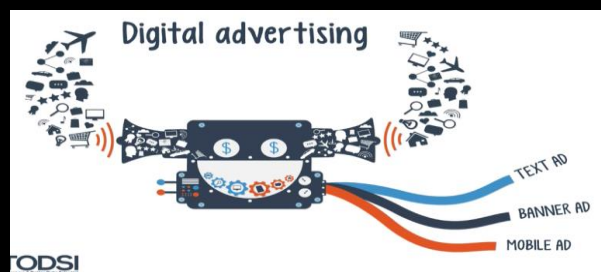
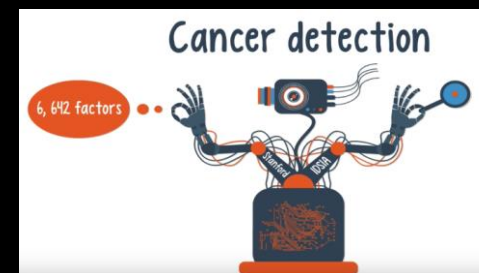
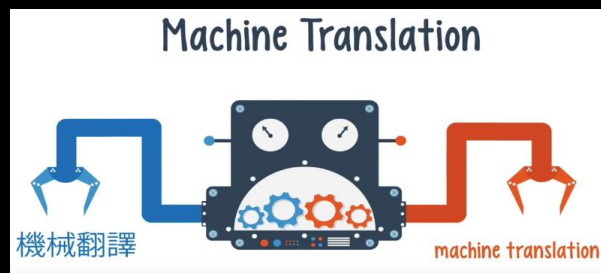


