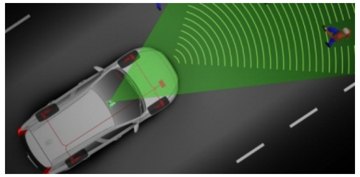
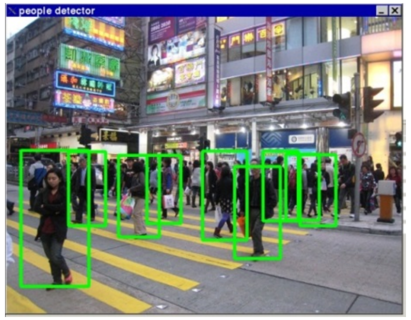
SYNOPSYS®
Silicon to Software™

Designing Scalable Multi Processor Embedded Vision Solutions

Dr Yankin Tanurhan
VP of Engineering, SG

July 2017



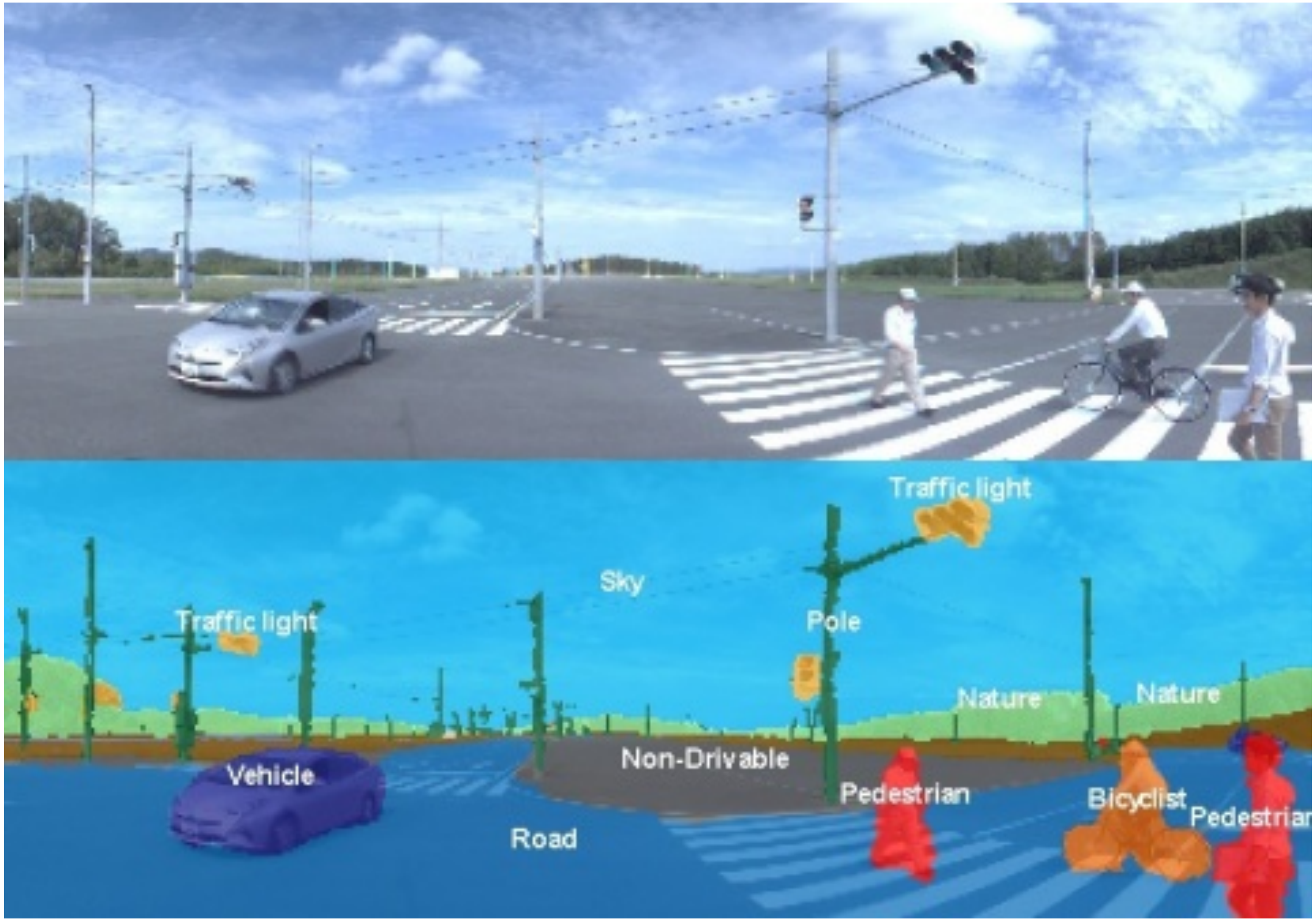


Embedded Vision Processor Outline

- Emerging Neural Network-based Applications
- DesignWare® EV6x Processor Family
  - Multi-core Vision SIMD engine
  - OpenVX and OpenCL C programming tools
  - Reference applications and libraries
- Third Generation CNN Engine
  - Features
  - Performance scaling
  - Programming tools

Emerging Neural Network-based Applications

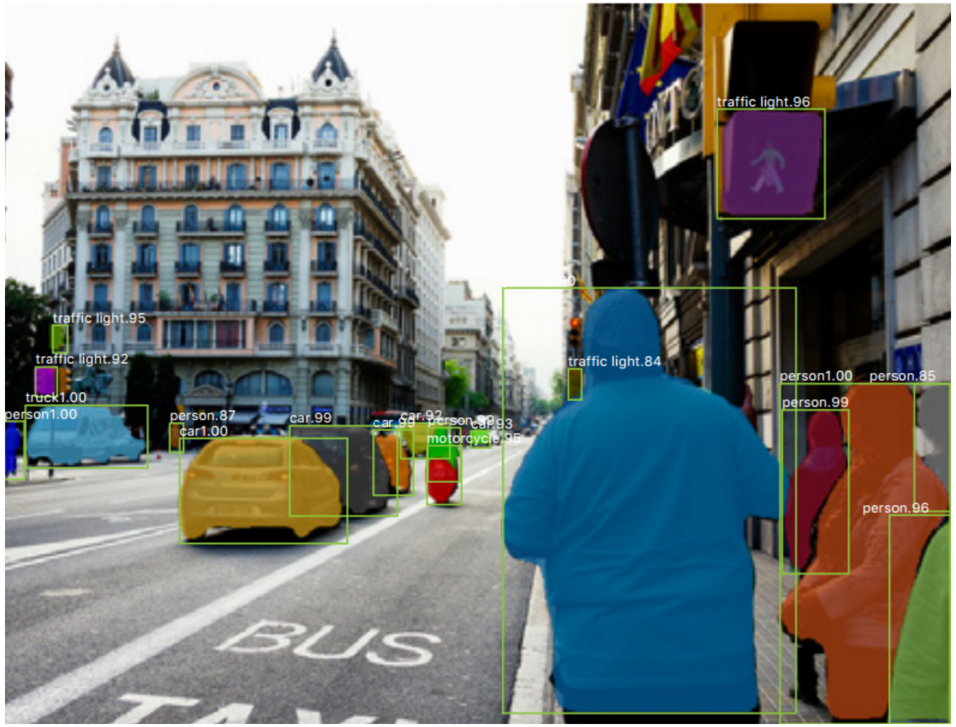
Scene Segmentation



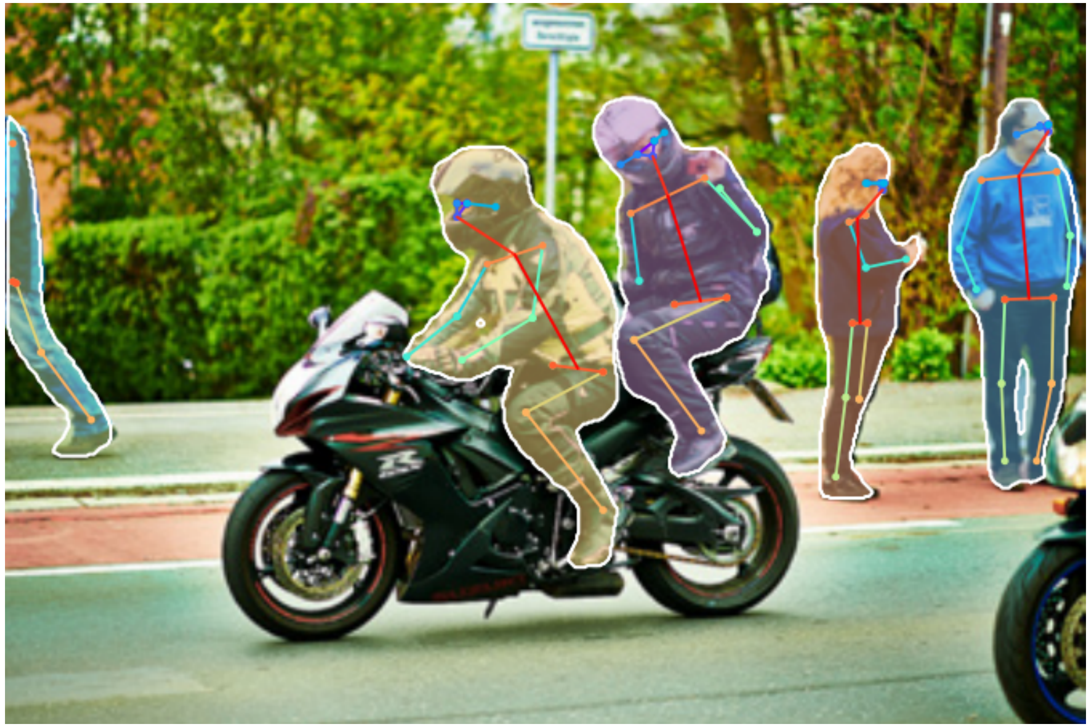
Source: Press Release by Toshiba and Denso, 17 Oct. 2016

Instance Segmentation and Keypoint Detection

Microsoft COCO Dataset 300K Images, 80 Object Categories, Keypoints on 100,000 people

