# DMS疲劳检测设计

# 1. 背景

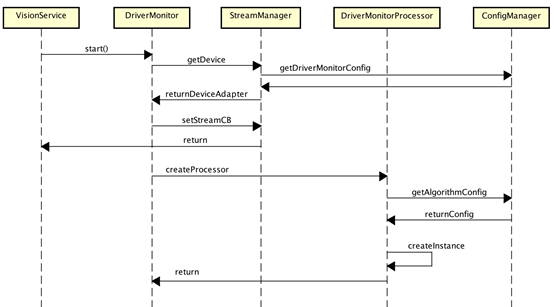
DMS检测包括，疲劳(闭眼/哈欠)、分心、抽烟和打电话等行为的检测。

本文主要涉及visiond中DMS相关模块的架构设计，**不包含算法部分**；

# 2. 架构设计

架构设计图：

driver设计3流程图：



# 3. 离线视频测试

## 3.1 DMS视频剪切、标注要求

### 3.1.1 裁剪采集视频的整体要求：

1. 要求每个视频仅包含某类（闭眼/打哈欠/抽烟/打电话/正常）的动作，且时长要求大于2s

1. 裁剪的视频段第一帧需已经是动作的进行状态，且尽量是能看到完整人脸，最后一帧时动作仍在持续或者即将结束

1. 测试视频整体保持驾驶姿态，目视前方，不应有低头垂眼。

### 3.1.2 各种状态的具体要求：

1. 闭眼：眼睛保持闭合

1. 打哈欠：打哈欠状态尽量真实（正常的一个哈欠会持续2s以上）为了模仿而仅张嘴的视频不应标注为打哈欠，打哈欠时不能用手捂住嘴。若是几乎无间隔的多次打哈欠，不需要分开剪裁。视频第一帧建议从嘴张到一半开始，最后一帧为嘴完全闭合前。

1. 抽烟：需要有真实的香烟，且手扶香烟，手贴在嘴边，烟在嘴中

1. 打电话：手持电话贴在耳边，若中途换手需要将视频剪成两段，分别从左/右手持电话贴在耳边作为开始的第一帧

1. 正常：不存在以上状态，并且不在低头、点头、摇头的状态中

## 3.2 DMS配置

1. 修改算法测试类型为4

1. 修改输出日志的格式为1

1. 确认"algo\_process\_fps" 值为10. DMS和场景动作设置不同

1. 修改最小车速为0，或者模拟车速大于20

1. 输入源改为"simulator" "default" -> "simulator"

1. 修改DMS算法模型，使用mnn模型

# 算法测试类型，0 改为 4  
"algo\_test\_modules": 0, // 0: nothing, 1: gesture,2: face, 3: for gesture&face, 4: for DMS  
  
# 输出日志的格式，0 改为 1  
"assess\_algo": 0, // 0: disable, 1: simple, 2: complex  
  
"algo\_process\_fps" : 10, // DMS 要求用10，不要修改  
  
# 最小车速，20 改为 0. 用视频测试时，不需要速度限制  
"enable\_min\_speed": 20, // 启动驾驶员检测的最小车速 kmph  
  
# DMS模型“head\_Model”: 设置为1，使用mnn模型  
 "driver\_head" : {  
 "head\_Model" : 1, // 0:none, 1:mnn, 2:tvm  
   
# 输入源“dev\_type”: "default" 改为 "simulator"  
 "drivermonitor\_camera": {  
 "dev\_type": "default",  
 },

可以参考修改后的配置文件：[📎vision\_config.json](https://yuque.antfin.com/attachments/lark/0/2021/json/19256916/1624526692027-de432cc3-0e6d-40be-bd50-44dcff35d78b.json)

## 3.3 测试DMS 步骤

目前分心算法需要预先标定摄像头角度。

#创建视频目录  
adb -host shell mkdir /mnt/data/video  
#创建日志输出目录  
adb -host shell mkdir -p /var/visiond/debugoutput  
adb -host shell touch /var/visiond/debugoutput/record.txt  
adb -host shell chmod 777 /var/visiond/debugoutput/record.txt  
  
