GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

「嵌入式和IoT」专场

本期议题：ncnn：有着开源初心的AI推理框架

倪辉(nihui) 2021年07月31日

腾讯公司旗下顶级人工智能实验室——优图实验室

GOTC

聚焦计算机视觉

专注人脸识别、图像识别、OCR等领域开展技术研发和行业落地。

推动产业升级

始终专注基础研究、产业落地两条腿走路的战略，助力行业降本增效。

践行科技向善

致力于通过视觉AI技术解决社会问题，帮助社会群体。

腾讯顶级人工智能实验室
腾讯优图实验室于2012年正式成立
我们一直以来专注于视觉技术的研究与落地

腾讯公司旗下顶级人工智能实验室——优图实验室

GOTC

腾讯优图位列榜首

思否2020年度技术品牌影响力企业

Ncnn荣获十大开源新锐项目

InfoQ发布2020年中国技术力量年度榜单

年度卓越人工智能企业及产品

2020年中国科学院《互联网周刊》金I奖

视觉AI进入全球第4，蝉联中国第1

Gartner2021年度《Magic Quadrant for Cloud AI Developer Services》研究报告

最佳产品Top10

量子位2020年度中国人工智能年度评选

介绍 ncnn

GOTC

一个高性能神经网络前向计算框架

- https://github.com/Tencent/ncnn
- 2017年7月对外开源，腾讯优图实验室首个AI类开源项目
- 帮助开发者将AI算法轻松部署到端侧高效执行

  - 跨平台部署，PC/手机/云端/前端
  - 无任何第三方依赖
  - 高性能，汇编级优化，模型量化加速
  - API稳定易用，更新维护积极
  - CI/CD完备，代码健壮
  - 架构简单，方便裁剪和二次开发
  - 社区活跃，衍生开源项目丰富

支持的部署平台和硬件

- 操作系统
  - Windows/Linux/MacOS
  - Android/iOS/其他嵌入式系统
- CPU
  - x86/arm/mips/risc-v/loongarch
- GPU
  - nvidia/amd/intel
  - apple/arm-mali/qcom-adreno
- Webassembly
  - chrome/firefox/safari/edge
  - android-webview/ios-safari

介绍 ncnn

GOTC

支持的深度学习框架和模型

- caffe/mxnet/pytorch(onnx)/darknet/keras/tensorflow(mlir)
  - 经典和常见的CNN: VGG/GoogleNet/ResNet/DenseNet/SENet/FPN/...
  - 轻量级CNN: SqueezeNet/MobileNetV1/V2/V3/ShuffleNetV1/V2/MNasNet/...
  - 人脸检测: MTCNN/RetinaFace/SCRFD...
  - 通用物体检测: Faster-RCNN/R-FCN/...
  - 轻量级物体检测: SSD/SSDLite/YOLOV2/V3/V4/V5/X/NanoDet/...
  - 分割: FCN/PSPNet/UNet/YOLACT/...
  - 人体姿态估计: SimplePose/HRNet...
  - OCR：ChineseOCRLite/PaddleOCR
  - 自然语言处理：seq2seq

