

目录

目录	1
概述	2
LinkSVP简介	2
UseLinkIVE	3
准备工作	3
运行结果	4
UseHisilVE	6
准备工作	6
运行结果	7
UseOpencv	9
准备工作	9
运行结果	10
MotionDetect	12
准备工作	12
运行结果	13
FaceDetect	15
准备工作	15
运行结果	16
FaceRecognition	18
准备工作	18
运行结果	19
PedestrianDetect	21
准备工作	21
运行结果	22

概述

该工程集合了若干关于机器视觉的基础演示例程，适用于灵派科技全系产品，可以帮助用户快速的熟悉如何使用LinkSVP库进行机器视觉相关的产品开发。

LinkSVP简介

LinkSVP(Smart Vision Platform)整合了海思HiSVP、HiIVE、NCNN、OpenCV等，未来还会支持tensorflow。LinkSVP基于LinkLib，因此，借助LinkLib的便捷性，开发者无需处理底层视频传输，可以更专注于算法的设计与优化。开发者基于LinkFrame类创建的对象，只需要设置目标算法需要的分辨率、帧率，系统会自动处理底层视频接口，开发者只需要在视频帧到达时调用其算法即可。

借助NCNN的整合，即使是在不支持NNIE的芯片上，也能进行简单的深度学习运算。

在尝试这些例程的同时，建议您先阅读《[LinkSDK在线文档](#)》，以了解代码中各模块的功能与参数。

建议体验顺序如下：

- UseLinkIVE，使用封装好的IVE库进行开发的二值化例子
- UseHisiIVE，使用海思官方的IVE库进行开发的二值化例子
- UseOpencv，使用opencv进行开发的二值化例子
- MotionDetect，LinkIVE库进行移动侦测
- FaceDetect，综合使用LinkIVE、NCNN、opencv进行人脸检测追踪

UseLinkIVE

该示例程序演示了如何使用LinkIVE进行图像二值化操作

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 将带有HDMI输出功能的设备(如摄像机、笔记本、机顶盒等)接入评估板的HDMI-A接口
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行UseLinkIVE程序

运行结果



UseHisilVE

该示例程序演示了如何使用海思官方的IVE进行图像二值化操作

由于LinkIVE还在完善中，如果遇到LinkIVE无法满足的需求，可以直接调用官方IVE函数。

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 将带有HDMI输出功能的设备(如摄像机、笔记本、机顶盒等)接入评估板的HDMI-A接口
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行UseHisilVE程序

运行结果



UseOpencv

该示例程序演示了如何使用opencv进行图像二值化操作

当IVE系统的功能无法满足需求时，我们可以借助强大的opencv来实现，但是请注意观察cpu性能

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 将带有HDMI输出功能的设备(如摄像机、笔记本、机顶盒等)接入评估板的HDMI-A接口
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行UseOpencv程序

运行结果



MotionDetect

该示例程序演示了如何使用LinkIVE进行一个相对完整的移动检测功能开发

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 将带有HDMI输出功能的设备(如摄像机、笔记本、机顶盒等)接入评估板的HDMI-A接口
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行MotionDetect程序

