CEVA

CEVA TECHNOLOGY
SYMPOSIUM SERIES

12th YEAR

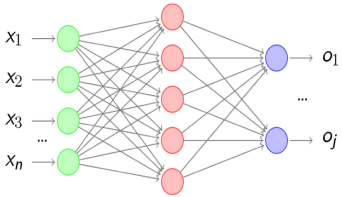
Embedded 3D intellisense & edge AI computing

嵌入式3D智能感知和边缘AI计算

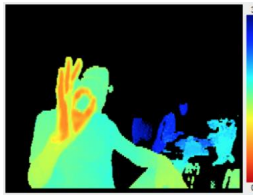
Redong Ying, Associate Professor,
School of Electronic Information and Electrical Engineering,
Shanghai JiaoTong University

内容概要

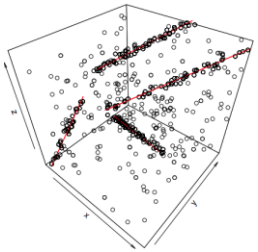
机器视觉是当前AI技术爆发的一个中心



视觉AI对CPU的运算要求



3D传感器为视觉AI带来了了那些变革

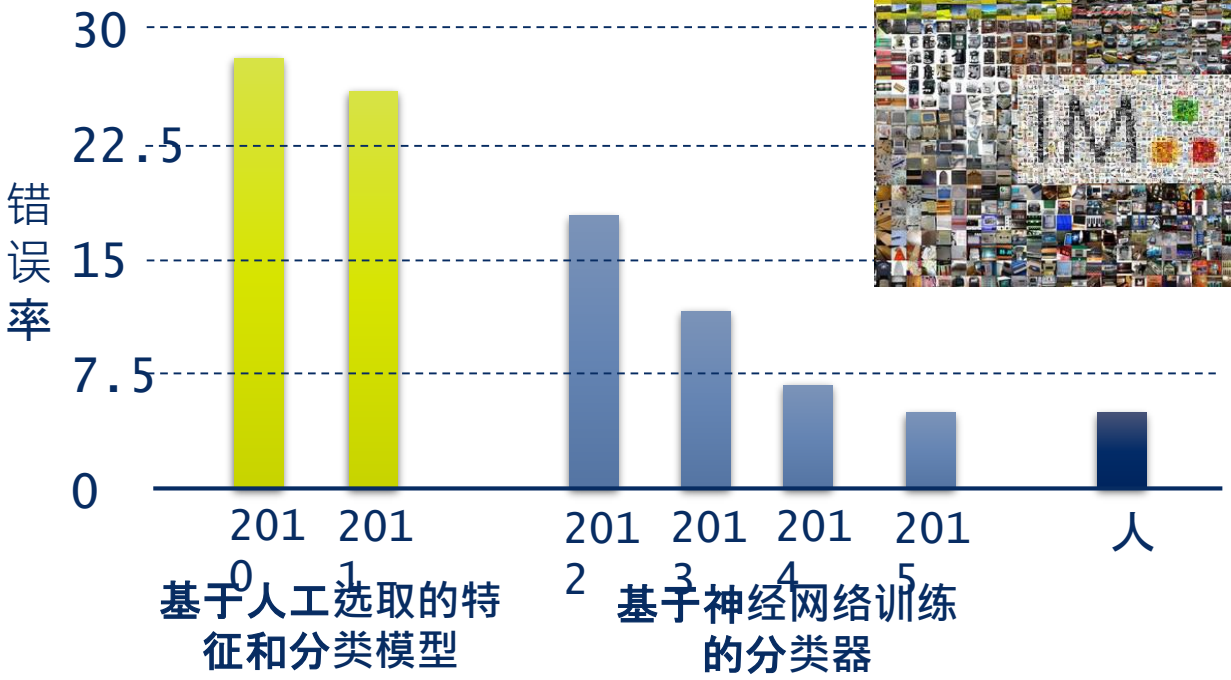


3D-AI运算对CPU的新需求

结论

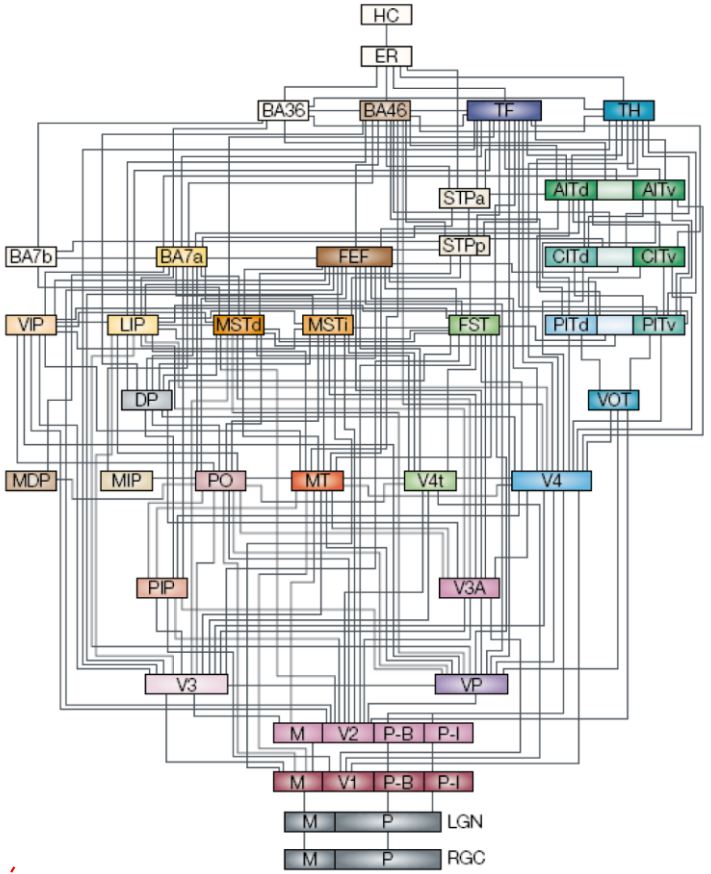
3D传感器将成为视觉AI嵌入化、边沿化的核心助推器之一

2010年以来AI发展



AI爆发的中心点是视觉技术

- 眼睛是人脑的主要信息来源
- 视觉在很长时间内还会是AI焦点



Daniel J. Felleman and David C. Van Essen,
"*Distributed Cerebral Processing in the Primate Cerebral Cortex*,"