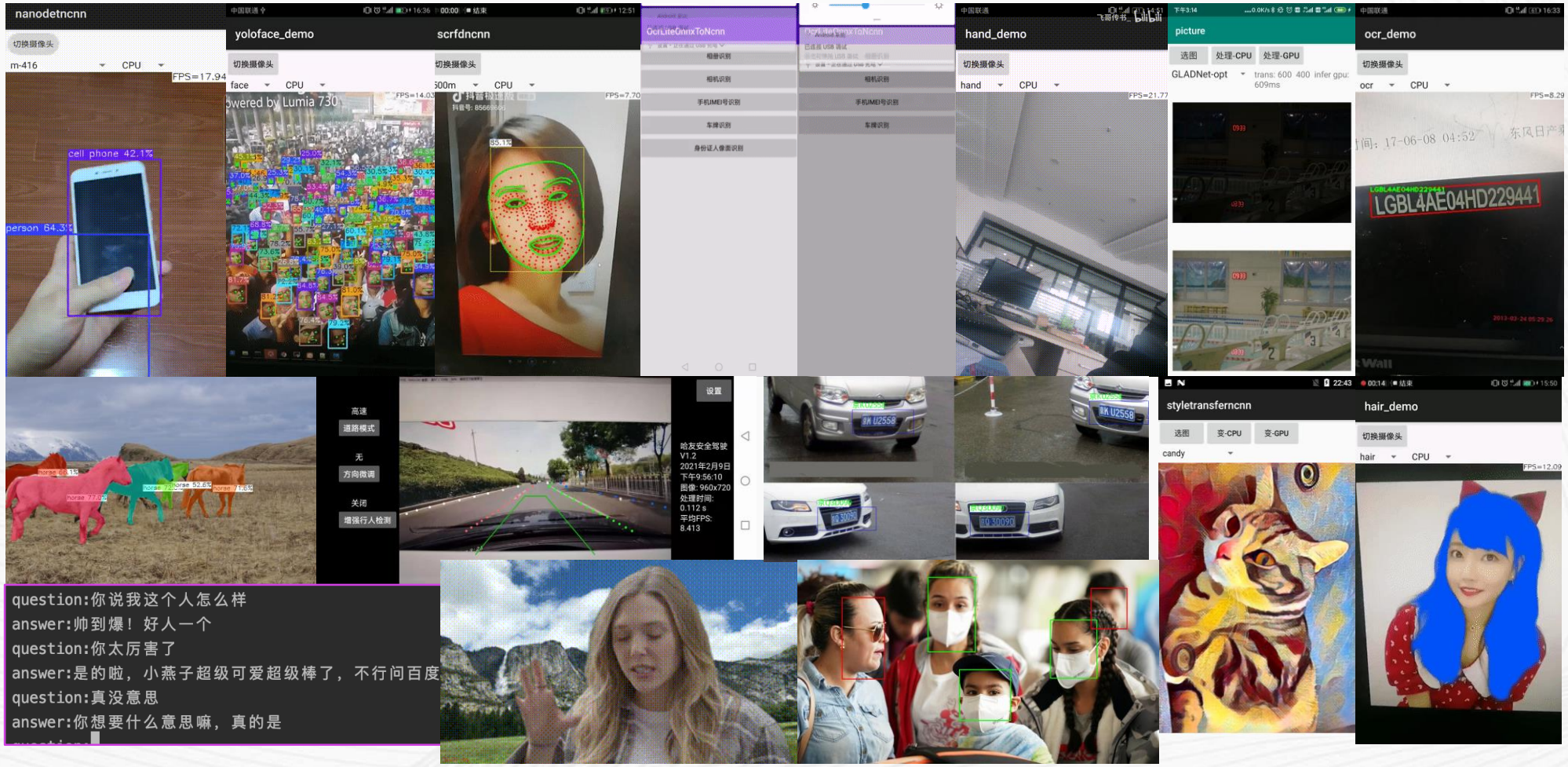
开源应用

- 人脸检测，人脸关键点定位
- 3D建模，人脸识别
- 车辆检测，车牌检测
- 车牌OCR
- 通用物体检测识别，实例分割
- 通用中文OCR
- 人脸属性分类，口罩检测
- 人像分割，头发分割
- 车道线检测
- 疲劳驾驶检测
- 图像超分，视频插帧
- 去雾，增强
- 风格化
- 智能中文对话
- ……

