sdk封装需要对封装环境，封装的步骤，要非常熟悉。对C++的语法，还有opencv，进行一些简单了解即可。我总结了一下我遇到一些问题，总结如下：

1.onnx导出

需要安装环境：

```
apt-get update
apt-get install protobuf-compiler libprotoc-dev
pip install onnx=1.11.0
```

执行导出命令

```
cd /project/train/src_repo/yolov5/
python export.py --weights /project/train/models/weights/best.pt   --opset-version  11
--include onnx
```

2.封装的代码修改

主要是对SampleAlgorithm.cpp，SampleAlgorithm.hpp，Yolov5TrtInfer.cpp，Yolov5TrtInfer.hpp进行修改

修改主要是两个地方，一个是对输入的尺寸进行resize，为了便于观察尺寸的变换，我们可以将其进行输出

```
bool Yolov5TrtInfer::doPreprocess(const cv::Mat& cInMat)
{
    m_iInWidth = cInMat.cols;
    m_iInHeight = cInMat.rows;
    //等比例缩放
    cv::Mat cTmpResized;

    if( m_iInWidth >= m_iInHeight )
    {
        m_fRecoverScale = static_cast<float>(m_iInWidth) / m_cModelInputSize.width;
        cv::resize( cInMat, cTmpResized, cv::Size( m_cModelInputSize.width,
m_cModelInputSize.width * static_cast<float>(m_iInHeight) / m_iInWidth) );
        m_iPadDeltaY = (  m_cModelInputSize.height - cTmpResized.rows ) / 2;
        m_iPadDeltaX = 0;
    }
    else
    {
        m_fRecoverScale = static_cast<float>(m_iInHeight) / m_cModelInputSize.width;
        cv::resize( cInMat, cTmpResized, cv::Size( m_cModelInputSize.width *
static_cast<float>(m_iInWidth) / m_iInHeight, m_cModelInputSize.height ) );
        m_iPadDeltaX = ( m_cModelInputSize.width - cTmpResized.cols) / 2;
        m_iPadDeltaY = 0;
    }
    std::cout<<"cTmpResized:"<<cTmpResized.size()<<std::endl;

    //填充
    memset(m_cPasteBoard.data, 114,  m_cModelInputSize.width *
m_cModelInputSize.height * 3 );
    cTmpResized.copyTo(m_cPasteBoard.rowRange(m_iPadDeltaY, m_iPadDeltaY +
cTmpResized.rows).colRange(m_iPadDeltaX, m_iPadDeltaX + cTmpResized.cols));
    std::cout<<"m_cPasteBoard:"<<m_cPasteBoard.size()<<std::endl;
    //BGR2RGB /255 HWC->HWC
    cv::cvtColor(m_cPasteBoard, m_cRGBMat, cv::COLOR_BGR2RGB);
//     cv::imwrite("hh.jpg", m_cRGBMat);
    m_cRGBMat.convertTo(m_Normalized, CV_32FC3, 1/255.);
    cv::split(m_Normalized, m_InputWrappers);
//     std::cout<<"290"<<std::endl;
    return true;
}
```

