**一、cropandpaste（抠图和粘贴）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acldvppVpcCropAndPasteAsync异步接口，将输入图片指定位置指定大小部分图片抠出来，粘贴到输出图片指定位置。对于异步接口，还需调用[aclrtSynchronizeStream](mk:@MSITStore:C:\\Users\\ascend\\Documents\\WeChat%20Files\\hu1015138540\\FileStorage\\File\\2020-06\\应用开发指导.chm::/zh-cn_topic_0238843731.html" \l "ZH-CN_TOPIC_0238843731)接口阻塞Host运行，直到指定Stream中的所有任务都完成。

•调用acldvppCreateRoiConfig接口创建图片crop和paste配置数据。

•缩放后输出图片内存根据YUV420SP格式计算，计算公式：对齐后的宽\*对齐后的高\*3/2

**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。**

acldvppCreateChannel(dvppChannelDesc\_);

**//5.将图片读入内存，inDevBuffer\_表示存放输入图片的内存, inDevBufferSize表示内存大小，输入内存要提前申请**

acldvppMalloc(&inBufferDev, inBufferSize);

**//6. 创建输入图片的描述信息，并设置各属性值，vpcInputDesc\_是acldvppPicDesc类型**

vpcInputDesc\_ = acldvppCreatePicDesc();

acldvppSetPicDescData(vpcInputDesc\_, inBufferDev); // JpegD -> vpcCropAndPaste

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

acldvppSetPicDescWidth(vpcInputDesc\_, inputWidth);

acldvppSetPicDescHeight(vpcInputDesc\_, inputHeight);

acldvppSetPicDescWidthStride(vpcInputDesc\_, jpegOutWidthStride);

acldvppSetPicDescHeightStride(vpcInputDesc\_, jpegOutHeightStride);

acldvppSetPicDescSize(vpcInputDesc\_, jpegOutBufferSize);

**//7. 申请输出内存vpcOutBufferDev\_，内存大小vpcOutBufferSize\_根据计算公式得出**

acldvppMalloc(&vpcOutBufferDev\_, vpcOutBufferSize\_);

**//8. 创建输出图片的描述信息，并设置各属性值，vpcOutputDesc\_是acldvppPicDesc类型**

**//如果抠图的输出图片作为模型推理的输入，则输出图片的宽高要与模型要求的宽高保持一致**

void \*vpcOutBufferDev\_ = nullptr;

int dvppOutWidth = outputWidth;

int dvppOutHeight = outputHeight;

int dvppOutWidthStride = AlignmentHelper(outputWidth, widthAlignment);;

int dvppOutHeightStride = AlignmentHelper(outputHeight, heightAlignment);;

uint32\_t vpcOutBufferSize\_ =

dvppOutWidthStride \* dvppOutHeightStride \* sizeAlignment / sizeNum;

aclError aclRet = acldvppMalloc(&vpcOutBufferDev\_, vpcOutBufferSize\_);

vpcOutputDesc\_ = acldvppCreatePicDesc();

INFO\_LOG("acldvppCreatePicDesc w=%d/%d,h=%d/%d,vpcOutBufferSize=%d",dvppOutWidth,dvppOutWidthStride,dvppOutHeight,dvppOutHeightStride,vpcOutBufferSize\_);

acldvppSetPicDescData(vpcOutputDesc\_, vpcOutBufferDev\_);

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

acldvppSetPicDescWidth(vpcOutputDesc\_, dvppOutWidth);

acldvppSetPicDescHeight(vpcOutputDesc\_, dvppOutHeight);

acldvppSetPicDescWidthStride(vpcOutputDesc\_, dvppOutWidthStride);

acldvppSetPicDescHeightStride(vpcOutputDesc\_, dvppOutHeightStride);

acldvppSetPicDescSize(vpcOutputDesc\_, vpcOutBufferSize\_);

**//9. 执行异步抠图贴图，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**

acldvppVpcCropAndPasteAsync(dvppChannelDesc\_, vpcInputDesc\_,

vpcOutputDesc\_, cropArea\_, pasteArea\_, stream\_);

ret = aclrtSynchronizeStream(stream\_);

**//10. 解码结束后，释放资源，包括输入/输出图片的描述信息、输入/输出内存**

( (void)acldvppDestroyRoiConfig(cropArea\_);

cropArea\_ = nullptr;

(void)acldvppDestroyRoiConfig(pasteArea\_);

pasteArea\_ = nullptr;

(void)acldvppDestroyPicDesc(vpcInputDesc\_);

vpcInputDesc\_ = nullptr;

(void)acldvppDestroyPicDesc(vpcOutputDesc\_);

vpcOutputDesc\_ = nullptr;

(void)acldvppDestroyChannel(dvppChannelDesc\_);

(void)acldvppDestroyChannelDesc(dvppChannelDesc\_);

dvppChannelDesc\_ = nullptr;

SaveOutputFile(outPicDesc.picName.c\_str(), vpcOutBufferDev\_, vpcOutBufferSize\_);

if (vpcOutBufferDev\_ != nullptr) {

(void)acldvppFree(vpcOutBufferDev\_);

vpcOutBufferDev\_ = nullptr;

}

aclrtDestroyStream(stream\_);

stream\_ = nullptr;

aclrtDestroyContext(context\_);

context\_ = nullptr;

aclrtResetDevice(deviceId\_);

INFO\_LOG("end to reset device is %d", deviceId\_);

aclFinalize();

INFO\_LOG("end to finalize acl");

**3、代码流程图**