# 基于Jetson的深度学习优化及应用

主讲人：宋健明

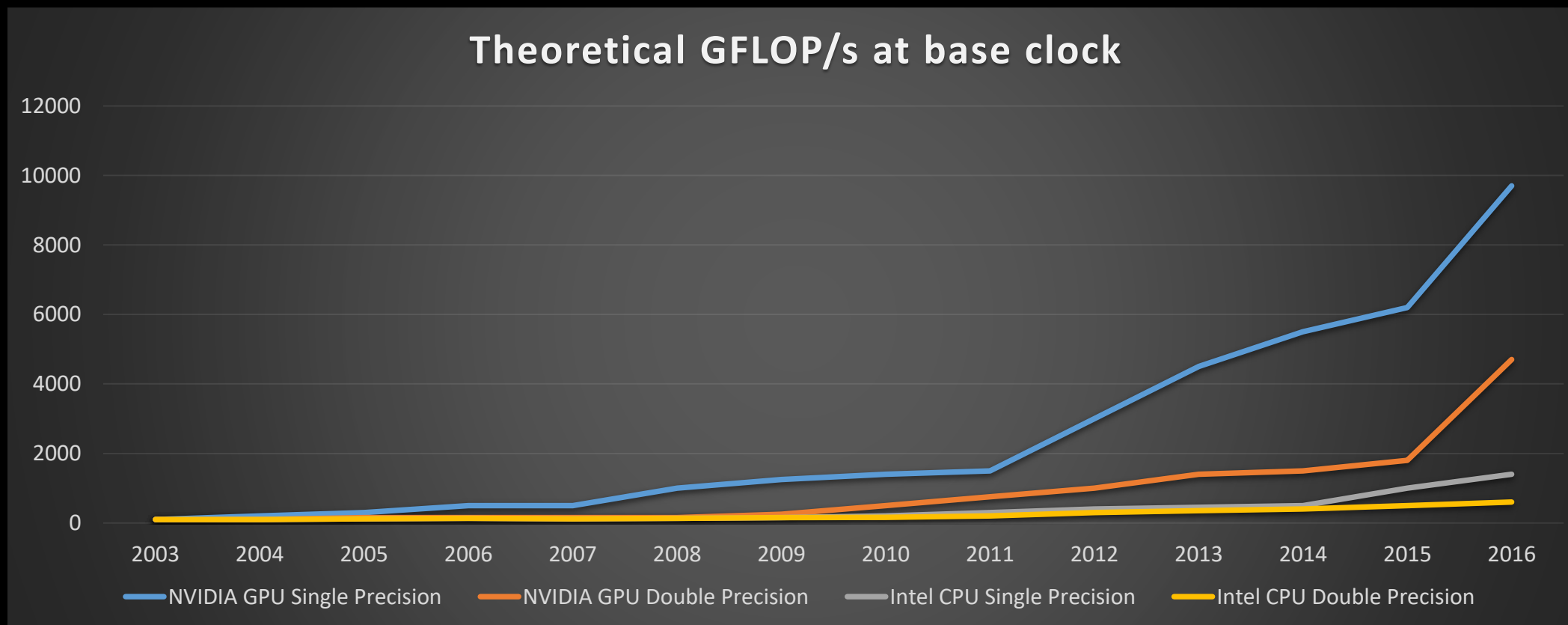
# 深度学习的广泛应用



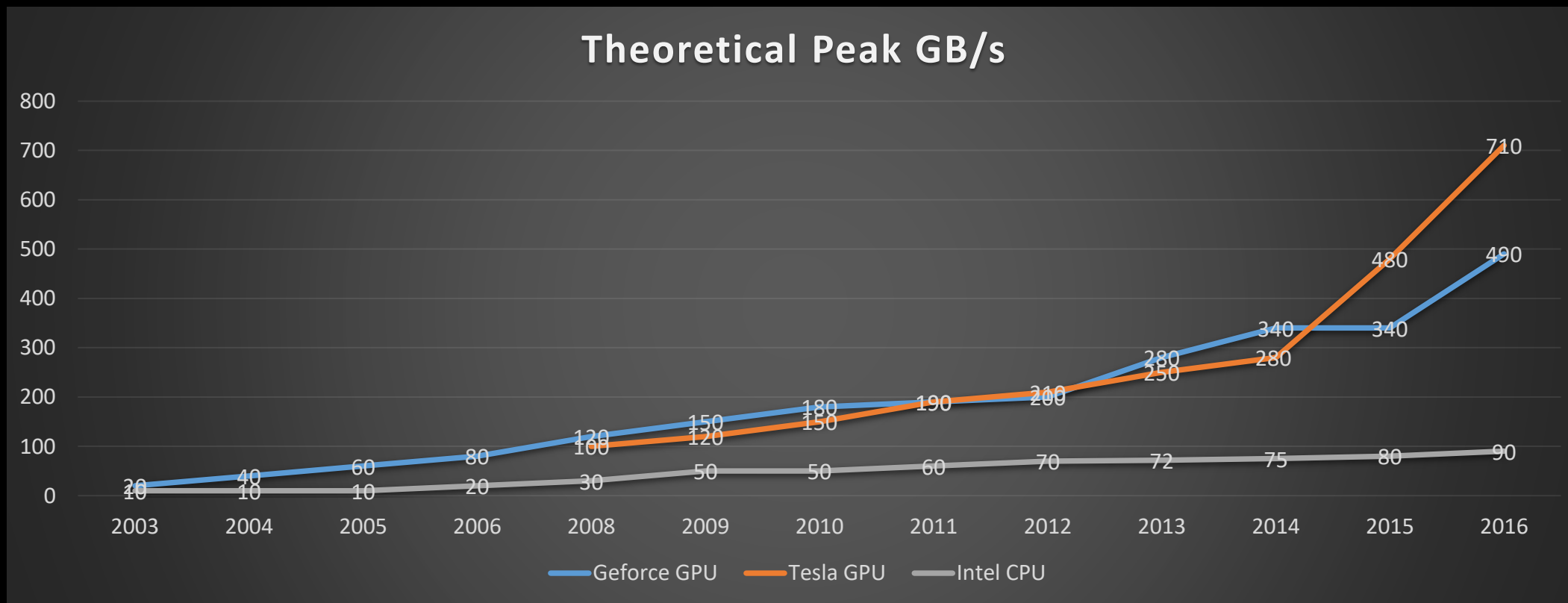
# 深度学习的问题

- 计算量大，需要使用高性能的硬件
- 模型体积大，参数多
- 无法在嵌入式设备中使用

# 深度学习与GPU

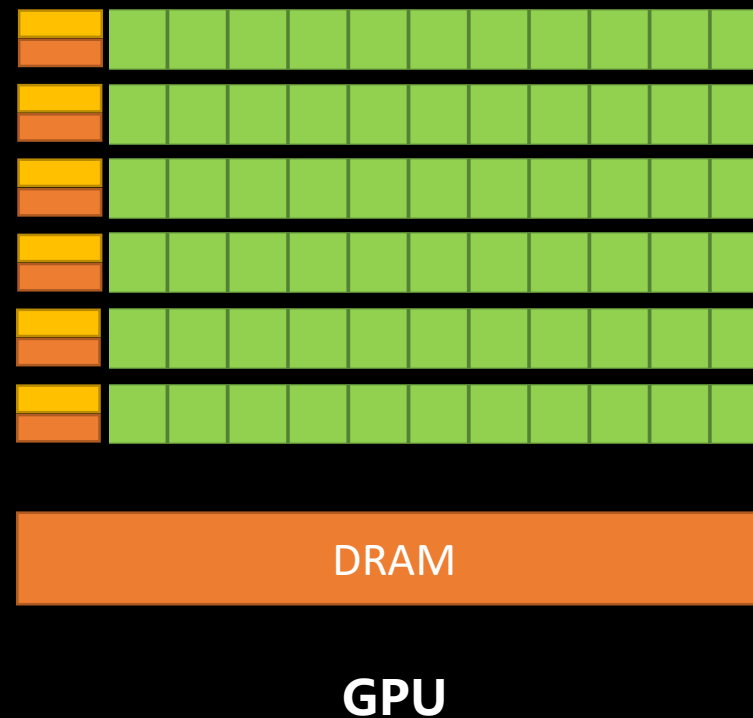
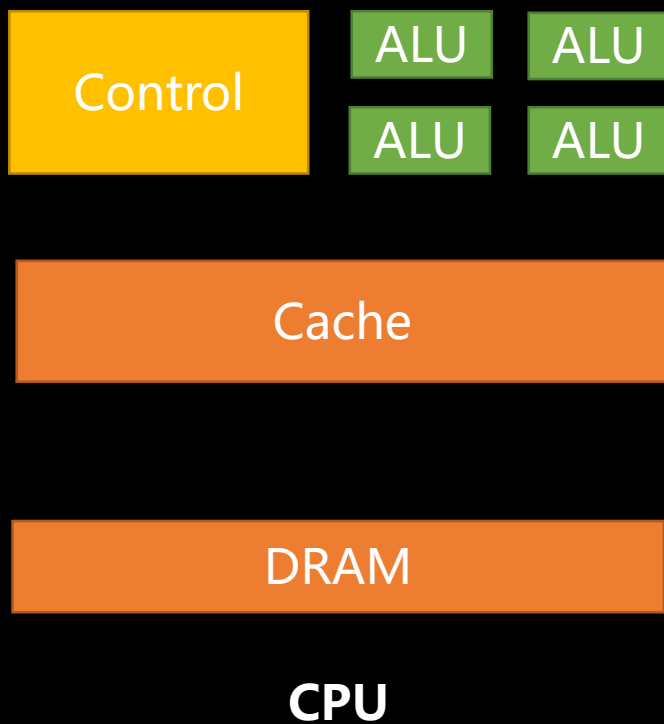