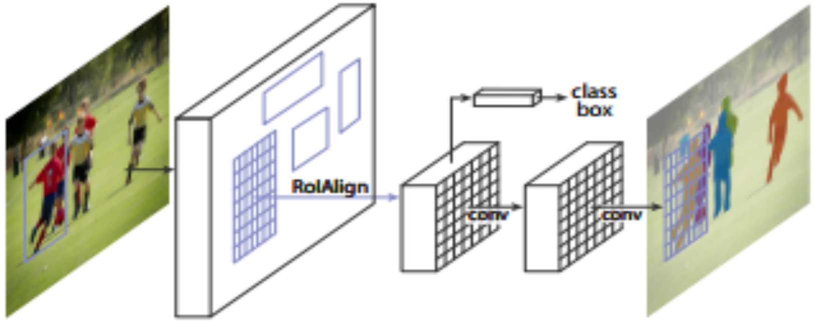


Instance Segmentation

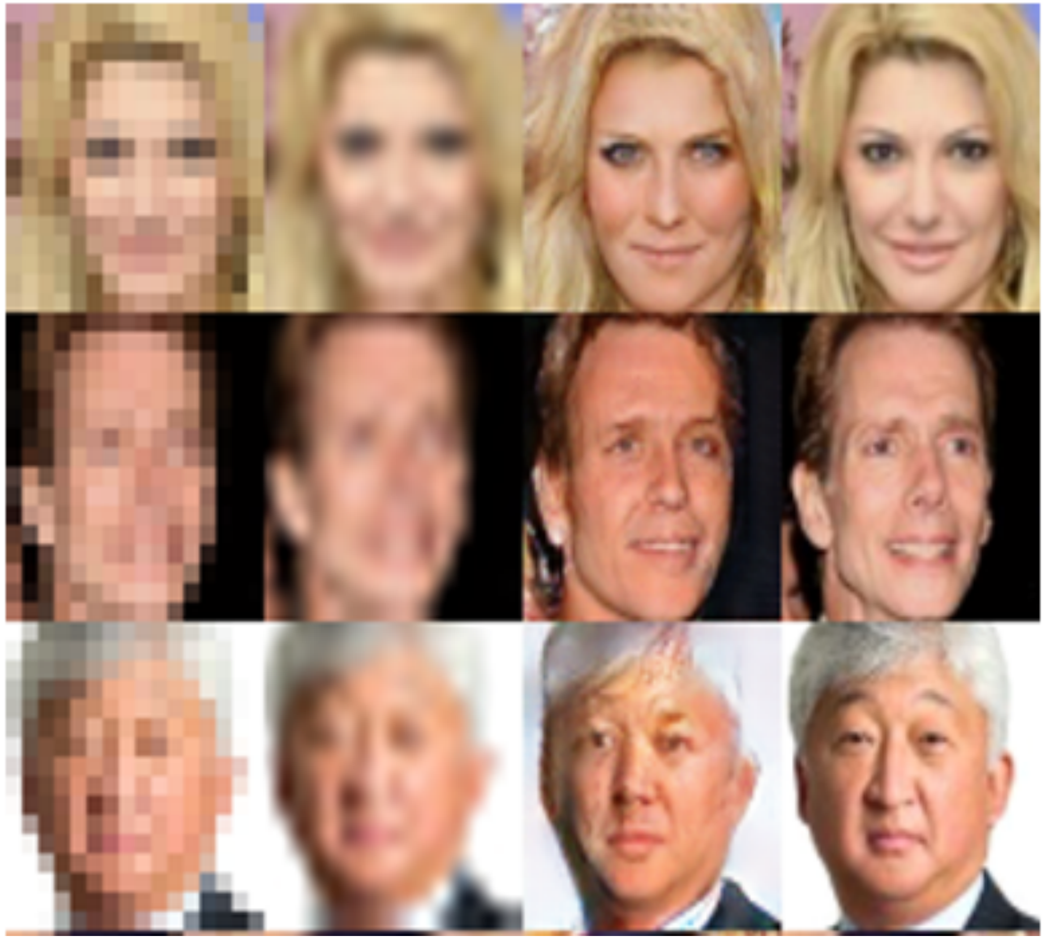


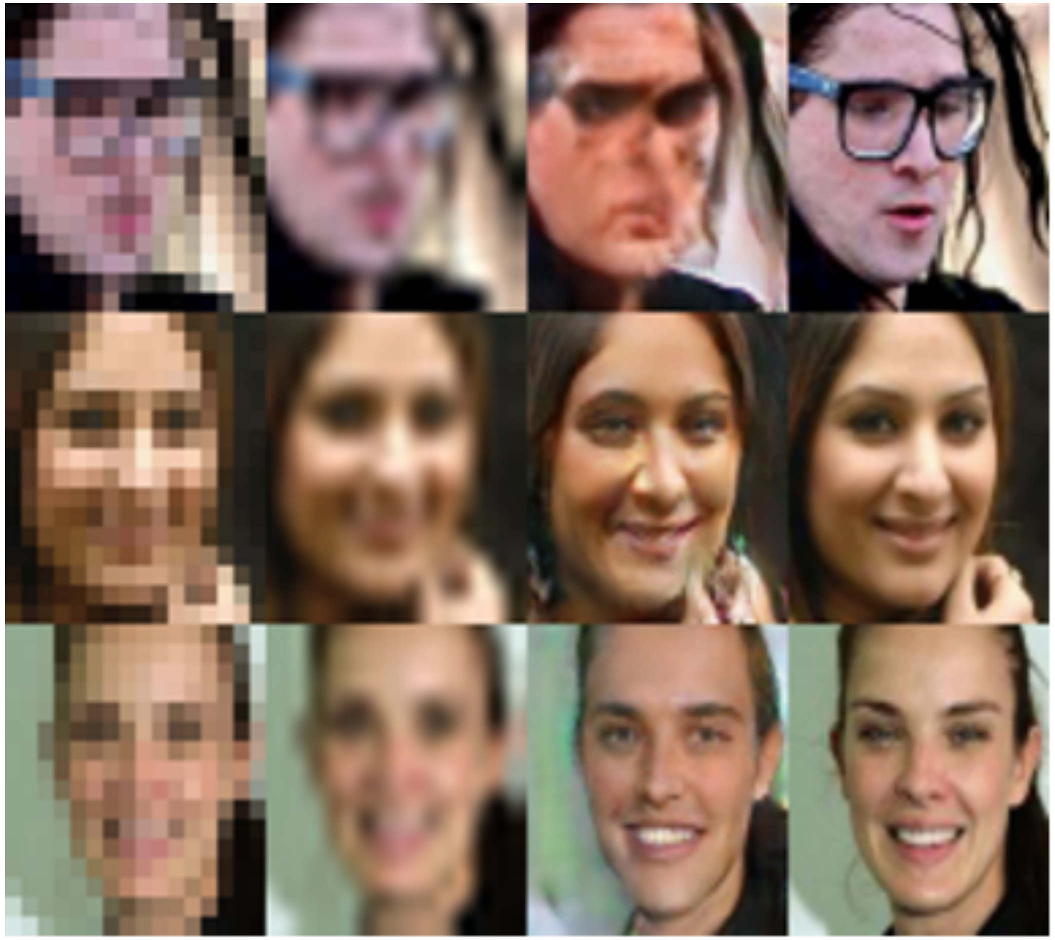
Keypoints on People

Source "Mask R-CNN", He et al. Facebook AI Research



Super resolution using CNN   600 GMAC for one 4K frame

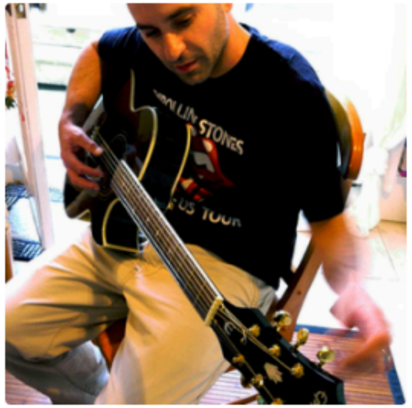




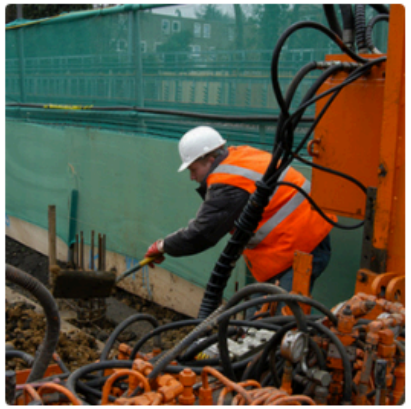
“Image Super-Resolution Using Deep Convolutional Networks (2016), C. Dong et al.”

Image Caption Generation with RNNs

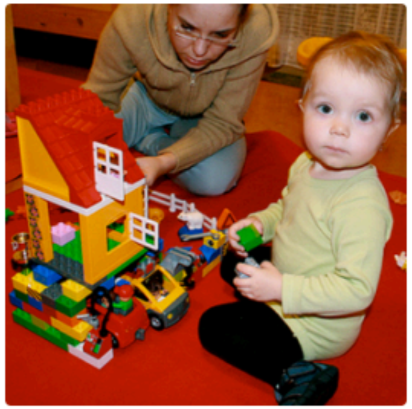
Recurrent Neural Networks: CNN + LSTM (Long-term Short-Term Memory)



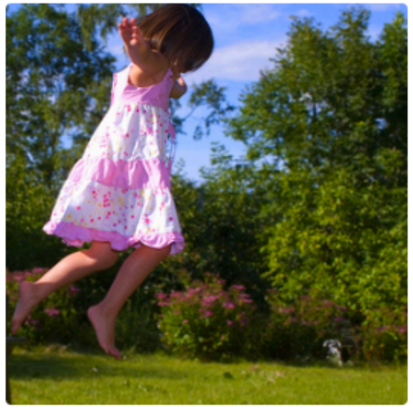
"man in black shirt is playing guitar."



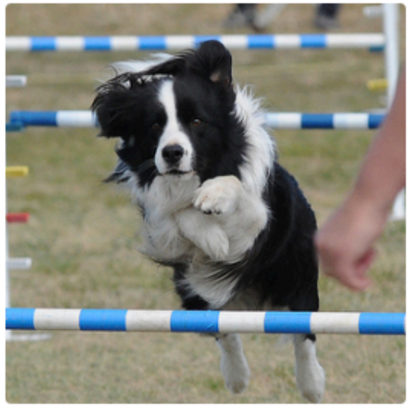
"construction worker in orange safety vest is working on road."



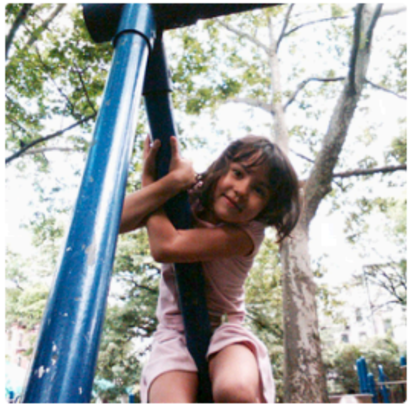
"two young girls are playing with lego toy."



"girl in pink dress is jumping in air."



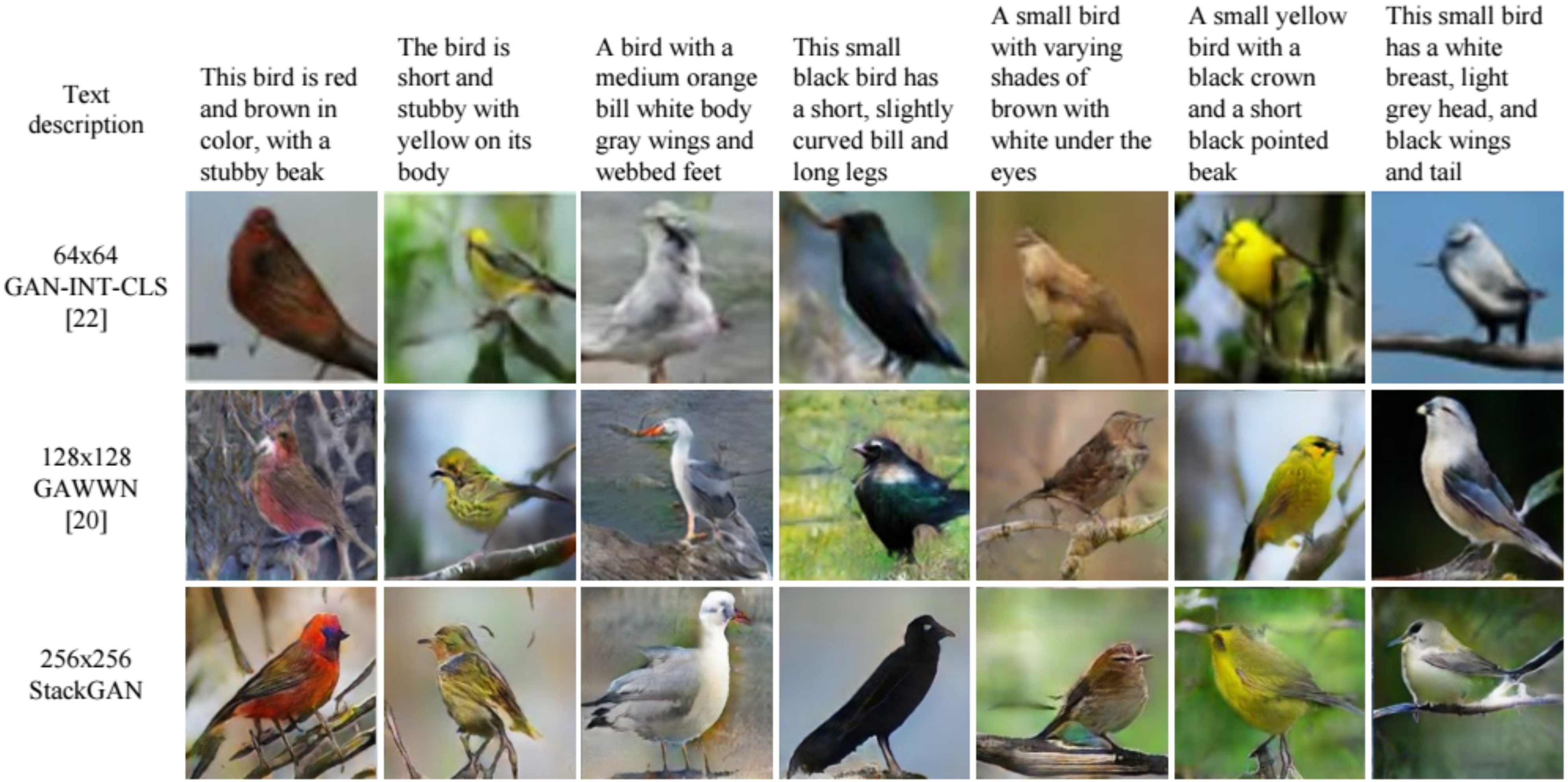
"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."