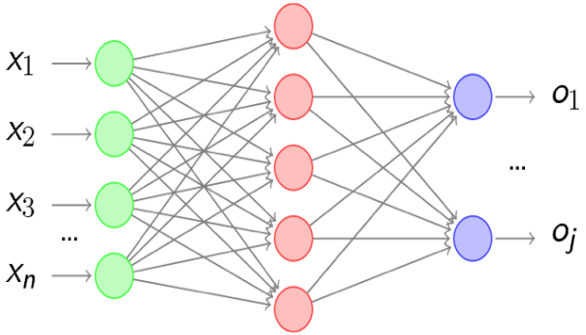
视觉AI的算法

- 人工设计算法

手动特征提取和人工设计的结构化分析算法

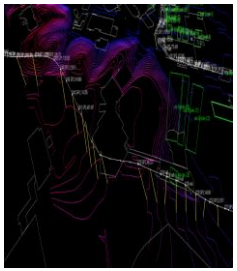
- 数据自动学习算法

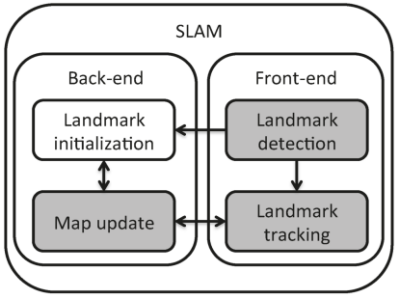
构建训练算法，通过数据训练自动提取特征并自动找出识别的算法参数或结构，最热门的是神经网络方法



视觉AI的主要运算

- 人工设计法：以SLAM为例





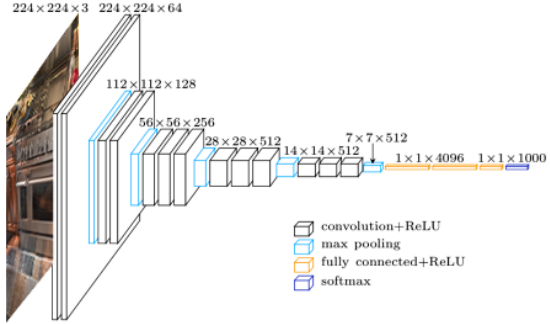
主要运算：

- 2D滤波和特征提取
- 坐标变换
- 概率滤波
- 点匹配搜索
- 参数估计和非线性优化

对应底层运算

- 2D卷积
- 小矩阵乘法
- 矩阵或向量乘法
- 矩阵分解
- 匹配和搜索

- 数据自动学习法：以VGG16深度卷积神经网络为例



主要运算：

- 2D卷积
- 全连接层网络计算
- 降采样
- 非线形变换

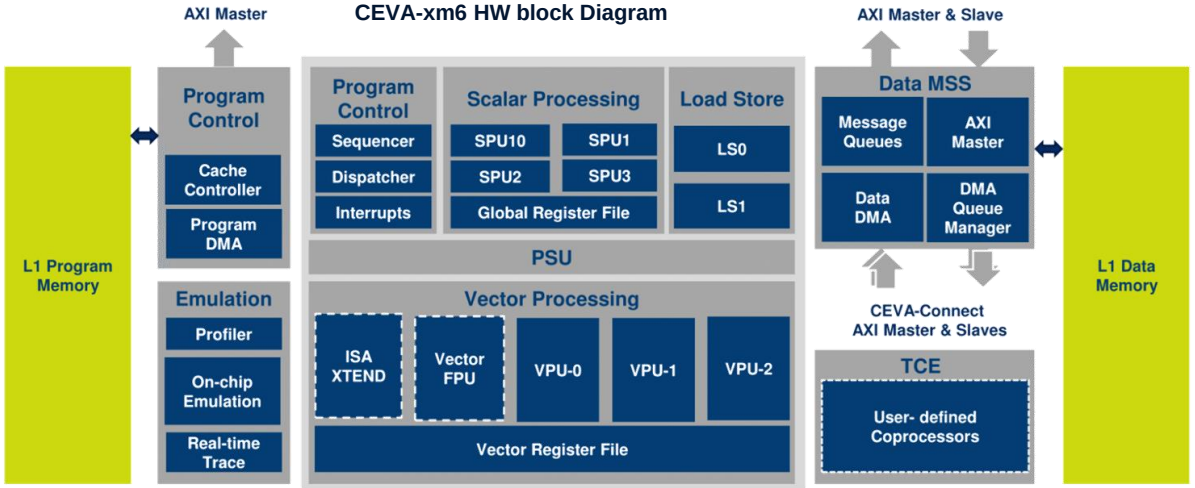
对应底层运算

- 2D卷积
- 非线性映射
- 大矩阵或向量乘法
- 数据比较和搜索

K. Simonyan, A. Zisserman, "*Very Deep Convolutional Networks for Large-Scale Image Recognition*"

视觉AI运算对嵌入式边缘计算的需求

- 视觉AI主要运算
  - ☺2D卷积
  - ☺小矩阵乘法
  - ☺非线性映射
  - ☺大矩阵或向量乘法
  - ☺矩阵分解(转矩阵/向量乘法)
  - ☺数据比较和搜索
  - ☺点匹配和搜索



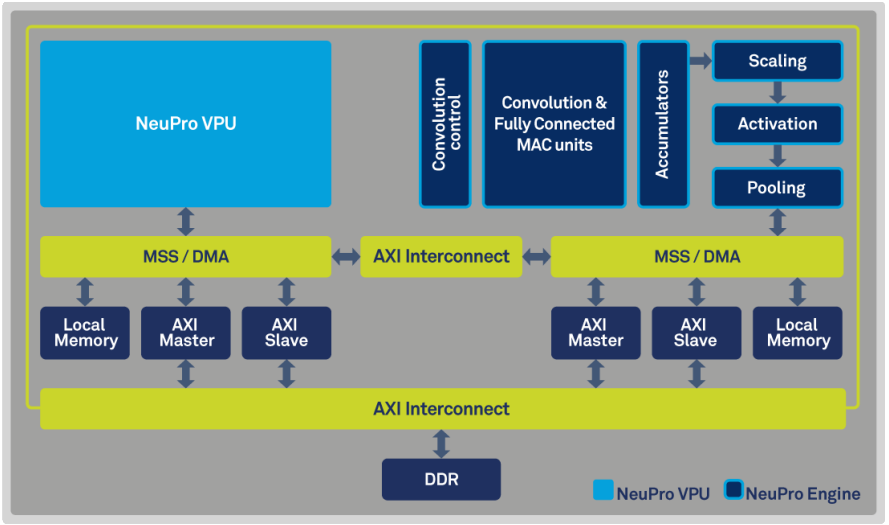
- 底层运算的矢量化和并行化保障AI运算效率
- 执行效率超过传统PC机CPU

视觉AI运算对嵌入式边缘计算的需求

- 针对神经网络的边沿计算适配
  - 通道剪枝(channel pruning)
  - 权重稀疏化(weight sparsification)
  - 权重量化(weight quantization)