# 深度学习与GPU



# 深度学习与GPU

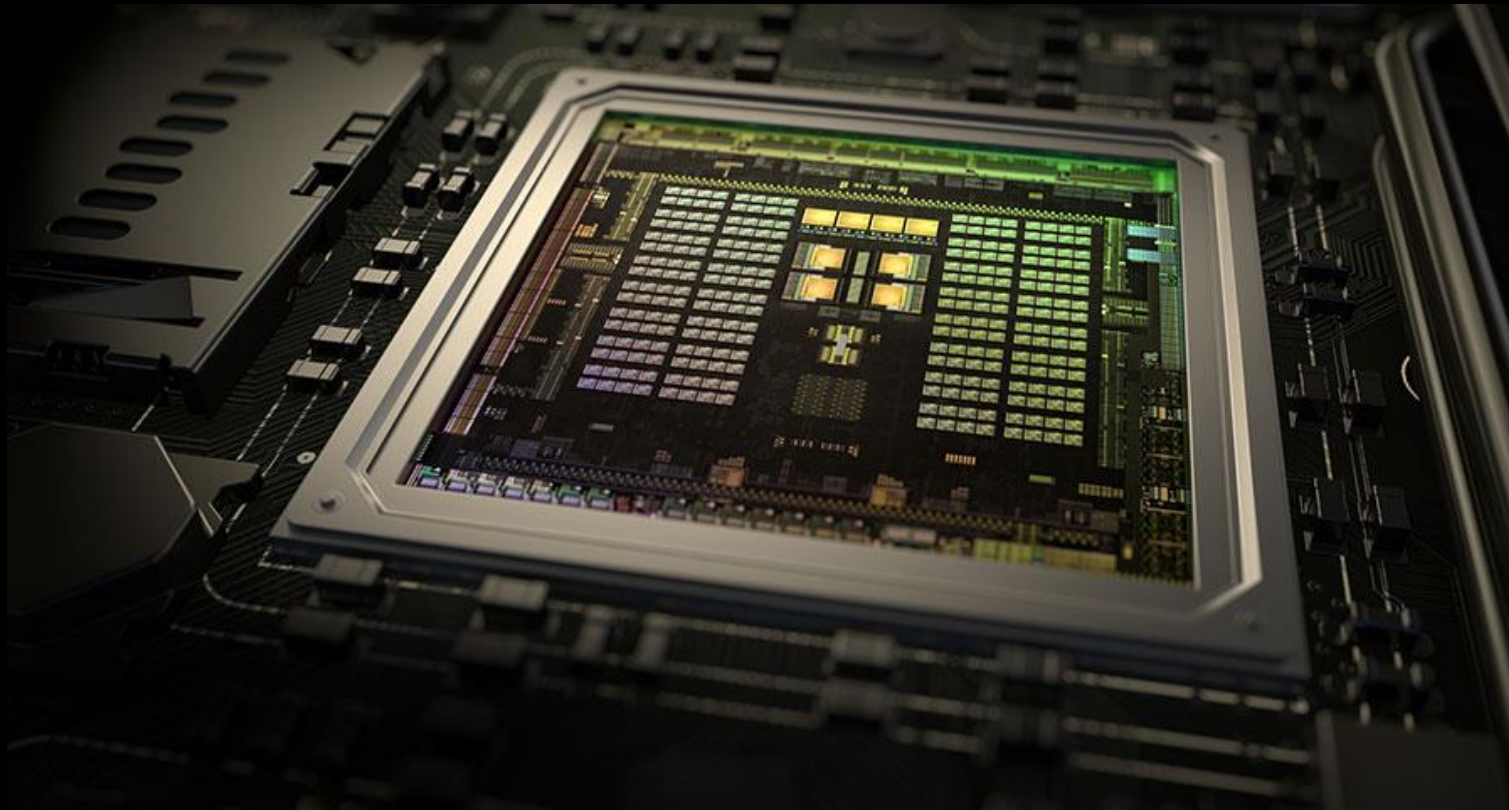
- CPU与GPU的区别



# GPU特点

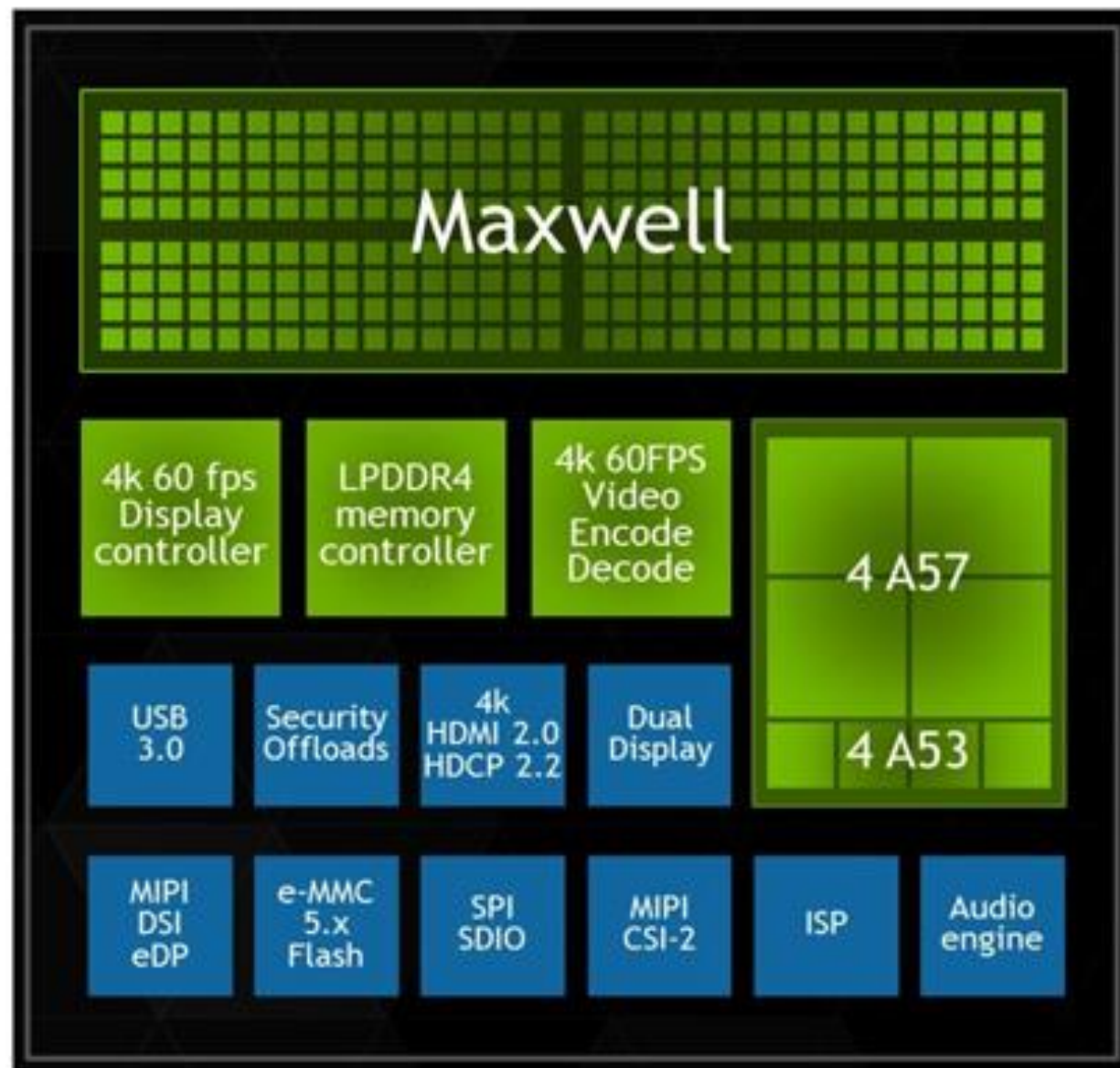
- 1、计算单元多
- 2、控制单元少
- 3、每一个计算单元有独立的cache，寄存器数量多
- 4、创建线程消耗资源少
- 5、使用高速内存

# 深度学习在嵌入式平台中的应用





# Jetson 平台架构



# 深度学习在嵌入式平台应用的挑战

- 1、模型体积大，参数多，计算量大
- 2、全浮点运算
- 3、占用内存多
- 4、计算延时大

# 深度学习-模型优化

## 重构模型

- 1、减小filter大小，使用1x1卷积代替大卷积核
- 2、使用卷积替代pooling
- 3、减少每层的输出节点的数量，但是相应需要加深网络的层数
- 4、使用CRelu激活函数，进一步减少每层的输出节点的数量
- 5、使用分层子网络(Inception结构)，提高网络对不同尺度物体的泛化能力，同时减少参数
- 6、使用Hypernet

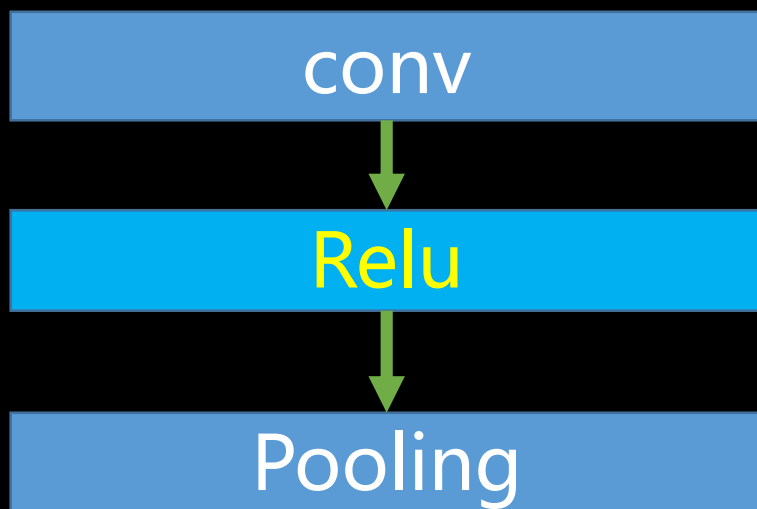
# 深度学习-模型优化

## 1x1卷积核优点

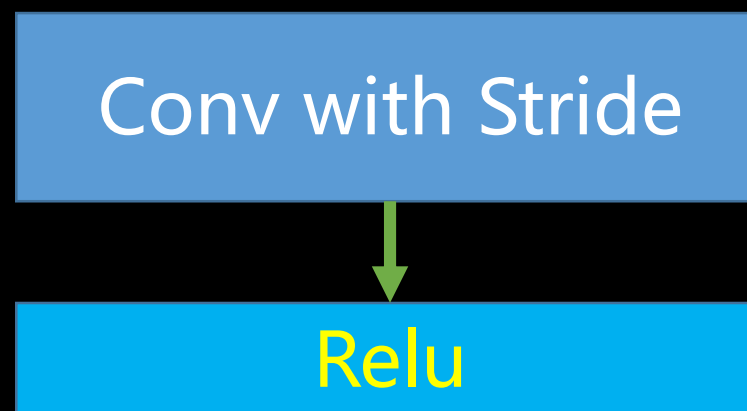
- 1、实现跨通道的交互核信息整合
- 2、实现卷积核通道的降维核升维
- 3、对于单通道的特征图核单个卷积核，1x1卷积类似与对特征图进行比例缩放
- 4、降低计算量