New Trends (Still academic): GANs

Generative Adversarial Networks – Generation of Images from Text Descriptions



Text description

This bird is red and brown in color, with a stubby beak

The bird is short and stubby with yellow on its body

A bird with a medium orange bill white body gray wings and webbed feet

This small black bird has a short, slightly curved bill and long legs

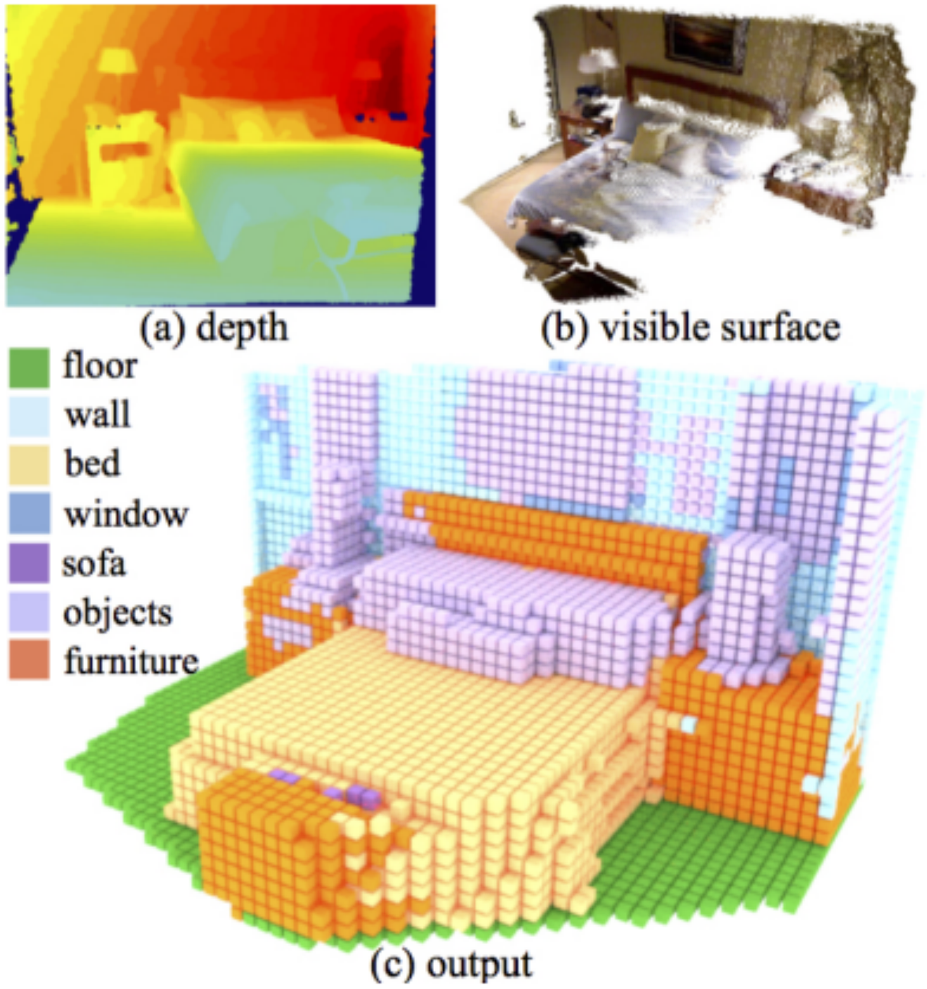
A small bird with varying shades of brown with white under the eyes

A small yellow bird with a black crown and a short black pointed beak

This small bird has a white breast, light grey head, and black wings and tail

New Trends (still academic)

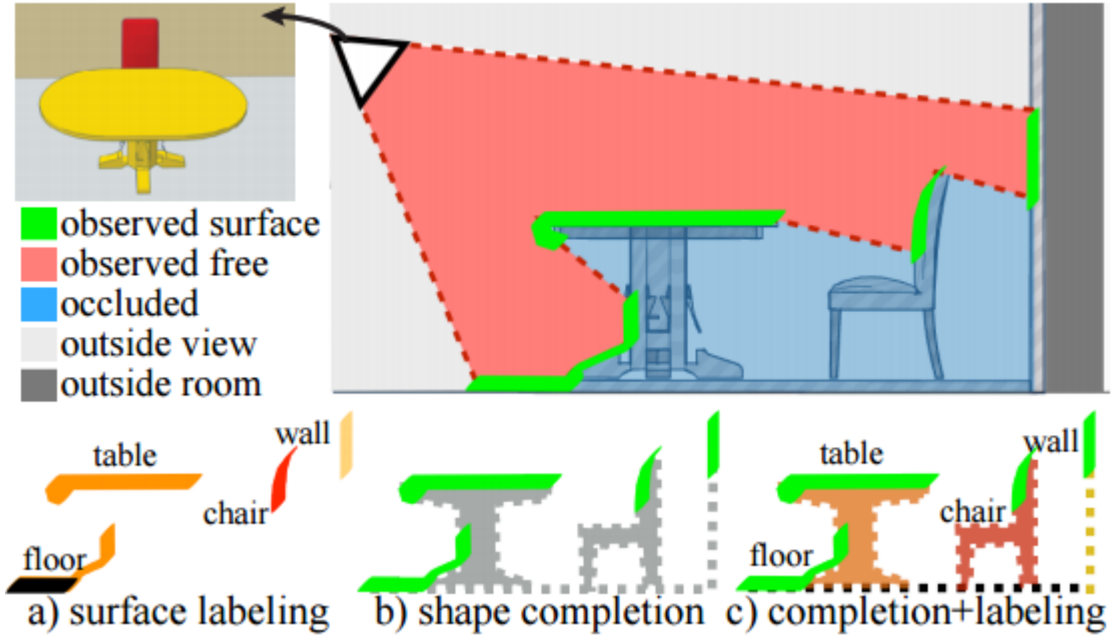
Semantic Scene Completion from a Single Depth Image



(a) depth

(b) visible surface

(c) output



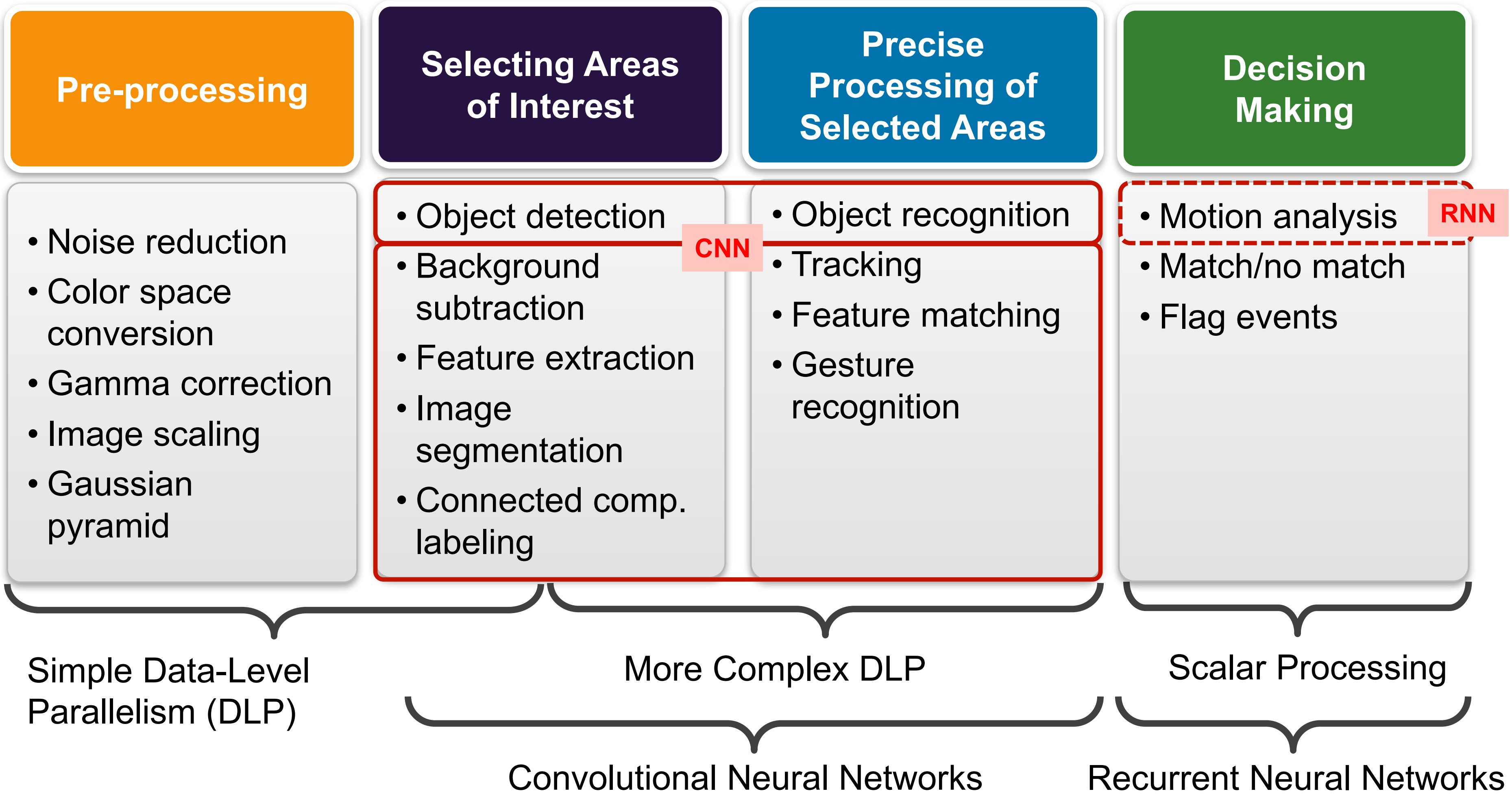
a) surface labeling

b) shape completion

c) completion+labeling

Embedded Vision Solutions

- Combining the best of traditional vision and deep learning approaches
- Combining scalar, vector processing with specialized CNN engines



Pre-processing
- Noise reduction
- Color space conversion
- Gamma correction
- Image scaling
- Gaussian pyramid

Selecting Areas of Interest
- Object detection
- Background subtraction
- Feature extraction
- Image segmentation
- Connected comp. labeling

CNN

Precise Processing of Selected Areas
- Object recognition
- Tracking
- Feature matching
- Gesture recognition

Decision Making
- Motion analysis
- Match/no match
- Flag events

RNN

Simple Data-Level Parallelism (DLP)

