HARMAN
A SAMSUNG COMPANY

REAL TIME OBJECT DETECTION ON LOW POWER EMBEDDED PLATFORMS

GEORGE JOSE, HARMANX INDIA

28 OCT, 2019

1. Objectives

2. Network Design Challenges On Embedded Platforms

3. Object Detection : An Overview

4. Real Time Object Detection Neural Networks : Survey

5. Embedded Platforms : Insights

6. Proposed Network Design and Verification on TDA2PX

7. Results & Conclusion

HARMAN

Design neural network which performs object detection on embedded platforms

with following objectives

High Accuracy | High FPS | Low Latency | Less Power

- Bringing deep learning applications to edge platforms
- Devices constrained by computation speed and memory bandwidth
- Energy efficiency required to function in battery operated devices
- For applications like drone surveillance systems, self driving cars etc

HARMAN

1. Objectives

2. Network Design Challenges On Embedded Platforms

3. Object Detection : An Overview

4. Real Time Object Detection Neural Networks : Survey

5. Embedded Platforms : Insights

6. Proposed Network Design and Verification on TDA2PX

7. Results & Conclusion

Low Power Embedded Platforms : Challenges

HARMAN

Key Design Challenges:

- Constrained by both computation and memory
- Not enough parallelism because of lesser cores
- Some network layers not supported in embedded platforms

Comparison of NVIDIA 1080Ti GPU vs TI TDA2PX EVE Processor

Parameter	NVIDIA 1080Ti	TI TDA2PX EVE	Factor
Clock Frequency	1.50 GHz	0.90 GHz	1.6
FLOPs	11.3 TFLOPs	0.03 TFLOPs	376
Memory Bandwidth	27.30 Tbps	0.38 Tbps	72
# Cores	3500	2	1750
Power	250 W	< 10 W	25

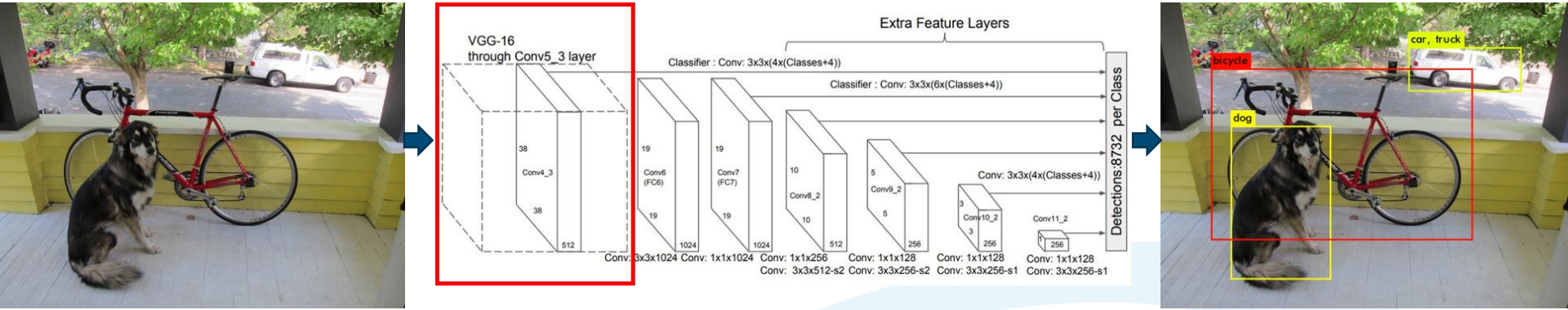
FLOPs – Floating Point Operations/second

1. Objectives
2. Network Design Challenges On Embedded Platforms
3. **Object Detection : An Overview**
4. Real Time Object Detection Neural Networks : Survey
5. Embedded Platforms : Insights
6. Proposed Network Design and Verification on TDA2PX
7. Results & Conclusion

Object Detection : Introduction

Object detection consists of two sub tasks:

- **Predict the bounding box** coordinates for objects present in the image **(Regression)**
- **Identify the class of the object** present in bounding box predicted **(Classification)**



