**!cd yolov5 && python3 train.py --img 320 --batch 16 --epochs 200 --data dataset.yaml --weights yolov5s.pt**

--weights yolov5s.pt通过指定数据集、批量大小、图像大小以及预训练（推荐）或随机初始化--weights '' --cfg yolov5s.yaml（不推荐） ，在 COCO128 上训练 YOLOv5s 模型。

运行下面的命令训练coco128.ymal，训练5epochs。可以有两种训练方式

--cfg yolov5s.yaml --weights ''：从头开始训练

--cfg yolov5s.yaml --weights yolov5s.pt：从预训练的模型加载开始训练

YOLOv5在coco128上训练5epochs的命令：

**python train.py --img 640 --batch 16 --epochs 5 --data ./data/coco128.yaml --cfg ./model yoyov5s.yaml --weights**

训练的更多可选参数：

--epochs：训练的epoch，默认值300

--batch-size：默认值16

--cfg：模型的配置文件，默认为yolov5s.yaml

--data：数据集的配置文件，默认为data/coco128.yaml

--img-size：训练和测试输入大小，默认为[640, 640]

--rect：rectangular training，布尔值

--resume：是否从最新的last.pt中恢复训练，布尔值

--nosave：仅仅保存最后的checkpoint，布尔值

--notest：仅仅在最后的epoch上测试，布尔值

--evolve：进化超参数（evolve hyperparameters），布尔值

--bucket：gsutil bucket，默认值''

--cache-images：缓存图片可以更快的开始训练，布尔值

--weights：初始化参数路径，默认值''

--name：如果提供，将results.txt重命名为results\_name.txt

--device：cuda设备，例如：0或0,1,2,3或cpu，默认''

**torch.hub.load(github, model, force\_reload=False, \*args, \*\*kwargs)**

参数:

* ****github**** – 必需，一个字符串对象，格式为“repo\_owner/repo\_name[:tag\_name]”，可选 tag/branch。如果未做指定，默认的 branch 是 master 。比方说: 'pytorch/vision[:hub]'
* ****model**** – 必须，一个字符串对象，名字在hubconf.py中定义。
* ****force\_reload**** – 可选， 是否丢弃现有缓存并强制重新下载。默认是：False。
* \*****args**** – 可选， 可调用的model的相关args参数。
* ****\*\*kwargs**** – 可选， 可调用的model的相关kwargs参数。

| **返回:** | **一个有相关预训练权重的单一模型。** |
| --- | --- |

**torch.hub.set\_dir(d)**

也可以将hub\_dir设置为本地目录来保存中间模型和检查点文件。

如果未设置此参数,环境变量TORCH\_HUB\_DIR 会被首先搜寻，~/.torch/hub 将被创建并用作后备。

**os.path.join(path, \*paths)**

智能地拼接一个或多个路径部分。

返回值是 path 和 \*paths 的所有成员的拼接，其中每个非空部分后面都紧跟一个目录分隔符，最后一个部分除外，这意味着如果最后一个部分为空，则结果将以分隔符结尾。如果某个部分为绝对路径，则之前的所有部分会被丢弃并从绝对路径部分开始继续拼接。

在 Windows 上，遇到绝对路径部分（例如 r'\foo'）时，不会重置盘符。如果某部分路径包含盘符，则会丢弃所有先前的部分，并重置盘符。请注意，由于每个驱动器都有一个“当前目录”，所以 os.path.join("c:", "foo") 表示驱动器 C: 上当前目录的相对路径 (c:foo)，而不是 c:\foo。

**os.path.relpath(path[, start])**

返回从当前目录或 start 目录（可选）到达 path 之间要经过的相对路径。这仅仅是对路径的计算，不会访问文件系统来确认 path 或 start 的存在性或属性。

开始 默认为 **[os.curdir](https://docs.python.org/zh-cn/2.7/library/os.html" \l "os.curdir" \o "os.curdir)**

可用性：Windows、Unix。

**os.getenv(key, default=None)**

如果存在则返回环境变量键的值，如果不存在则 返回默认值。键，默认值和结果是 str。请注意，由于uses ，映射也同样在导入时被捕获，并且该函数可能不会反映未来的环境变化。[getenv()](https://docs.python.org/zh-cn/3/library/os.html?highlight=os getenv" \l "os.getenv" \o "os.getenv)[os.environ](https://docs.python.org/zh-cn/3/library/os.html?highlight=os getenv" \l "os.environ" \o "操作系统环境)[getenv()](https://docs.python.org/zh-cn/3/library/os.html?highlight=os getenv" \l "os.getenv" \o "os.getenv)