More Complex DLP

Scalar Processing

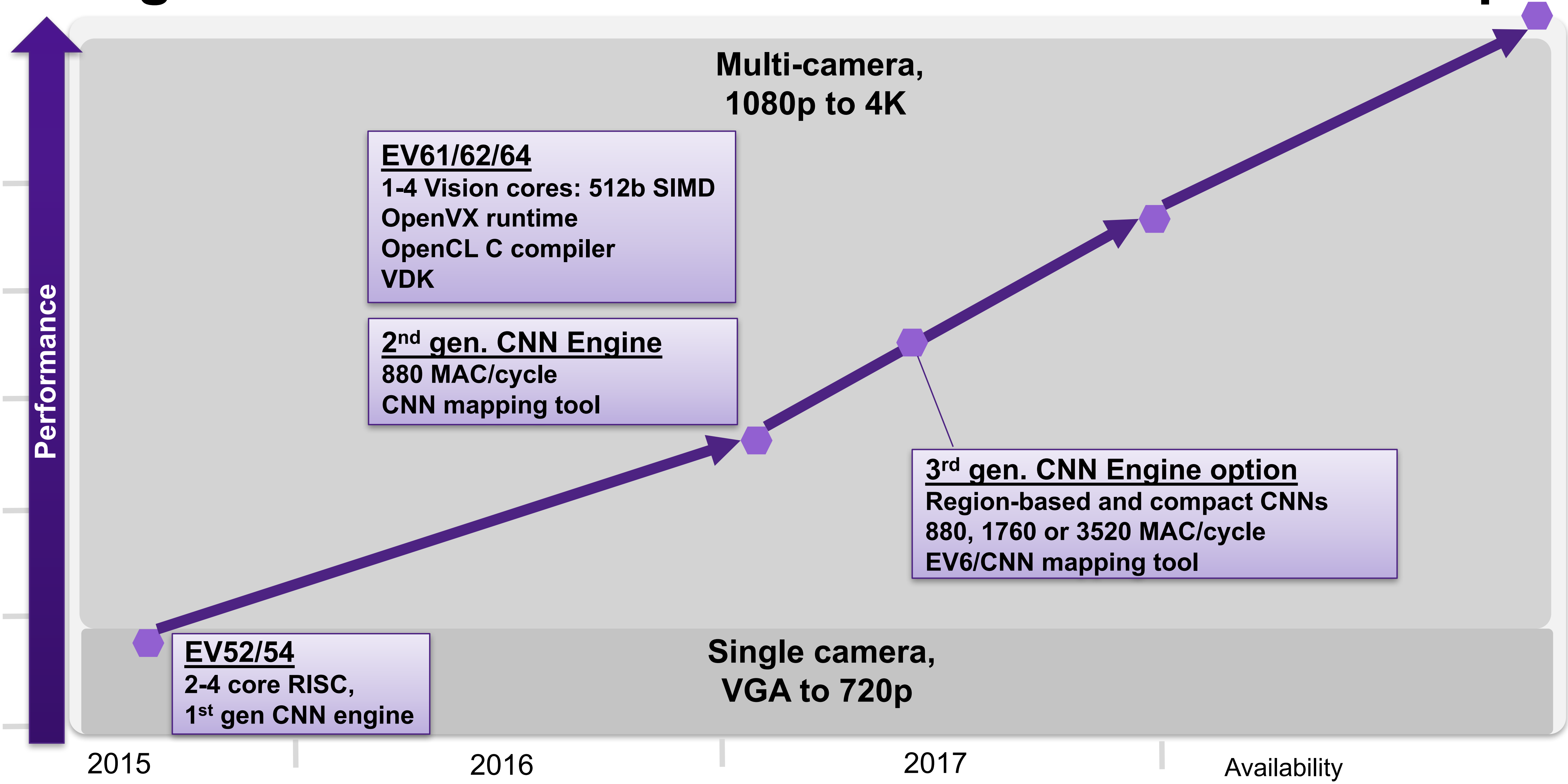
Convolutional Neural Networks

Recurrent Neural Networks

DesignWare® EV6 Processor Family

- *Vision-specific wide SIMD engine*
- *High-performance OpenCL C compiler, OpenVX Runtime*

DesignWare EV Embedded Vision Processor Roadmap



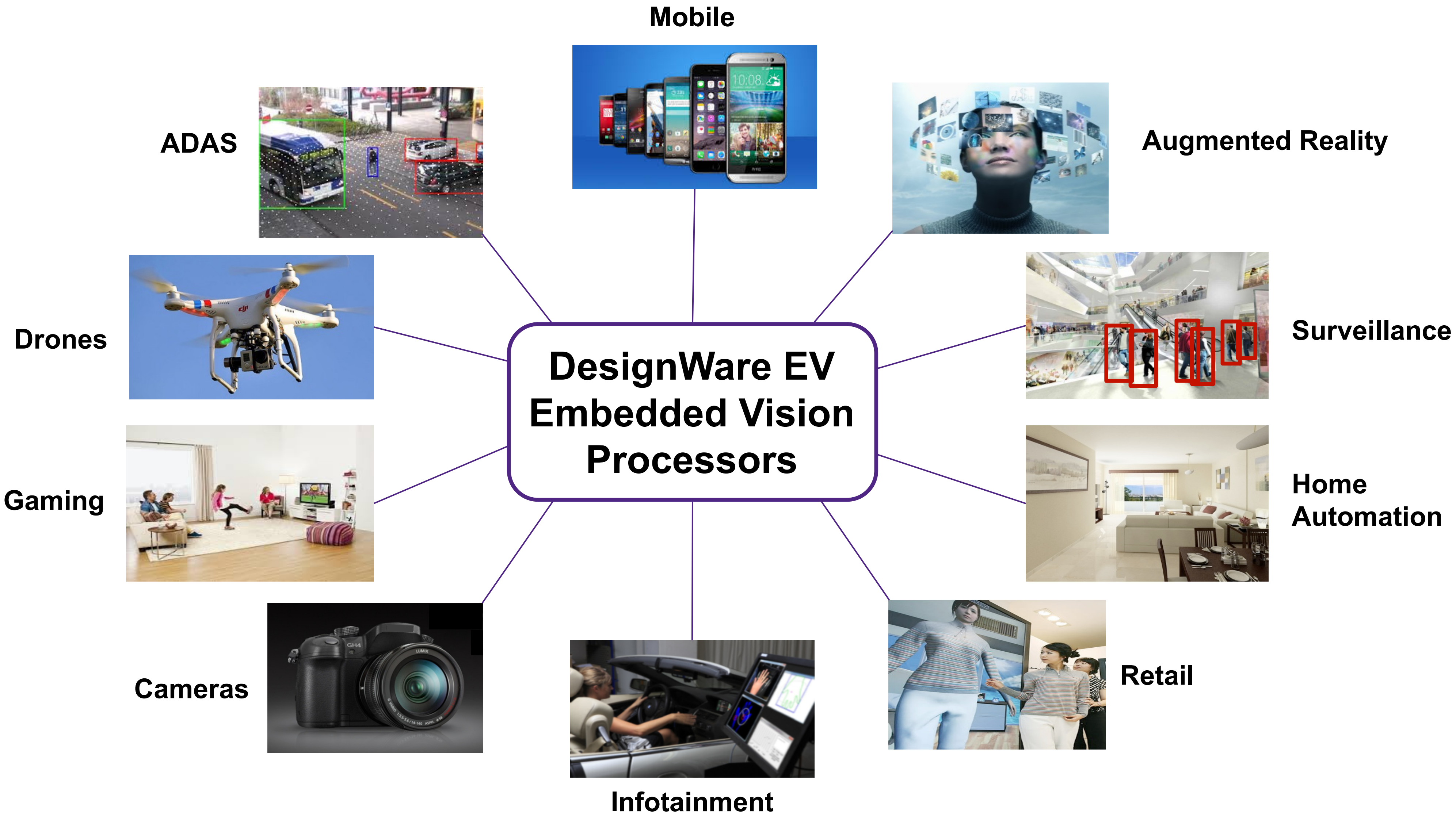
EV52/54
2-4 core RISC,
1st gen CNN engine

EV61/62/64
1-4 Vision cores: 512b SIMD
OpenVX runtime
OpenCL C compiler
VDK

2nd gen. CNN Engine
880 MAC/cycle
CNN mapping tool

3rd gen. CNN Engine option
Region-based and compact CNNs
880, 1760 or 3520 MAC/cycle
EV6/CNN mapping tool

Target Vision Applications



Mobile

ADAS

Augmented Reality

Drones

DesignWare EV Embedded Vision Processors

Surveillance

Gaming

Home Automation

Cameras

Retail

Infotainment

Preliminary – Subject to Change

EV6x Processor Benefits

Most Integrated Solution

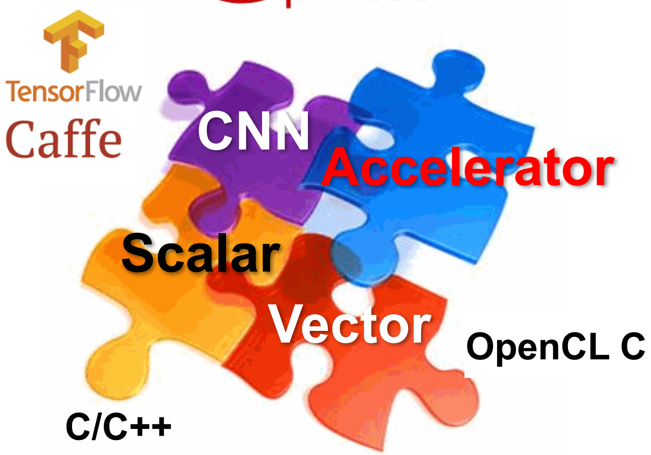
Embedded Vision Libraries

OpenCV

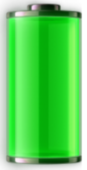
OpenVX

Standard Programming model

OpenVX



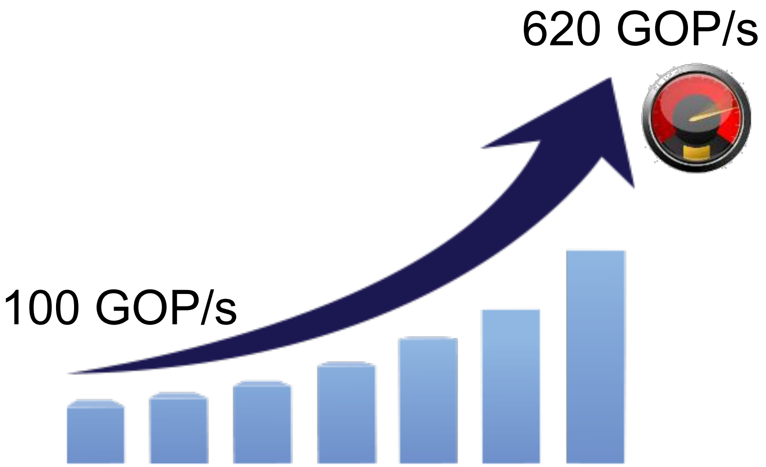
Low power:
Over 1200 GMAC/s/W in CNN engine (16 nm FFC)



Low area:
<1 mm^2 for EV61-vector with CNN engine (16 nm FFC)

High productivity

Highly Scalable Vector Engine

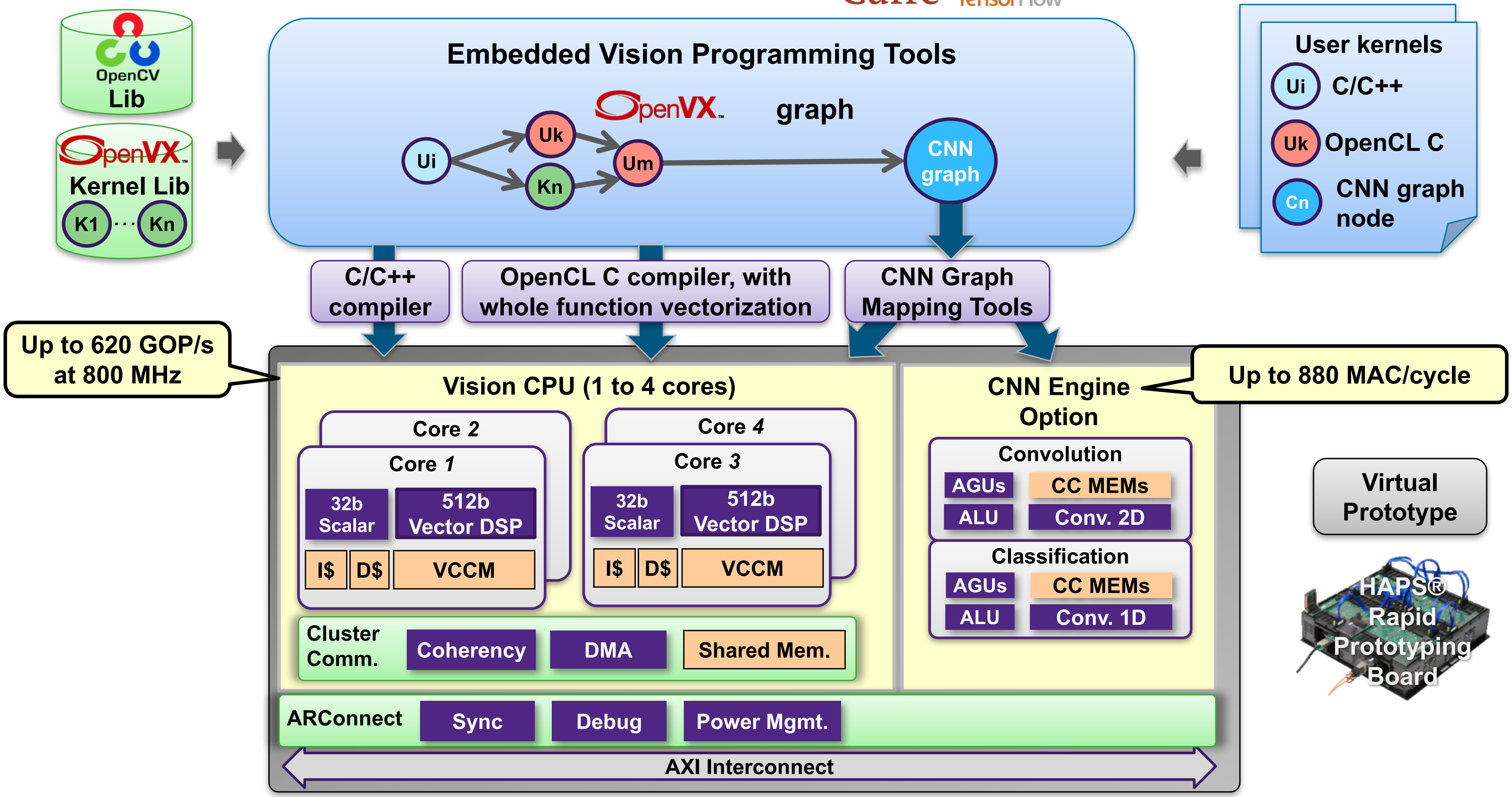


High-performance CNN:

Up to 880 MAC/cycle

EV6x with CNN Engine

Caffe TensorFlow



OpenCV Lib

OpenVX Kernel Lib

K1 ··· Kn

Embedded Vision Programming Tools

OpenVX graph

Ui → Uk, Kn → Um → CNN graph

User kernels

Ui C/C++

Uk OpenCL C

Cn CNN graph node

C/C++ compiler

OpenCL C compiler, with whole function vectorization

CNN Graph Mapping Tools

Up to 620 GOP/s at 800 MHz

Vision CPU (1 to 4 cores)

Core 2

Core 1

32b Scalar

512b Vector DSP

I$ D$ VCCM

Core 4

Core 3

32b Scalar

512b Vector DSP

I$ D$ VCCM

Cluster Comm.

