

# Machine Learning at Edge

A. Afsharrad, E. Sharifian

## **A Literature Review**

Supervisor: Dr. M. Maddah-Ali

September 30, 2020

# Overview

## 1 Introduction

## 2 General Ideas

- Network Binarization
- Network Pruning
- Knowledge Distillation
- Network Quantization
- Low-Rank Approximation
- Other Ideas

## 3 Products on the Market

- Google Coral
- Intel Neural Compute Stick
- Xnor.ai and AI2GO

Introduction



General Ideas



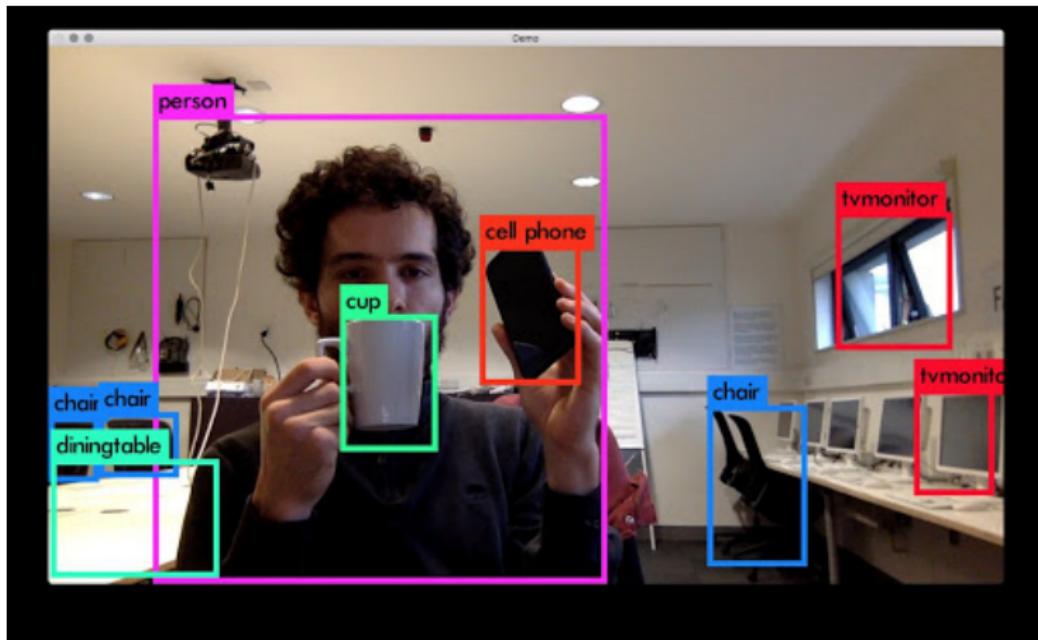
Products on the Market



# Introduction

# Motivation

Neural networks are everywhere.



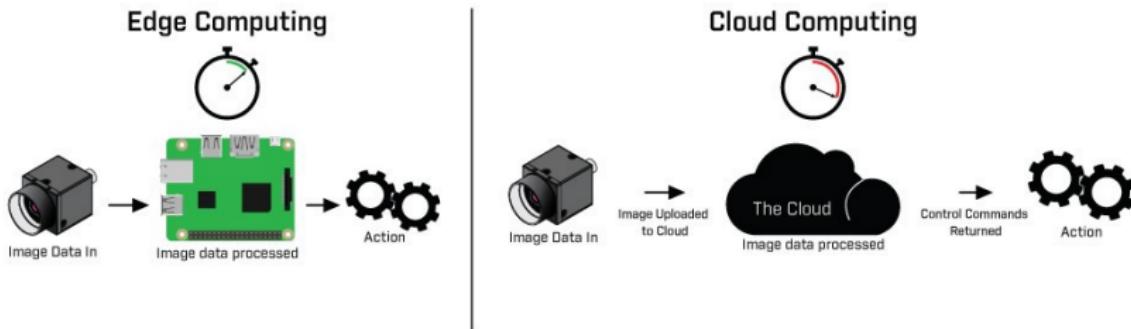
# Motivation

Neural networks are everywhere.  
Even in art!



# Machine Learning at Edge

And implementing ML models on edge devices seems more important than before.



But the problem is the very limited resources available on edge devices.