# 深度学习-模型优化

## 卷积替代pooling



标准卷积层

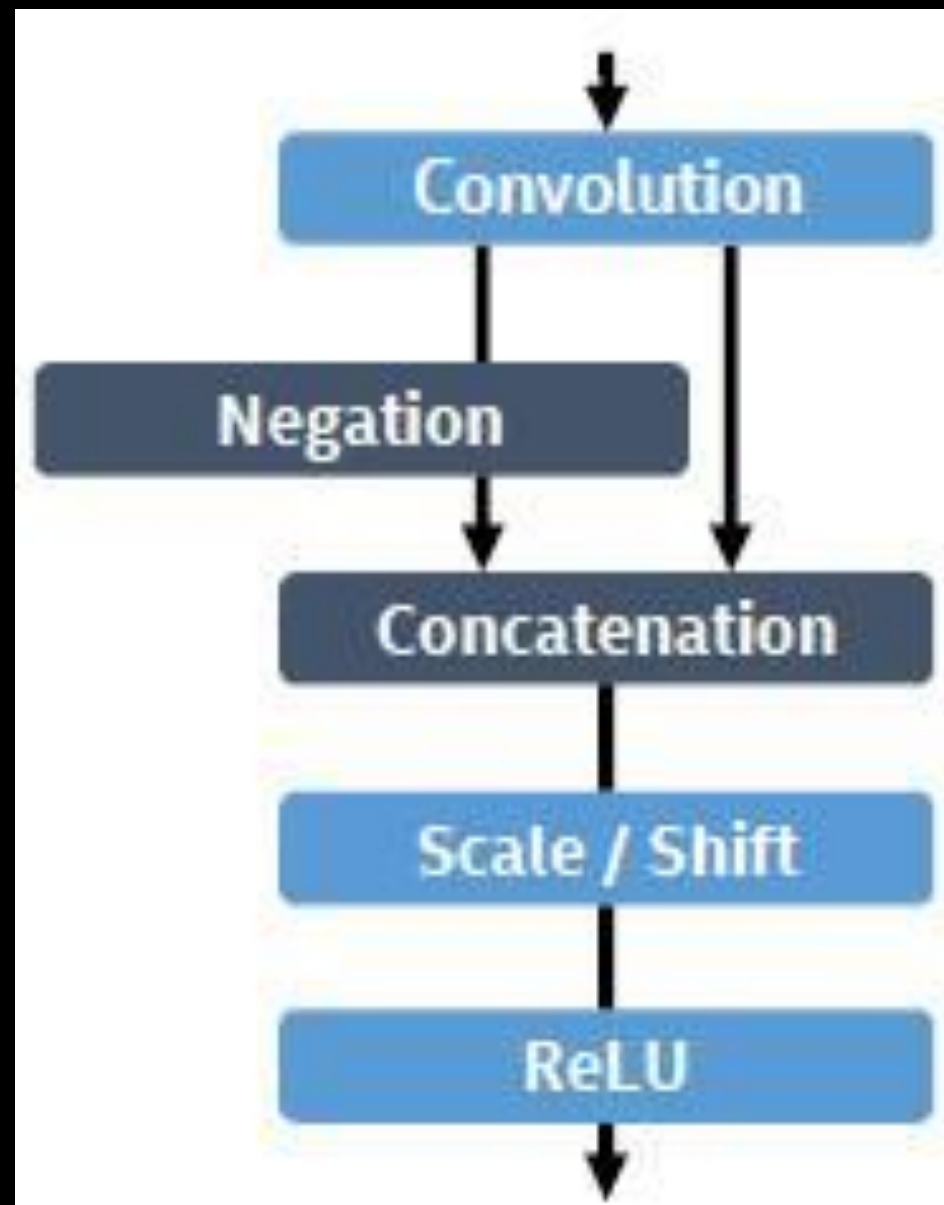


卷积替代Pooling

# 深度学习-模型优化

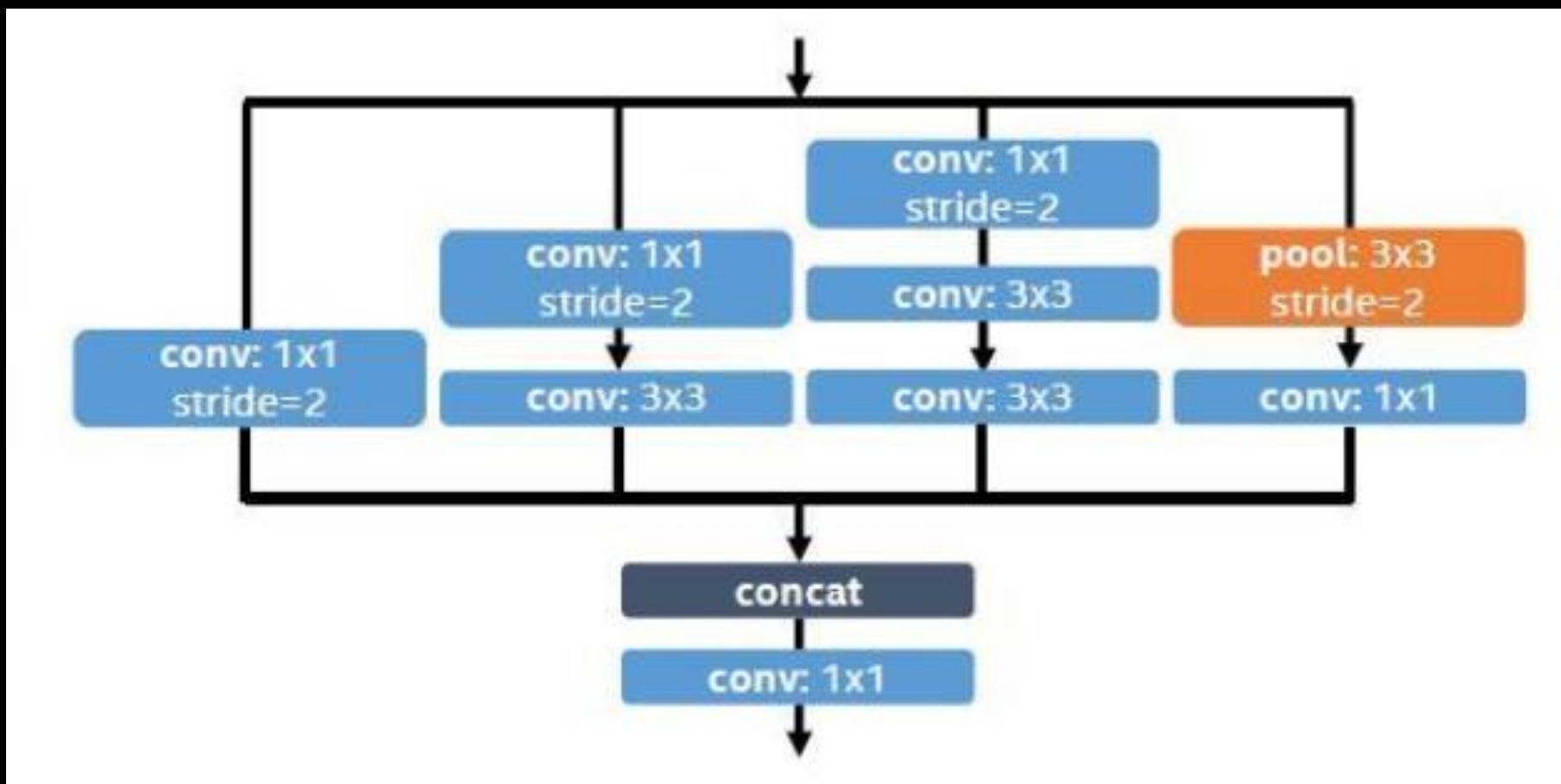
## CRelu激活

$$CReLU(x) = ([x]_+, [-x]_+)$$



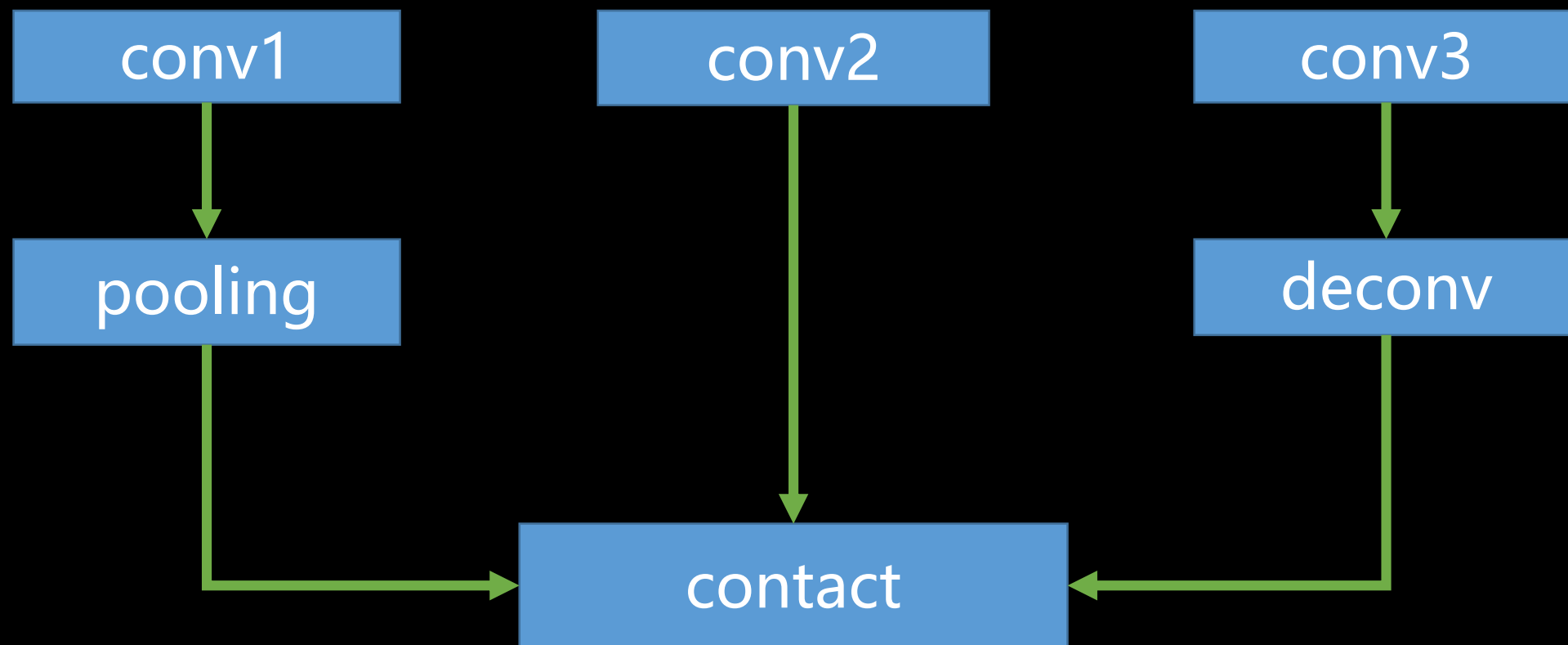
# 深度学习-模型优化

## Inception结构



# 深度学习-模型优化

## HyperNet结构





# 深度学习-模型优化

## 参数压缩

- 1、使用svd算法，压缩全连接层
- 2、使用剪枝算法，将小于权值的节点直接移除

