CEVA

CDNN2

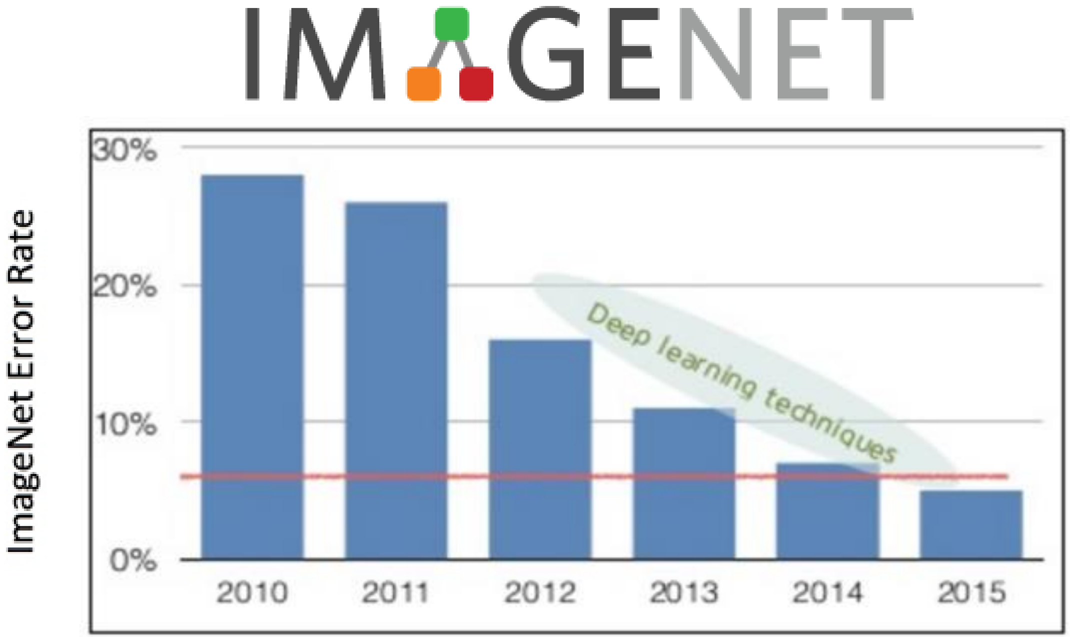
CEVA Deep Neural Network

Liran Bar, Director of Product Marketing (Liran.Bar@ceva-dsp.com)