在Unix系统上，键和值会使用 [sys.getfilesystemencoding()](https://docs.python.org/zh-cn/3/library/sys.html" \l "sys.getfilesystemencoding" \o "sys.getfilesystemencoding) 和``'surrogateescape'`` 错误处理进行解码。如果你想使用其他的编码，使用 。[os.getenvb()](https://docs.python.org/zh-cn/3/library/os.html?highlight=os getenv" \l "os.getenvb" \o "os.getenvb)

[可用性](https://docs.python.org/zh-cn/3/library/intro.html" \l "availability): 大部分的Unix系统，Windows。

**numpy.squeeze(a, axis=None)**

从a中删除长度为 1 的轴。

参数:

* ****a**** – 必选，类似数组的，输入数据。
* ****axis**** – 可选，无或整数或整数元组。

选择形状中长度为 1 的条目的子集。如果选择了形状条目大于 1 的轴，则会引发错误。

| **返回** | **Squeezed：ndarray** |
| --- | --- |

输入数组，但删除了长度为 1 的所有维度或维度的子集。这始终是 a本身或进入a的视图。请注意，如果所有轴都被挤压，则结果是 0d 数组而不是标量。

| **注意** | **ValueError** |
| --- | --- |

如果轴不是无，并且被挤压的轴的长度不是 1

**cv2.VideoCapture()**

从视频文件或摄像机中捕获视频的类。

**cv2.VideoCapture() → <VideoCapture object>**

**cv2.VideoCapture(filename) → <VideoCapture object>**

**cv2.VideoCapture(device) → <VideoCapture object>**

参数:

* **filename**- 打开的视频文件的名称
* **device** – 打开的视频捕获设备的 id（即相机索引）。如果连接了单个摄像头，则只需传递 0。

**VideoCapture::open[¶](http://www.opencv.org.cn/opencvdoc/2.3.2/html/modules/highgui/doc/reading_and_writing_images_and_video.html?highlight=cv2.videocapture" \l "videocapture-open" \o "Permalink to this headline)**

参数：

* **filename**- 打开的视频文件的名称
* **device** – 打开的视频捕获设备的 id（即相机索引）。

这些方法首先调用**VideoCapture::release()**来关闭已经打开的文件或相机。

**VideoCapture::isOpened**

如果视频捕获已经初始化，则返回 true。

cv2.VideoCapture.isOpened ( )

如果先前对VideoCapture构造函数或VideoCapture::open的调用成功，则该方法返回 true。

**VideoCapture::release**

关闭视频文件或捕获设备。

**cv2.VideoCapture.release()**

后续VideoCapture::open()和VideoCapture析构函数会自动调用这些方法。

C 函数还释放内存并清除\*capture指针。

**VideoCapture::grab**

从视频文件或捕获设备中抓取下一帧。

cv2.VideoCapture.grab() → successFlag

cv.GrabFrame(capture) → int

方法/函数从视频文件或相机中抓取下一帧，并在成功的情况下返回真（非零）。

该功能的主要用途是在多摄像头环境中，尤其是在摄像头没有硬件同步的情况下。也就是说，您为每个摄像头调用VideoCapture::grab()，然后调用较慢的方法VideoCapture::retrieve()来解码并从每个摄像头获取帧。这样就消除了去马赛克或运动 jpeg 解压缩等的开销，并且从不同相机检索到的帧将在时间上更接近。

此外，当连接的摄像头是多头（例如，立体摄像头或 Kinect 设备）时，从中检索数据的正确方法是先调用VideoCapture::grab，然后再调用VideoCapture::retrieve()一个或更多次使用不同的通道参数值。见<https://code.ros.org/svn/opencv/trunk/opencv/samples/cpp/kinect_maps.cpp>

**VideoCapture::retrieve**

解码并返回抓取的视频帧。

cv2.VideoCapture.retrieve([image[, channel]]) → successFlag, image

cv.RetrieveFrame(capture) → iplimage

方法/函数解码并重新处理刚刚抓取的帧。如果没有抓取到任何帧（相机已断开连接，或者视频文件中没有更多帧），则方法返回 false 并且函数返回 NULL 指针。Note