# 深度学习-模型优化

## 效果

- 1、参数降低为原来的1/30
- 2、速度提升10-20倍
- 3、性能下降不到1%，可以基本忽略性能损失

# 深度学习-平台优化

## Jetson平台特点

- 1、内存大
- 2、显存共用内存
- 3、内存传输速率低
- 4、搭载GPU加速芯片，支持FP16指令

# 深度学习-平台优化

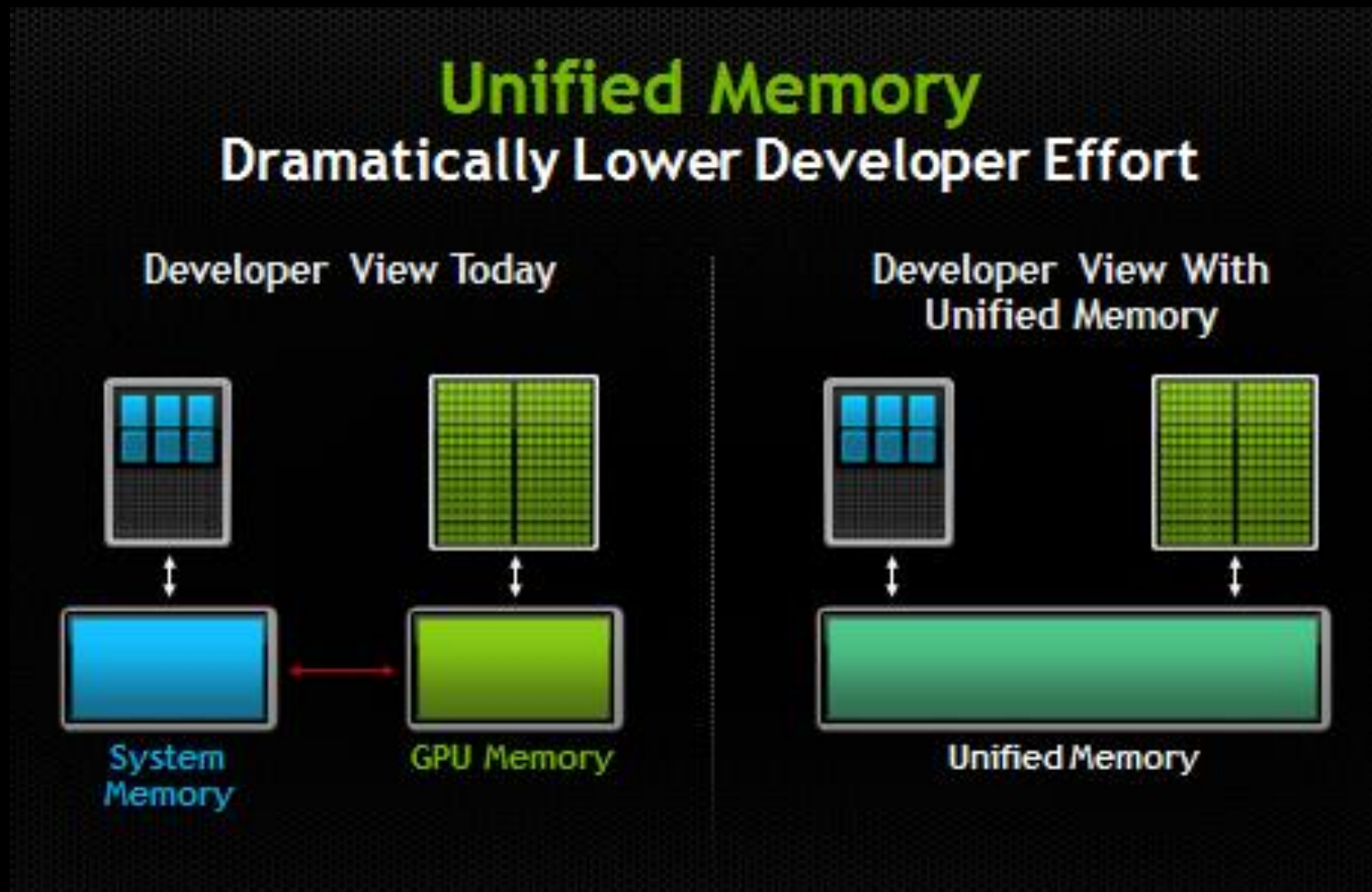
## 优化注意事项

- 1、减少内存和显存的数据同步
- 2、尽量使用GPU
- 3、使用CUDA流操作和异步操作
- 4、使用FP16指令

# 深度学习-平台优化

## 数据流优化

- 1、使用unified memory或者cudaMemcpyAsync