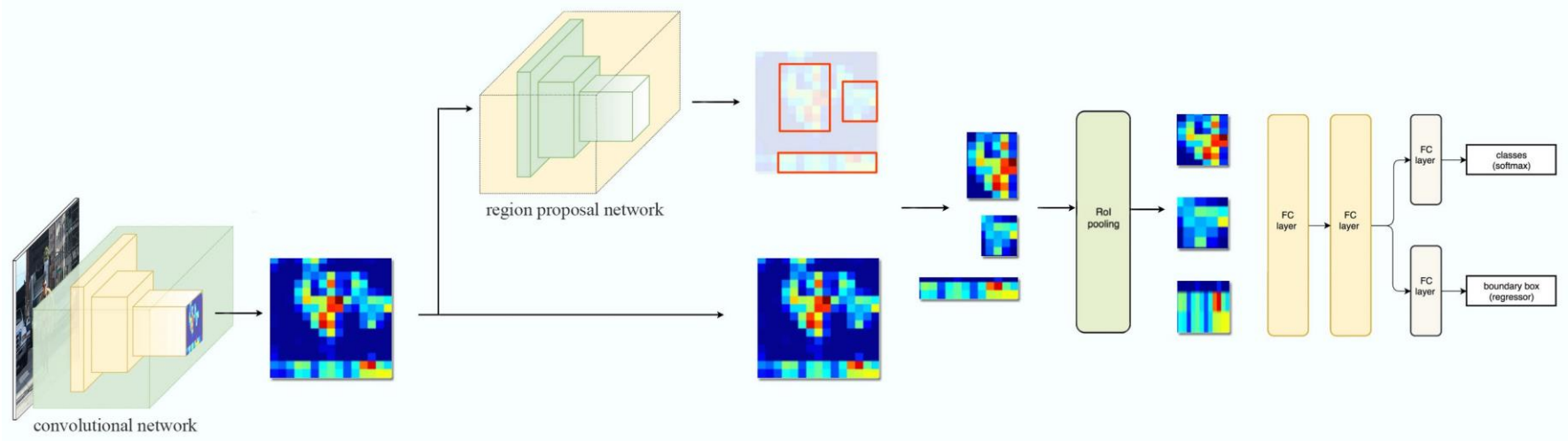
Object Detection network consists of two parts:

- Detection Architecture
- Backbone Feature Extractor

https://arxiv.org/pdf/1512.02325.pdf

Detection Architecture

Two Stage Object Detection Architecture

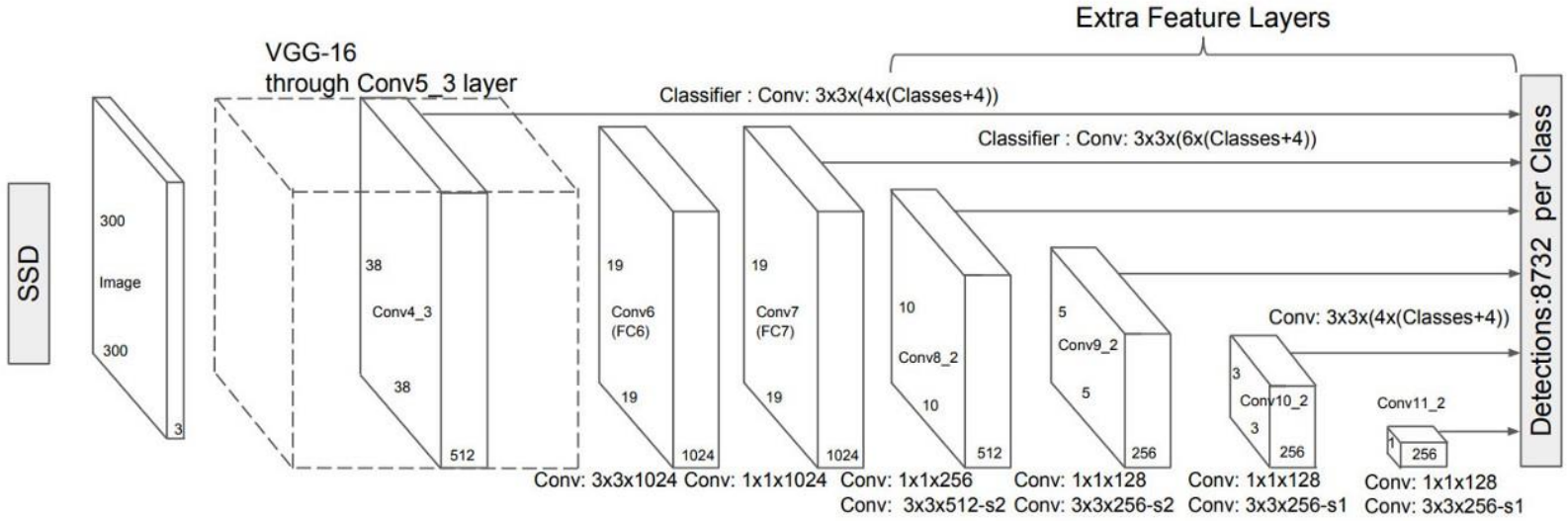


https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9

- Region proposal network predicts possible object regions in image
- Object classification performed on feature map after ROI pooling
- ROI pooling converts varying object feature map to fixed size
- Examples: Faster RCNN, R-FCN

HARMAN

Single Stage Object Detection Architecture

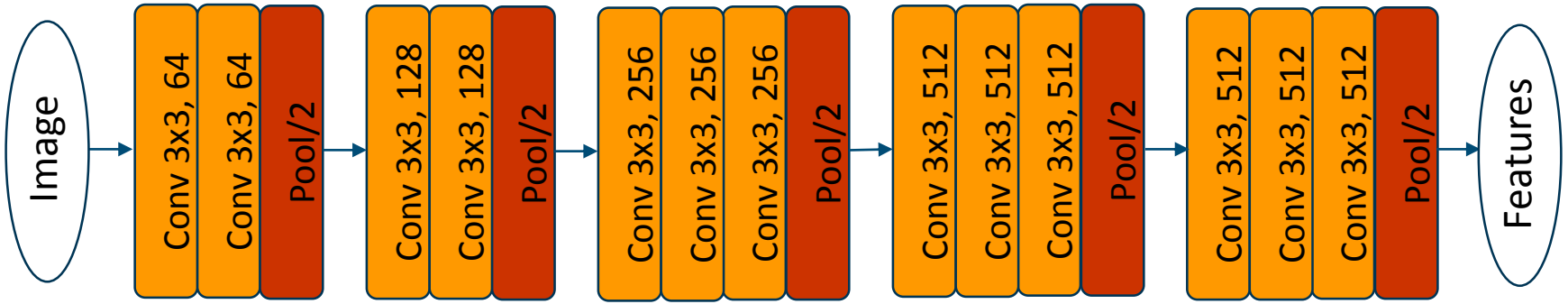


https://arxiv.org/pdf/1512.02325.pdf

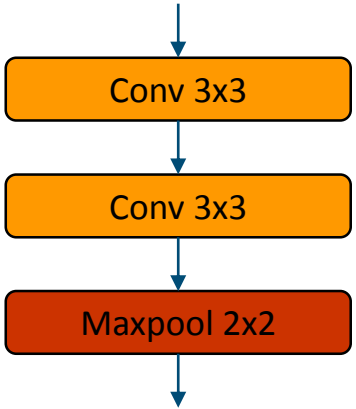
- Directly predicts bounding box coordinates and class confidence scores
- Typically faster than two stage object detection architectures in GPUs
- Examples: SSD, YOLO