还有遇到下面的报错是输出头的问题，需要将代码有多个输出的进行注释：

```
Yolov5TrtInfer::Yolov5TrtInfer(const std::string& strModelName)
{
    Logger gLogger;
    //根据tensorrt pipeline 构建网络
    IBuilder* builder = createInferBuilder(gLogger);
    builder->setMaxBatchSize(1);
    const auto explicitBatch = 1U << static_cast<uint32_t>
(NetworkDefinitionCreationFlag::kEXPLICIT_BATCH);
    INetworkDefinition* network = builder->createNetworkV2(explicitBatch);

    nvonnxparser::IParser* parser = nvonnxparser::createParser(*network, gLogger);
    parser->parseFromFile(strModelName.c_str(), static_cast<int>
(ILogger::Severity::kWARNING));
    LOG(INFO) << "model name is " << strModelName;
    IBuilderConfig* config = builder->createBuilderConfig();
    config->setMaxWorkspaceSize(1ULL << 30);

    m_CudaEngine = builder->buildEngineWithConfig(*network, *config);
    m_CudaContext = m_CudaEngine->createExecutionContext();

    // 分配输入输出的空间,DEVICE侧和HOST侧
    m_iInIndex = m_CudaEngine->getBindingIndex( IN_NAME );
    m_iOutIndexS1 = m_CudaEngine->getBindingIndex(OUT_NAME_S1);
    m_iOutIndexS2 = m_CudaEngine->getBindingIndex(OUT_NAME_S2);
    m_iOutIndexS3 = m_CudaEngine->getBindingIndex(OUT_NAME_S3);
    m_iOutIndexS4 = m_CudaEngine->getBindingIndex(OUT_NAME_S4);


    Dims dims_i = m_CudaEngine->getBindingDimensions(m_iInIndex);
    LOG(INFO) << dims_i.d[0] << " " << dims_i.d[1] << " " << dims_i.d[2] << " " <<
dims_i.d[3];
    int size = dims_i.d[0] * dims_i.d[1] * dims_i.d[2] * dims_i.d[3];
    m_cModelInputSize = cv::Size(dims_i.d[3], dims_i.d[2]);
    cudaMalloc(&m_ArrayDevMemory[m_iInIndex], size * sizeof(float));
    m_ArrayHostMemory[m_iInIndex] = malloc(size * sizeof(float));
    //方便NHWC到NCHW的预处理
    m_InputWrappers.emplace_back(dims_i.d[2], dims_i.d[3], CV_32FC1,
m_ArrayHostMemory[m_iInIndex]);
    m_InputWrappers.emplace_back(dims_i.d[2], dims_i.d[3], CV_32FC1,
m_ArrayHostMemory[m_iInIndex] + sizeof(float) * dims_i.d[2] * dims_i.d[3] );
    m_InputWrappers.emplace_back(dims_i.d[2], dims_i.d[3], CV_32FC1,
m_ArrayHostMemory[m_iInIndex] + 2 * sizeof(float) * dims_i.d[2] * dims_i.d[3]);
    m_InputWrappers.emplace_back(dims_i.d[2], dims_i.d[3], CV_32FC1,
m_ArrayHostMemory[m_iInIndex] + 3 * sizeof(float) * dims_i.d[2] * dims_i.d[3]);
    m_ArraySize[m_iInIndex] = size *sizeof(float);




    //output s1
```

```
    dims_i = m_CudaEngine->getBindingDimensions(m_iOutIndexS1);
    LOG(INFO) << dims_i.d[0] << " " << dims_i.d[1] << " " << dims_i.d[2] << " " <<
dims_i.d[3];
    size = dims_i.d[0] * dims_i.d[1] * dims_i.d[2];
    cudaMalloc(&m_ArrayDevMemory[m_iOutIndexS1], size * sizeof(float));
    m_ArrayHostMemory[m_iOutIndexS1] = malloc( size * sizeof(float));
    m_ArraySize[m_iOutIndexS1] = size *sizeof(float);
//     m_vecYoloParams.push_back({ dims_i.d[0],dims_i.d[1],dims_i.d[2],{ {10,13},
{16,30}, {33,23} } });
    m_vecYoloParams.push_back({ dims_i.d[0],dims_i.d[1],dims_i.d[2],dims_i.d[3],{
{10,13}, {16,30}, {33,23} } });
    // //output s2
    // dims_i = m_CudaEngine->getBindingDimensions(m_iOutIndexS2);
    // LOG(INFO) << dims_i.d[0] << " " << dims_i.d[1] << " " << dims_i.d[2] << " " <<
dims_i.d[3]<<dims_i.d[4];
    // size = dims_i.d[0] * dims_i.d[1] * dims_i.d[2] * dims_i.d[3]*dims_i.d[4];
    // cudaMalloc(&m_ArrayDevMemory[m_iOutIndexS2], size * sizeof(float));
    // m_ArrayHostMemory[m_iOutIndexS2] = malloc( size * sizeof(float));
    // m_ArraySize[m_iOutIndexS2] = size *sizeof(float);
    // m_vecYoloParams.push_back({ dims_i.d[0],dims_i.d[1],dims_i.d[2],dims_i.d[3],{
{30,61}, {62,45}, {59,119} } });

    //output s3
    // dims_i = m_CudaEngine->getBindingDimensions(m_iOutIndexS3);
    // LOG(INFO) << dims_i.d[0] << " " << dims_i.d[1] << " " << dims_i.d[2] << " " <<
dims_i.d[3]<<dims_i.d[4];
    // size = dims_i.d[0] * dims_i.d[1] * dims_i.d[2] * dims_i.d[3]*dims_i.d[4];
    // cudaMalloc(&m_ArrayDevMemory[m_iOutIndexS3], size * sizeof(float));
    // m_ArrayHostMemory[m_iOutIndexS3] = malloc( size * sizeof(float));
    // m_ArraySize[m_iOutIndexS3] = size *sizeof(float);
    // m_vecYoloParams.push_back({ dims_i.d[0],dims_i.d[1],dims_i.d[2],dims_i.d[3],{
{116,90}, {156,198}, {373,326} } });
    // cudaStreamCreate(&m_CudaStream);


    //output s4
    // dims_i = m_CudaEngine->getBindingDimensions(m_iOutIndexS4);
    // LOG(INFO) << dims_i.d[0] << " " << dims_i.d[1] << " " << dims_i.d[2] << " " <<
dims_i.d[3];
    // size = dims_i.d[0] * dims_i.d[1] * dims_i.d[2] * dims_i.d[3]*dims_i.d[4];
    // cudaMalloc(&m_ArrayDevMemory[m_iOutIndexS4], size * sizeof(float));
    // m_ArrayHostMemory[m_iOutIndexS4] = malloc( size * sizeof(float));
    // m_ArraySize[m_iOutIndexS4] = size *sizeof(float);
    // m_vecYoloParams.push_back({ dims_i.d[0],dims_i.d[1],dims_i.d[2],dims_i.d[3],{
{116,90}, {156,198}, {373,326} } });
    // cudaStreamCreate(&m_CudaStream);
    m_bUninit = false;


    m_cPasteBoard = cv::Mat(m_cModelInputSize, CV_8UC3, cv::Scalar(128, 128, 128));
    parser->destroy();
```

```
    network->destroy();
    config->destroy();
    builder->destroy();
}
```

```
[W] [TRT] TensorRT was linked against cuDNN 8.1.0 but loaded cuDNN 8.0.5
[W] [TRT] TensorRT was linked against cuDNN 8.1.0 but loaded cuDNN 8.0.5
[W] [TRT] TensorRT was linked against cuDNN 8.1.0 but loaded cuDNN 8.0.5
[E] [TRT] INVALID_ARGUMENT: Cannot find binding of given name: 339
[E] [TRT] INVALID_ARGUMENT: Cannot find binding of given name: 392
[E] [TRT] INVALID_ARGUMENT: Cannot find binding of given name: 445
I0728 14:10:46.001539    74 Yolov5TrtInfer.cpp:50] 1 3 640 640
I0728 14:10:46.002719    74 Yolov5TrtInfer.cpp:67] 1 25200 22 0
[E] [TRT] Parameter check failed at: engine.cpp::getBindingDimensions::2177, condition:
bindIndex >= 0 && bindIndex < getNbBindings()
I0728 14:10:46.002929    74 Yolov5TrtInfer.cpp:76] 0 0 0 00
[E] [TRT] Parameter check failed at: engine.cpp::getBindingDimensions::2177, condition:
bindIndex >= 0 && bindIndex < getNbBindings()
I0728 14:10:46.002950    74 Yolov5TrtInfer.cpp:85] 0 0 0 00
[E] [TRT] Parameter check failed at: engine.cpp::getBindingDimensions::2177, condition:
bindIndex >= 0 && bindIndex < getNbBindings()
I0728 14:10:46.002981    74 Yolov5TrtInfer.cpp:96] 0 0 0 0
I0728 14:10:46.005992    74 ji.cpp:94] [SDKLOG] SamplePredictor init OK.
I0728 14:10:46.006016    74 Algo.cpp:48] sdk mode : 0
```

