**一、JPEGE（图片编码）功能使用指导**

**1、基本原理**

•调用acldvppCreateChannel接口创建图片数据处理的通道。

•调用acldvppJpegEncodeAsync异步接口，将YUV格式图片编码成.jpg图片。对于异步接口，还需调用aclrtSynchronizeStream接口阻塞Host运行，直到指定Stream中的所有任务都完成。

•调用acldvppJpegPredictEncSize接口根据输入图片描述信息、图片编码配置数据预估图片编码后所需的输出内存的大小。

**2、基本流程**

**//1.ACL初始化**

const char \*aclConfigPath = "../src/acl.json";

aclInit(aclConfigPath)；

**//2.运行管理资源申请,包括Device、Context、Stream，stream\_是aclrtStream类型**

aclrtSetDevice(deviceId\_)；

aclrtCreateContext(&context\_, deviceId\_);

aclrtCreateStream(&stream\_);

**//3.创建图片数据处理通道时的通道描述信息，dvppChannelDesc\_是acldvppChannelDesc类型**

dvppChannelDesc\_ = acldvppCreateChannelDesc();

**//4.创建图片数据处理的通道**

aclRet = acldvppCreateChannel(dvppChannelDesc\_);

**//5.申请内存**

**//5.1 输入内存**

**//申请Host内存inputHostBuff，存放YUV格式的图片数据**

**//申请Device内存inputDevBuff**

**//将通过aclrtMemcpy接口将Host的图片数据传输到Device，数据传输完成后，需及时调用aclrtFreeHost接口释放Host内存**

aclrtMallocHost(&inputHostBuff, PicBufferSize);

acldvppMalloc(&inputDevBuff, PicBufferSize);

aclrtMemcpy(inputDevBuff, PicBufferSize, inputHostBuff, PicBufferSize, ACL\_MEMCPY\_HOST\_TO\_DEVICE);

**//5.2 输出内存，申请Device内存encodeOutBufferDev\_,存放编码后的输出数据**

acldvppMalloc(&encodeOutBufferDev\_, outBufferSize)

**//6. 创建编码输入图片的描述信息，并设置各属性值**

//encodeInputDesc\_是acldvppPicDesc类型

encodeInputDesc\_ = acldvppCreatePicDesc();

acldvppSetPicDescData(encodeInputDesc\_, reinterpret\_cast<void \*>(inDevBuffer\_));

acldvppSetPicDescFormat(encodeInputDesc\_, PIXEL\_FORMAT\_YUV\_SEMIPLANAR\_420);

acldvppSetPicDescWidth(encodeInputDesc\_, inputWidth\_);

acldvppSetPicDescHeight(encodeInputDesc\_, inputHeight\_);

acldvppSetPicDescWidthStride(encodeInputDesc\_, encodeInWidthStride);

acldvppSetPicDescHeightStride(encodeInputDesc\_, encodeInHeightStride);

acldvppSetPicDescSize(encodeInputDesc\_, inDevBufferSizeE\_);

**//7. 创建图片编码放配置数据，设置编码质量**

**//编码质量范围[0, 100]，其中level 0编码质量与level 100差不多，而在[1, 100]内数值越小输出图片质量越差。**

jpegeConfig\_ = acldvppCreateJpegeConfig();

acldvppSetJpegeConfigLevel(jpegeConfig\_, 100);

**//8. 执行异步解码，再调用aclrtSynchronizeStream接口阻塞Host运行，直到指定Stream中的所有任务都完成**

acldvppJpegEncodeAsync(dvppChannelDesc\_, encodeInputDesc\_, encodeOutBufferDev\_,

&outBufferSize, jpegeConfig\_, stream\_);

aclrtSynchronizeStream(stream\_);

**//9.申请Host内存hostPtr，将编码后的输出图片回传到Host，再将Host内存中的数据写入文件,写完文件后，需及时调用aclrtFreeHost接口释放Host内存**

aclrtMallocHost(&hostPtr, dataSize);

aclrtMemcpy(hostPtr, dataSize, encodeOutBufferDev\_, outBufferSize, ACL\_MEMCPY\_DEVICE\_TO\_HOST);

**//11. 编码结束后，释放资源，包括编码输入/输出图片的描述信息、编码输入/输出内存**

acldvppDestroyPicDesc(encodeInputDesc\_);

(void)acldvppFree(inputDevBuff);

(void)acldvppFree(encodeOutBufferDev\_);

**3、代码流程图**