Coherency

DMA

Shared Mem.

CNN Engine Option

Up to 880 MAC/cycle

Convolution

AGUs

CC MEMs

ALU

Conv. 2D

Classification

AGUs

CC MEMs

ALU

Conv. 1D

ARConnect

Sync

Debug

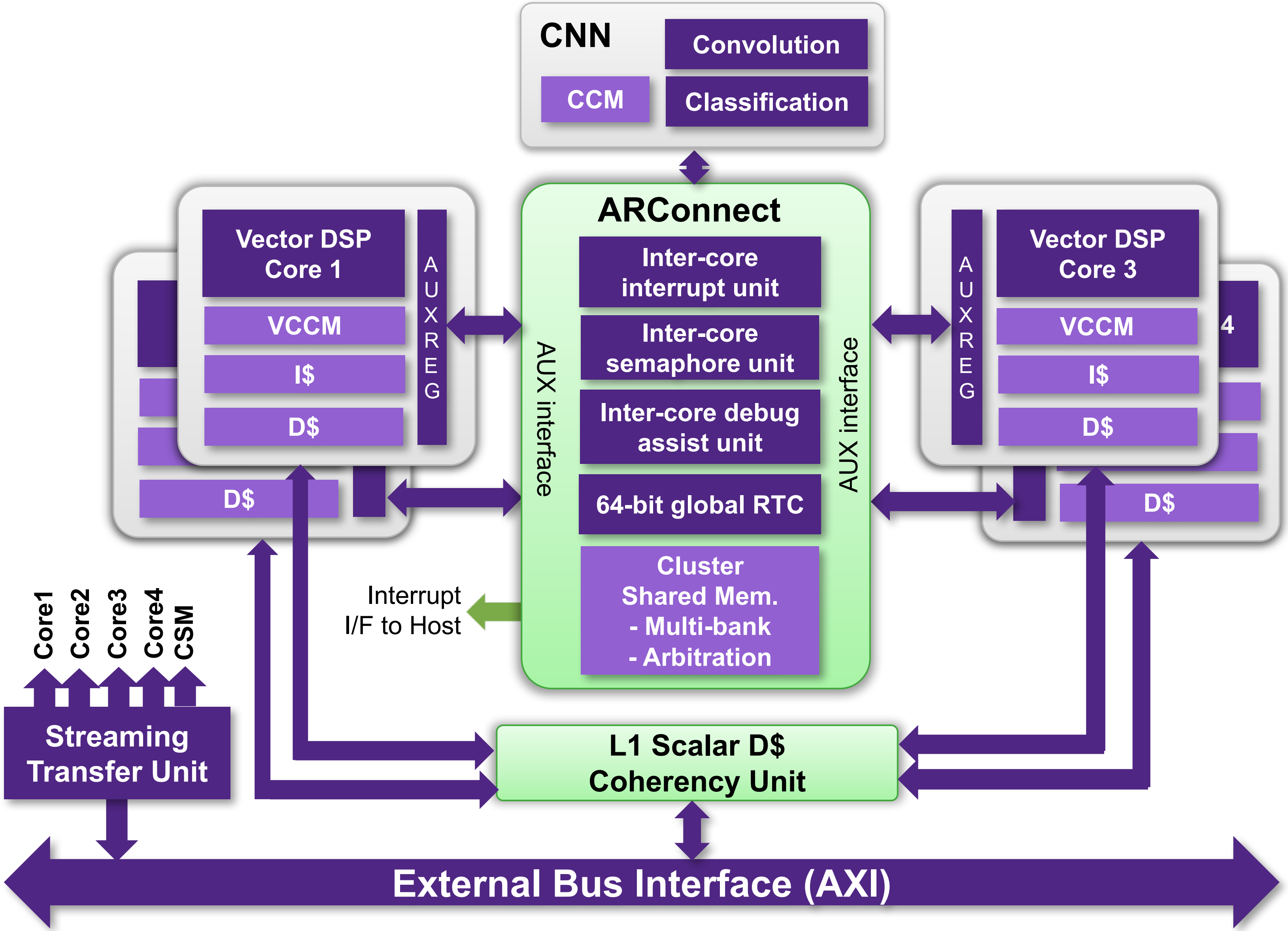
Power Mgmt.

AXI Interconnect

Virtual Prototype

HAPS® Rapid Prototyping Board

EV6x Scalability

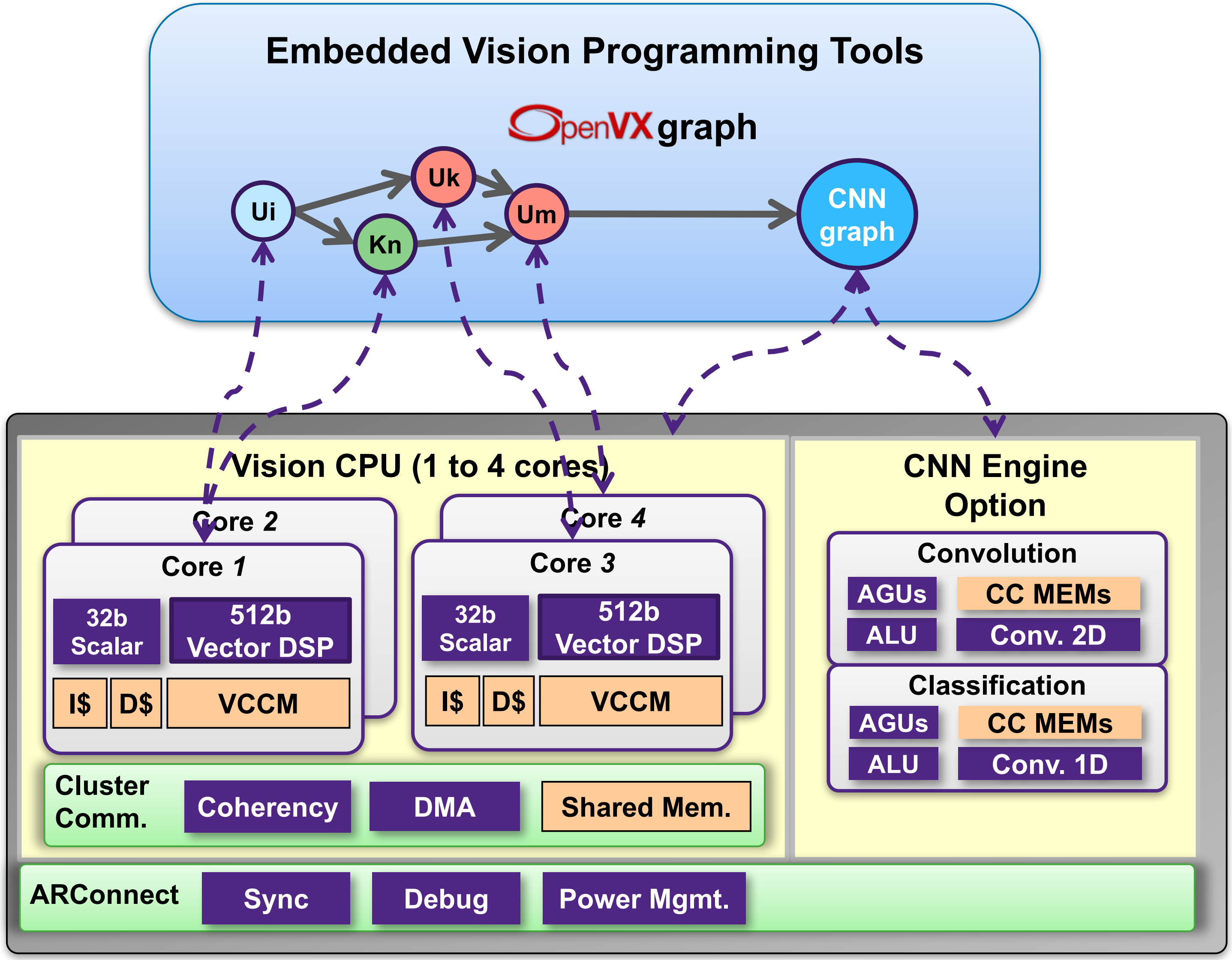


- Explicitly managed memory for high-performance pixel processing
- Support for high efficiency multi-core synchronization and data communication
- Cache-coherent L1 memory for high productivity control code

EV Programming Tools

Based on Embedded Vision standards

OpenVX Graph Mapping in EV Processor

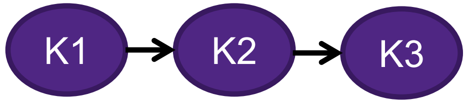


- Runtime performs OpenVX node to processor core assignment and load balancing
  - Option for user-guided assignment
  - Frame or tile-based
- Automatic insertion of communication buffers and memory allocation
  - Option for user-guided memory allocation
  - Extensible to customer H/W accelerators

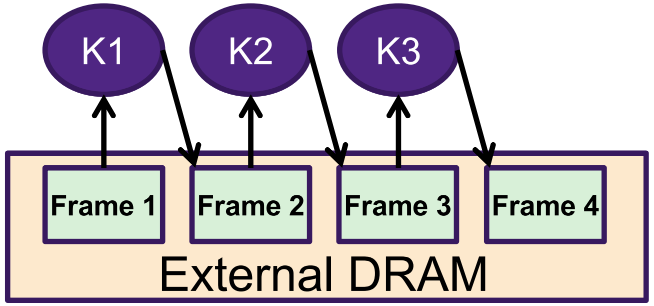
OpenVX Tiling in EV Processor

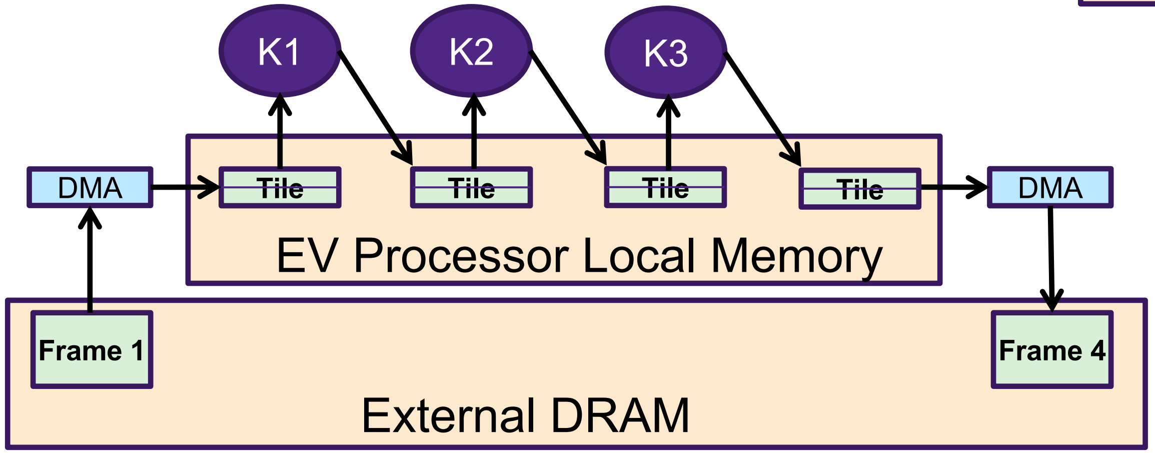
Reducing memory size and power

- Logical Model
  - Data flow between Kernels



- Classical OpenCL Kernel Implementation
  - Host-Device frame buffer movement
  - Efficiency/memory size/power issues!