Github ncnn 开源应用精选

GOTC

https://github.com/zchrissirhcz/awesome-ncnn



m5stack unitv2 使用 ncnn + nanodet 工业场景良品筛选

GOTC

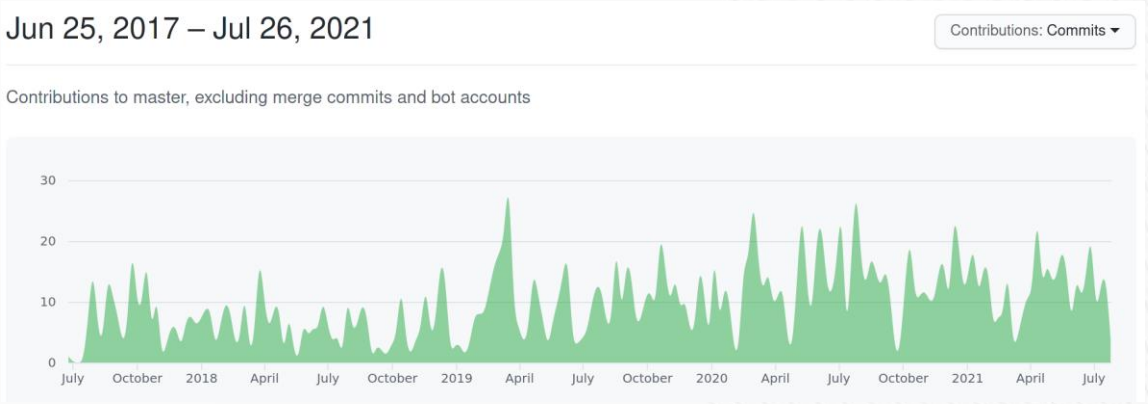
https://www.bilibili.com/video/BV1Vq4y1x7o1

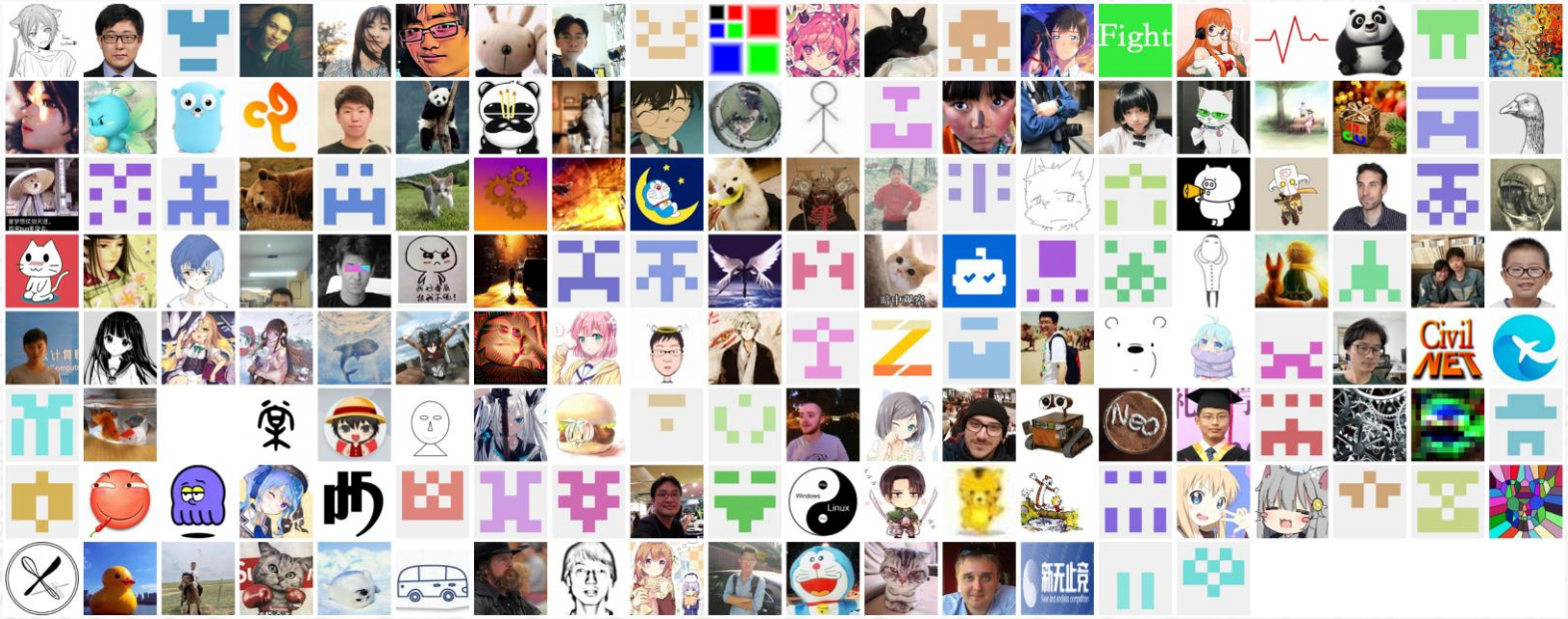
https://www.bilibili.com/video/BV1Vq4y1x7o1