Introduction  
○○○

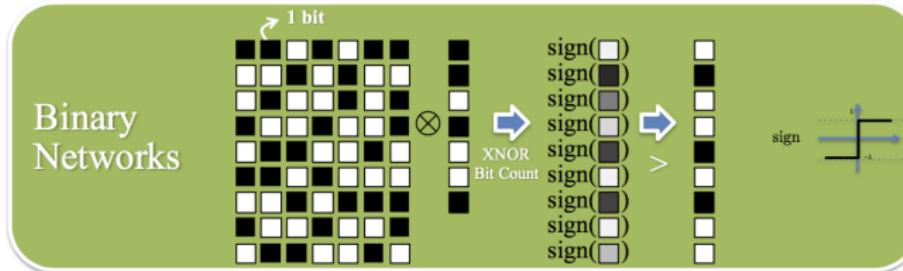
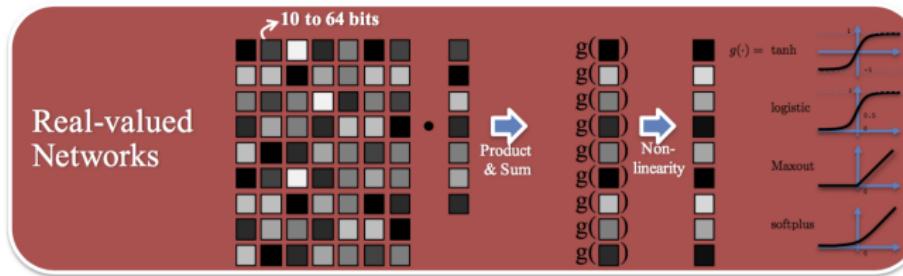
General Ideas  
●○○○○○○○○○○○○○○○○

Products on the Market  
○○○○○○○○○○

## General Ideas

# Network Binarization

**Idea:** Replace network weights and/or inputs with binary values



# Selected Paper: XNOR-Net

## XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

Mohammad Rastegari<sup>1(✉)</sup>, Vicente Ordonez<sup>1</sup>, Joseph Redmon<sup>2</sup>,  
and Ali Farhadi<sup>1,2</sup>

**Key Idea:** Approximate any vector  $\mathbf{W} \in \mathbb{R}^n$  with  $\alpha \in \mathbb{R}$  and  $\mathbf{B} \in \{-1, +1\}^n$  as  $\mathbf{W} = \alpha\mathbf{B}$ .

# Selected Paper: XNOR-Net – Methodology

**Key Idea:** Approximate any vector  $\mathbf{W} \in \mathbb{R}^n$  with  $\alpha \in \mathbb{R}$  and  $\mathbf{B} \in \{-1, +1\}^n$  as  $\mathbf{W} = \alpha\mathbf{B}$ .

**Method:** Solve the following optimization problem

$$\alpha^*, \mathbf{B}^* = \operatorname{argmin}_{\alpha, \mathbf{B}} \|\mathbf{W} - \alpha\mathbf{B}\|^2$$

**Result:**

$$\mathbf{B}^* = \operatorname{Sign}(\mathbf{W}) \quad , \quad \alpha^* = \frac{1}{n} \|\mathbf{W}\|_{\ell_1}$$

# Selected Paper: XNOR-Net – Results

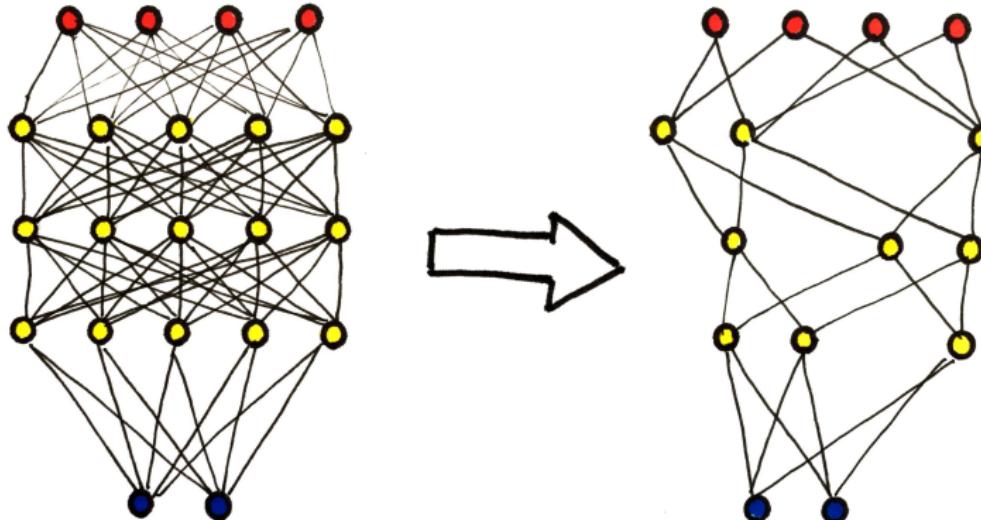
- $58\times$  faster convolution operations
- $32\times$  memory saving
- better than previous work, comparable to state-of-the-art:

**Table 1.** This table compares the final accuracies (Top1 - Top5) of the full precision network with our binary precision networks; Binary-Weight-Networks (BWN) and XNOR-Networks (XNOR-Net) and the competitor methods; BinaryConnect (BC) and BinaryNet (BNN).

Classification accuracy (%)									
Binary-weight				Binary-input-binary-weight				Full-precision	
BWN		BC [11]		XNOR-Net		BNN [11]		AlexNet [1]	
Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
<b>56.8</b>	<b>79.4</b>	35.4	61.0	<b>44.2</b>	<b>69.2</b>	27.9	50.42	56.6	80.2

# Network Pruning

**Idea:** Remove unimportant connections, filters, and layers from a network.



# Selected Paper: An Entropy-based Pruning method

## An Entropy-based Pruning Method for CNN Compression

Jian-Hao Luo      Jianxin Wu

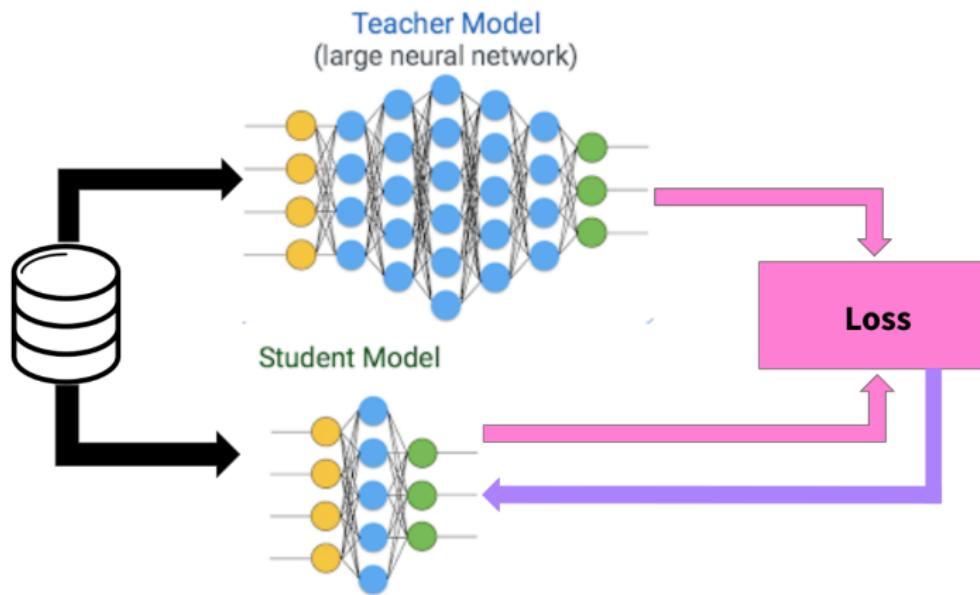
National Key Laboratory for Novel Software Technology  
Nanjing University, China

{luojh, wujx}@lamda.nju.edu.cn

**Key Idea:** The importance of each filter is evaluated by the proposed entropy-based method first. Then several unimportant filters are discarded to get a smaller CNN model.

# Knowledge Distillation

**Idea:** Transfer the knowledge from a cumbersome (teacher) network to a smaller (student) model for deployment.



# Selected Paper: Distilling the Knowledge in a Neural Net

---

## Distilling the Knowledge in a Neural Network

---

Geoffrey Hinton<sup>\*†</sup>

Google Inc.

Mountain View

geoffhinton@google.com

Oriol Vinyals<sup>†</sup>

Google Inc.

Mountain View

vinyals@google.com

Jeff Dean

Google Inc.

Mountain View

jeff@google.com

**Key Idea:** After training a cumbersome neural network for a specific task, one can transfer the knowledge to a light-weight network which is more suitable for deployment.

# Selected Paper: Distilling the Knowledge in a Neural Net

**Key Idea:** After training a cumbersome neural network for a specific task, one can transfer the knowledge to a light-weight network which is more suitable for deployment.

