

# 视频分析/图像分类接口

本接口将包含**图像分析**和**在线训练**两部分，目前仅提供图像分析接口。

## 图像分析接口

### 1 单幅图片的分析/分类：

client通过POST上传图像数据，server分析数据，并通过json返回分析结果。

#### **request:**

```
POST /pic HTTP/1.1
Content-Type: image/jpeg
Content-Length: xxxx
Host: xxx
```

[image data]

#### **response**

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
```

```
{
  [ "top1": "单人-讲台区-手指向-学生区", "score1": 0.87002,
    "top2": "单人-讲台区-看-学生区", "score2": 0.0100,
    "top3": "单人-讲台区-看-无", "score3": 0.00093
  ]
}
```

**解释：**request时，必须使用 POST，url固定是 /pic，body 是图像的二进制数据。  
response 的 body 为 json 格式，里面是一个列表，列出了最有可能的前三类分类结果。topN 为分类的文字描述（这部分是否需要统一定义？），scoreN为机器预测的可能性。

### 2 受控视频分析

client通过POST提交需要分析的url（目前包括 rtsp:// rtmp:// file://），某些选项（每隔多长时间分析一次之类的），server 如果启动分析成功，返回 sessionid，失败则返回错误。

创建分析任务：

#### **request**

```
POST /stream HTTP/1.1
Content-Type: text/json
```

```
{
  "url": "rtsp://172.16.1.89/av0_0",
  "interval": 3.0,
  "topN": 3
}
```

### response

HTTP/1.0 200 OK

Content-Type: application/json; charset=UTF-8

```
{
  "error": 0,
  "sessionid": "xxxxx"
}
```

**解释：**request时，通过json格式的 body 指定分析的url，每次分析的间隔时间秒（分析的是图像！），topN 可以指定查询返回时，每帧图像的最有可能的前N中分类结果。response中，如果 error = 0，则将提供 sessionid，给后面的调用使用。

### 查询分析任务状态

#### request

GET /stream/<sessionid>/query?start=xxx&stop=xxx HTTP/1.1

#### response

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

```
{
  "state": "running",
  "results": [
    { "stamp": xxx,
      "top1": yyy, "score1": zzz,
      ...
    }
    { "stamp": xxx,
      "top1": yyy, "score1": zzz,
      ....
    }
    ....
  ]
}
```

**解释：**request 指定查询时间范围，response 返回指定时间段内的图像分类结果，通过 json 格式返回。state 为当前状态，包含：running, stopped，如果stopped说明url提供的流已经结束。

结束分析任务：

**request**

DELETE /stream/<sessionid> HTTP/1.1

**response:**

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

```
{  
  ...  
}
```

**解释：**request DELETE 使用对应的 sessionid，response 的 body 为完整的分析结果，格式如查询的结果。调用该函数后，sessionid 将失效。

列出所有分类任务

**request**

GET /stream HTTP/1.1

**response**

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

```
{  
  "num": 3,  
  "sessions": [  
    {  
      "begin": 1465869659.877,  
      "cf_images": 0,  
      "sid": "1",  
      "boot_cfg": {  
        "interval": 3,  
        "topn": 3,  
        "url": "rtsp://172.16.1.189"  
      },  
      "running": true,  
      "last_stamp": -1  
    },  
    {  
      "begin": 1465869696.534,  
      "sid": "3",  
      "cf_images": 49,  
      "running": false,  
      "last_stamp": 24.4079999924,  
    }  
  ]  
}
```

```

    "boot_cfg" : {
        "interval" : 3,
        "url" : "c:/users/sunkw/desktop/videos/51.mp4",
        "topn" : 3
    }
},
{
    "begin" : 1465869666.409,
    "running" : true,
    "last_stamp" : 386.111000061,
    "boot_cfg" : {
        "topn" : 3,
        "url" : "rtsp://172.16.1.89",
        "interval" : 3
    },
    "sid" : "2",
    "cf_images" : 131
}
]
}

```

**解释：** request的url与创建session一样。response中返回session列表，基本结构：

```

struct SessionDescr {
    int sid;      // sessionid, 与创建 session 返回的 sessionid 一致；

    struct {
        char *url;
        float interval;
        int topn;
    } boot_cfg; // 创建session 时提供的参数

    double begin;      // 创建 session 时的主机时间
    bool running;      // 分析是否正在进行中，但某些直播情况下，这个值不能保证准确，
                        // 可以通过检查 cf_images 和 last_stamp 参数判断是否正确。
    double last_stamp; // 分析的图像中最后一帧图像的时间戳，如果-1，说明没有分析结果
    int cf_images;      // 被分析的图像数目
};

```

### 3 批分析任务

创建任务

**request**

POST /batch HTTP/1.1

Content-Type: application/json; charset=UTF-8

```
{
```

```
    "notify_url": "http://xxxx",
    "urls": [
        "rtsp://...",
        "rtmp://...",
        "/mnt/media/...mp4",
        ....
    ]
}
```

**response:**

HTTP/1.1 OK 200

Content-Type: application/json; charset=UTF-8

```
{
    "error": 0,
    "batchid": "xxx"
}
```

**解释：**request 的body中提供 notify\_url，当批任务完成后，server将调用该函数，进行通知，通知的格式再定吧。response的body为json格式，描述是否创建成功，一般来说，只能同时跑一个批任务。

取消批任务：再定吧