运行结果



```
~/demo # ./MotionDetect
LinkSDK 1.0.0 build 20190801_9648
mmz_userdev:get_mmbInfo_safe: mmb(0x78606000) not found!
System unmap mmz memory failed!
mmz_userdev:get_mmbInfo_safe: mmb(0x78606000) not found!
System free mmz memory failed!
mmz_userdev:get_mmbInfo_safe: mmb(0x7872A600) not found!
System unmap mmz memory failed!
mmz_userdev:get_mmbInfo_safe: mmb(0x7872A600) not found!
System free mmz memory failed!
[OutputV0_1] show 1920 x 1080 vid 0
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 230400))("h", QVariant(double, 0.997222))("w", QVariant(double, 0.998437))("x", QVariant(double, 0))("y", QVariant(double, 0))))
[OutputV0_1] lntfB 1920 1080
[InputV1_1] V1 set 1920 x 1080 P 60 @ 8 , 32
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 55153))("h", QVariant(double, 0.997222))("w", QVariant(double, 0.651563))("x", QVariant(double, 0.346875))("y", QVariant(double, 0))))
QVariant(QVariantMap, QMap("area", QVariant(uint, 10189))("h", QVariant(double, 0.480556))("w", QVariant(double, 0.101562))("x", QVariant(double, 0))("y", QVariant(double, 0.0694444))))
QVariant(QVariantMap, QMap("area", QVariant(uint, 12314))("h", QVariant(double, 0.444444))("w", QVariant(double, 0.134375))("x", QVariant(double, 0))("y", QVariant(double, 0.552778))))
QVariant(QVariantMap, QMap("area", QVariant(uint, 1840))("h", QVariant(double, 0.233333))("w", QVariant(double, 0.05625))("x", QVariant(double, 0.154688))("y", QVariant(double, 0.763889))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 2221))("h", QVariant(double, 0.0833333))("w", QVariant(double, 0.210938))("x", QVariant(double, 0.542188))("y", QVariant(double, 0.913889))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 1194))("h", QVariant(double, 0.0583333))("w", QVariant(double, 0.207813))("x", QVariant(double, 0.535937))("y", QVariant(double, 0.913889))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 2988))("h", QVariant(double, 0.141667))("w", QVariant(double, 0.196875))("x", QVariant(double, 0.5375))("y", QVariant(double, 0.855556))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 864))("h", QVariant(double, 0.0277778))("w", QVariant(double, 0.201563))("x", QVariant(double, 0.553125))("y", QVariant(double, 0.861111))))
QVariant(QVariantMap, QMap("area", QVariant(uint, 1048))("h", QVariant(double, 0.0777778))("w", QVariant(double, 0.090625))("x", QVariant(double, 0.55625))("y", QVariant(double, 0.919444))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 3562))("h", QVariant(double, 0.144444))("w", QVariant(double, 0.24375))("x", QVariant(double, 0.528125))("y", QVariant(double, 0.852778))))
#####
QVariant(QVariantMap, QMap("area", QVariant(uint, 2859))("h", QVariant(double, 0.113889))("w", QVariant(double, 0.23125))("x", QVariant(double, 0.545312))("y", QVariant(double, 0.803333))))
```

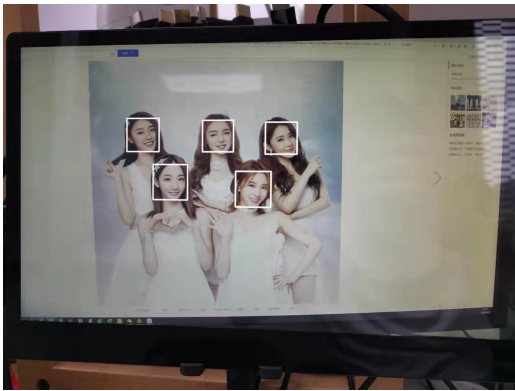
FaceDetect

该示例程序演示了如何综合使用LinkIVE、NCNN、IVE进行一个人脸检测与追踪的功能开发

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 将带有HDMI输出功能的设备(如摄像机、笔记本、机顶盒等)接入评估板的HDMI-A接口
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行FaceDetect程序

运行结果



FaceRecognition

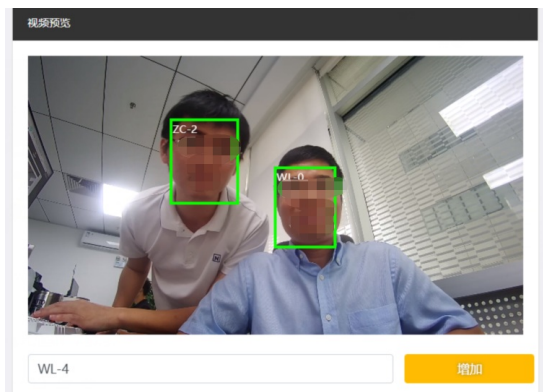
该示例程序演示了在3519A、3559A平台上如何利用NNIE性能进行人脸识别

该项目包括主程序、网页、模型数据、人脸数据、rpc通信等环节，是一个完整的人脸识别工程示例

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 连接好镜头、对焦，放置到适合采集人脸的位置
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，运行Face目录
- 运行FaceDetectRecognition程序
- 打开浏览器，访问开发板ip
- 登陆，账号密码均为admin
- 进入高级功能>人脸识别
- 采集阶段需要保持画面中只有一个人脸，进行多角度采集
- 每个角度取不同的命名，比如正脸XX-0，左转15度XX-1，抬头15度XX-2之类的
- 随后可以在界面上看到识别结果
- 识别阶段可以多人，但是需要距离镜头较近才能识别

运行结果



PedestrianDetect

该示例程序演示了在3519A、3559A平台上如何利用NNIE性能进行行人检测

准备工作

- 参照用户手册搭建开发环境、编译LinkSVP工程、配置网络启动参数。
- 连接好镜头、对焦，放置到适合拍摄行人的位置
- 将评估板的HDMI-OUT接显示器(支持1080P即可，程序默认输出1080P60)。
- 上电，进入/root/demo目录
- 运行PedestrianDetect程序

运行结果