真开源，真赋能

GOTC

中心化的开源赋能	去中心化的开源赋能
单一公司背景	全面社区化
全职开发团队	多企业/个人参与
服务公司业务	探索前沿技术创新
大一统的系统	组件化，嵌入式
中心化的生态	去中心化的生态





我来写，我也行

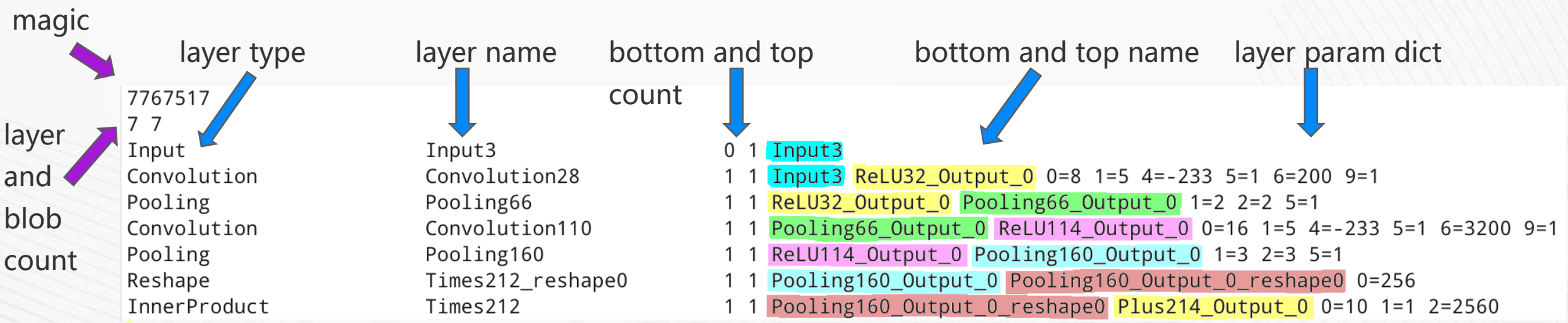
GOTC

加载模型

```
ncnn::Net net;
net.load_param("mnist.param");
net.load_model("mnist.bin");
```

1. read and parse param and bin file
2. create all layer instances
3. foreach layer

   call layer load_param() + load_model()



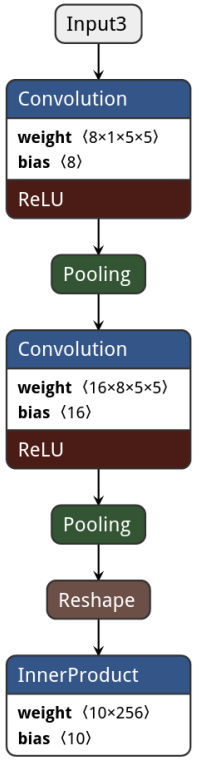
我来写，我也行

GOTC

推理

```
ncnn::Extractor ex = net.create_extractor();
ex.input("data", in);
ex.extract("prob", out);
```

1. solve producer chain
2. call producer layer forward()
3. save output to extractor cache




```
Layer::forward
Convolution::forward
Convolution_arm::forward
conv5x5s2_neon
convolution_sgemm_neon
```

is-A 继承关系

根据输入和参数选择最优化的实现

你的代码很好，现在是我的啦

使用 Github release 编译好的二进制包

https://github.com/Tencent/ncnn/releases

1. 下载解压
2. 配路径

```
project(yolox)
cmake_minimum_required(VERSION 3.10)

find_package(OpenCV REQUIRED)

set(ncnn_DIR "<ncnn_install_dir>/lib/cmake/ncnn")
find_package(ncnn REQUIRED)

add_executable(yolox yolox.cpp)
target_link_libraries(yolox ncnn ${OpenCV_LIBS})
```