- 架构改进：
  （MobileNet/SqueezeNet/ShuffleNet/ Distilling）

CEVA-NeuPro HW Block Diagram



解决视觉AI对嵌入式边缘计算的压力

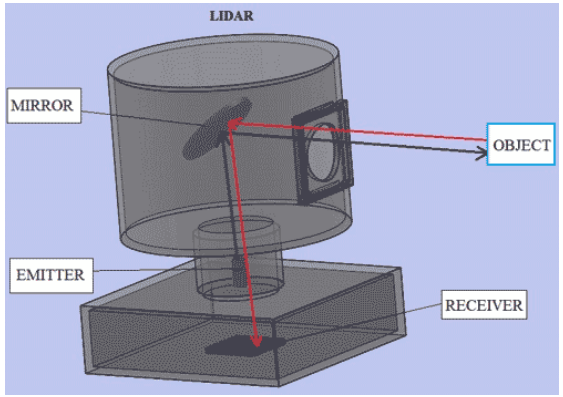
- 已有模型适配
  剪枝、权重稀疏化、量化
- 架构更新
  MobileNetV2, SqueezeNet
  等轻量级神经网络架构
- 硬件运算能力提升

3D传感器技术

3D传感器

- 激光LiDAR
- 结构光/光编码深度相机
- TOF深度相机

3D传感器——激光LiDAR



原理

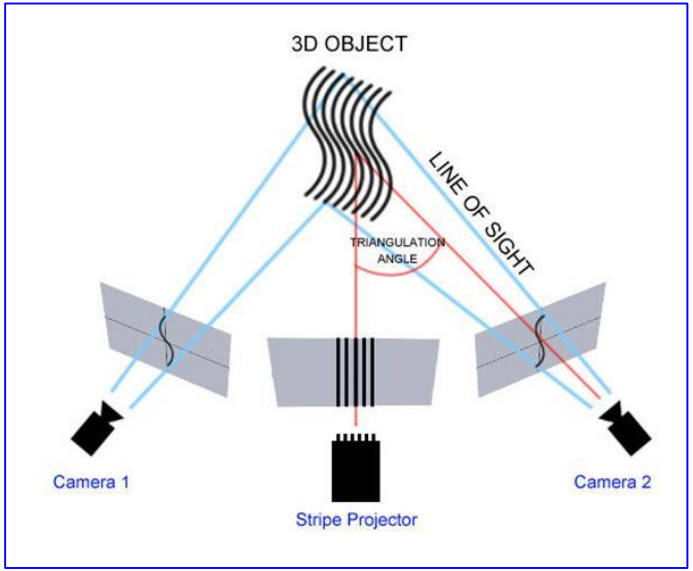
- **利用**单点激光测距原理
- **通**过扫描装置，测量多点数据，**得到深度**图

特点

- **高精度/低噪声/**远距离
- **高成本/低**帧率**/激光功率限制，通常有旋**转运动部件
- **常用于3D建模和自**动驾驶的平面扫描

3D传感器——结构光深度相机

微软的kinect早期产品使用结构光技术



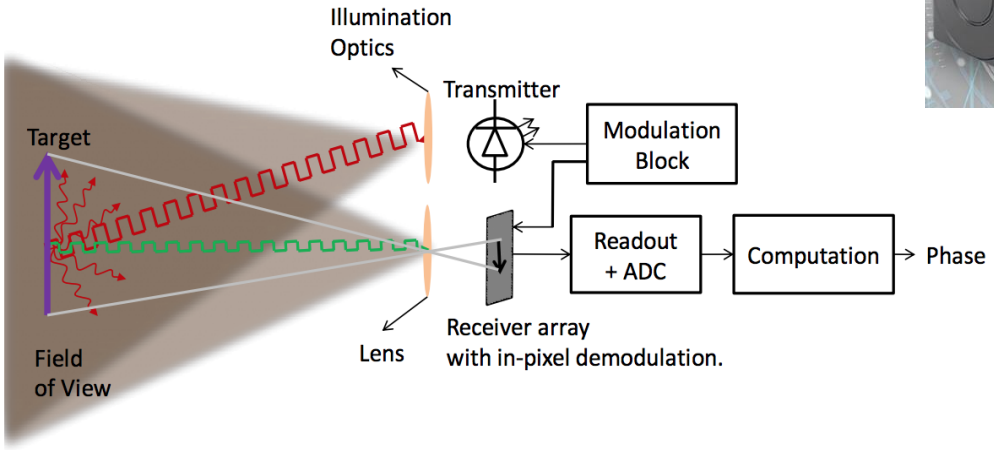
原理

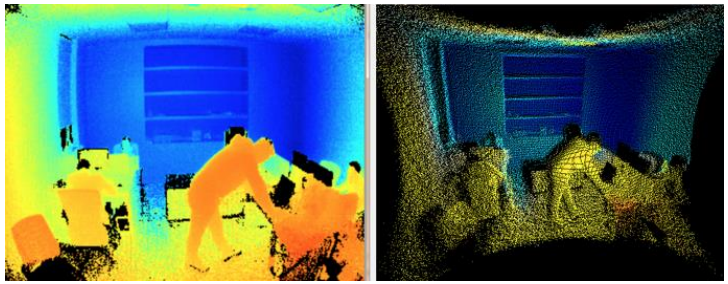
- “投影仪”（通常是红外的）投射条纹或者伪随机点到物体
- 根据不同视角拍摄到的条纹扭曲计算曲面形状

特点

- 相机成本较低(相比摄像头)
- 受环境光影响小(红外光源, 比RGB好些, 但比激光差)/运算量大

3D传感器——TOF深度相机





原理说明

- 发射光被高频信号调制（3～100MHz）
- **光**检测器输出的调制信号和发送信号有相位差
- **相位差由光**线从发射到接受的整个“飞行”时间决定
- **接收端将相位差折算成**时间差，**再折算距离，于是得到被**测对象距离
- **通**过接收端的透镜，**分离不同空**间位置的反射光