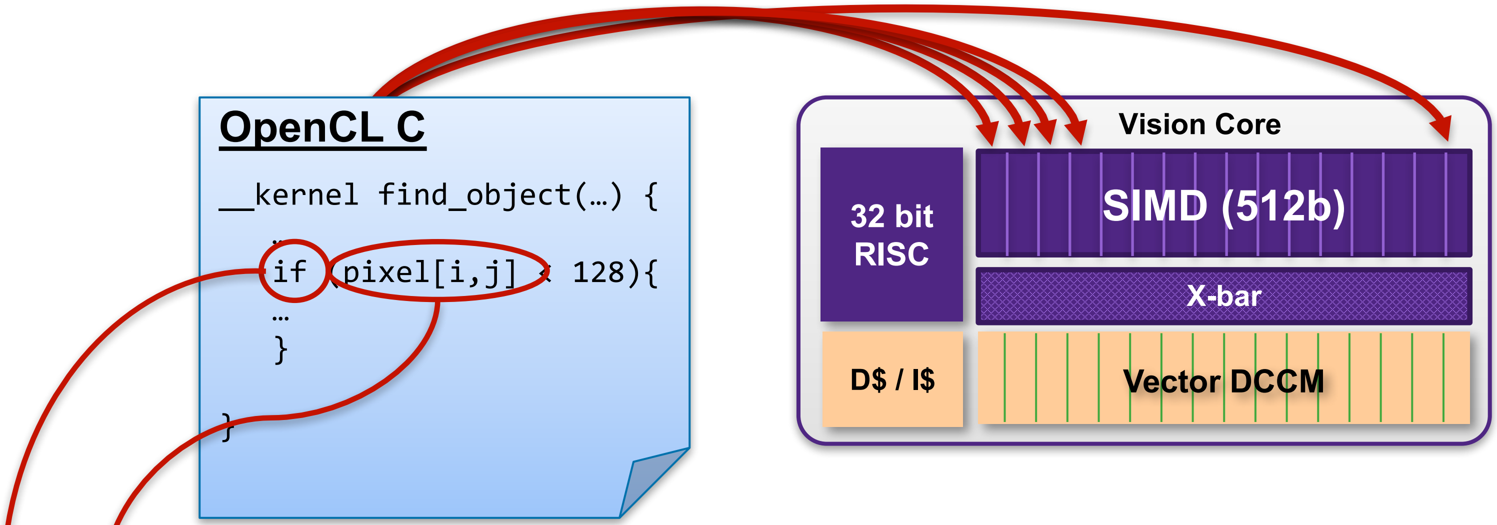


- EV Proc. tiled implementation
  - Data “tunneled” through small(er) local vector memory
  - Enhanced OpenVX/OpenCL runtime
  - Runtime calls kernels directly
  - No round-trip to host

OpenCL™ C Whole Function Vectorization

OpenCL 2.0, embedded profile

The compiler maps OpenCL C kernel on all the SIMD lanes



OpenCL C

```
__kernel find_object(…) {
   …
   if (pixel[i,j] < 128){
   …
   }
}
```

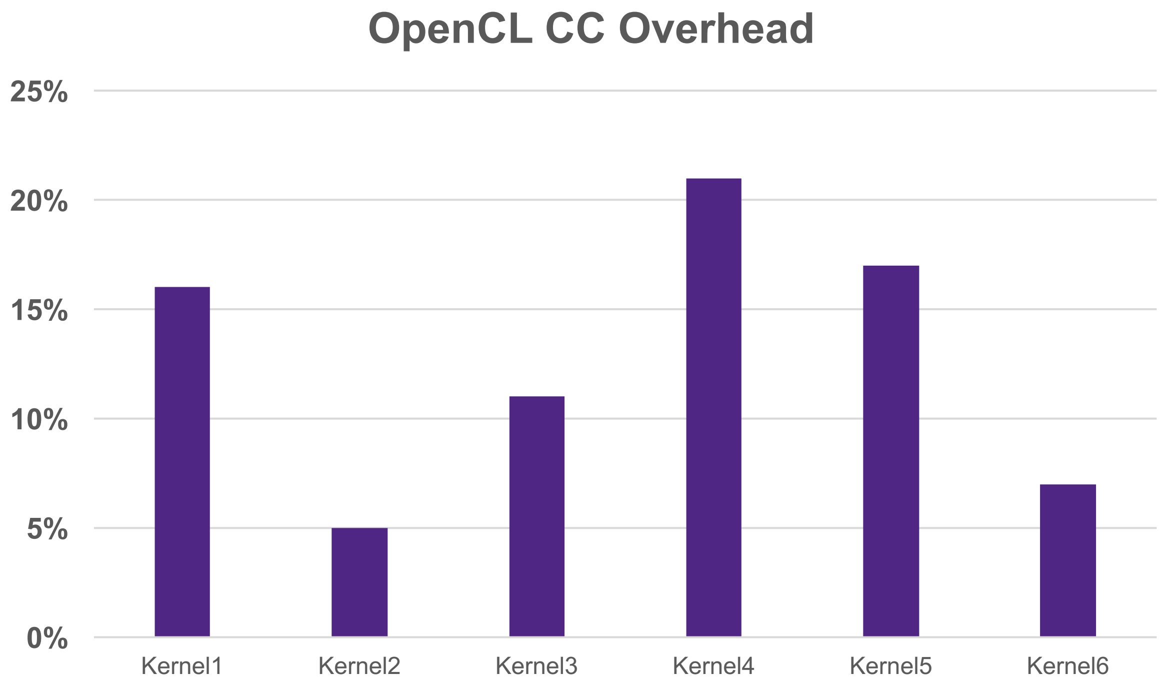
- Lanes execute the same program on different data
  - Every lane works on a different pixel, image patch, decision tree,….
- Every lane can do independent load/stores to the shared Vector DCCM with the X-bar (Scatter-Gather)
- Lane-dependent control-flow is mapped to predicated execution

Preliminary – Subject to Change

OpenCL C compiler efficiency

- Experiments used Synopsys MetaWare OpenCL C compiler
- Overhead measured relative to manually optimized assembly code

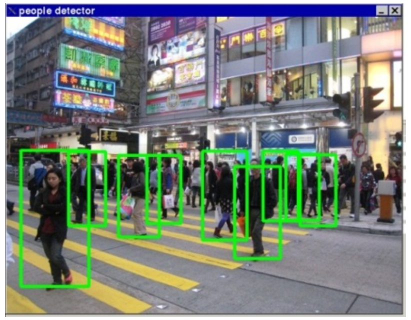
- Features used
  - Wide vectors with multiple data types
  - Predicated scatter/gather built-ins
  - Cross lane reductions/shuffles
  - SIMD based optimized built-ins library
  - Explicit vectorization

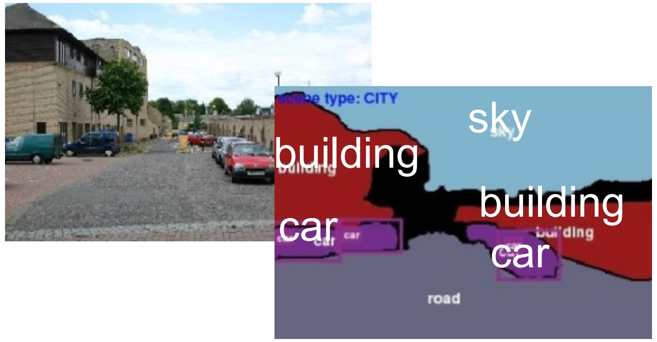


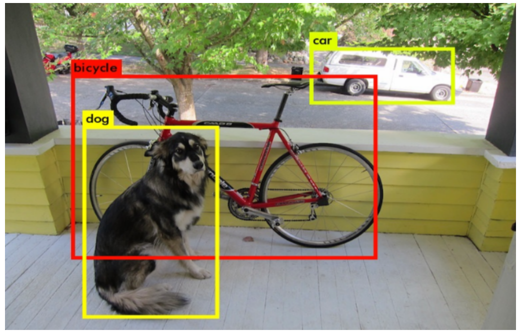
EV6x Third Generation CNN Engine for Neural Network Based Vision Applications

- Leading performance, power and area

- Fully customer programmable

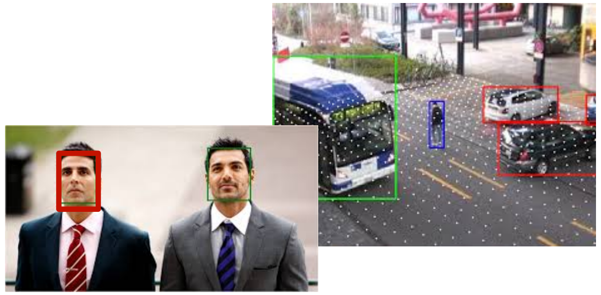


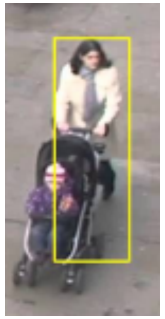


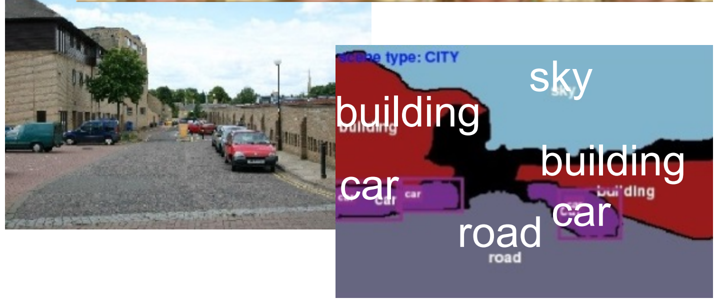


CNN for a Wide Range of Vision Applications

- Image classification, search similar images
- Object detection, classification & localization
  - –Any type of object(s), depending on training phase
- Face recognition
- Visual attention
- Facial expression recognition
- Gesture recognition / hand tracking
- Resolution upscaling
- Scene recognition and labelling, semantic segmentation
  - –Sky, mountain, road, tree, building, …

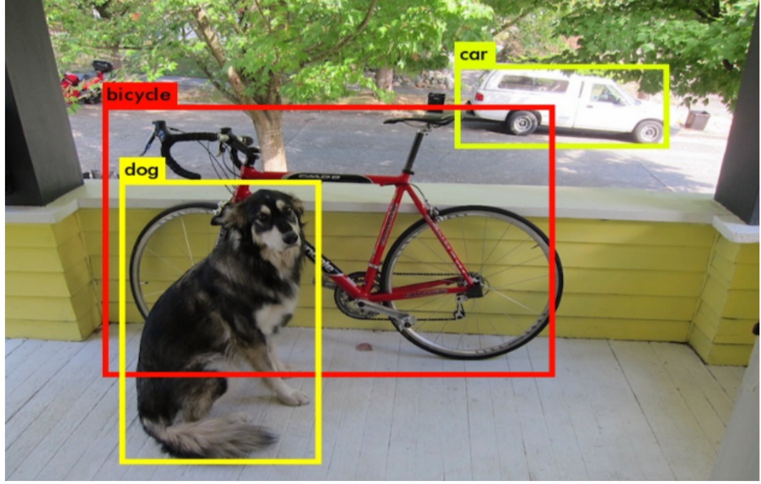




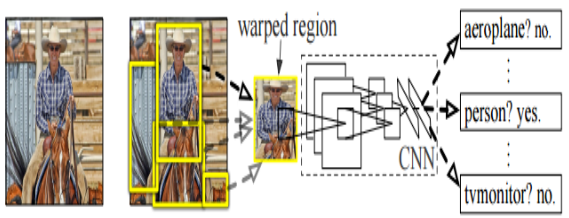


Object Detection with CNNs

Detection: bounding boxes + classification

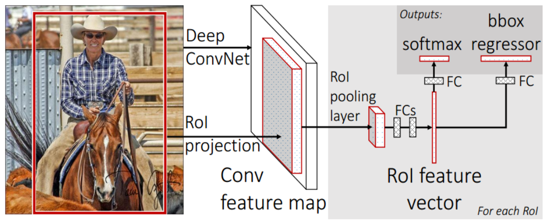


R-CNN



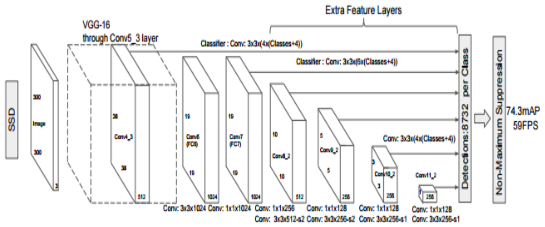
On CCN for finding regions + Full CNN per Region