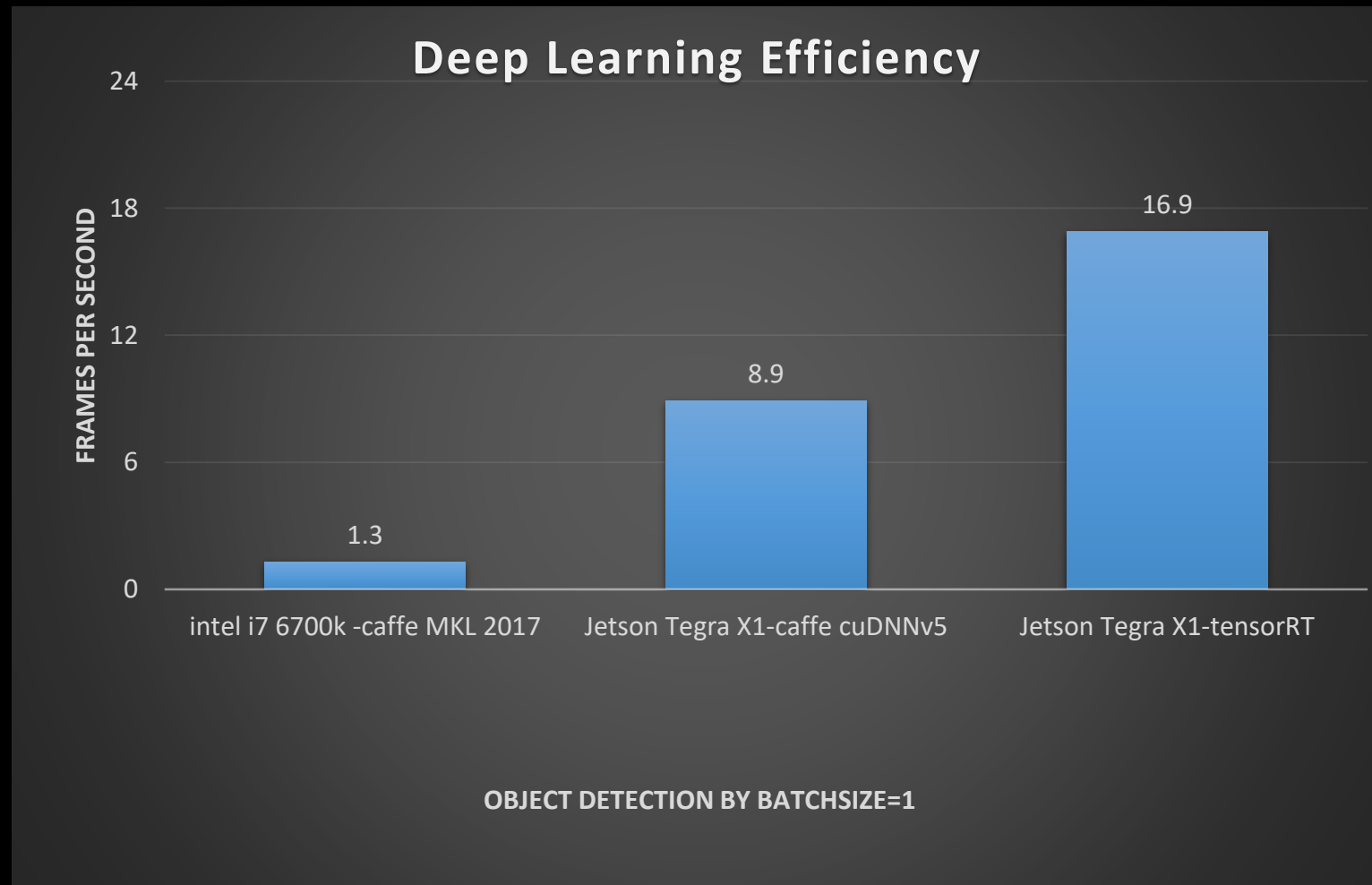
# 深度学习-平台优化

- 2、由于图像传感器数据需要预处理，可将数据通过DMA预载入GPU做预处理，减少数据同步次数
- 3、使用cuda Stream流操作，使用异步函数调用，减少同步次数
- 4、关键算法GPU并行化，减少数据同步，提升速度

# 深度学习-平台优化

## FP16指令集优化

- 1、提升1倍计算速度
- 2、减少GPU资源使用
- 3、降低GPU功耗



# 深度学习-平台优化

● 如何使用FP16：

- 1、CUDA内建函数，直接调用FP16指令
- 2、使用cublas调用FP16操作，需要转换数据类型
- 3、使用cudnn调用FP16封装函数，需要转换数据类型
- 4、使用tensorRT，不需要转换数据类型