3.注意训练编码，封装编码的时候，千万不能把/project/train/src_repo，/project/ev_sdk的git信息替换掉，不然后面很多的问题，会让你蚌埠住。比如我下面列举这几个

（1）每天502的错，每天都要重建环境。

（2）训练onnx导出，训练日志会显示训练pt的日志，这种情况下可以查看终端日志，终端日志会显示是否导出成功。

（3）sdk封装环境下修改的代码没同步到调试环境下。

4.报test-ji-api这种错，可以找c++的平台的人员帮忙看看，我这里是他们远程帮忙修改了Configuration.hpp一个地方，但是忘记是哪了。。。

```
E0729 15:25:16.633805  3911 test.cpp:81] *** Aborted at 1659079516 (unix time) try "date -d @1659079516" if you are u
sing GNU date ***
E0729 15:25:16.634519  3911 test.cpp:81] PC: @     0x7fd0a384d562 cfree
E0729 15:25:16.634730  3911 test.cpp:81] *** SIGSEGV (@0x100000019) received by PID 3911 (TID 0x7fd0a6162000) from PI
D 25; stack trace: ***
E0729 15:25:16.635058  3911 test.cpp:81]     @    0x7fd0a5724390 (unknown)
E0729 15:25:16.635322  3911 test.cpp:81]     @    0x7fd0a384d562 cfree
E0729 15:25:16.635617  3911 test.cpp:81]     @    0x7fd0a5d78f20 Configuration::checkAndUpdateVecStr()
E0729 15:25:16.635922  3911 test.cpp:81]     @    0x7fd0a5d7ba92 Configuration::ParseAndUpdateArgs()
E0729 15:25:16.636238  3911 test.cpp:81]     @    0x7fd0a5d75a5a SampleAlgorithm::Init()
E0729 15:25:16.636520  3911 test.cpp:81]     @    0x7fd0a5d83865 ji_create_predictor
E0729 15:25:16.636607  3911 test.cpp:81]     @          0x41157c Algo::Init()
E0729 15:25:16.636670  3911 test.cpp:81]     @          0x40ad22 test_for_ji_calc_image()
E0729 15:25:16.636718  3911 test.cpp:81]     @          0x40c9d6 main
E0729 15:25:16.637024  3911 test.cpp:81]     @    0x7fd0a37e9840 __libc_start_main
E0729 15:25:16.637076  3911 test.cpp:81]     @          0x409269 _start
E0729 15:25:16.637349  3911 test.cpp:81]     @                0x0 (unknown)
/usr/local/ev_sdk/runt.sh: line 23:  3911 Segmentation fault      (core dumped) ./test-ji-api -f 1 -i /home/data/1043
/
```