July 2016 – Please contact us for the update material

www.ceva-dsp.com

Deep Learning Performance Improvements

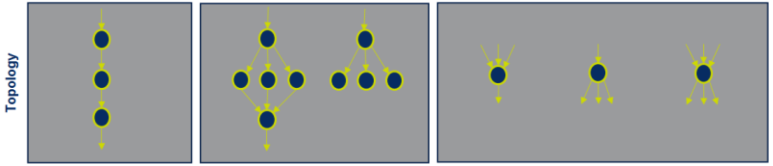


Human Performance

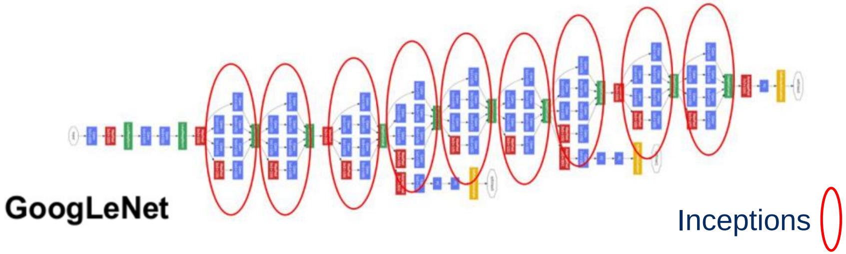
Source: Nervana Systems

The Neural Networks Challenge

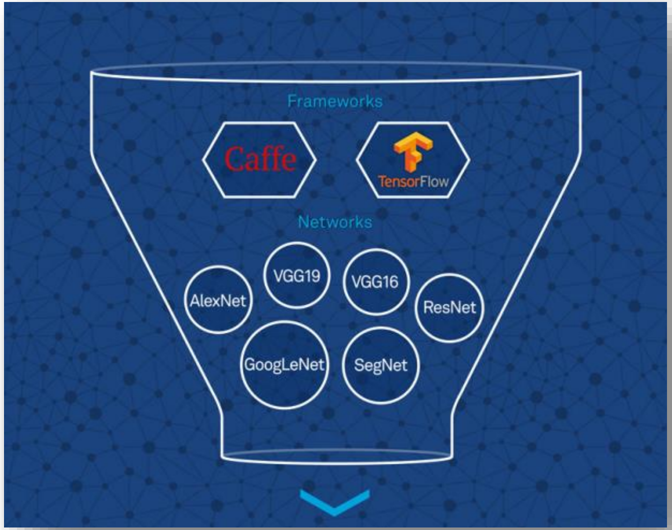
- Various training **frameworks**: Caffe, TensorFlow, Torch etc.
- Various **layers**: Convolutional, Normalization, Pooling etc.
- Various network **topologies**



- Need to deal with network inside a network



- Need to have optimized solution for variable sizes of ROI



Leading Deep Learning Frameworks

BVLC Caffe

- A well-known and widely used machine-vision library
- Implementation of fast convolutional nets to C and C++
- Made with expression, speed, and modularity in mind
- Used by researchers, academy, and commercial companies

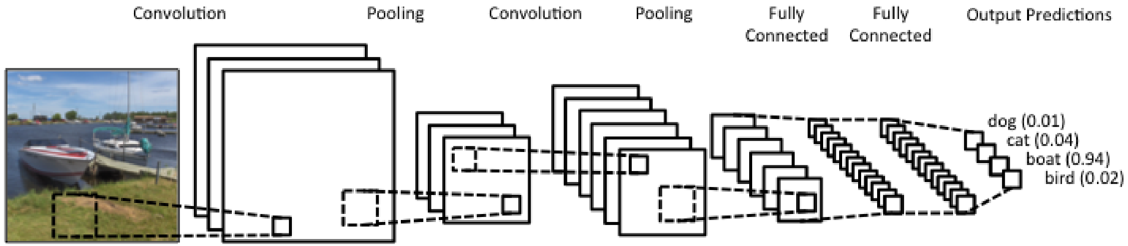
Google TensorFlow

- Relatively new alternative to Caffe, supported and promoted by Google
- Scalable to work both for research and commercial purpose without making any changes
- A software library for numerical computation using data flow graphs

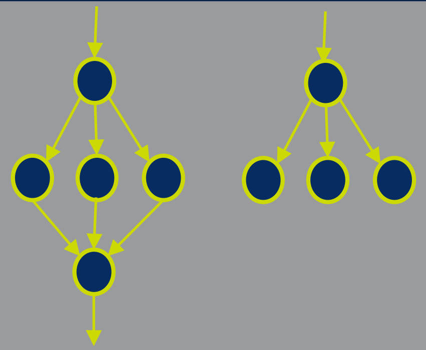
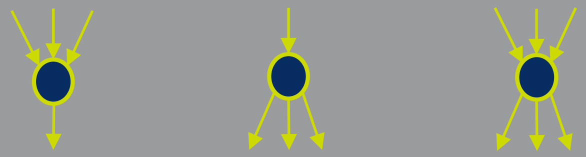
Leading Neural Network Layers

- Convolutional
- Normalization
- Pooling (Average and Max)
- Fully Connected
- Activation (ReLU, Parametric ReLU, TanH, Sigmoid)
- Deconvolution
- Concatenation
- Upsample
- Argmax
- Softmax

Flexible embedded solution is required to cope with the evolving and leading neural network layers



Deep Learning Topologies

	Linear Networks	Multiple Layers Per Level	Multiple-Input-Multiple-Output
Topology			 (a) (b) (c)
Networks	AlexNet VGG-19 VGG-16 VGG_S	GoogLeNet	GoogLeNet SegNet ResNet