特点

- **相机成本**较低(**相比**摄像头)
- **运算量大(但比双目小)/受**环境光**影响小**(红外光源，**比激光差**)

为何使用3D？

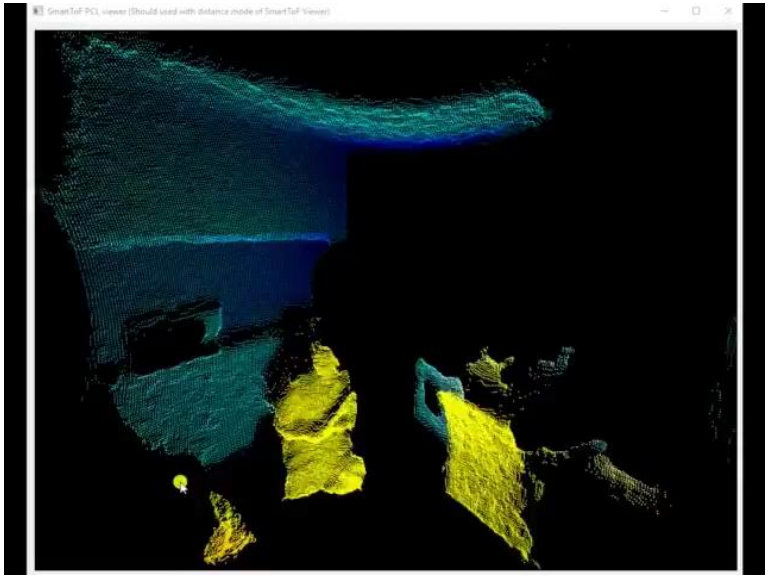
- 低功耗/低运算量/强大AI效果

3D传感器如何帮助视觉AI的嵌入化和边缘化?