自动处理相关依赖

libgomp/libomp 多线程运行库

libvulkan Vulkan API调用库

- ncnn-20210720-android-shared.zip
- ncnn-20210720-android-vulkan-shared.zip
- ncnn-20210720-android-vulkan.zip
- ncnn-20210720-android.zip
- ncnn-20210720-full-source.zip
- ncnn-20210720-ios-bitcode.zip
- ncnn-20210720-ios-vulkan-bitcode.zip
- ncnn-20210720-ios-vulkan.zip
- ncnn-20210720-ios.zip
- ncnn-20210720-macos-vulkan.zip
- ncnn-20210720-macos.zip
- ncnn-20210720-ubuntu-1604-shared.zip
- ncnn-20210720-ubuntu-1604.zip
- ncnn-20210720-ubuntu-1804-shared.zip
- ncnn-20210720-ubuntu-1804.zip
- ncnn-20210720-ubuntu-2004-shared.zip
- ncnn-20210720-ubuntu-2004.zip
- ncnn-20210720-webassembly.zip
- ncnn-20210720-windows-vs2015-shared.zip
- ncnn-20210720-windows-vs2015.zip
- ncnn-20210720-windows-vs2017-shared.zip
- ncnn-20210720-windows-vs2017.zip
- ncnn-20210720-windows-vs2019-shared.zip
- ncnn-20210720-windows-vs2019.zip

你的代码很好，现在是我的啦

GOTC

ncnn 源代码文件夹，成为项目代码的一部分

```
project(yolox)
cmake_minimum_required(VERSION 3.10)

find_package(OpenCV REQUIRED)

option(NCNN_DISABLE_RTTI "" ON)
option(NCNN_DISABLE_EXCEPTION "" ON)

option(WITH_LAYER_absval "" OFF)
option(WITH_LAYER_argmax "" OFF)

add_subdirectory(ncnn)

add_executable(yolox yolox.cpp)
target_link_libraries(yolox ncnn ${OpenCV_LIBS})
```

添加 ncnn 文件夹前

设置 ncnn 编译参数

裁剪不需要的层

https://github.com/Tencent/ncnn

1. 下载 ncnn 源代码
2. 放到项目里

自动与项目一起编译 ncnn 源代码

你的代码很好，现在是我的啦

GOTC

我只想调用下 ncnn 优化好的算子

```cpp
#include <ncnn/layer.h>

ncnn::Mat inout(2);
inout[0] = 33.f;
inout[1] = -2.f;
{
    ncnn::Layer* softmax = ncnn::create_layer("Softmax");

    ncnn::ParamDict pd;
    pd.set(0, 0); // axis = 0
    softmax->load_param(pd);

    softmax->create_pipeline(opt);

    softmax->forward_inplace(inout, opt);

    softmax->destroy_pipeline(opt);

    delete softmax;
}
```

1. 创建算子
2. 加载参数
3. 加载权重
4. 创建pipeline
5. 调用forward
6. 清理

你的代码很好，现在是我的啦

GOTC

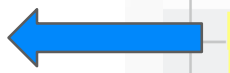
我想把 ncnn 优化好的代码，拿过来，成为我项目里的代码

加载模型时，卷积权重 winograd 预变换

- 输入 kernel，fp32，outch-inch-kh-kw
- 输出 kernel_tm_pack8，conv3x3 stride1 pack8 fp16 专用

```
static void conv3x3s1_winograd64_transform_kernel_pack8_fp16sa_neon(
    const Mat& kernel, Mat& kernel_tm_pack8, int inch, int outch);

static void conv3x3s1_winograd64_pack8_fp16sa_neon(
    const Mat& bottom_blob, Mat& top_blob,
    const Mat& kernel_tm, const Mat& _bias, const Option& opt);
```



> 主文件夹 > osd > ncnn > src > layer > arm

名称

- convolution_3x3_pack4to1_bf16s.h
- convolution_3x3_pack4to1.h
- convolution_3x3_pack8_fp16s.h
- convolution_3x3_pack8to1_fp16s.h
- convolution_3x3_pack8to1_int8.h
- convolution_3x3_pack8to4_fp16s.h
- convolution_3x3_pack8to4_int8.h