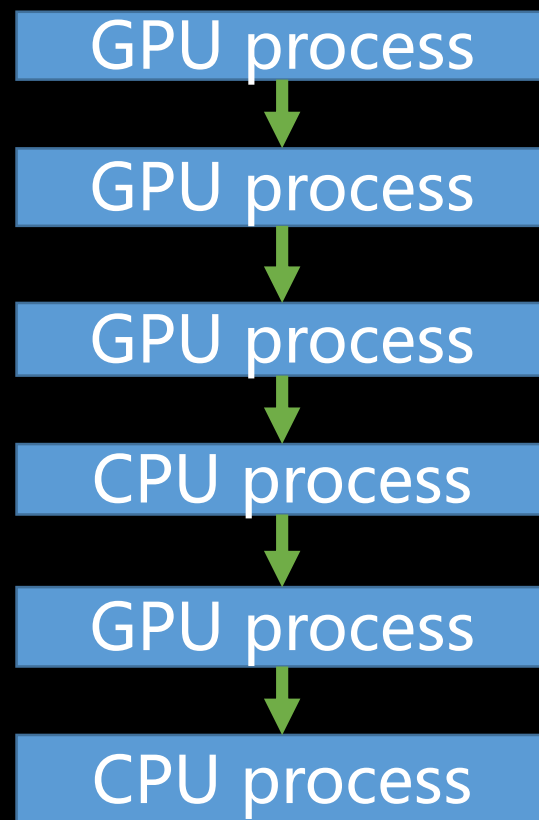
# 深度学习-平台优化

## 总结

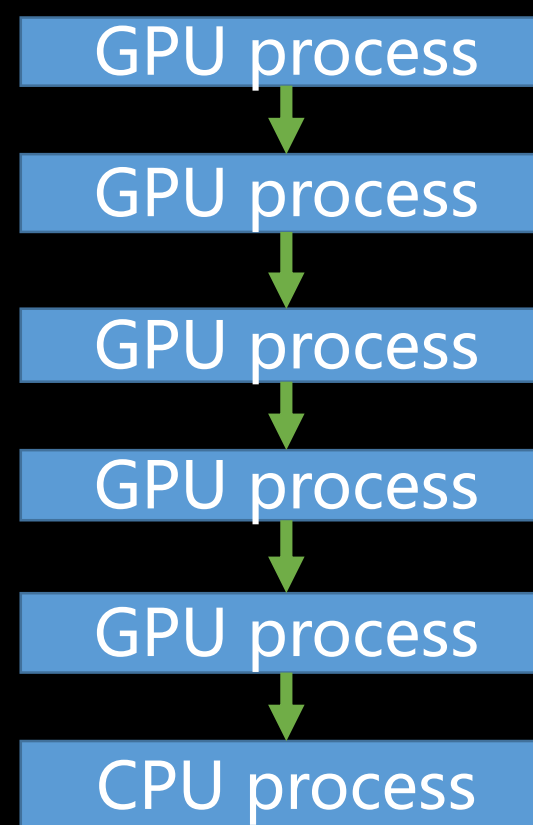
- 1、速度提升100%
- 2、GPU占用率降低20%
- 3、GPU平均温度降低5度

# 深度学习-平台优化

超长流水线：



**错误的流水**



**正确的流水**

# 深度学习-平台优化

超长流水线注意事项：

- 1、使用CUDA stream操作以及异步操作，减少延迟
- 2、使用多条流水线，降低数据同步的延迟
- 3、载入核函数的开销较大，尽量让一个核函数有更多的操作

# Cal mCar 案例展示

# CaImCar 智能车载视觉系统-主要功能

深度学习，机器视觉，小型化嵌入式系统

精准感知、预测车辆行驶过程中的交通情况

- 驾驶行为实时分析
- 驾驶场景实时分析



# CalmCar 智能车载视觉系统-架构

## CalmCar系统

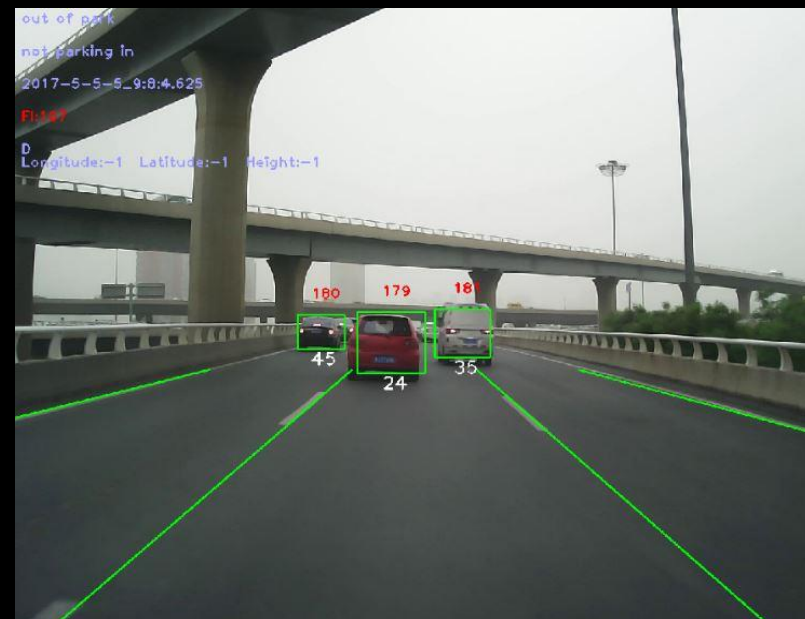
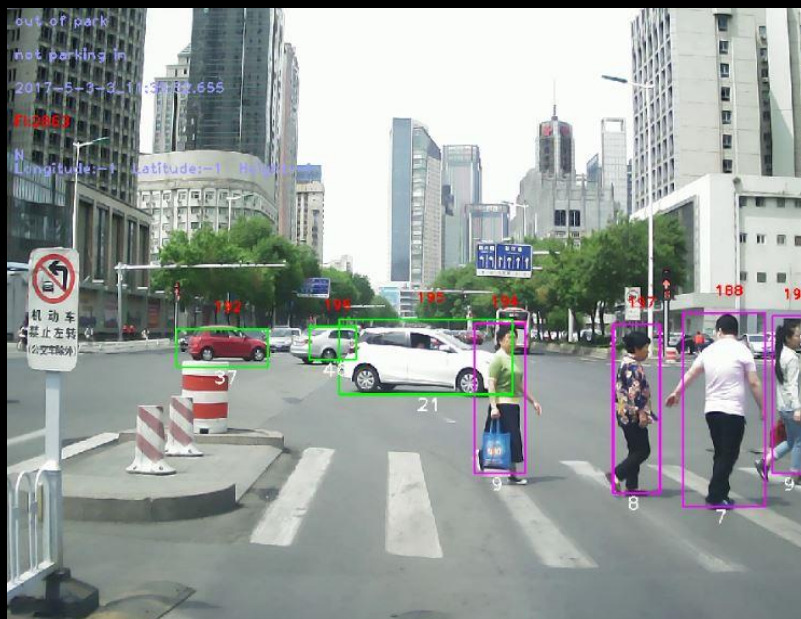
车载线束 (Harness)

图像传感器 (Image Sensor)

嵌入式处理单元 (ECU)