- 以极低的运算量为视觉AI算法提供更加直接和简洁的解决思路

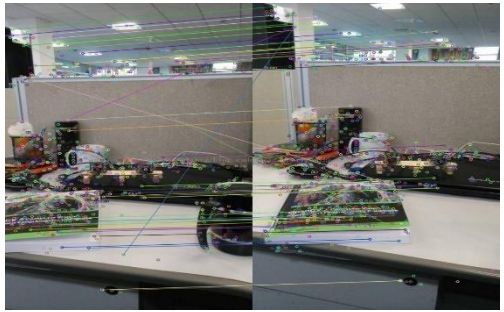
3D传感器如何帮助视觉AI的嵌入化/边缘化？
——快速实现视觉对象分割

从2D图像目标和环境混合，难以区分

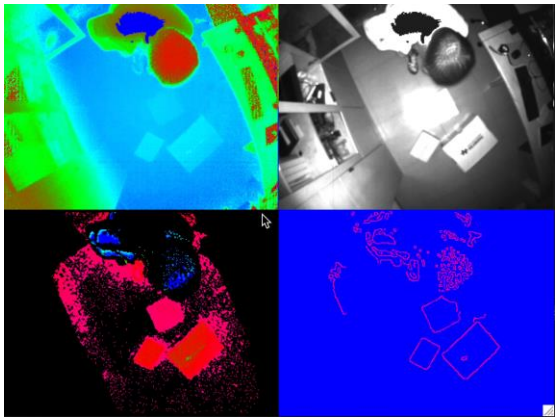


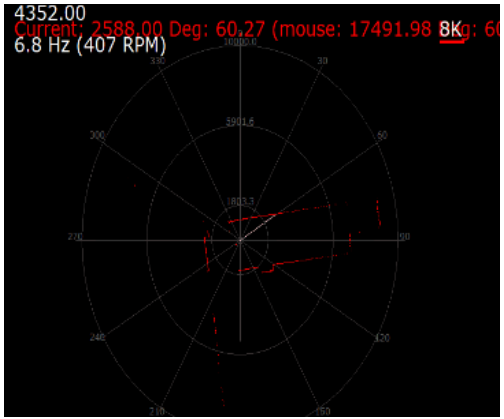
从3D数据根据距离信息直接分割目标

3D传感器如何帮助视觉AI的嵌入化/边缘化?
——识别和测量物理尺寸



从2D图像获取尺度信息困难,
需要大量运算

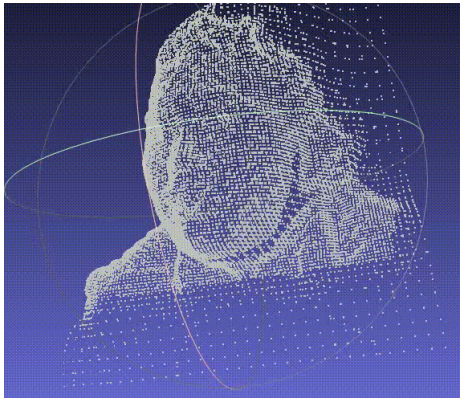




基于3D直接测量获得物理尺寸

3D传感器如何帮助视觉AI的嵌入化/边缘化?
——身份识别

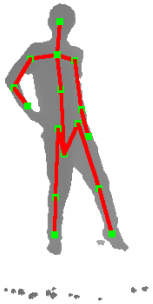
2D图像受光照影响，**易受欺骗**

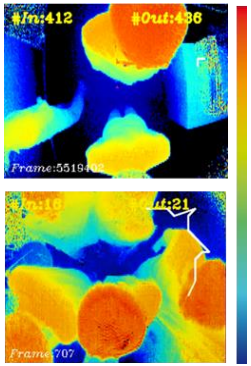


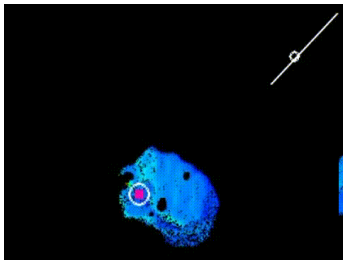
3D数据提供立体人脸数据，**不受光照影响**

3D传感器如何帮助视觉AI的嵌入化/边缘化?
——行为识别

2D人体行为识别需要强大的GPU运算





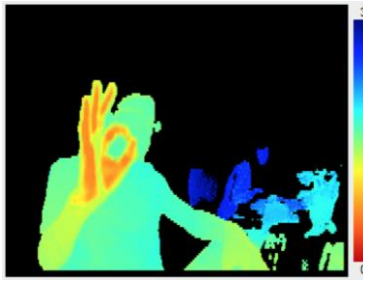


深度图进行3D人体行为识别只需要极低的CPU运算

相比传统的2D-AI，3D-AI有哪些算法上的改变？

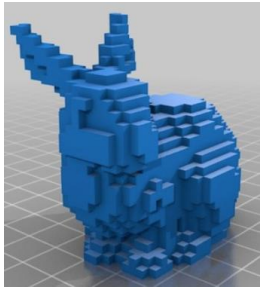
- 数据结构
- 主要的运算

3D传感器的数据组织形式



深度图

2维数组，存放每个像素到相机镜头的距离

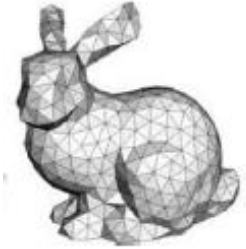


体像素

3维数组，每个元素和3D空间坐标对应，存放改位置是否有物体占据的标志

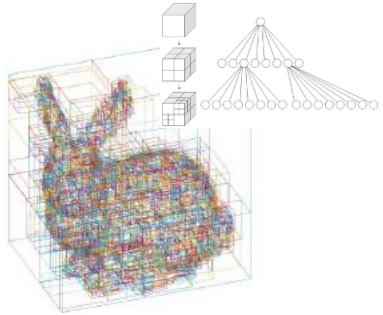
点云

对物体表面采样，记录每个采样点



三角剖分

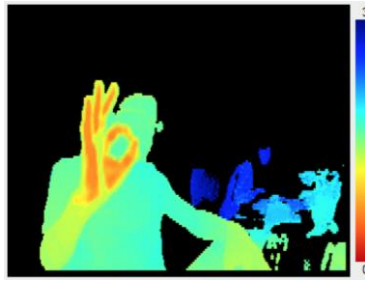
用三角形拼接拟合物体表面，物体形状通过这些三角形存储



八叉树

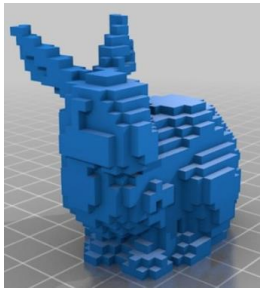
将空间按尺度分割成大小立方体，记录立方体内是否存在物体点云。按树结构存储。

3D传感器的数据组织形式



▶深度图

☺最接近2D图像数据格式
☺利于应用2D图像处理算法
☹不能反映完整几何体，常被认为是2.5D

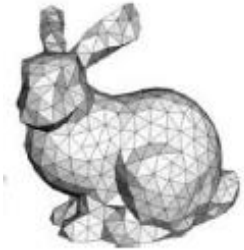


▶体像素

☺查询检索方便
☹存储效率低

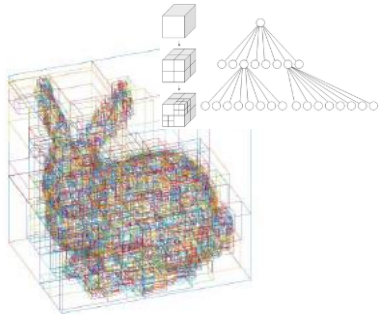
▶点云

☺存储效率高，
☹按距离检索数据点困难



▶三角剖分

☺存储效率高，
☺利于3D渲染显示，
☹按距离检索数据点困难



▶八叉树

☺存储效率较高，
☹数据结构复杂，
☺利于按距离检索。

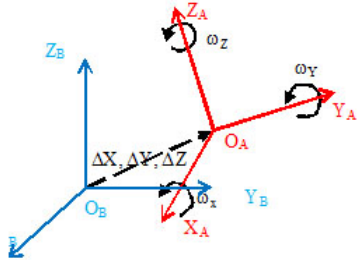
由于存储效率高和数据结构简单，多数实时3D传感器模块支持深度图和点云数据输出。对嵌入式系统“友好”