Neural Network Embedded Challenges

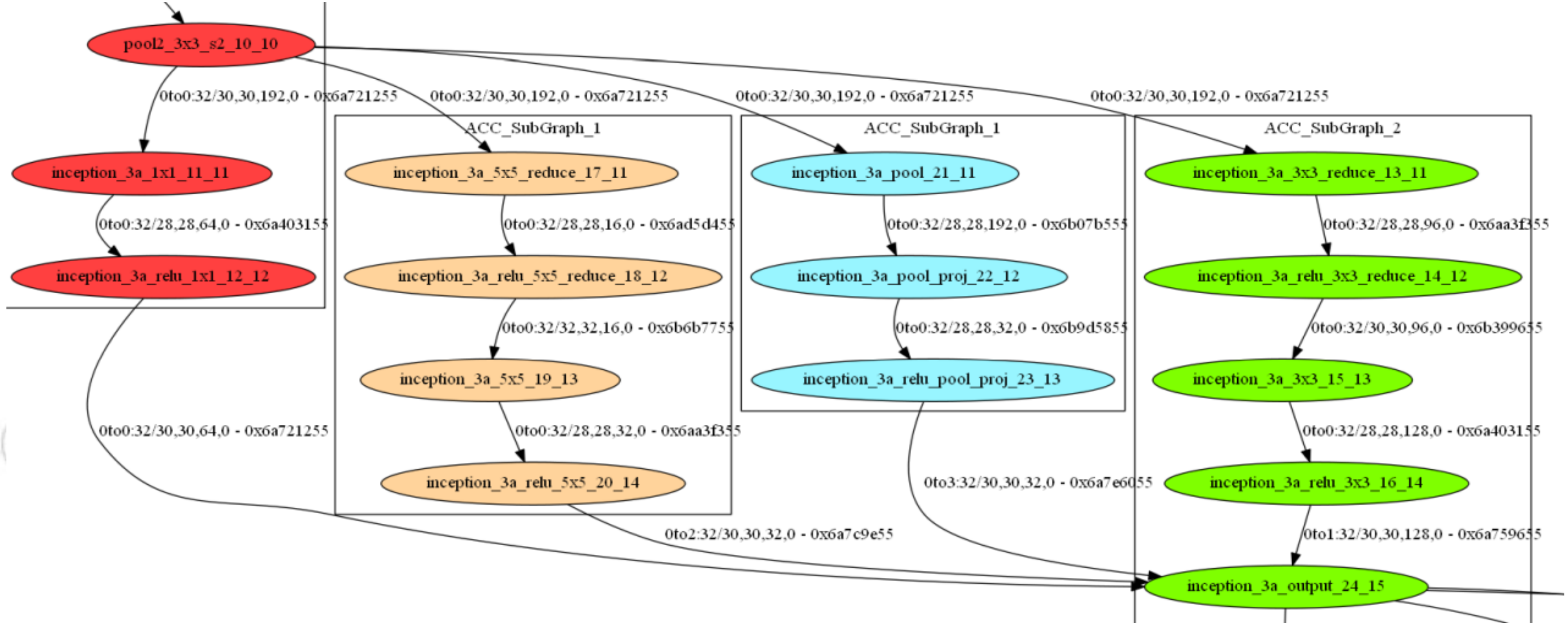
Implementing a deep neural network in embedded systems is a challenging task!

- Very high bandwidth consuming and computing bottleneck
  - Data transfer in/out the DDR – Input/output maps
  - Convolution and Fully Connected data weights from DDR
  - Processing multi ROI with the same network
  - Internal memory size limitation on embedded platforms
- Porting and optimizing neural networks to embedded system consumes time!
  - Special programing knowledge (intrinsic, assembly)
  - Specific experience in the embedded platform (instructions, hardware capabilities)
  - Fixed HW solutions are not flexible to cope with the evolving neural networks

Long "Time-To-Market"

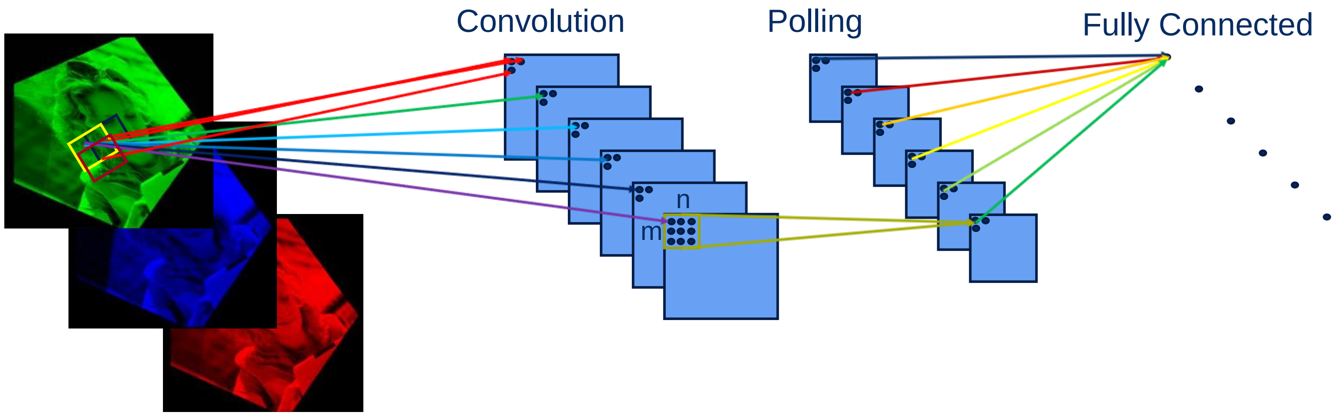
GoogleNet Challenge

- How do you run a network like GoogLeNet with minimum transaction through DRR?



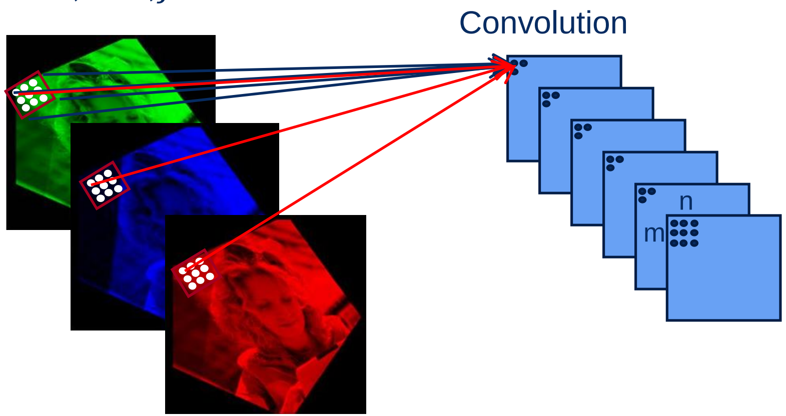
What can be done?

- **Reducing bandwidth**
  - In the convolution layer, each output is calculated by the same inputs
    - Weights matrix are shared between output results in the same map (in order not to load the weights more than once)
    - The input data can be reused to avoid useless transactions from DDR



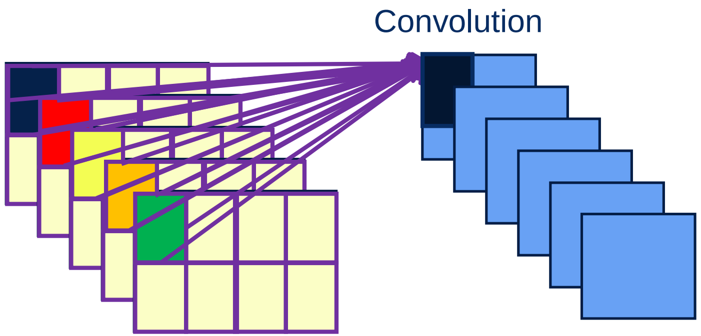
What can be done? – Cont.

- **Maximum multiply accumulate utilization**
  - Differentiate between large inputs to small input maps and the number of maps from each type
  - Large size maps - $X^{l}_{c,i,j} = \sum_{c=0}^{C} \sum_{i=0}^{H} \sum_{j=0}^{W} W^{l}_{c,m,n} X^{l-1}_{c,i+m,j+n}$
  - Large amount of maps with small size (last layers) - $X^{l}_{c,i,j} = \sum_{i=0}^{H} \sum_{j=0}^{W} \sum_{c=0}^{C} W^{l}_{c,m,n} X^{l-1}_{c,i+m,j+n}$



What can be done? – Cont.

- **Overcome small internal memory size**
  - Try to preserve the principle of “All inputs must be in the internal memory” by tile division
  - Divide all input maps to identical tile sizes



What can be done? – Cont.

- Use compression algorithms and prior knowledge to reduce bandwidth to and from the external memory
  - Using algorithms like huffman coding
  - Work in pipeline to save BW
  - Identify when some of the calculation can be saved
  - Share data between calculations
  - Recognize when the focus should be on the weights and when it should be on the map size – network dependent
  - Compress and decompress better over time (learn from frame by frame execution)

CEVA Deep Neural Network (CDNN2)

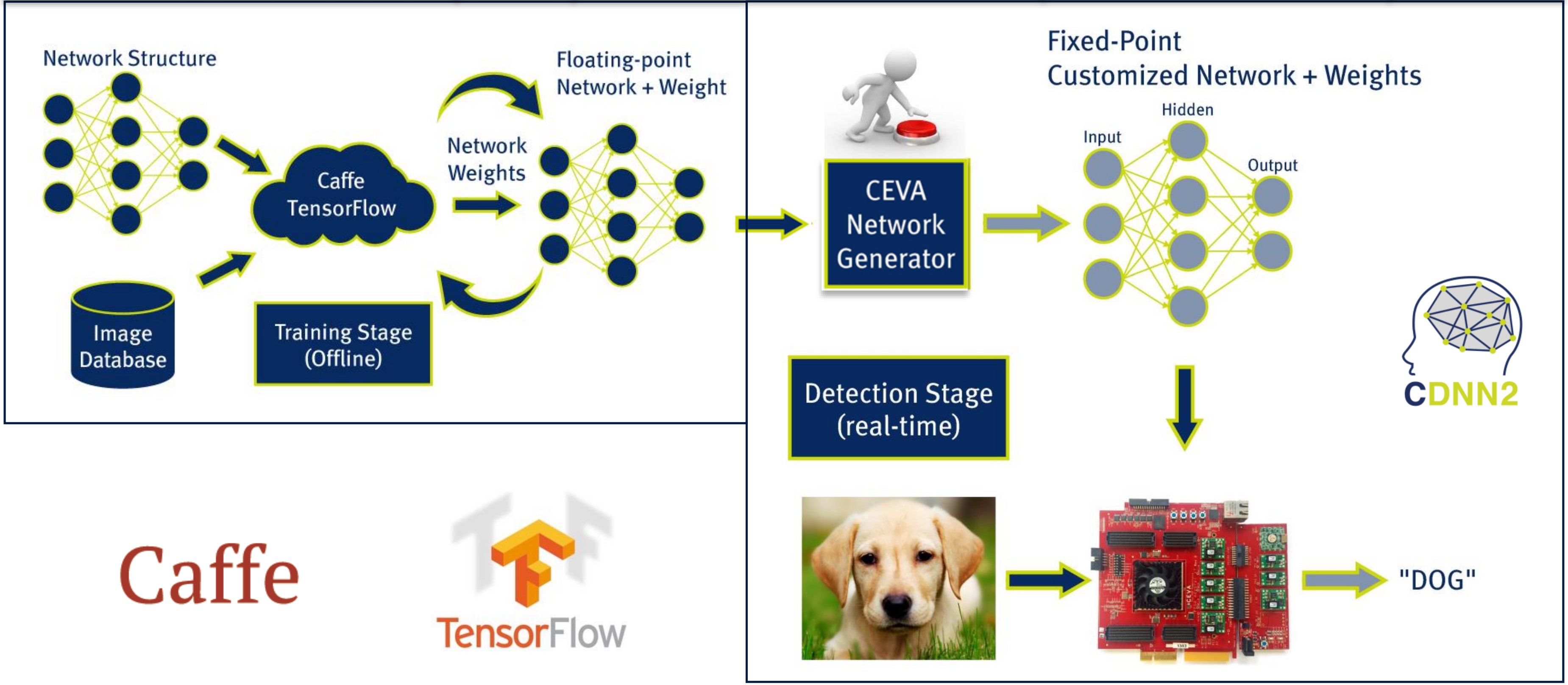
- 2nd gen SW framework support
  - Caffe and TensorFlow Frameworks
  - Various networks*
  - All network topologies
  - All the leading layers
  - Variable ROI
  - “Push-button” conversion from pre-trained networks to optimized real-time
  - Accelerates machine learning deployment for embedded systems
  - Optimized for CEVA-XM4 vision DSP