**videoCapture::get**

返回指定的VideoCapture属性。

cv2.VideoCapture.get(propId) → retval

cv.GetCaptureProperty(capture, propId) → double

**参数：**

|  |  |
| --- | --- |
|  | **propId** –属性标识符。它可以是以下之一：   * **CV\_CAP\_PROP\_POS\_MSEC**视频文件的当前位置（以毫秒为单位）或视频捕获时间戳。 * **CV\_CAP\_PROP\_POS\_FRAMES**下一个要解码/捕获的帧的基于 0 的索引。 * **CV\_CAP\_PROP\_POS\_AVI\_RATIO**视频文件的相对位置：0 - 影片开始，1 - 影片结束。 * **CV\_CAP\_PROP\_FRAME\_WIDTH**视频流中帧的宽度。 * **CV\_CAP\_PROP\_FRAME\_HEIGHT**视频流中帧的高度。 * **CV\_CAP\_PROP\_FPS**帧速率。 * **CV\_CAP\_PROP\_FOURCC**编解码器的 4 字符代码。 * **CV\_CAP\_PROP\_FRAME\_COUNT**视频文件中的帧数。 * **CV\_CAP\_PROP\_FORMAT**由retrieve()返回的 Mat 对象的格式。 * **CV\_CAP\_PROP\_MODE**表示当前捕获模式的后端特定值。 * **CV\_CAP\_PROP\_BRIGHTNESS**图像的亮度（仅适用于相机）。 * **CV\_CAP\_PROP\_CONTRAST**图像对比度（仅适用于相机）。 * **CV\_CAP\_PROP\_SATURATION**图像的饱和度（仅适用于相机）。 * **CV\_CAP\_PROP\_HUE**图像的色调（仅适用于相机）。 * **CV\_CAP\_PROP\_GAIN**图像增益（仅适用于相机）。 * **CV\_CAP\_PROP\_EXPOSURE**曝光（仅适用于相机）。 * **CV\_CAP\_PROP\_CONVERT\_RGB**布尔标志，指示图像是否应转换为 RGB。 * **CV\_CAP\_PROP\_WHITE\_BALANCE**目前不支持 * **CV\_CAP\_PROP\_RECTIFICATION**立体相机的校正标志（注意：目前仅支持 DC1394 v 2.x 后端） |

**注意：当查询**VideoCapture类使用的后端不支持的属性时，返回值 0。

**VideoCapture::set**

在VideoCapture中设置一个属性。

cv2.VideoCapture.set(propId, value) → retval

cv.SetCaptureProperty(capture, propId, value) → None

参数：

|  |  |
| --- | --- |
|  | * **propId –属性标识符。它可以是以下之一：** * **CV\_CAP\_PROP\_POS\_MSEC视频文件的当前位置，以毫秒为单位。** * **CV\_CAP\_PROP\_POS\_FRAMES下一个要解码/捕获的帧的基于 0 的 索引。** * **CV\_CAP\_PROP\_POS\_AVI\_RATIO视频文件的相对位置：0 - 影片开始，1 - 影片结束。** * **CV\_CAP\_PROP\_FRAME\_WIDTH视频流中帧的宽度。** * **CV\_CAP\_PROP\_FRAME\_HEIGHT视频流中帧的高度。** * **CV\_CAP\_PROP\_FPS帧速率。** * **CV\_CAP\_PROP\_FOURCC编解码器的 4 字符代码。** * **CV\_CAP\_PROP\_FRAME\_COUNT视频文件中的帧数。** * **CV\_CAP\_PROP\_FORMAT由retrieve()返回的 Mat 对象的格式。** * **CV\_CAP\_PROP\_MODE表示当前捕获模式的后端特定值。** * **CV\_CAP\_PROP\_BRIGHTNESS图像的亮度（仅适用于相机）。** * **CV\_CAP\_PROP\_CONTRAST图像对比度（仅适用于相机）。** * **CV\_CAP\_PROP\_SATURATION图像的饱和度（仅适用于相机）。** * **CV\_CAP\_PROP\_HUE图像的色调（仅适用于相机）。** * **CV\_CAP\_PROP\_GAIN图像增益（仅适用于相机）。** * **CV\_CAP\_PROP\_EXPOSURE曝光（仅适用于相机）。** * **CV\_CAP\_PROP\_CONVERT\_RGB布尔标志，指示图像是否应转换为 RGB。** * **CV\_CAP\_PROP\_WHITE\_BALANCE目前不支持** * **CV\_CAP\_PROP\_RECTIFICATION立体相机的校正标志（注意：目前仅支持 DC1394 v 2.x 后端）** * **value – 属性的值。** |
|  |  |

|  |  |
| --- | --- |
|  |  |