模型推理时，卷积计算，使用上面预变换权重 kernel_tm

- 输入 bottom_blob，fp16，inch/8-h-w，pack8
- 输出 top_blob，fp16，outch/8-h-w，pack8

https://github.com/Tencent/ncnn/wiki/element-packing

pack8：c-h-w 内存布局变换为 c/8-h-w-8

- 提升跨channel数据访问连续性
- SIMD友好，数据对齐，与vector寄存器宽度一致

模型是商业机密

GOTC

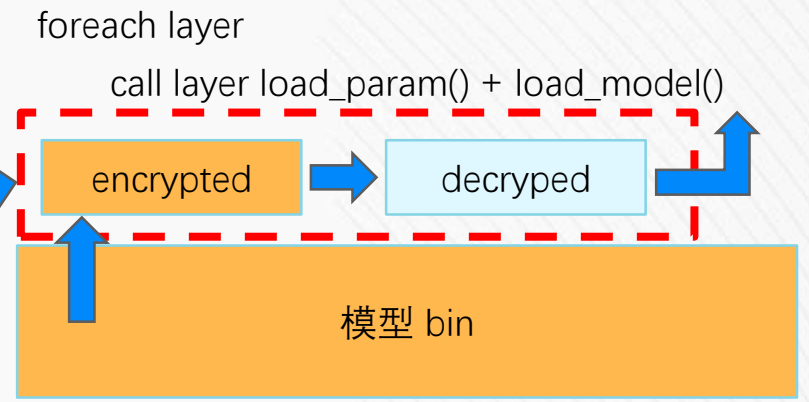
重点：内存中任何一刻都不能存有完整的明文模型

错误示例

- 使用 ncnn2mem 工具把模型转换为代码编译进二进制程序中
  - 查找二进制 magic 即可获得模型存储位置
- 使用第三方库（如 OpenSSL）离线加密，运行时解密，使用内存加载API
  - 运行时监测内存分配，查找 magic 即可获得整块模型内存

```
#include "datareader.h"

class MyEncryptedDataReader : public ncnn::DataReader
{
public:
    MyEncryptedDataReader(const char* filepath, unsigned char _key);
    ~MyEncryptedDataReader();
    virtual size_t read(void* buf, size_t size) const;
private:
    FILE* fp;
    unsigned char key;
};
```