(*) Including AlexNet, GoogLeNet, ResNet, SegNet, VGG, NIN and others

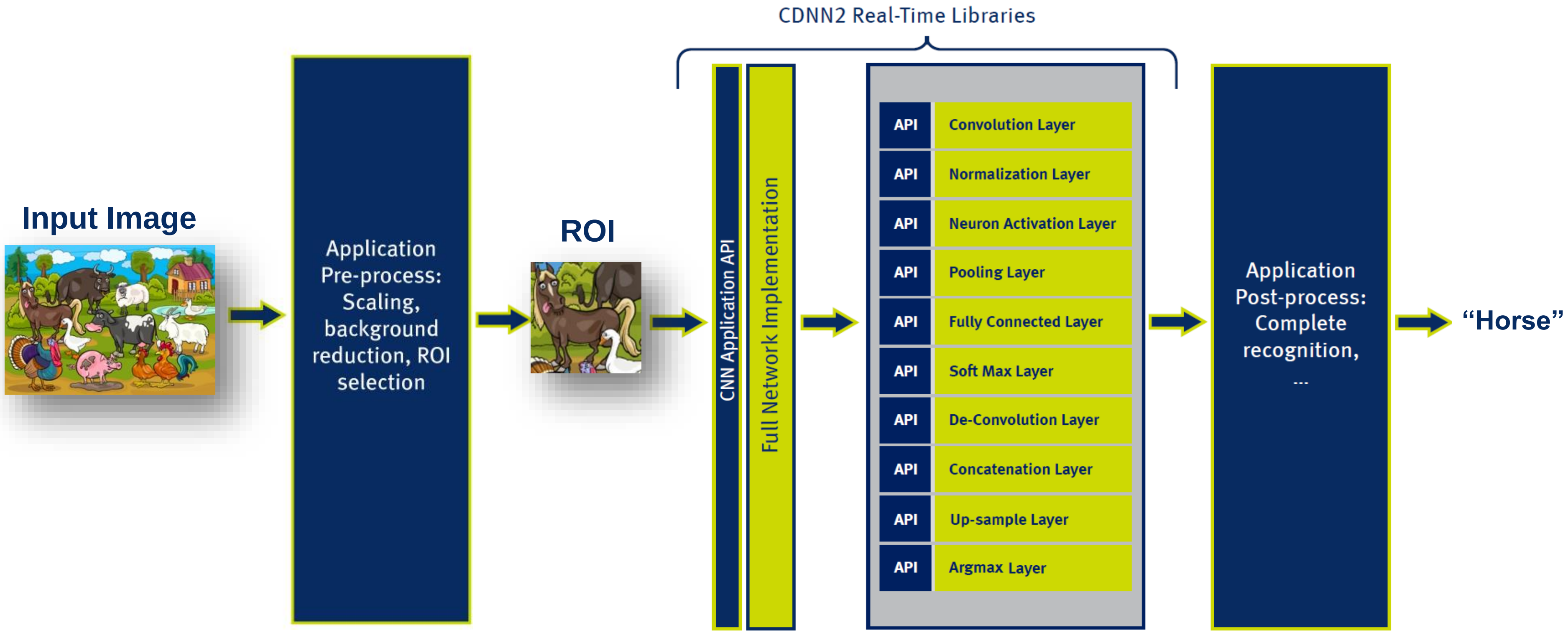
CDNN2 Usage Flow

CEVA

OEM / Partner (offline)	**CEVA (offline + real-time)**

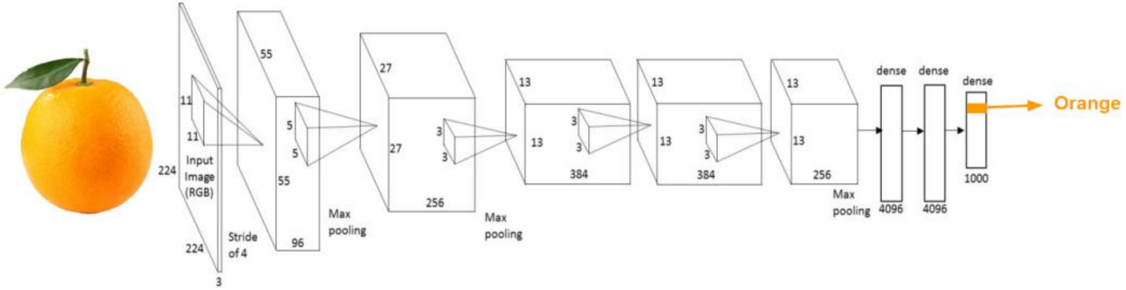


Real-Time CDNN2 Application Flow



AlexNet - Network Performance

- Network specification
  - Full forward classification case image measurement (single iteration)
  - 24 layers, 224x224 network input size
  - 11x11, 5x5 and 3x3 convolution filters
- Memory bandwidth
  - Pre-trained network: 253Mbytes floating point
  - Post CDNN2 (optimized for CEVA-XM4): 16Mbytes fixed-point
  - Including weights and data



CEVA Network Generator

CDNN2 CEVA

Image Database
(For Training)

Caffe
TensorFlow

Pre-Trained Network
(*Deploy.caffemodel*)