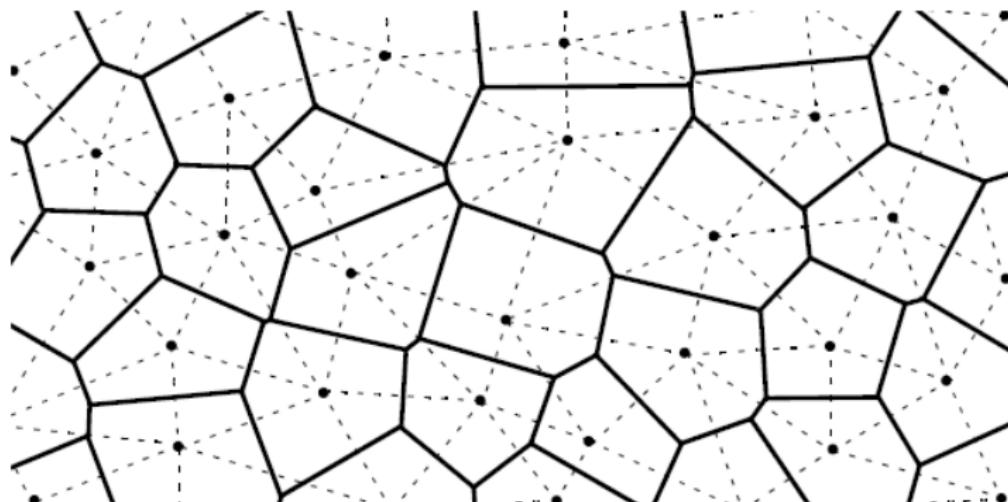
**Method:** Define the cost function for training the student model so that it would imitate the teacher.

*Example:* for a classification task with  $n$  classes:

- $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$  the probability vectors each model assigns to each class for a given sample
- Cost function  $C = d(\mathbf{p}, \mathbf{q})$   
( $d$  is a suitable metric for probability distributions)

# Network Quantization

**Idea:** quantize each network parameter (or vector of parameters) as an element drawn from a finite set, i.e., the codebook, so as to discard the original value for model compression.



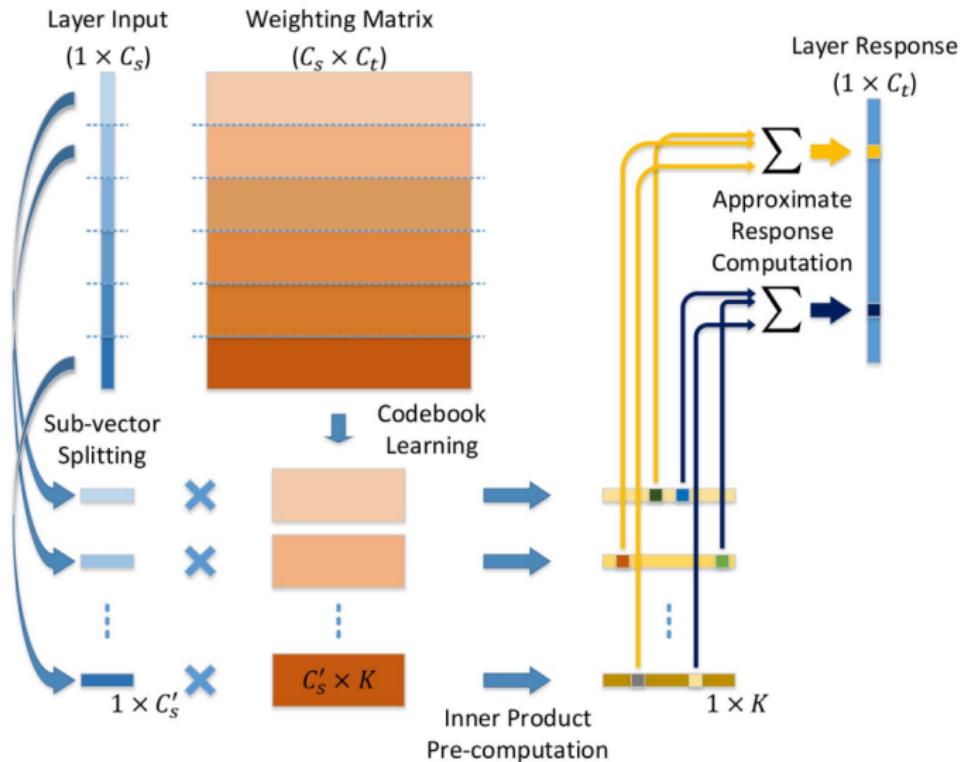
# Selected Paper: Quantized CNN

## Quantized CNN: A Unified Approach to Accelerate and Compress Convolutional Networks

Jian Cheng, Jiaxiang Wu<sup>✉</sup>, Cong Leng, Yuhang Wang, and Qinghao Hu

**Key Idea:** product quantization treats the feature space as the Cartesian product space composed by multiple subspaces. In each subspace, we learn a subcodebook.

