

行为识别的数据标注

数据标注

- 数据标注就是使用自动化工具通过分类、画框、注释等等对收集来的数据进行标记以形成可供计算机识别分析的优质数据的过程。
- 数据标注的对象主要分为文本、图片、音频、视频四个种类。
- 数据标注的过程：
 - 1.数据清洗;
 - 2.数据标注;
 - 3.标注检验。

HMDB-51



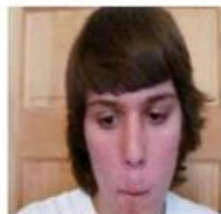
brush
hair



cartwheel



catch



chew



clap



climb



climb
stairs



dive



draw
sword



dribble



drink



eat



fall
floor



fencing



flic
flac



golf



hand
stand



hit



hug



jump



kick

UCF-101



行为识别常用数据集

- 行为分类数据集（裁剪后的视频片段，视频级标签）：UCF-101、HMDB-51、Kinetics-400、Kinetics-600 和 something-to something。
- 行为检测数据集（针对定位任务，单帧标签）：UCF101-24、JHMDB-21、AVA（多动作多标签）。
- 骨骼数据集（视频分类任务+3D骨骼坐标点）：NTU RGB+D-60, NTU RGB+D-120。






常用数据标注工具

- 1.Labelme <https://github.com/wkentaro/labelme>
- 2.LabelImg <https://github.com/tzutalin/labelImg>
- 3.OpenCV/CVAT www.github.com/opencv/cvat
- 4. LabelHub www.Labelhub.cn
- 5.VoTT

数据标注工具(VoTT)

- VoTT是微软发布的用于图像目标检测的标注工具，它是基于javascript开发的，因此可以跨Windows和Linux平台运行，并且支持从图片和视频读取。此外，它还提供了基于CNTK训练的faster-rcnn模型进行自动标注然后人工矫正的方式，这样大大减轻了标注所需的工作量。其分为V1和V2两个版本，目前V1仅用于BUG修复，不再添加新功能支持，因此建议大家直接用V2.
- <https://github.com/Microsoft/VoTT/releases>

Assets 5

 vott-2.2.0-darwin.dmg	134 MB
 vott-2.2.0-linux.snap	122 MB
 vott-2.2.0-win32.exe	93.8 MB
 Source code (zip)	
 Source code (tar.gz)	

VoTT的使用介绍


```

import os
import json
import cv2
from bounding_box import bounding_box as bbox_plot
Img_Dir = 'C:\\Users\\wu411299174\\Desktop\\target\\vott-json-export'
Save_Dir = 'C:\\Users\\wu411299174\\Desktop\\results'
Draw_Tags = ['run']
def make_video(imgs_dir, img_size=(1920, 1080), fps=5):
    file_path = os.path.join(Save_Dir, 'test.mp4')
    filelist = os.listdir(imgs_dir)
    fourcc = cv2.VideoWriter_fourcc('D', 'I', 'V', 'X')
    video = cv2.VideoWriter(file_path, fourcc, fps, img_size)
    for item in filelist:
        if item.endswith('.jpg'):
            item = imgs_dir + '/' + item
            img = cv2.imread(item)
            video.write(img)
    video.release()
def draw_one_img(img_name, img_timestamp, labels_list):
    img_path = os.path.join(Img_Dir, img_name)
    img = cv2.imread(img_path, cv2.IMREAD_COLOR)
    for label in labels_list:
        tag = label['tags'][0]
        if not tag in Draw_Tags:
            continue
        bbox = label['boundingBox']
        left = bbox['left']
        top = bbox['top']
        right = left + bbox['width']
        bottom = top + bbox['height']
        bbox_plot.add(img, left, top, right, bottom, tag, 'red')
        save_name = 'time%s.jpg' % format(img_timestamp, '.2f').replace('.', '')

```

```

        save_path = os.path.join(Save_Dir, save_name)
        cv2.imwrite(save_path, img)
        print(save_name, "ok")
def main():
    lb_json = os.path.join(Img_Dir, 'test-export.json')
    with open(lb_json, 'r') as f:
        labels_dict = json.load(f)
    assets = labels_dict['assets']
    size = None
    for k, v in assets.items():
        img_name = v['asset']['name']
        img_timestamp = v['asset']['timestamp']
        if size is None:
            w = v['asset']['size']['width']
            h = v['asset']['size']['height']
            size = (w, h)
        labels_list = v['regions']
        draw_one_img(img_name, img_timestamp, labels_list)
    make_video(Save_Dir, img_size=size)
if __name__ == '__main__':
    main()

```

感谢大家的耐心聆听！