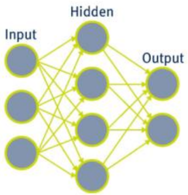
Network Structure
(Deploy.prototxt)

Normalization Image
(*Deploy.binaryproto*)

Network Labeling
(Deploy.labels)

CEVA
Network
Generator

CEVA-XM4
Optimized
Network



Real-Time Network Generator Demo

Live CDNN2 demo:
https://www.youtube.com/watch?v=SXINFryLM3Q&feature=youtu.be


Age and Gender Classification using Convolutional Neural Networks

Gil Levi Tal Hassner

The Open University of Israel

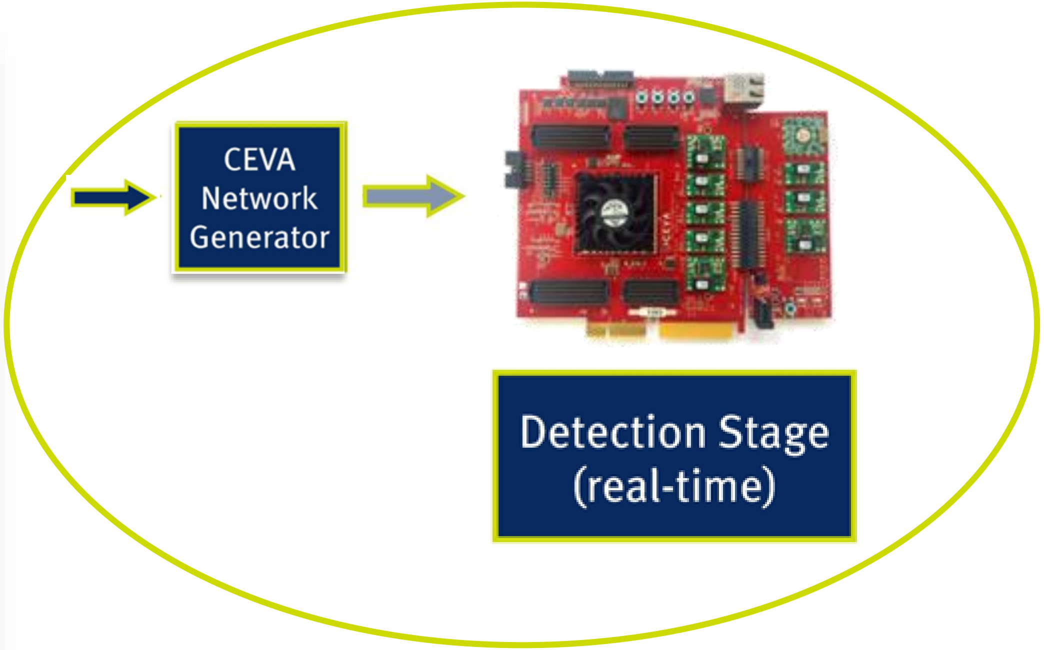
Figure 1. Faces from the Adience benchmark for age and gender classification. These images represent some of the challenges of age and gender estimation from real-world, unconstrained images. Most notably, extreme blur (low-resolution), occlusions, out-of-plane pose variations, expressions and more..