Backbone Feature Extractors

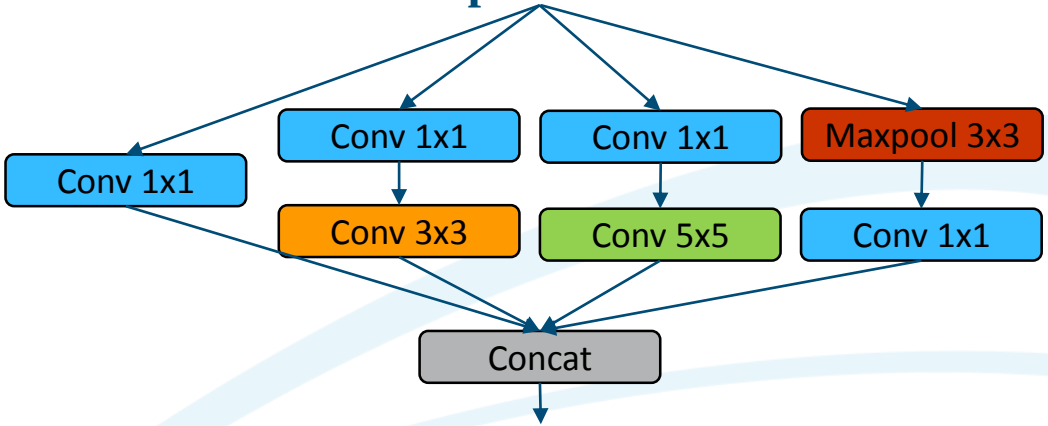
VGG Feature Extractor



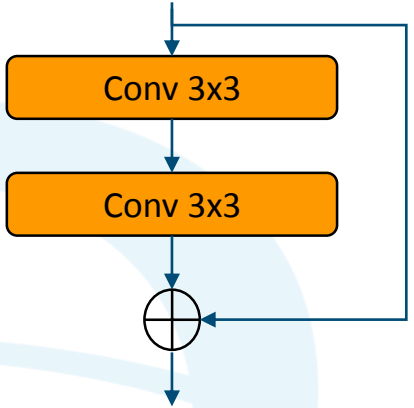
VGG Block



Inception Block

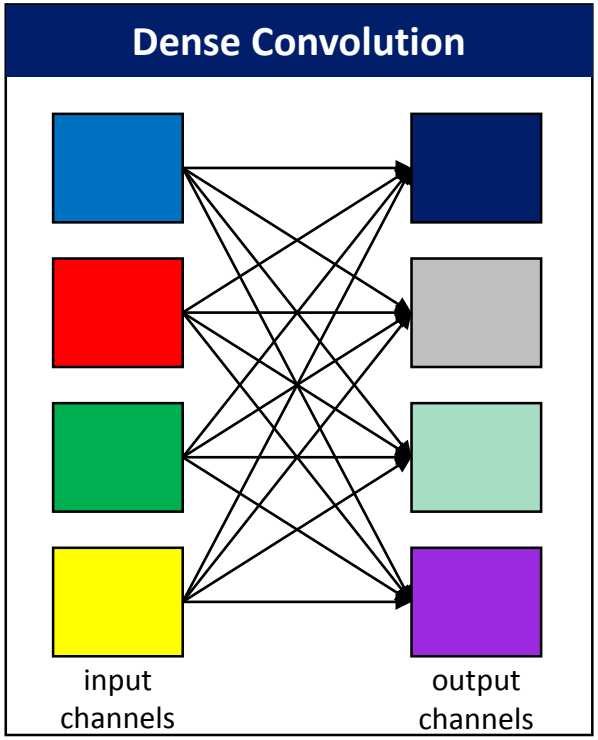


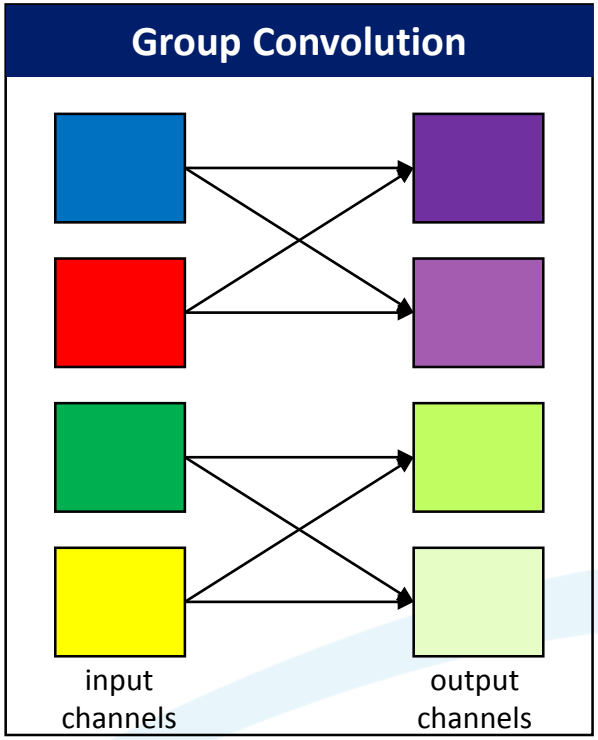
ResNet Block

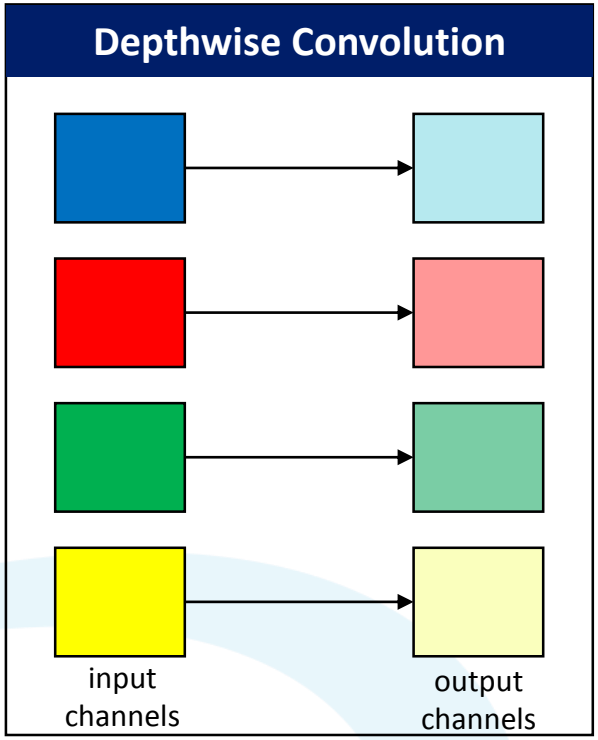


Computationally intensive, hence not suited for real time embedded applications

Dense convolutions : Problems







Reducing Computations

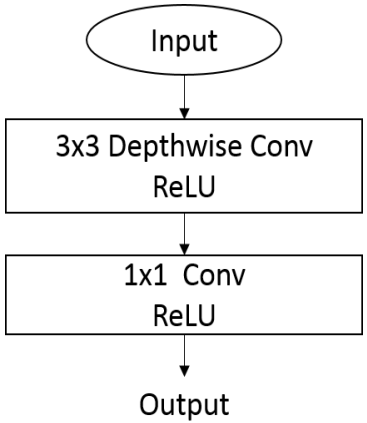
- Use alternatives like group or depthwise convolutions instead of dense convolution
- Squeeze the channels before dense convolution using 1x1 filters (pointwise convolution)

1. Objectives
2. Network Design Challenges On Embedded Platforms
3. Object Detection : An Overview
4. **Real Time Object Detection Neural Networks : Survey**
5. Embedded Platforms : Insights
6. Proposed Network Design and Verification on TDA2PX
7. Results & Conclusion