Faster R-CNN



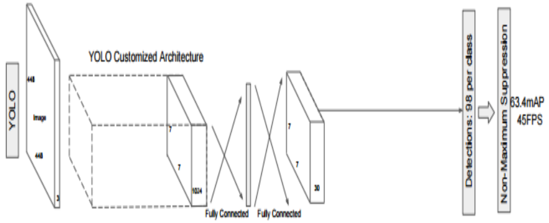
Reuse some of the region CNN for the Classification

SSD



One CNN tapping into multiple Scales for differ object sizes

Yolo V2

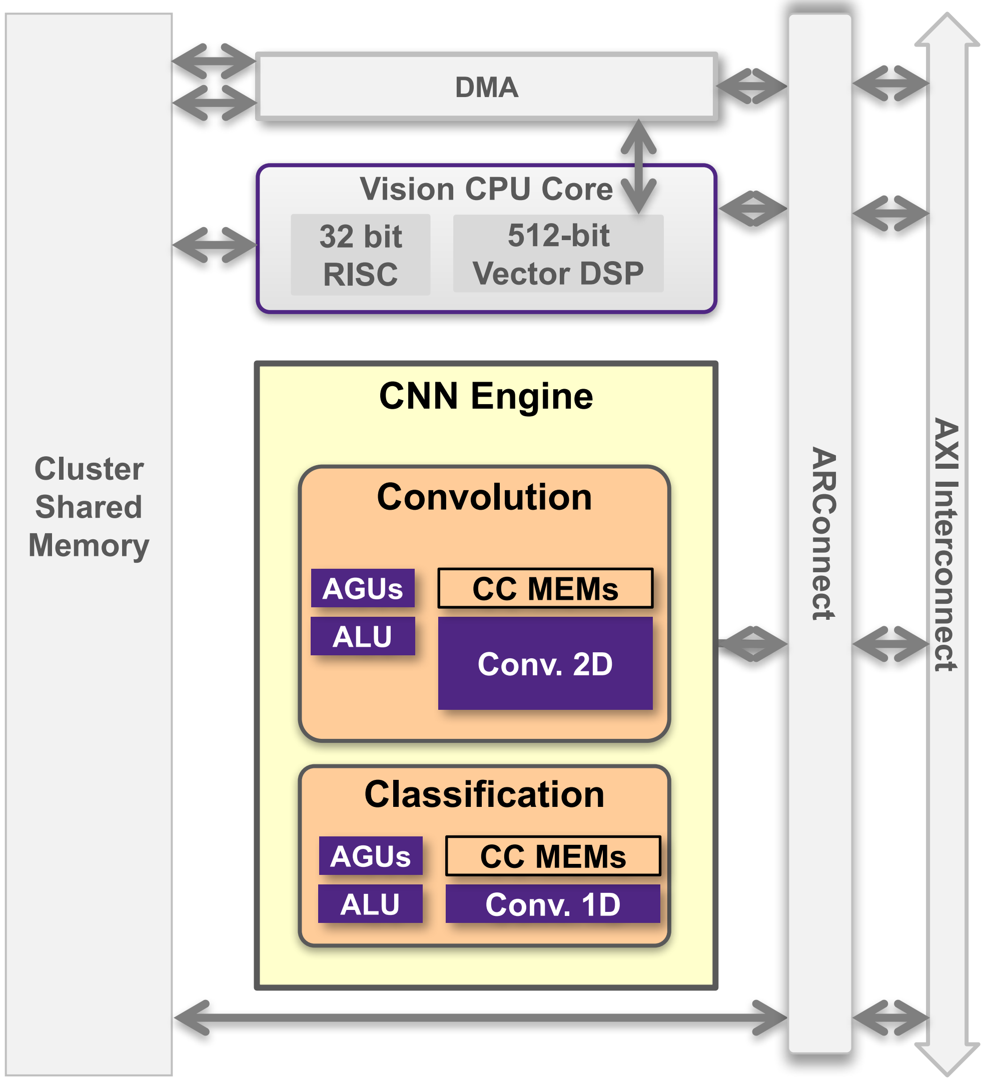


One CNN

New algorithms are not only faster, and more accurate, but also simpler!

Source “Fast(er) R-CNN”, Ross Girshick, Microsoft,   “SSD: Single Shot MultiBox Detector” Liu et al. “YOLO9000: Better, Faster, Stronger

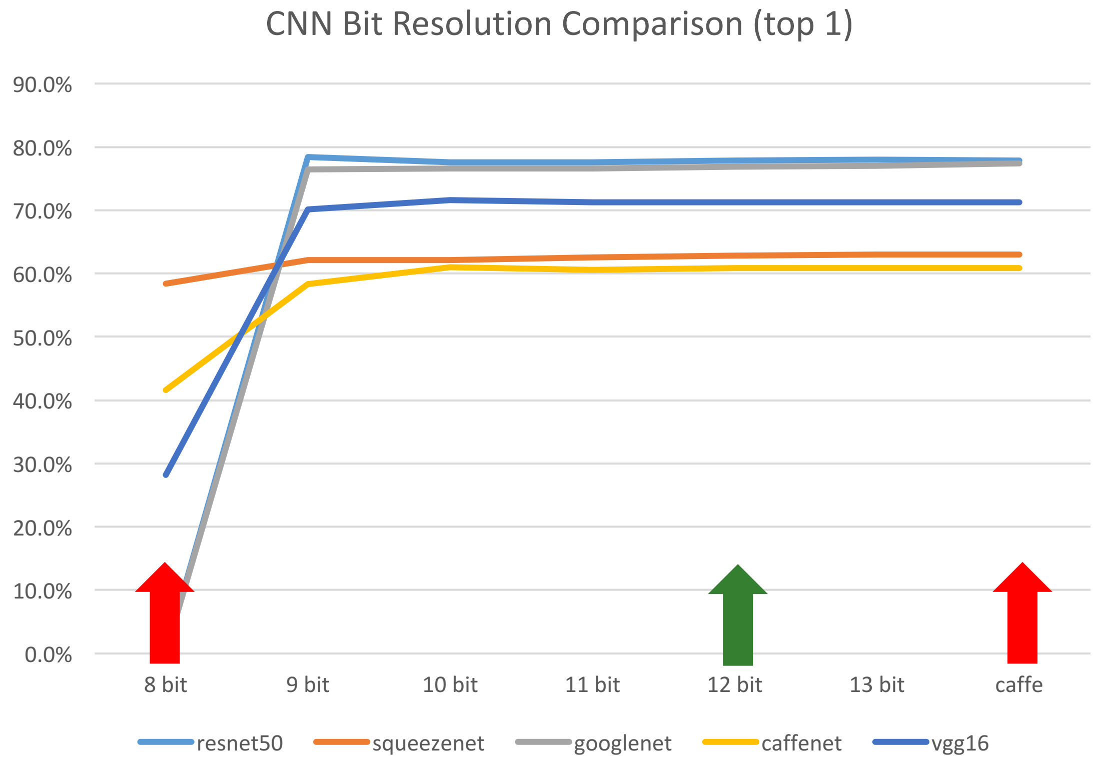
High-Performance EV6x CNN Engine

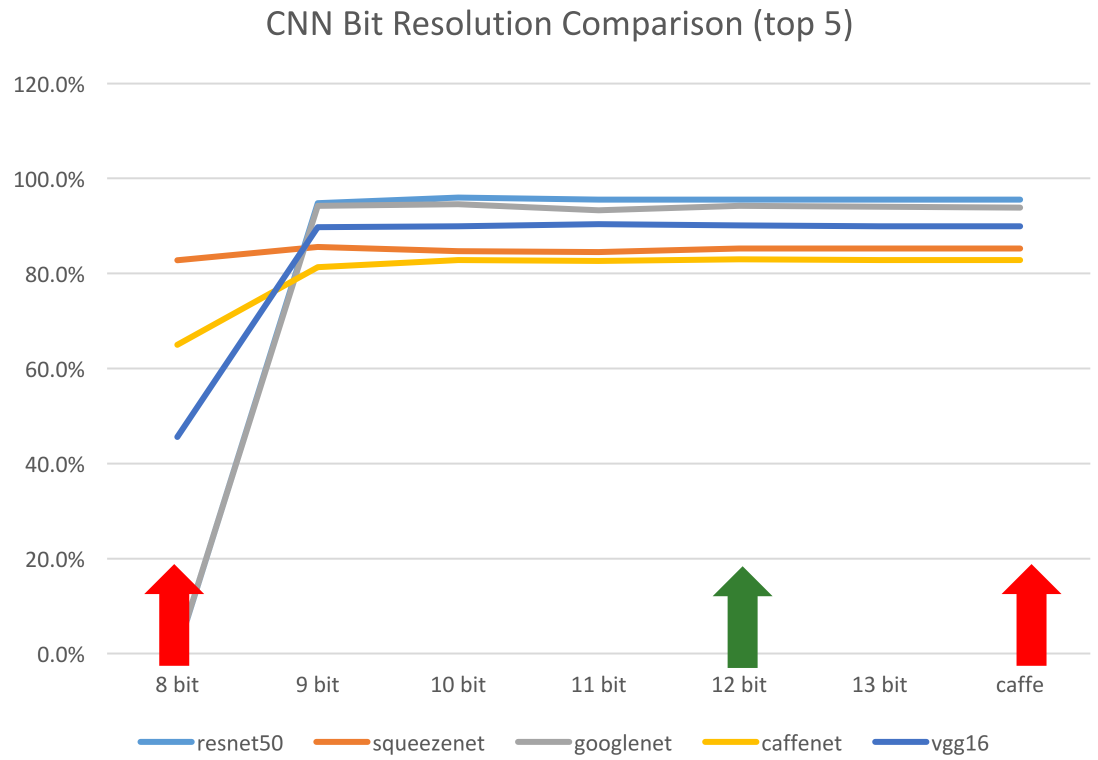


- Dedicated EV6x CNN Engine delivers high performance from 880 to 3520 MACs/cycle
- Fully programmable to support full range of fixed point CNN graphs
- State-of-the-art power-efficiency >1200 GMAC/s/W
- Supports resolutions up to 4K
- Real-time, high quality image classification, object detection, semantic segmentation
- Operates in parallel with Vision CPUs increasing efficiency and throughput

Bit width impact on Detection Accuray

Functional simulation model with varying bit widths (ILSVRC Graphs / Caffe Trained Models)