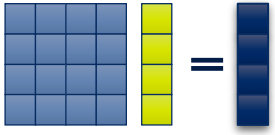
3D传感器数据的运算

- 几何变换
- 点数据处理
- 3D-AI模式识别

3D传感器数据的运算——几何变换

- 空间坐标变换

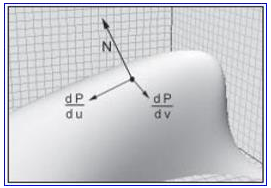




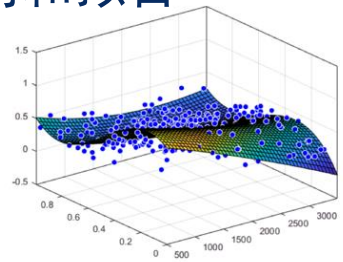
4x4矩阵乘法

- 这些算法的共同特点——依赖于小矩阵乘法和矩阵运算

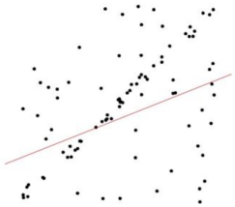
- 曲线/面的识别和拟合



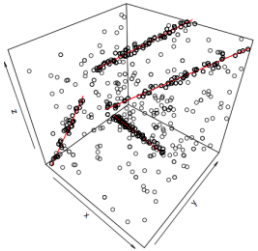
法向量计算：3x3矩阵运算



方程拟合：
常用矩阵特征值分解



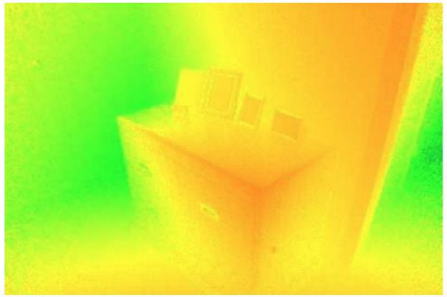
RANSAC算法：
随机采样及曲线/面拟合

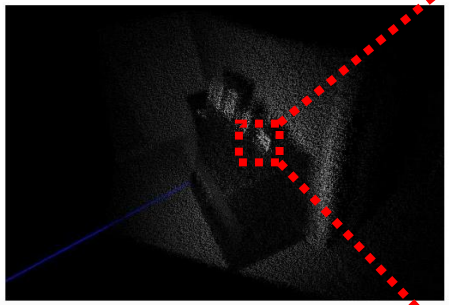


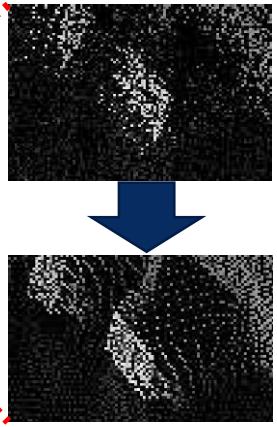
Hough算法：参数空间搜索

3D传感器数据的运算——点数据处理

- 噪声滤波

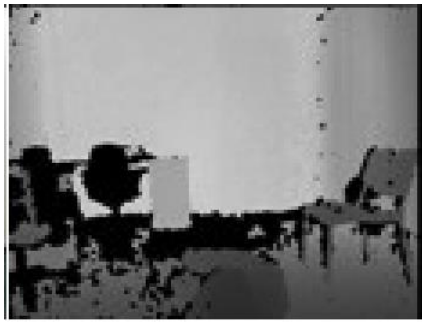




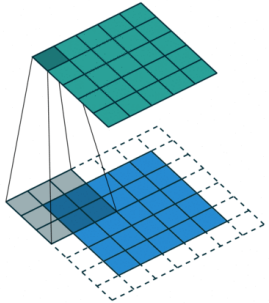


TI, SLOA230, “*Filtering for 3D Time-of-Flight Sensors*”

- 空洞处理



Suolan Liua, Chen Chen, Nasser Kehtarnavaz, "*A Computationally Efficient Denoising and Hole-Filling Method for Depth Image Enhancement*"



- 主要运算包括：
  平滑滤波（卷积）、法向量计算、曲面拟合等

3D传感器在嵌入式计边缘计算系统中的可用性

- 数据传输

  使用深度图或者点云格式能够实时传送3D视频，和传统摄像头模块数据率相当。比如30fps的QVGA图像，16-bit的深度图数据速率是：4MBps

- 数据处理

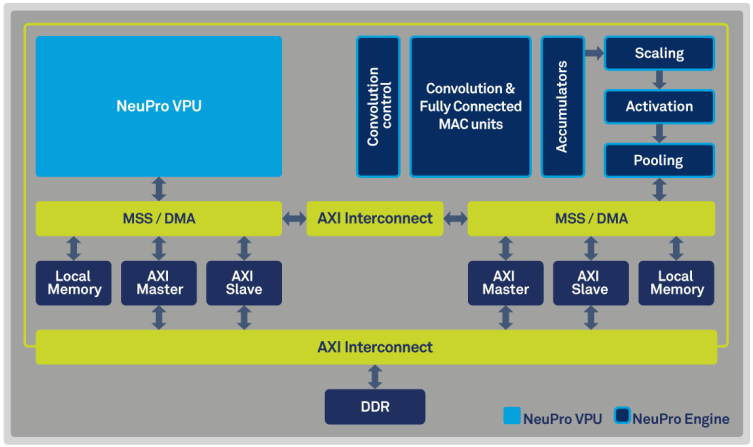
  之前讨论的3D数据处理算法大部分已经在2D数字图像处理的算法库中实现，对硬件的运算压力和2D数字图像处理相当

  ☺小矩阵乘法

  ☺小矩阵分解（可以转换成小矩阵乘法）

  ☺2D卷积

CEVA-NeuPro HW Block Diagram



3D-AI对计算有更高要求，现有的处理器硬件能够满足吗？

3D-AI模式识别算法

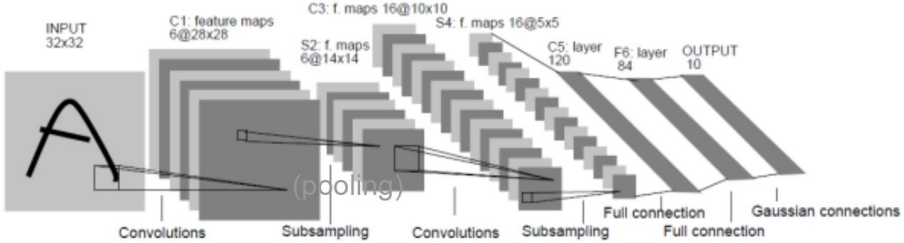
两个路线方向

- 沿用2D的运算架构
- 针对3D运算特点构建专用算法

2D Deep Learning: AlexNet, VGG, GoogleNet, STN, ResNet, DenseNet, …

3D Deep Learning: VoxNet, Octree CNN, Kdnetwork, PointNet, PointNet++, …

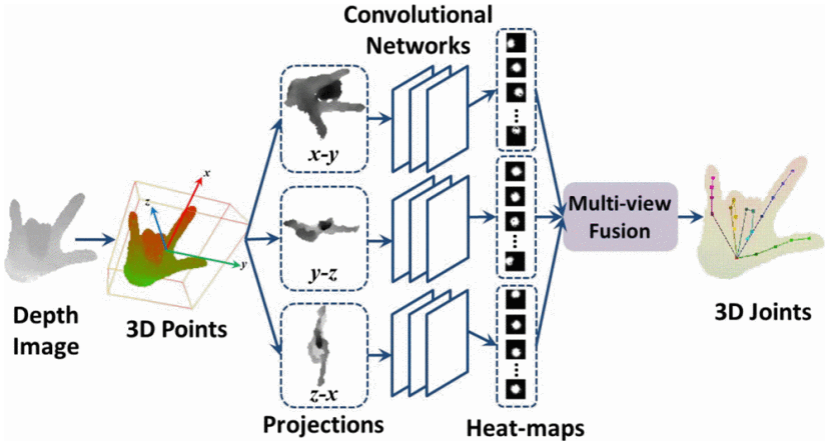
3D-AI模式识别算法——2D算法在3D数据上直接应用



用于2D图像识别的卷积神经网络结构示例

LeNet (1998 by LeCun et al.)