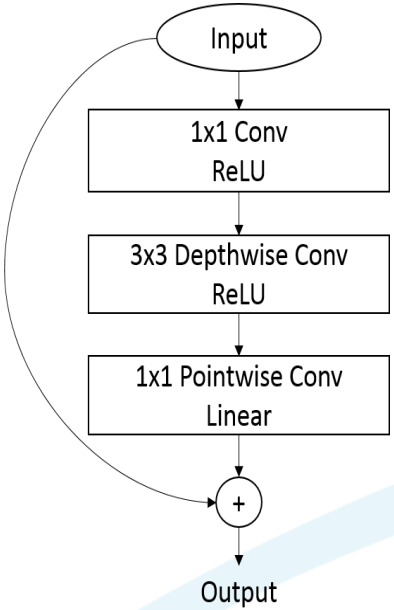
Real Time Object Detection Networks

HARMAN

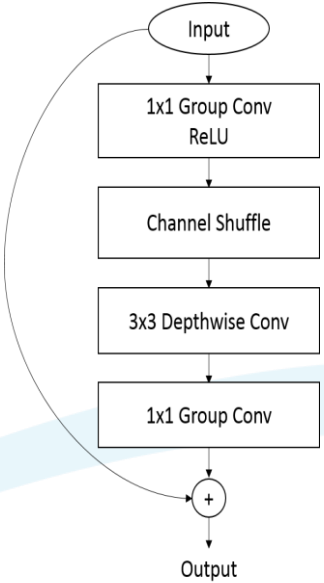
MobileNetv1



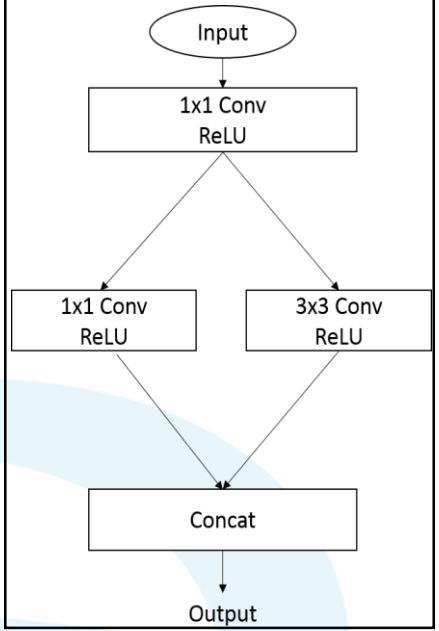
MobileNetv2



ShuffleNet



SqueezeNet



Performance on embedded platform

HARMAN

Benchmarking data by TI of various backbone architectures on a single EVE core at 635MHz for TDA2PXSoC. (D) represents dense model and (S) represents sparse model

Network topology	Image Size	MMACs	Latency
MobileNetv1	224x224	567.70	559.18ms
SqueezeNetv1	227x227	390.80	237.60ms
JacintoNet11 v2 (D)	224x224	405.81	203.23ms
JacintoNet11 v2 (S)	224x224	107.54	103.23ms

Key Observations:

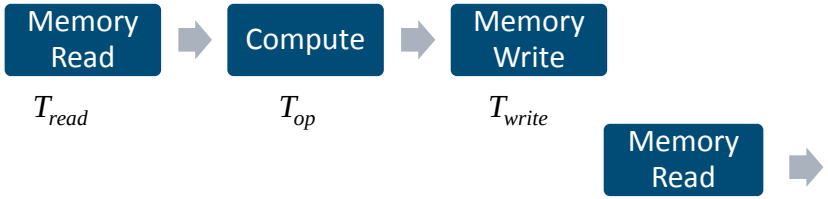
- Lower MMACs doesn't imply faster inference speeds
- Sparse convolutions help achieve around 2x speedup

** Jacintonet11 is a feature extractor developed by TI for faster inference on this platform*

1. Objectives
2. Network Design Challenges On Embedded Platforms
3. Object Detection : An Overview
4. Real Time Object Detection Neural Networks : Survey
5. **Embedded Platforms : Insights**
6. Proposed Network Design and Verification on TDA2PX
7. Results & Conclusion

Process Execution Pipeline

Single Thread Process Execution



Let T_{mem} be the total time required for memory operations:

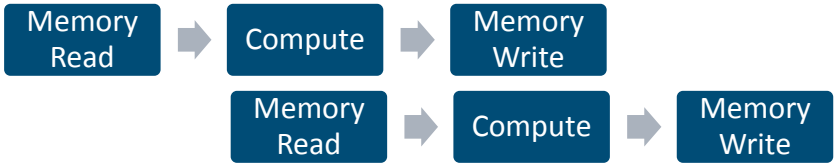
$$T_{mem} = T_{read} + T_{write}$$

Time taken to complete the process :

$$T_{proc} = T_{mem} + T_{op}$$

Inefficient resource utilization

Multi Thread Process Execution



Fetch memory for next cycle, while compute for current cycle is executing (Instruction pipelining)