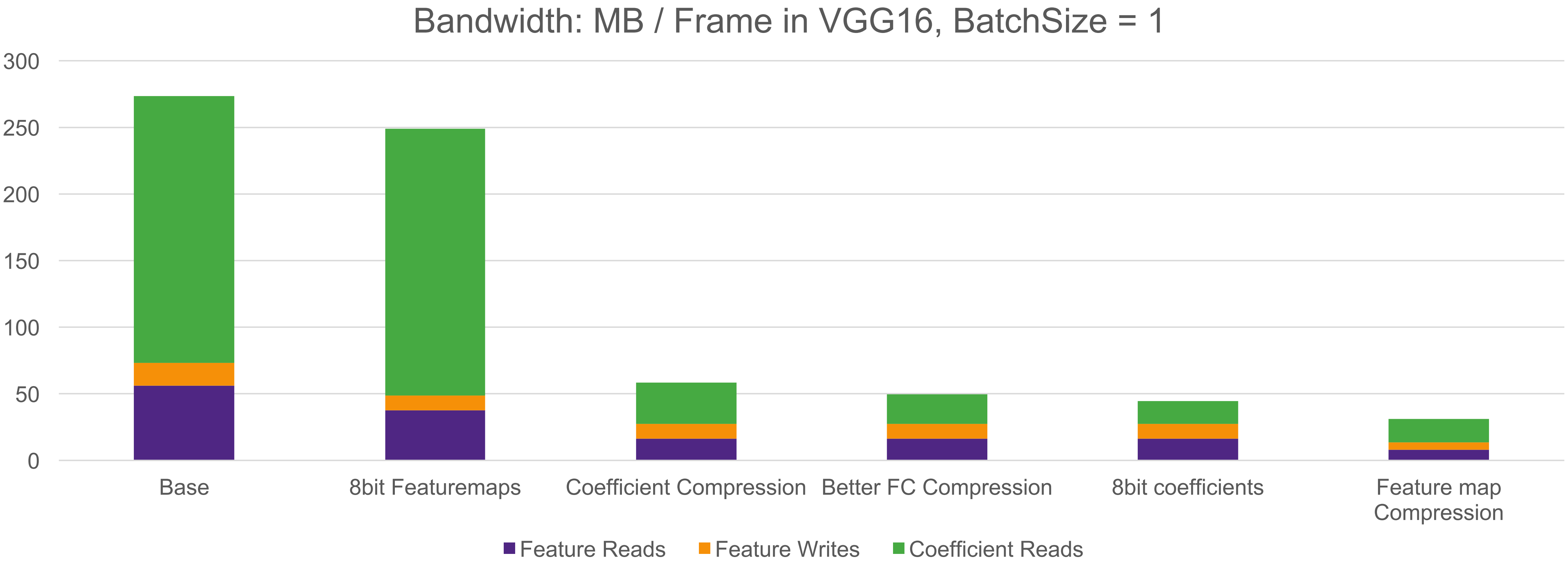




TOP1	8 bit	9 bit	10 bit	11 bit	12 bit	13 bit	caffe
resnet50	0.2%	78.4%	77.6%	77.6%	77.8%	78.0%	77.8%
squeezenet	58.4%	62.2%	62.2%	62.6%	62.8%	63.0%	63.0%
googlenet	0.2%	76.4%	76.6%	76.6%	76.8%	77.0%	77.4%
caffenet	41.6%	58.4%	61.0%	60.6%	60.8%	60.8%	60.8%
vgg16	28.2%	70.2%	71.6%	71.2%	71.2%	71.2%	71.2%

TOP5	8 bit	9 bit	10 bit	11 bit	12 bit	13 bit	caffe
resnet50	0.6%	94.8%	96.0%	95.6%	95.6%	95.6%	95.6%
squeezenet	82.8%	85.6%	84.8%	84.6%	85.2%	85.2%	85.2%
googlenet	0.8%	94.2%	94.6%	93.4%	94.2%	94.0%	93.8%
caffenet	65.0%	81.4%	82.8%	82.6%	83.0%	82.8%	82.8%
vgg16	45.6%	89.8%	90.0%	90.4%	90.2%	90.0%	90.0%

Bandwidth Reduction: VGG16 Example



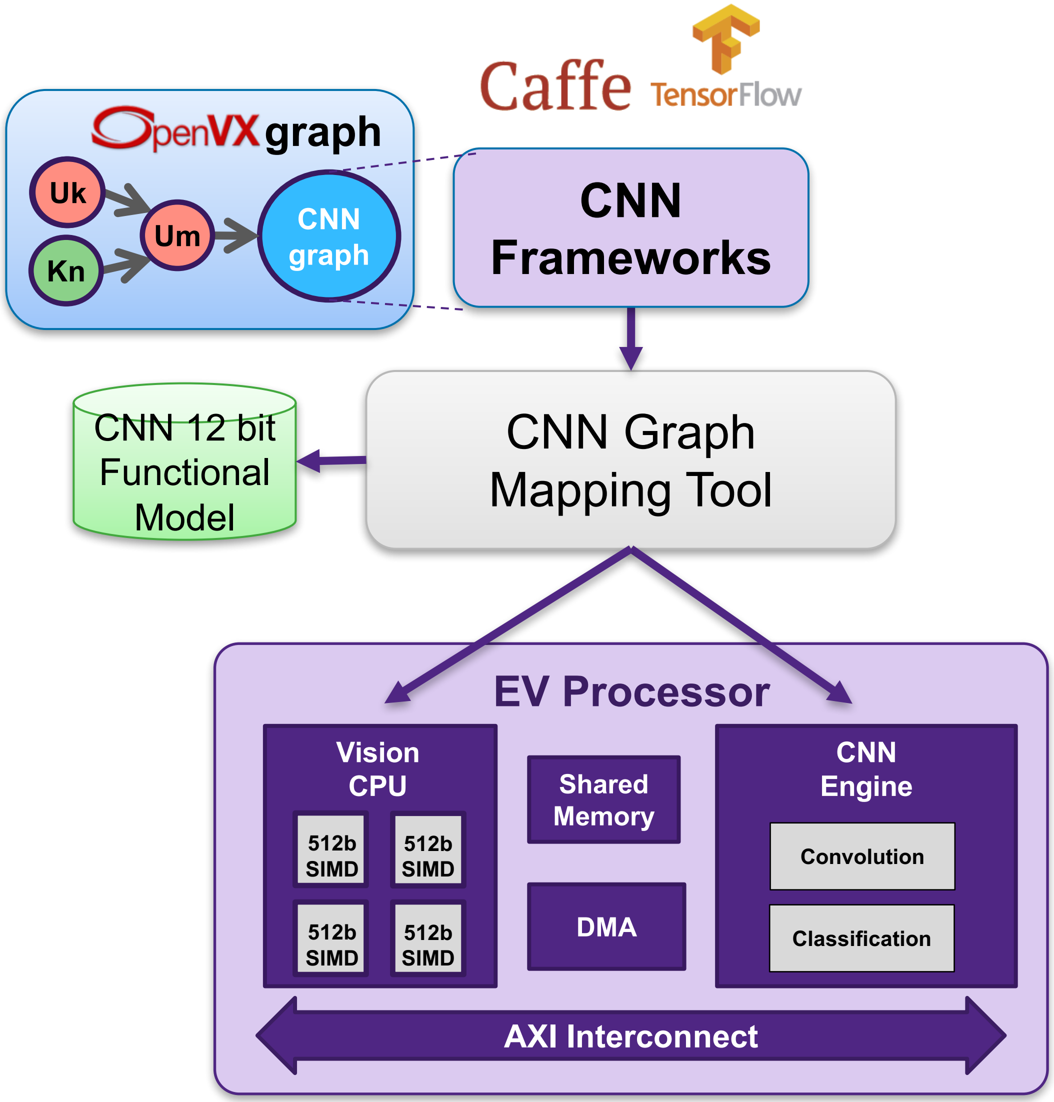
9X bandwidth reduction

Note: Single Batch VGG16 is worst-case scenario for Coefficients Bandwidth. More modern graphs have much less coefficients.

CNN Mapping Tool

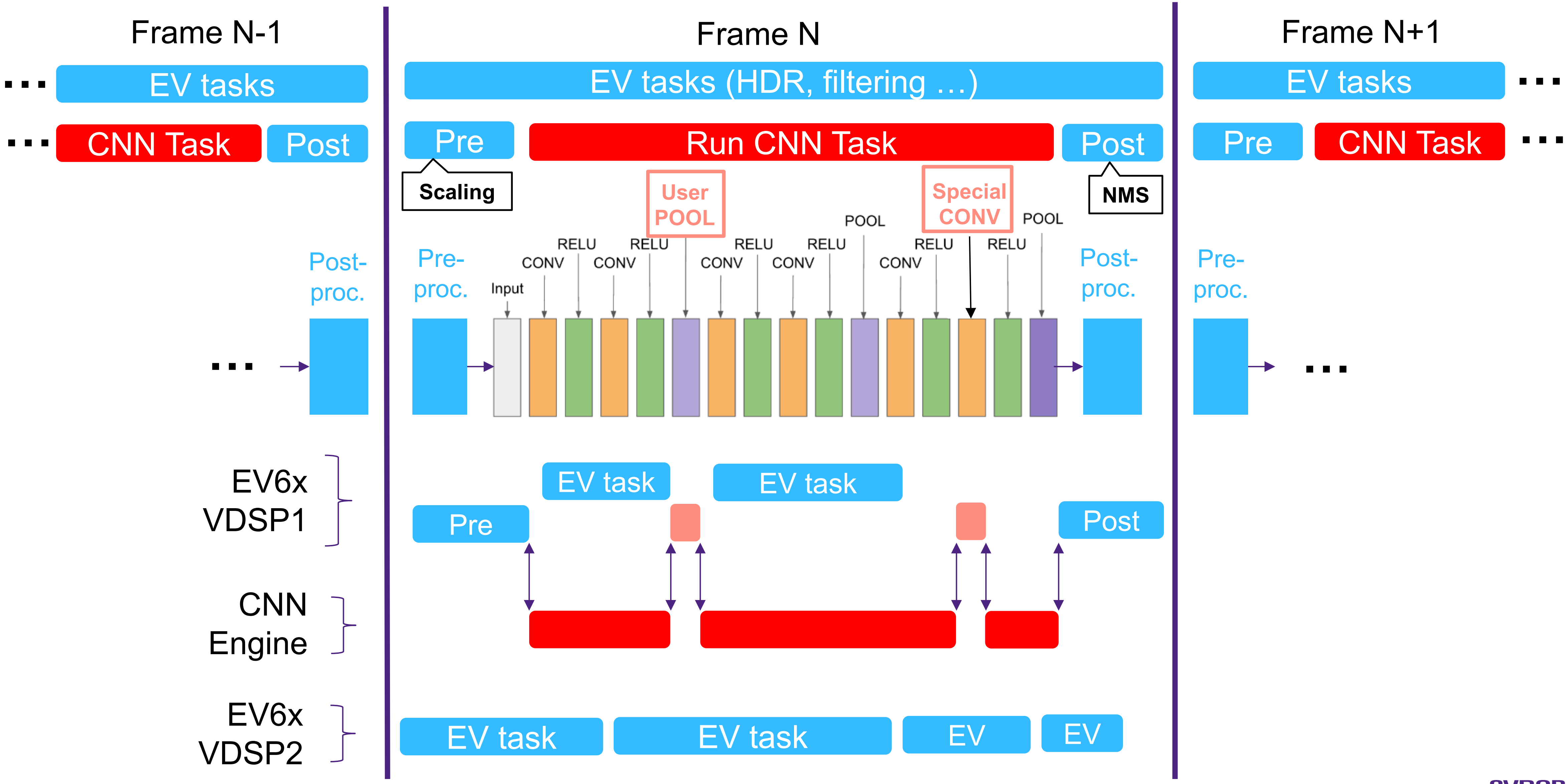
Preliminary – Subject to Change

Mapping to EV6x Cores and CNN engine



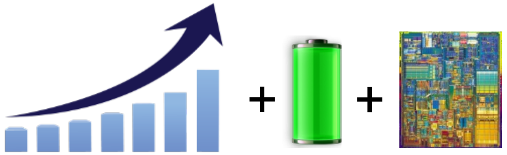
- Push-button CNN Mapping tool
  - Accepts Caffe Graphs with supported features
  - Import of Tensorflow graphs
- Native functions mapped to CNN engine
  - Automatic conversion to 12 bit dynamic fixed point
- Distributed execution on EV6x core(s)
  - Flexibility for new functionalities
    - New CNN innovations
    - RNN (LSTM, Quasi-RNN)
  - Support of rare or legacy functionalities
    - Loss layers
    - Local Response Normalization
    - All pooling layers that are not natively supported by CNN engine
  - Higher performance functions on vector core(s)
  - Non-performance critical features on scalar core
  - Customer-defined custom CNN layer
    - Programmed in standard C/C++ or OpenCL C

Distributed execution on EV6x core(s) and CNN engine



DesignWare EV6x Summary

- Highly integrated and scalable solutio
  - Scalar + Vector DSP + CNN Engine
  - Designed for heterogeneous multicore processing



- State-of-the art PPA
  - <1 mm^2 for EV61 Vector DSP and CNN engine (16 nm FFC)
  - CNN Engine delivers over 1200 GMAC/s/W (16 nm FFC)

OpenCL C

- High productivity toolset
  - OpenVX, OpenCL C with whole function vectorization, OpenCV libraries
  - Automatic CNN graph mapping tool
  - Future-proof with distributed processing
- Synopsys – a global partner in IP licensing and EDA tools

SYNOPSYS®
Silicon to Software™

Thank You