

# Api readme

Tuesday, August 7, 2018

16:29

Client端调用方式(参见sudo client.py) :

1. 调用url: [http://10.128.34.1:9000/run\\_model](http://10.128.34.1:9000/run_model)
2. Input ( 模型的参数 ) :
  - a. Saved as json ( like a dict)
  - b. plist\_info (dict) ( 参见hw.json ) :
    - i. key: img\_name,
    - ii. value(dict):
      - 1) 'gps\_info', 'exif\_info' (dict)
      - 2) 'label' (str)
  - c. 'user\_id': user id (should be unique for each request)
  - d. 'thread\_id': (should be unique for each request)
  - e. 'count': counter of threads

```
f. json_input = dict()
    json_input['plist_info'] = raw_json
    json_input['user_id'] = user
    json_input['thread_id'] = thr_id
    json_input['count'] = count
```

3. Output (返回值) :
  - a. 最后分组的image name, grouped by different scenes , 返回给前端。
  - b. Format: 看前端需要吧 , 目前是list

模型 ( 参见api\_album.py) :

1. **!!! 目前只能用云从的内网访问**

2. 调用url: 10.128.34.1:9000/run\_model
3. 所用包 : flask
4. 多线程 : 理论上可以多线程
5. 模型位置 : 云从server上的docker中 :
  - a. Server: 10.128.34.1
  - b. Docker: uathena/tf\_py35\_with\_dlib\_cv2\_gpu:album\_project  
Image Id: 0c5c2c5d871f
  - c. Container id: 6c80dc0693f6
6. 启动模型(一般情况下 , 这个模型都是开着的.....just in case...)
  - a. docker exec -it 6c80dc0693f6 bash
  - b. python api\_album.py
  - c. will combine start docker and start server together in future