#上传测试视频  
adb -host push <你的测试视频目录> /mnt/data/video  
  
#打开疲劳检测（打呵欠、闭眼）  
adb -host shell vdsend setDmsDrowsy 1  
#打开分心检测  
adb -host shell vdsend setDmsDistract 1  
#打开抽烟检测  
adb -host shell vdsend setDmsSmoke 1  
#打开打电话检测  
adb -host shell vdsend setDmsCall 1  
  
#重启visiond  
adb -host shell systemctl restart visiond

## 3.4 DMS结果

日志文件输出在 /var/visiond/debugoutput/record.txt。 格式如下：

/mnt/data/video/Crop/ir/drowsy/1.avi distract 0 headLower 0 eye 1700 smoke 0 yawn 0 phone 0  
/mnt/data/video/Crop/ir/drowsy/2.avi distract 0 headLower 0 eye 1700 smoke 0 yawn 0 phone 0

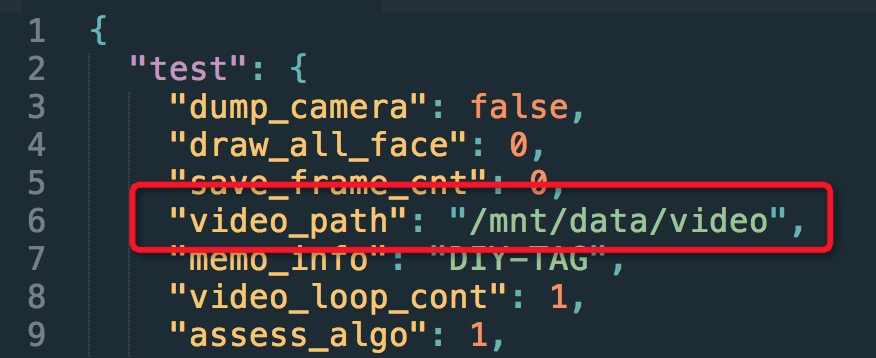
每个文件对应一行输出，distract——分心，headLower——低头，eye——闭眼，smoke——抽烟，yawn——打呵欠，phone——打电话。动作后面的数字表示首次检测到这个动作的时间，单位毫秒。如果动作后面的数字是0，表示没有检测到这个动作。

## 3.5 修改测试视频的目录（可选）

默认从/mnt/data/video读取视频，也可以修改为其它目录

修改 /usr/lib/visiond/vision\_config.json 里的 "video\_path"  参数，指向你的视频根目录。然后重启visiond。

visiond会遍历指定目录下的所有视频和图片文件



## 3.6 vdsend测试DMS命令

在车机上执行命令vdsend可以看到help。

# vdsend  
vdsend command [params]  
 vdsend DMS\_Calibration 读取DMS标定  
 vdsend startDms 启动DMS模块  
 vdsend stopDms 关闭DMS模块  
 vdsend setDmsDrowsy [0 | 1] 疲劳检测开关  
 vdsend setDmsDistract [0 | 1] 分心检测开关  
 vdsend setDmsSmoke [0 | 1] 抽烟检测开关  
 vdsend setDmsCall [0 | 1] 打电话检测开关  
 vdsend setGazeCalibrate [string] 分心摄像头角度标定  
 vdsend getDmsDrowsy 读取疲劳检测开关状态  
 vdsend getDmsDistract 读取分心检测开关状态  
 vdsend getDmsSmoke 读取抽烟检测开关状态  
 vdsend getDmsCall 读取打电话检测开关状态  
 vdsend getGazeCalibrate 读取分心摄像头角度标定  
 vdsend jarvis [string] 模拟DMS检测结果信号  
 vdsend fake speed 50 设置车速为50.一旦设置假数据后，pcan、实车的输入都无效

## 3.7 模拟DMS的提醒

### 1. 设置dms对车速相关限制为0  
setprop visiond.dms.test.travel\_status 0  
### 2. 设置dms在开车后xxx毫秒不触发提示  
setprop visiond.dms.test.pretime 0   
### 3. stop dms，清除DMS上一次的状态  
vdsend stopDms  
### 4. 删除jarvis数据库并重启jarvisd  
rm /private/jarvis/jarvis.db  
systemctl restart jarvisd  
### 5. 输入DMS事件，可以是离线视频的方式，也可以是camera输入的方式

参考：<https://yuque.antfin-inc.com/docs/share/f626b2c3-0866-4113-8aae-a662c751ba85?#>