标检查是否进入禁入区，若进入则将其ID加入入侵列表。

- Wazuh：https://github.com/wazuh/wazuh，是一个开源的安全监控平台。其核心是轻量级端点代理，可部署在需监控的系统上，实时收集安全数据并传输到中央管理服务器。通过规则引擎能识别潜在安全威胁和异常行为，可用于检测恶意软件、rootkits、未授权网络连接等，也能监控文件系统变动，在区域入侵检测方面有广泛的应用场景。

- N3tstat IDS：https://github.com/NetSPI/N3tstat，是一个轻量级的端点检测和响应（EDR）框架，专门用于实时高级持久性威胁（APT）检测，提供了传统入侵检测系统（IDS）解决方案所不具备的APT检测功能，可帮助网络安全分析人员和网络防御者执行主动威胁搜寻。

安全帽识别

<https://kkgithub.com/yurizzzzz/Helmet-Detection-YoloV5>

GitHub上有不少安全帽识别项目，以下是一些常见的项目介绍：

- YOLOv4-Hat-detection：基于YOLOv4模型，可高效识别图像或视频中的安全帽佩戴情况，适用于建筑工地、工厂等场景。项目提供了详细教程，用户可根据需求对模型微调，还可结合DeepSORT算法实现对工人的长时间监控。

- YOLOv3-Helmet-Detection：基于YOLOv3算法，能检测图像中是否有人佩戴安全帽。可用于工厂安全监控和交通违规检测等场景。为提高模型泛化能力，可对训练数据进行旋转、缩放等数据增强操作，还可通过调整模型参数优化性能。项目源码地址为https://github.com/BlcaKHat/yolov3-Helmet-Detection。

- Smart\_Construction：利用YOLOv5进行安全帽佩戴检测，使用YOLOv5s、YOLOv5m、YOLOv5l等模型训练安全帽佩戴检测数据集，代码和权重均已开源。数据集包含7581张图像，项目还提供了详细教程，适合入门学习，地址为https://github.com/PeterH0323/Smart\_Construction。

- Safety-Helmet-Detection：基于YOLOv8模型，可识别建筑工地工人是否佩戴安全帽。项目包含了模型训练、标注等相关内容，能帮助学习者了解目标检测环境搭建、AI标注、训练、运行的全过程，GitHub地址是https://github.com/jomarkow/Safety-Helmet-Detection。

- 一盔一带安全帽识别：安全帽识别功能基于YOLOv3深度学习框架，人脸识别功能使用基于dlib的深度学习框架，界面设计采用pyqt5库，数据库使用mysql。通过摄像头传入信息，可识别未戴安全帽的人员并结合人脸识别对照数据库信息，统计违规次数。

人员跌倒识别项目

GitHub上有许多人员跌倒识别项目，以下是一些常见的项目介绍：

- Fall-Detection-with-CNNs-and-Optical-Flow：由Adrian Nunez开发，基于卷积神经网络（CNN）和光流技术，通过视觉信息自动检测跌倒事件。项目依赖OpenCV、TensorFlow等库，适用于老年人护理、体育训练等场景。

- human-falling-detect-tracks：基于YOLOv4物体检测框架和DeepSORT算法，结合OpenPose库，能实时监测和追踪人体跌倒行为。具有实时性强、准确性高、环境适应性好等特点，可应用于智能家居安全、公共场所监控等领域。

- ism\_person\_openpose：结合YOLOv5人体检测和OpenPose姿态检测来实现摔倒行为识别，主要使用Python开发，适用于家庭监控、养老院等场景。开发者可根据项目文档说明，收集图片并进行分类训练，以检测其他姿势。

- fall-detection：基于PaddleDetection的跌倒检测项目，包含模型文件以及web端部署代码。模型可检测正常、即将跌倒、已经跌倒、坐姿等状态，根目录下的 app.gradio.py 为demo网页展示启动文件。

- openvino-csharp-api：项目采用OpenVINO部署行人检测算法、关键点检测算法以及摔倒检测算法，实现对行人摔倒自动识别，并可在英特尔开发套件上使用OpenVINO C# API结合应用场景部署多模型，适用于养老院等场所对老人的摔倒检测。

- 基于yolov8的行人跌倒检测系统：该项目基于YOLOv8，包含Python源码、onnx模型和评估指标曲线等，还提供了精美GUI界面。可检测“fall”和“nofall”类别，适用于Windows 10系统，基于Anaconda3和Python 3.8环境，依赖torch和ultralytics等库。