Abstract: Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition. In this paper we show that by learning representations through the use of deep-convolutional neural networks (CNN), a significant increase in performance can be obtained on these tasks. To this end, we propose a simple convolutional net architecture that can be used even when the amount of learning data is limited. We evaluate our method on the recent Adience benchmark for age and gender estimation and show it to dramatically outperform current state-of-the-art methods..

Reference: Gil Levi and Tal Hassner, *Age and Gender Classification using Convolutional Neural Networks*, IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015

Click here for the PDF
Click here for the BibTex

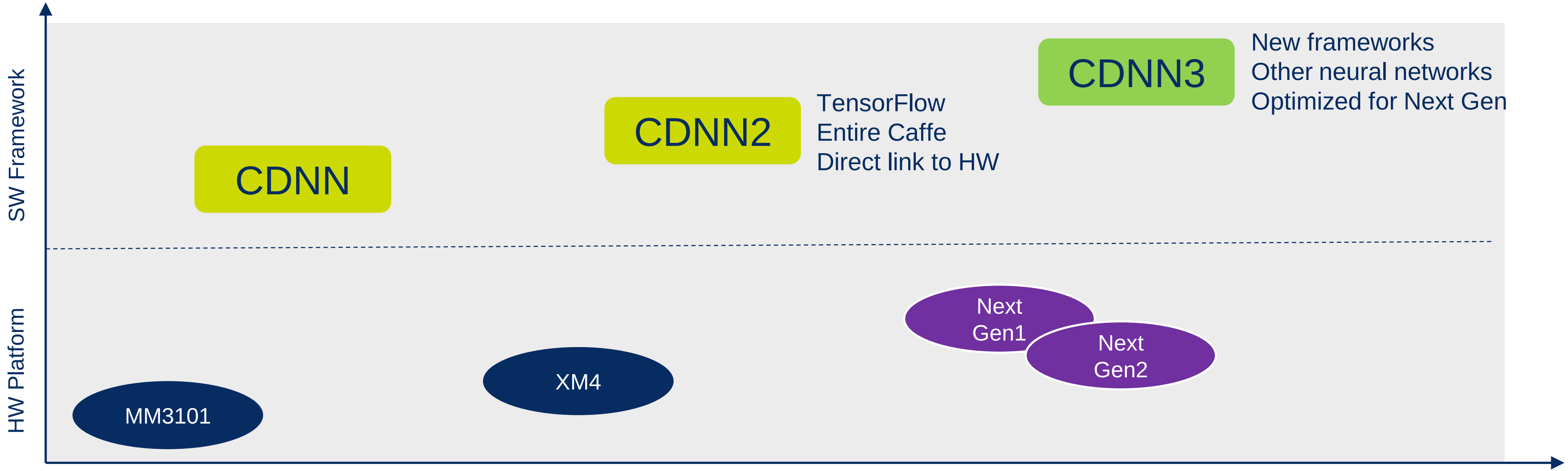
Downloading Age classification
Neural Network from the internet



Passing it via CEVA Network Generator and running it on the XM4 FPGA under 10 min !

Final Comments: HW + SW

- CEVA is at the forefront of development of Neural Network embedded platforms
- Normally, the HW platform is meaningless if not supported by the corresponding SW framework…



CEVA®

The DSP Powerhouse

Thank You

www.ceva-dsp.com