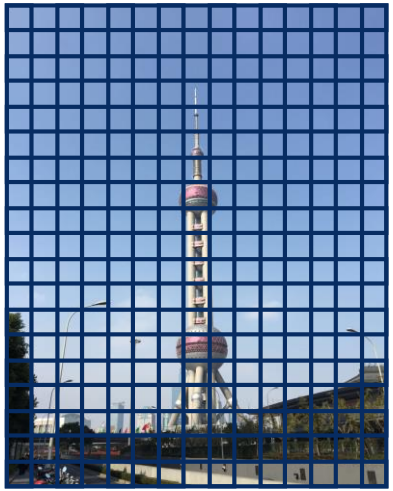
类似Multi-view CNN的路线



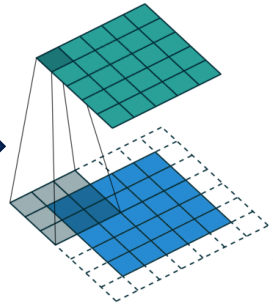
从3D点云生成特定视角的深度图，使用和2D图像识别相同的卷积神经网络结构实现手势识别

Liuhao Ge, Hui Liang, Daniel Thalmann, "*Robust 3D Hand Pose Estimation in Single Depth Images: From Single-View CNN to Multi-View CNNs*"

3D数据专用的神经网络——几何曲面上的网格卷积

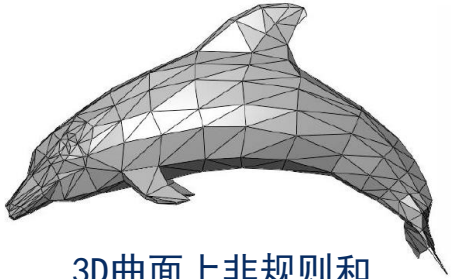


2D图上均匀和规则
网格采样

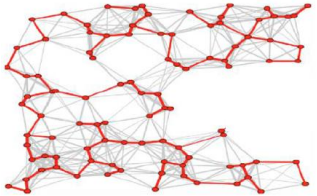


2D线性卷积
特征获取

小卷积核在大
特征图上卷积



3D曲面上非规则和
非均匀的采样



数据以图的形式组织，
使用图卷积

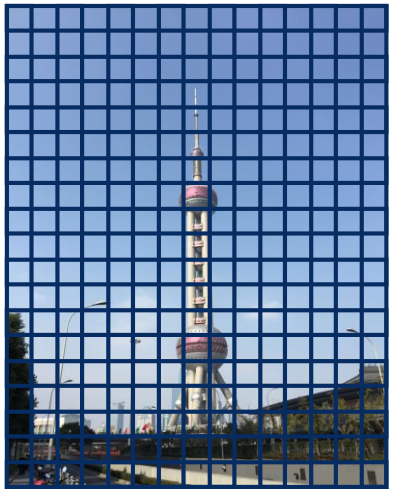
图卷积用矩阵多项式实现

$$\mathbf{y} = \left(\alpha_0\mathbf{I} + \alpha_1\mathbf{L} + \boldsymbol{\alpha_2}\mathbf{L^2} + \cdots + \boldsymbol{\alpha_K}\mathbf{L}^K\right)\mathbf{x}$$