# quantization process



# Low-Rank Approximation

**Idea:** Low-rank approximation for weight matrices or tensors.

A diagram illustrating the low-rank approximation of a matrix. On the left is a large square matrix symbol. An equals sign follows, followed by a summand. The first summand consists of a vertical bracket containing two horizontal bars. The top bar is labeled  $\mathbf{u}_2^1$  and the bottom bar is labeled  $\mathbf{u}_1^1$ . To the right of the plus sign is another summand, which is identical in structure to the first. To the right of the plus signs is the equation  $\mathbf{A} = \sum_{j=1}^r \mathbf{u}_1^j \circ \mathbf{u}_2^j$ .

A diagram illustrating the low-rank approximation of a tensor. On the left is a 3D cube symbol. An equals sign follows, followed by a summand. The first summand consists of a vertical bracket containing three horizontal bars. The top bar is labeled  $\mathbf{u}_3^1$ , the middle bar is labeled  $\mathbf{u}_2^1$ , and the bottom bar is labeled  $\mathbf{u}_1^1$ . To the right of the plus sign is another summand, which is identical in structure to the first. To the right of the plus signs is the equation  $\mathcal{A} = \sum_{j=1}^r \mathbf{u}_1^j \circ \mathbf{u}_2^j \circ \mathbf{u}_3^j$ .

# Modifying the Convolution Operation

**Key Idea:** use a variation of the classic convolution to achieve a smaller and/or faster network

**Selected Paper:** MobileNets

## MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Andrew G. Howard

Menglong Zhu

Bo Chen

Dmitry Kalenichenko

Weijun Wang

Tobias Weyand

Marco Andreetto

Hartwig Adam

Google Inc.

{howarda,menglong,bochen,dikalenichenko,weijunw,weyand,anm,hadam}@google.com

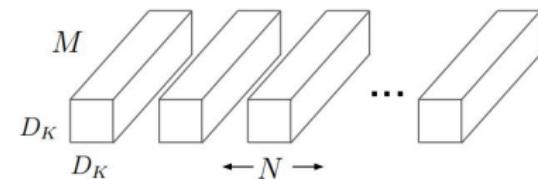
# Modifying the Convolution Operation

**Selected Paper:** MobileNets

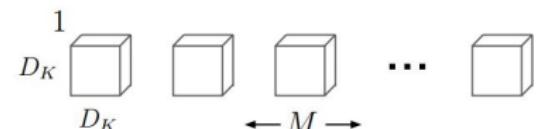
**Method:** use a depth-wise separable convolution

**Result:** a reduction computation factor of

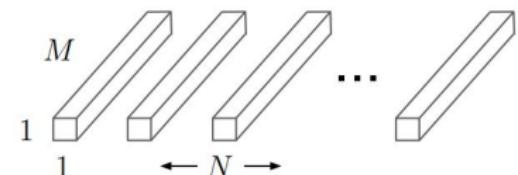
$$\frac{1}{N} + \frac{1}{D_K^2}$$



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c)  $1 \times 1$  Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

# Case Study: YOLO – a new approach to object detection

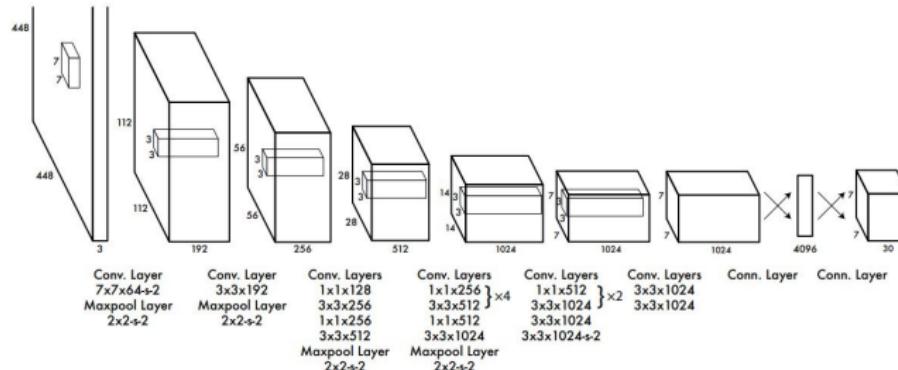
## You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon\*, Santosh Divvala\*†, Ross Girshick¶, Ali Farhadi\*†

