COCO-api instance

OverView

As we all know, the COCO2014/2017 dataset is widely used for object detection, instance segmentation, image description, key point detection, panoramic segmentation and many other tasks, the official has provided cocoapi's python, matlab, lua language interface, but in matlab using the interface provided by the program is very poor readability, not easy to intuitively clear, the use of people This program makes full use of the characteristics of the table type to enrich the expression of coco datasets, with only about 100 lines of code to achieve the "instances" level of API, without any third-party libraries, can be customized to modify the API, code readability.

众所周知，COCO2014/2017 数据集被广泛用于目标检测、实例分割、图像描述、关键点检测、全景分割等多种任务，官方已提供 cocoapi 的 python、matlab、lua 语言的接口，但在 matlab 中使用其提供的接口程序可读性非常差，不易直观清晰，使用的人较少，本程序充分利用 table 类型的特性，丰富展示 coco 数据集的表达，仅用 100 行左右代码实现了“instances”级别的 API，无需任何第三方库，可二次自定义修改 API，代码可读性强。

语法

[allCOCOdata,cocoDatastore,cocoNames] = cocoAPI(imagesDir,annotationFile,annoType)

功能：优雅的实现 coco2014，coco2017 数据集新接口

输入：

imagesDir，string 类型，输入 COCO 图像文件根目录

annotationFile，string 类型，与之对应的标注 json 文件

annoType，string 类型，标注类型，目前仅支持“instances”类别

输出：

allCOCOdata， table 类型，所有带有标注的完整信息，每行代表一副图像

cocoDatastore， TransformedDatastore object， 可就地迭代对象

cocoNames，categorical 类型，80 个类别

Requirements

- MatlabR2021a or higher
- coco2014 or 2017 datasets (images and annotations)

如何使用

一般做 object detection/segmention,训练任务都会用到可迭代的 datasets，比如对上述 API 返回数据 cocoDatastore 进行迭代使用，read 会从内存中读取部分数据，读取的数据直观上应当有以下返回形式数据格式：

data = read(cocoDatastore)

- RGB images (H x W x 3)
- Bounding boxes (NumObjects x 4, arranged as [x y w h])
- Labels (NumObjects x 1), categorical
- Masks (H x W x NumObjects)

Example

```
% coco path
imagesDir = "E:\all\coco2017\train2017";
annotationFile = "E:\all\coco2017\annotations_trainval2017\annotations\instances_train2
% coco-api
[allCOCOdata,cocoDatastore,coconames] = cocoAPI(imagesDir,annotationFile);% take very t
```

visualize

```
head(allCOCOdata) % Get top rows of table
```

```
ans = 8×13 table
```

...

	image_id	license	file_name	coco_url	height	width	date_captured
1	9	3	'000000000009.jpg'	'http://image...	480	640	'2013-11-19 20...
2	25	1	'000000000025.jpg'	'http://image...	426	640	'2013-11-16 14...
3	30	4	'000000000030.jpg'	'http://image...	428	640	'2013-11-24 03...
4	34	6	'000000000034.jpg'	'http://image...	425	640	'2013-11-18 16...
5	36	3	'000000000036.jpg'	'http://image...	640	481	'2013-11-18 06...
6	42	2	'000000000042.jpg'	'http://image...	478	640	'2013-11-18 09...
7	49	2	'000000000049.jpg'	'http://image...	500	381	'2013-11-14 20...
8	61	1	'000000000061.jpg'	'http://image...	488	640	'2013-11-18 04...

```
colormap = randi(255,length(coconames),3);
cocoDatastore = shuffle(cocoDatastore); % random
while cocoDatastore.hasdata()
    data = read(cocoDatastore);
    img = data{1};
    bboxs = data{2};
    labels = data{3};
    masks = data{4};

    colors = colormap(arrayfun(@(x)find(x==coconames),labels),:);
    draw = insertObjectAnnotation(img,'rectangle',bboxs,labels,...
        LineWidth=3,...
        Color=colors);
    for i = 1:size(masks,3)
        draw = labeloverlay(draw,masks(:,:,i),...
            Colormap=colors(i,:)/255,...
            Transparency = 0.5);
    end
```

```
    imshow(draw)
    break;
end
```