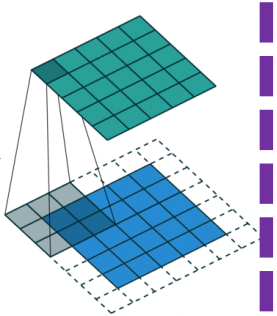
大矩阵乘法，运算量大

3D数据专用的神经网络——基于体像素数据的3D卷积

VoxNet

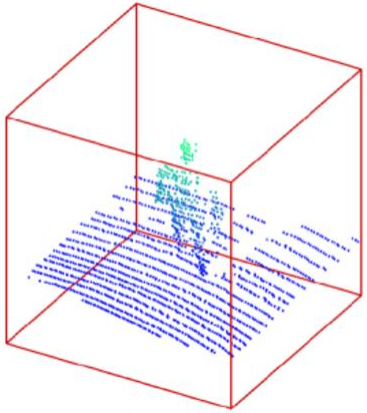


2D图上均匀和规则网格采样



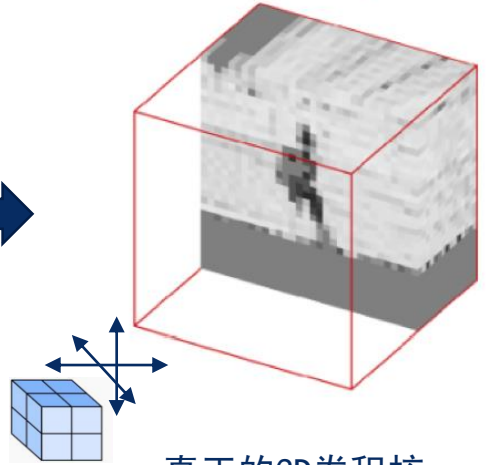
2D线性卷积特征获取

2D卷积核在2个方向上滑动



3D的体像素表示

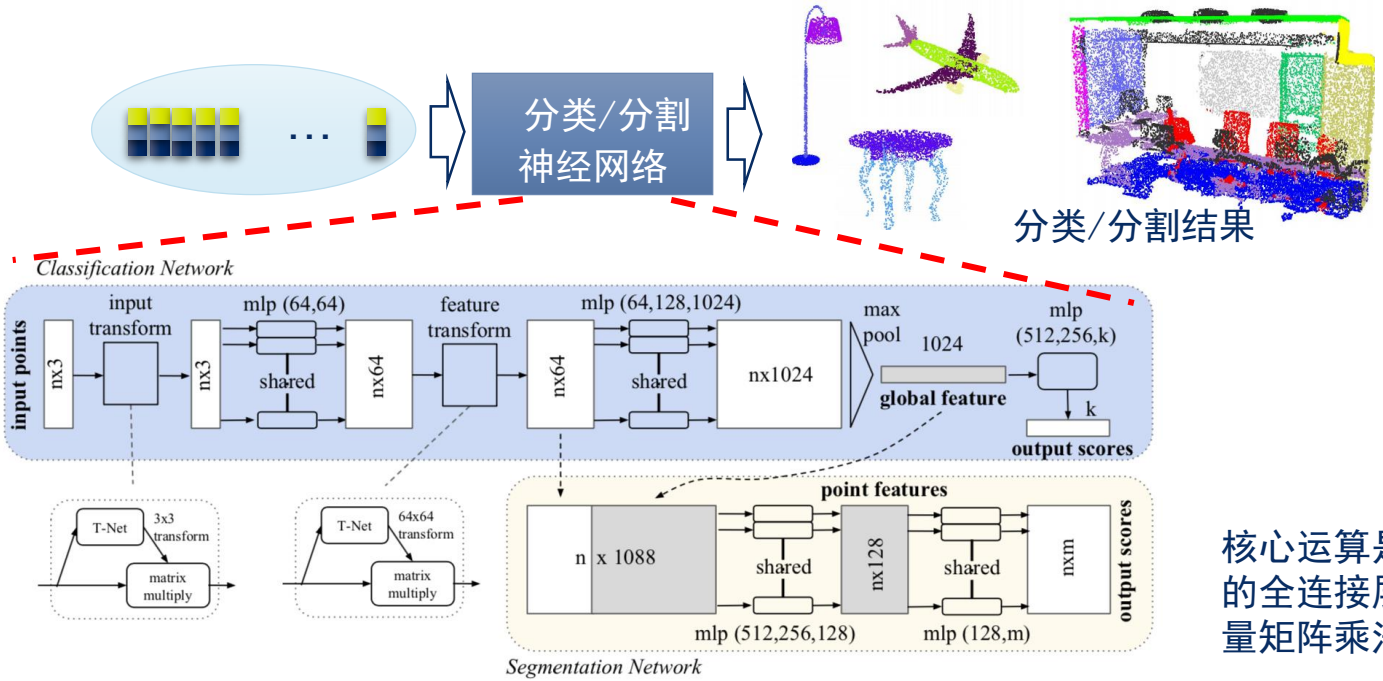
3D卷积核在3个维度滑动，存储和运算量大（有用八叉树改进的研究，O-CNN）



真正的3D卷积核

Daniel Maturana and Sebastian Scherer,
"VoxNet: *A 3D Convolutional Neural Network for Real-Time Object Recognition*"

3D数据专用的神经网络
——面向无序点云集合的网络PointNet



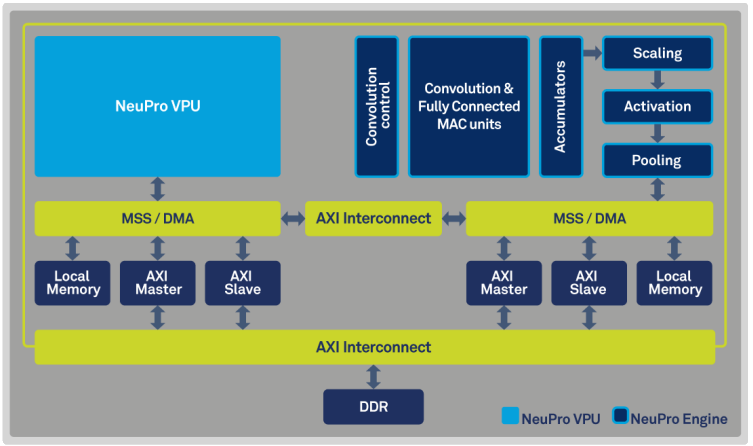
核心运算是神经网络的全连接层，对应大量矩阵乘法

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas, "*PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*"

嵌入式和边缘计算硬件适用于3D-AI吗？

- 大部分运算对硬件的需求和2D数字图像处理相同
  - ☺小矩阵乘法
  - ☺2D/3D卷积
  - ☺非线性映射
- 基于点云的部分运算需要新的硬件特性
  - ☹八叉树结构
  - ☹点云的最近邻搜索（k-d tree）
  - ☹图卷积(可以转换成矩阵乘法，但矩阵尺寸更大)
  - ☹基于体像素3D卷积的高效数据存储访问

CEVA-NeuPro HW Block Diagram



基于3D感知的嵌入式和边缘AI计算时代已经到来！

总结

- 3D传感器为AI应用提供的新的途径，更加低运算复杂度和更加好的识别效果
- 3D传感器帮助将AI嵌入到终端应用，是实现视觉相关的AI运算边沿化的重要助推器
- 对于3D-AI运算的优化和加速，现有的3D处理器引擎还有可以扩展的地方

Thank You

类脑智能
BATC Intelligence

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

www.ceva-dsp.com

http://bat.sjtu.edu.cn/

不同网络的运算量比较

Model	Model Size(MB)	MillionMult–Adds	MillionParameters
AlexNet[1]	>200	720	60
VGG16[2]	>500	15300	138
GoogleNet[3]	~50	1550	6.8
Inception–v3[4]	90–100	5000	23.2

不同网络采用Deep Compression后的参数压缩率

Network	Top-1 Error	Top-5 Error	Parameters	Compress Rate
LeNet-300-100 Ref	1.64%	-	1070 KB	
LeNet-300-100 Compressed	1.58%	-	**27 KB**	40×
LeNet-5 Ref	0.80%	-	1720 KB	
LeNet-5 Compressed	0.74%	-	**44 KB**	39×
AlexNet Ref	42.78%	19.73%	240 MB	
AlexNet Compressed	42.78%	19.70%	**6.9 MB**	35×
VGG-16 Ref	31.50%	11.32%	552 MB	
VGG-16 Compressed	31.17%	10.91%	**11.3 MB**	49×

Song Han, Huizi Mao, William J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding"

MobileNet的运算量比较

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

ShuffleNet与MobileNet在ImageNet上精度对比

Model	Complexity (MFLOPs)	Cls err. (%)	Δ err. (%)
1.0 MobileNet-224	569	29.4	-
ShuffleNet 2× ($g = 3$)	524	**29.1**	0.3
0.75 MobileNet-224	325	31.6	-
ShuffleNet 1.5× ($g = 3$)	292	**31.0**	0.6
0.5 MobileNet-224	149	36.3	-
ShuffleNet 1× ($g = 3$)	140	**34.1**	2.2
0.25 MobileNet-224	41	49.4	-
ShuffleNet 0.5× (arch2, $g = 8$)	40	**42.7**	6.7
ShuffleNet 0.5× (shallow, $g = 3$)	40	45.2	4.2

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, Jian Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices”

ShuffleNet与AlexNet在ARM平台上速度对比

Model	Cls err. (%)	FLOPs	224 × 224	480 × 640	720 × 1280
ShuffleNet 0.5× (arch2, $g = 3$)	43.8	40M	15.2ms	87.4ms	260.1ms
ShuffleNet 1× ($g = 3$)	34.1	140M	37.8ms	222.2ms	684.5ms
AlexNet [19]	42.8	720M	184.0ms	1156.7ms	3633.9ms

Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, William J. Dally "EIE: Efficient Inference Engine on Compressed Deep Neural Network"