人机交互 (HMI)

Objects Level 存储分析设备



# CaImCar 智能车载视觉系统-路试验证

## 准感知分

类：  
轿车

工程车

公交车

自行车

行人

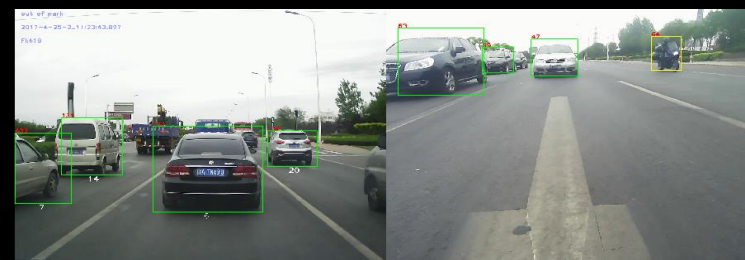
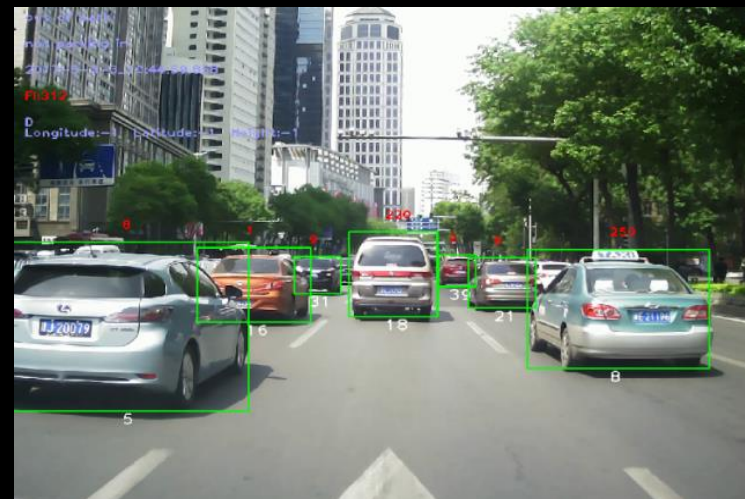
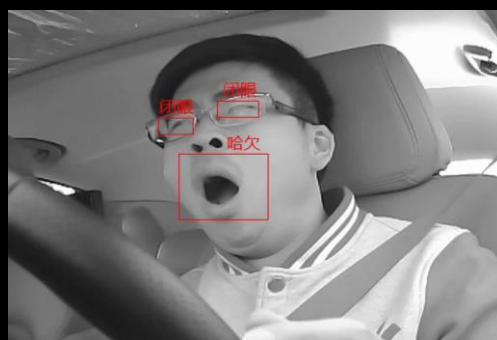
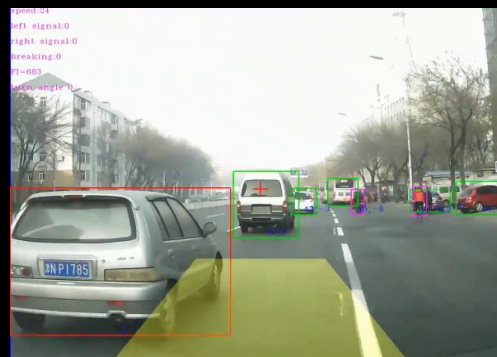
行车线

追踪车辆车牌识别

疲劳检测

限速牌，P牌识别

可行驶区域识别





# CaImCar 智能车载视觉系统-路试验证

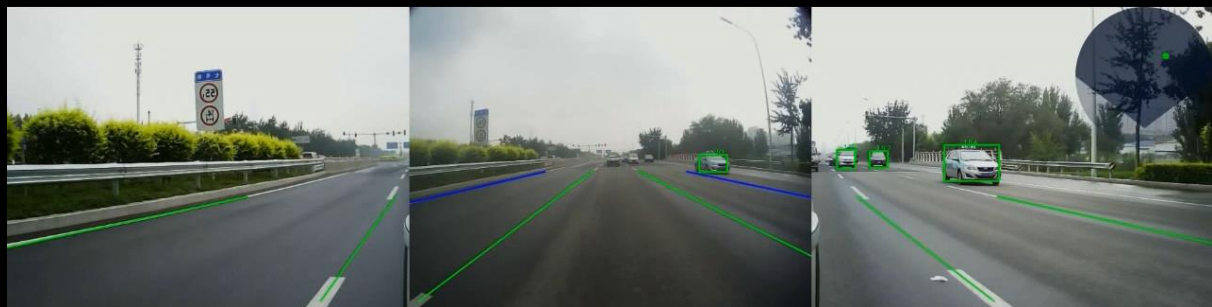
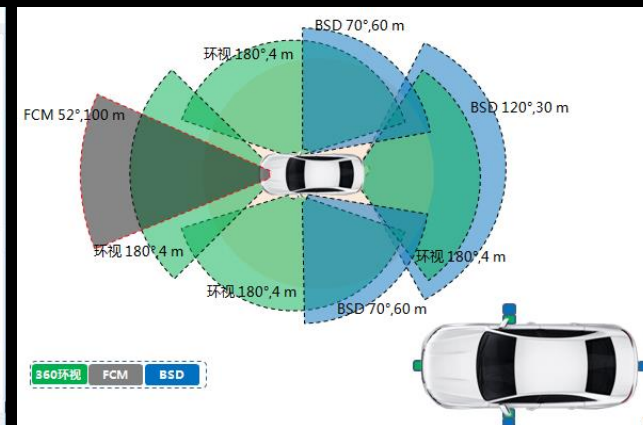
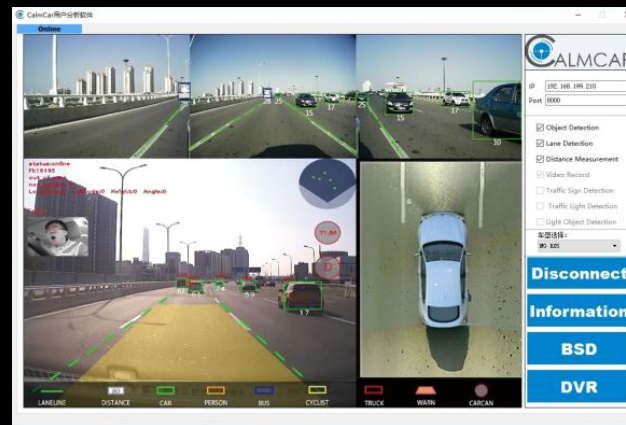
- 本车CAN信号 行驶速度、档位、发动机转速、雨刷信息、方向盘转角、转向灯、油门踏板、刹车等信息
- 前方物体信息 距离、相对速度、相对速度标准差、碰撞时间、检测物体类别、横向距离、物体宽度、行车线归类、前车车道变换方式等信息
- 目标信息 前方车辆切入ID、前方车辆切入碰撞时间、碰撞预警信号、自动紧急刹车预警、前方目标数量、左侧切入区域、右侧切入区域等





# CaImCar 智能车载视觉系统-数据采集与路试验证

- CalmCar 4相机与8相机架构
- 视觉传感器，GPS，MEMS加速度传感器，CAN信号高度同步
- 物体（Objects）与目标（Targets）实时采集与分析





---

**Thank  
you**

---