- ✓ 碎片化数据读取
- ✓ 边解密边加载
- ✓ 完全自定义读的方式

模型是商业机密

GOTC

不加密便是最好的加密

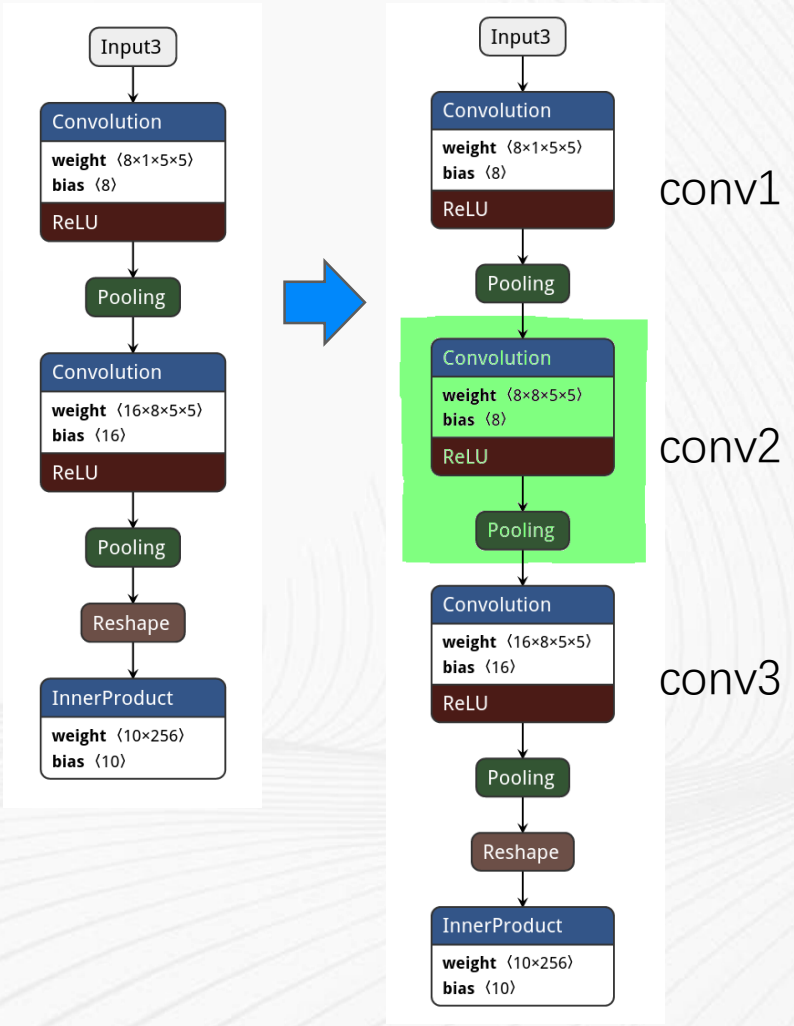
开源项目的破解，对于高手是比较容易的

- 更多的自定义op：MyBlock MyAttention MyNormailization
- 更多的无效模型层数：增加无用分支，增加无用层
- 提高破解难度：编译器混淆，减少导出的符号，减少常量字符串

```
ncnn::Extractor ex = net.create_extractor();
ex.input("data", in);
ex.extract("conv1", conv1);
ex.input("conv2", conv1);
ex.extract("prob", out);
```

将 conv1 输出当作 conv3 输入

跳过无用的 conv2 部分



Q & A 自问自答

GOTC

为什么不支持xxx的npu？

- 特制化芯片 vs 统一标准
- 固定 vs 可编程
- Android NNAPI & Apple CoreML
- OpenVX，Vulkan Machine Learning extension

怎么看xxx框架速度更快？

- 为业务模型、部署硬件特制优化
- 框架的可维护性，稳定性，开源模式

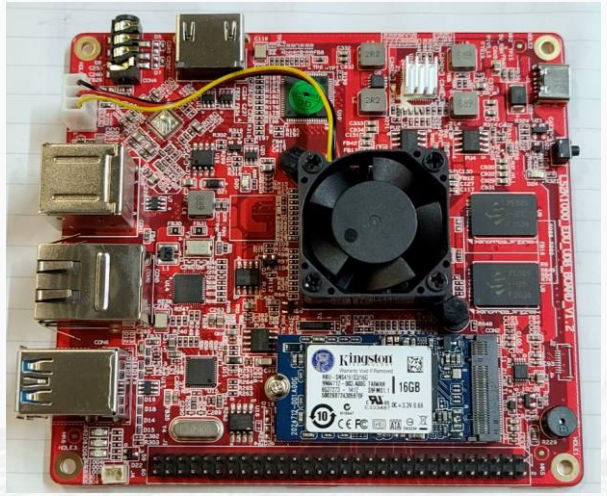
ncnn近期有何开发计划？

- 改善nlp和语音部署
- 更多CV功能
- 移植和优化我手上有的硬件

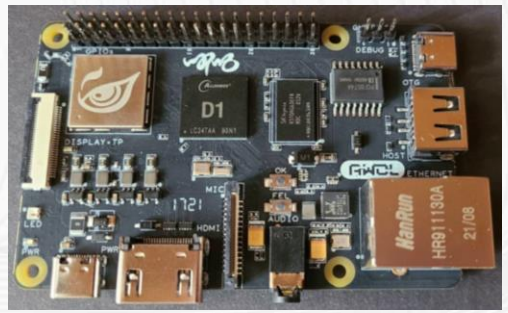
周易AIPU

K210

ESP32-C3



龙芯2K派



全志D1

Q & A

GOTC

技术交流社区

https://github.com/Tencent/ncnn

入群暗号：卷卷卷卷卷

GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE