# 1、python的安装

（1）先查看系统中是否自带python

$python3 --version 或者 $python –version

如果出现了python的版本号，则说明系统中自带python，无需要安装

（2）安装python3.6.9

$sudo apt install python3.6.9

（3）安装pip

$sudo apt-get install python3-pip

（4）更新pip版本

$python3 -m pip install --upgrade pip

# 2、部署一个python项目

## 2.1 从github上克隆出项目

参照4.1安装git

$git clone <https://github.com/theAIGuysCode/Object-Detection-API.git>

## 2.2安装项目所需的python依赖包

项目所需要的依赖包放在requirements.txt或requirements-gpu.txt（有显卡的情况下）。

requirements.txt

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | tensorflow==2.0.3  numpy  opencv-python==4.1.1.26  lxml  tqdm  flask  seaborn  pillow |

requirements-gpu.txt

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | tensorflow-gpu==2.1.2  numpy  opencv-python==4.1.1.26  lxml  tqdm  flask  seaborn  pillow |

$pip3 install -r requirements.txt

## 2.3下载yolov3官方模型

$wget https://pjreddie.com/media/files/yolov3.weights -O weights/yolov3.weights

## 2.4 将yolov3模型转换为Tensorflow模型

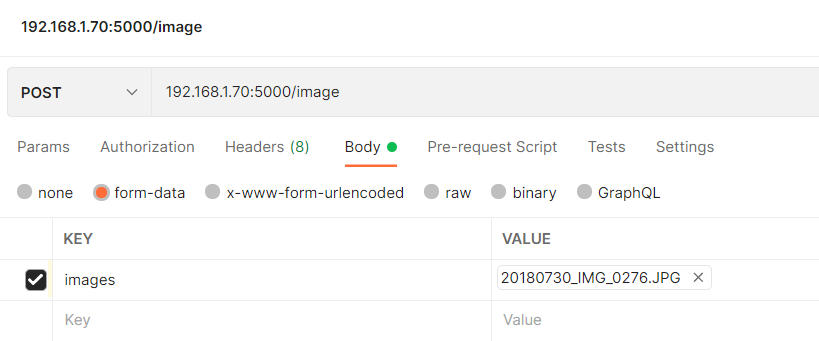
$python3 load\_weights.py

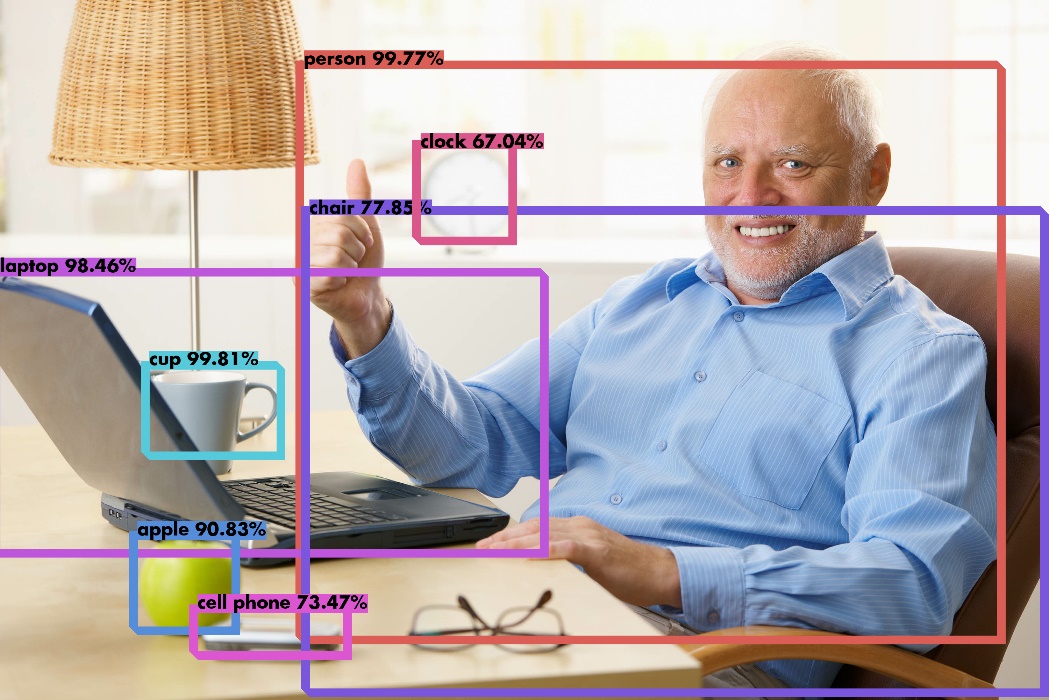
## 2.5 运行Flask应用程序启用图像识别AP

$python3 app.py

## 2.6 调用图像检测接口识别图像

能正确返回识别结果则说明部署成功





# 3、screen为多重视窗管理

* -A 　将所有的视窗都调整为目前终端机的大小。
* -d<作业名称> 　将指定的screen作业离线。
* -h<行数> 　指定视窗的缓冲区行数。
* -m 　即使目前已在作业中的screen作业，仍强制建立新的screen作业。
* -r<作业名称> 　恢复离线的screen作业。
* -R 　先试图恢复离线的作业。若找不到离线的作业，即建立新的screen作业。
* -s<shell> 　指定建立新视窗时，所要执行的shell。
* -S<作业名称> 　指定screen作业的名称。
* -v 　显示版本信息。
* -x 　恢复之前离线的screen作业。
* -ls或--list 　显示目前所有的screen作业。
* -wipe 　检查目前所有的screen作业，并删除已经无法使用的screen作业。

## 3.1 安装screen

$ sudo apt-get install screen

## 3.2 创建新视窗

创建名为one的视窗

$ screen –S one

## 3.3 在该视窗中部署图像检测接口

在该视窗中执行图像检测程序

$python3 app.py

然后按Ctrl+A+D切回主视窗，再次调用接口，看是否能正常调用；

输入$screen –r one切回刚创建的视窗

# 4、Git的安装使用

## 4.1 安装Git

$sudo apt-get install git

## 4.2 设置用户名(与github的账户无关)

$git config --global user.name "username"

$git config --global user.email "email-address"

## 4.3 初使化项目

把这个目录变成Git可以管理的仓库

$git init

## 4.4 往项目中添加文件

（1）文件添加到仓库

$git add README.md

（2）添加.gitignore文件，里面可以加入需要忽略的文件  
（3）把当前目录下所有未追踪的文件全部add了

$git add .

（4）把文件提交到仓库  
$git commit -m “first commit” filename

（4）把当前目录下所有未追踪的文件全部提交到仓库  
$git commit -m “first commit” ./

## 4.5 将项目推送到github

（1）查看是否配置过密钥

$cd ~/.ssh

（2）创建ssh

$ssh-keygen -t rsa -C 'xxx@xxx.com'

（3）关联远程仓库

$git remote add origin [git@github.com:\*\*\*/\*\*\*\*\*\*.git](mailto:git@github.com:***/******.git)

（4）把本地库的所有内容推送到远程库上

$git push -u origin master

（5）解决Support for password authentication 报错问题

https://www.mybj123.com/13423.html