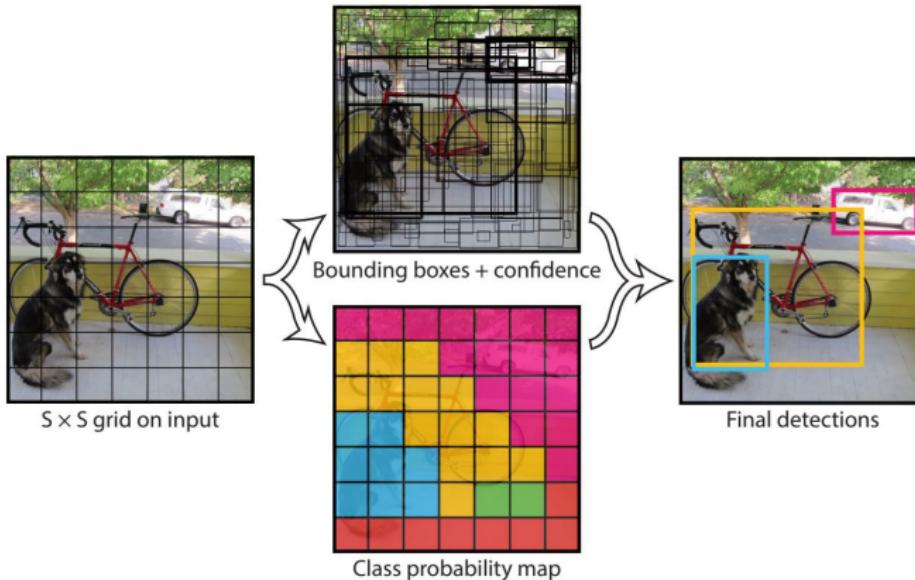
University of Washington\*, Allen Institute for AI†, Facebook AI Research¶

<http://pjreddie.com/yolo/>

**Key Idea:** Determine the bounding boxes and the object types simultaneously using a single neural network.



# Case Study: YOLO – a new approach to object detection



**Figure:** The system divides the image into an  $S \times S$  grid and for each grid cell predicts  $B$  bounding boxes, confidence for those boxes, and  $C$  class probabilities.

Introduction  
○○○

General Ideas  
○○○○○○○○○○○○○○○○

Products on the Market  
●○○○○○○○○○○

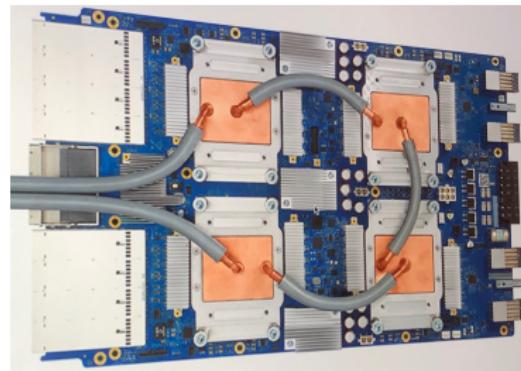
## Products on the Market

# Popular Edge Devices for Neural Nets

Edge device	GPU	CPU	ML software support
Coral SoM – Google	Vivante GC7000Lite	Quad ARM Cortex-A53 + Cortex-M4F	TensorFlow Lite, AutoML Vision Edge
Intel NCS2	Movidius Myriad X VPU (not GPU)		TensorFlow, Caffe, OpenVINO toolkit
Raspberry Pi 4	VideoCore VC6	Quad ARM Cortex-A72	TensorFlow, TensorFlow Lite
NVIDIA Jetson TX2	NVIDIA Pascal	Dual Denver 2 64-bit + quad ARM A57	TensorFlow, Caffe
RISC-V GAP8			TensorFlow
ARM Ethos N-77	8 NPUs in cluster, 64 NPUs in mesh		TensorFlow, TensorFlow Lite, Caffe2, PyTorch, MXNet, ONNX
ECM3531 A – Eta Compute	ARM Cortex-M3 + NXP CoolFlux DSP		TensorFlow, Caffe

# Google TPU

- Tensor Processing Unit
- an AI application-specific integrated circuit developed by Google
- In January 2019, Google made the Edge TPU available to developers with a line of products under the **Coral** brand.



# Google Coral

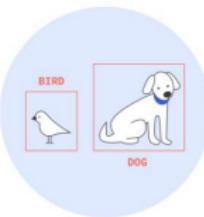
Coral is a complete toolkit to build products with local AI.



# Google Coral

Coral is a complete toolkit to build products with local AI.

Solutions for on-device intelligence



## Object detection

Draw a square around the location of various recognized objects in an image.



## Pose estimation

Estimate the poses of people in an image by identifying various body joints.



## Image segmentation

Identify various objects in an image and their location on a pixel-by-pixel basis.



## Key phrase detection

Listen to audio samples and quickly recognize known words and phrases.

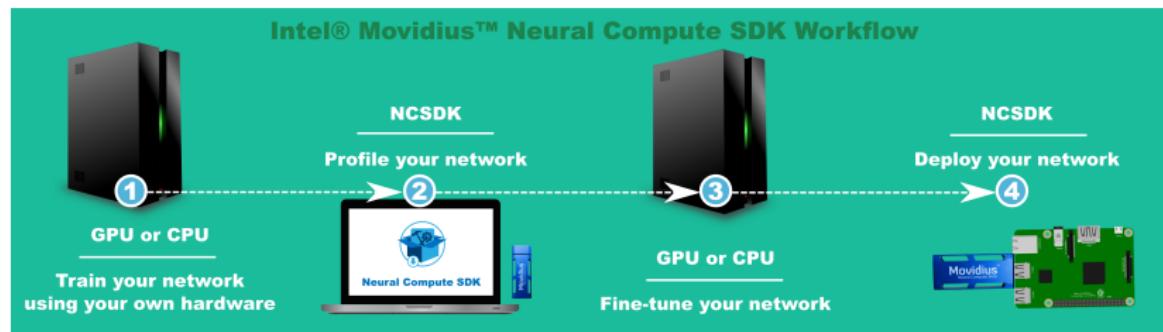
# Intel Neural Compute Stick

- The Intel Neural Compute Stick (NCS) is a tiny fanless deep learning device that you can use to learn AI programming at the edge.
- Movidius Neural Compute Stick enables rapid prototyping, validation and deployment of Deep Neural Network (DNN) inference applications at the edge



# Intel Neural Compute Stick – Workflow Schematic

The Intel Neural Compute Stick could pair with Raspberry Pi single board computer.



# Xnor.ai

**Xnor.ai:** a company founded in 2016 and bought by Apple for \$200M in 2020.



- brings state-of-the-art artificial intelligence to the edge
- Xnor's platform allows companies to run complex deep learning algorithms, formerly restricted to the cloud, locally on a range of devices including mobile phones, drones, and wearables.

## COMPANY BACKGROUND



Professor Ali Farhadi and Dr. Mohammad Rastegari founded Xnor.ai based on their pioneering research developed at the Allen Institute for Artificial Intelligence in Seattle, WA

### Publications

**Xnor-Net**: ImageNet classification using binary convolutional neural networks

**YOLO**: A new approach to object detection

**YOLO9000**: A state-of-the-art, real-time object detection system capable of detecting over 9,000 object categories

**Label Refinery**: Improving state-of-the-art accuracy by iteratively updating ground truth labels

# AI2GO

In 2019, Xnor released **AI2GO**, a do-it-yourself software platform for artificial intelligence.



A short introductory video:

<https://www.youtube.com/watch?v=jfAkV9JrErY>

# AI2GO

A slide from AI2GO introduction by Xnor.ai:



## Introducing AI2GO, a first-of-its-kind developer platform for AI on any device

- Software SDK for embedded devices (C++, Python, Swift)
- Xnor Bundles (XB): modular, pre-trained AI models combined with optimized inference engine runtime and APIs
- Hundreds of Deep Learning models tuned for popular use cases and devices
- Build & deploy your AI solution with state-of-the-art accuracy on the most resource-constrained edge devices
- No learning curve, no special equipment needed

Introduction

ooo

Xnor.ai and AI2GO

General Ideas

oooooooooooooooooooo

Products on the Market

oooooooooo●

# A journey through AI2GO website

## Select Your Hardware



These are hardware platforms we've trained models for.

 Click to select Linux x86_64	 Click to select Raspberry Pi 3	 Click to select Raspberry Pi 0	 Click to select Toradex
 Click to select Ambarella S5L	 Click to select Mac OS X	 Coming Soon Android	 Coming Soon iOS

# A journey through AI2GO website

## 1 Select your industry



Automotive

Photography &  
Video

Commercial

Media &  
Entertainment

Smart Home

### Smart Home Industry

#### What does it do?

Models for home automation, home security, and smart appliances. These models can be used inside and outside of the home to detect people, pets, and vehicles. There are models to determine what is in the kitchen by detecting kitchen objects, or in smart refrigerators to detect different types of food. Collectively these models determine what activities people are doing in and around your home.

#### Use cases

- Indoor Object Detector

- Indoor Object Classifier

- Pet Classifier

- Food Classifier

- Kitchen Object Detector

- Kitchen Object Classifier

- Outdoor Object Classifier

- Face Detector

- Outdoor Object Detector

- Pet Detector

# A journey through AI2GO website

2

## Filter by Smart Home Tasks

All Tasks

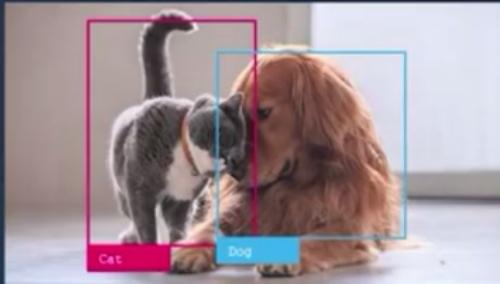
Detection

Multi-Class

Classification

# A journey through AI2GO website

## 3 Select a Smart Home Use Case



Smart Home Industry

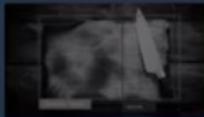
### Pet Detector

Provides Bounding boxes for Detection tasks

Used for indoor cameras to detect household pets



Indoor Object  
Detector



Kitchen Object  
Detector



Face Detector

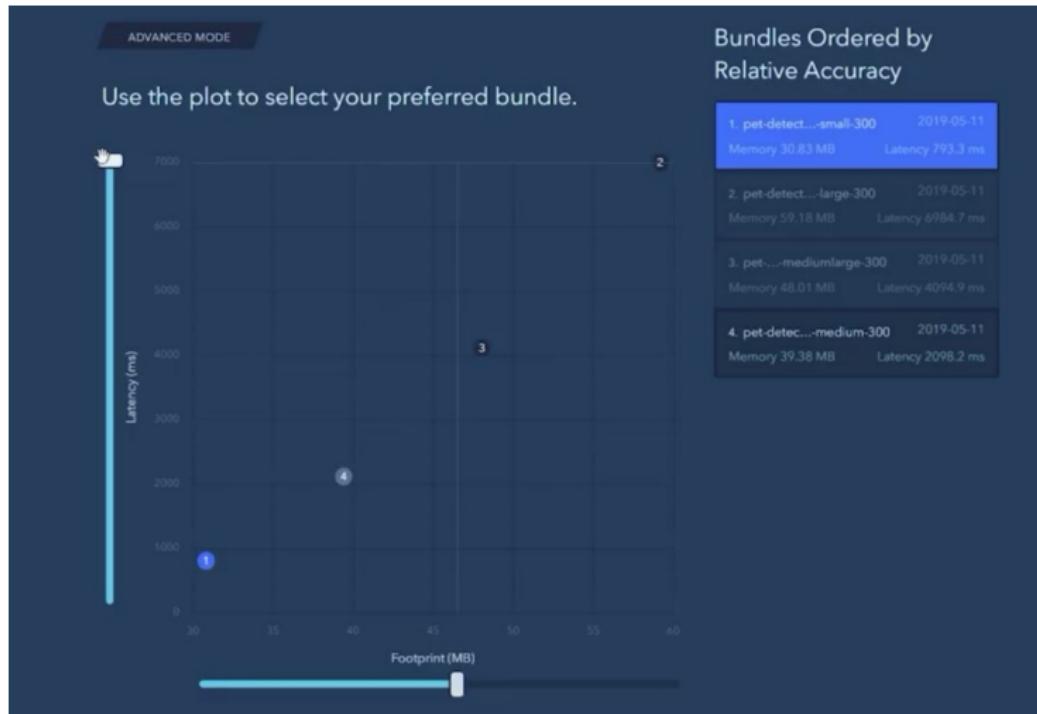


Outdoor Object  
Detector



Pet Detector

# A journey through AI2GO website



# A journey through AI2GO website