Effective throughput of the process :

$$1/max\ (T_{op}, T_{mem})$$

$T_{op} = \frac{\#ops}{BW_{math}}$ $T_{mem} = \frac{\#bytes}{BW_{mem}}$ BW_{math} – Processor Bandwidth (FLOPs) BW_{mem} – Memory Bandwidth (Gbps)

Math Limited vs Memory Limited Operation

HARMAN

- *Speed of an algorithm is determined by both computation and memory bandwidth of the hardware*

Math Limited Operation

For an operation to be math limited:

$T_{op} > T_{mem}$ or $\frac{\#ops}{\#bytes} > \frac{BW_{math}}{BW_{mem}}$

More computations performed for each byte accessed

Examples: Dense convolutions, Fully connected layer

Memory Limited Operation

For an operation to be memory limited:

$T_{mem} > T_{op}$ or $\frac{\#ops}{\#bytes} < \frac{BW_{math}}{BW_{mem}}$

Few computations performed for each byte accessed

Examples: Depthwise convolution, Pooling, Element wise operation layers

Energy Efficiency & Latency

SRAM	DRAM
• On-chip cache • Expensive and faster • Limited storage capacity • Consume less power	• Off-chip main memory • Cheaper, but slower • Higher storage capacity • Consumes more power

Cache Switching

Occurs when data not found in SRAM and needs to access DRAM

Associated Problems with DRAM access

- Increased power consumption
- Higher latency

Network Design – key contributors

- Large intermediate feature/activation maps
- Layers with large number of weights/parameters

Network Design Strategy

HARMAN

Design Strategy

1. Understand the hardware to find optimal layers for maximizing speed.
    - Group / pointwise / depthwise convolutions
    - Math limited vs memory limited
2. Striking a balance between spatial resolution and channel depth
    - Smaller feature maps
    - Optimal filter sizes
3. Make use of the hardware optimizations present in the platform
    - SIMD support
    - Quantization
4. Explore different model compression techniques
    - Sparsification
    - Network pruning

1. Objectives
2. Network Design Challenges On Embedded Platforms
3. Object Detection : An Overview
4. Real Time Object Detection Neural Networks : Survey
5. Embedded Platforms : Insights
6. **Proposed Network Design and Verification on TDA2PX**
7. Results & Conclusion

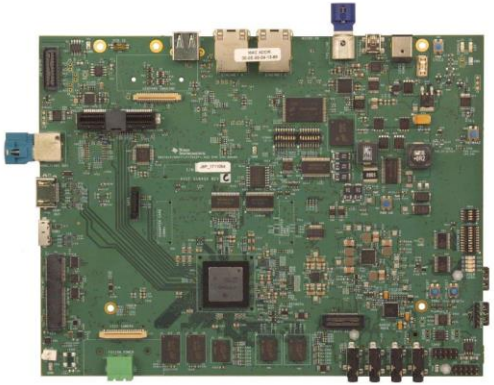
TDA2PX Platform

HARMAN

Developed by Texas Instruments For ADAS Applications

Two EVE Processors

Two A15 ARM Processors



Dual Cortex M4 Processors

Two DSP Processors

Development Flow

Model Training

- Offline float-32 bit training on Tensorflow or Caffe framework

Model Translation

- Converting to a fixed point (supports 4-bit to 12-bit) model supported by TIDL library

Model Inference

- Running the quantized network model on the TDA2PX SoC

Design Strategy For TDA2PX

HARMAN

(I) Finding the optimal layers	
Architecture	**Limitations**
MobileNets	Depthwise convolutions are memory limited Deeper networks leading to high latency
SqueezeNets	Squeeze -> Expansion (1x1 & 3x3 filters) -> Concat Deeper and memory intensive network
ShuffleNets	Shuffle operations are memory intensive Shuffle operations not supported by TIDL library
Solution	• Use group convolutions to reduce computations • Use pointwise convolutions to aggregate channel information instead of channel shuffle

Design Strategy For TDA2PX

HARMAN

(II) Striking balance between spatial width and channel dimension

- Major computational complexity lies at initial part of network
- Large feature maps leads to frequent cache switching

Solution:

- ❖ Balance the computation throughout the network
- ❖ Reduce channels in the initial layers of the network
- ❖ Gradually increase channels after downsampling

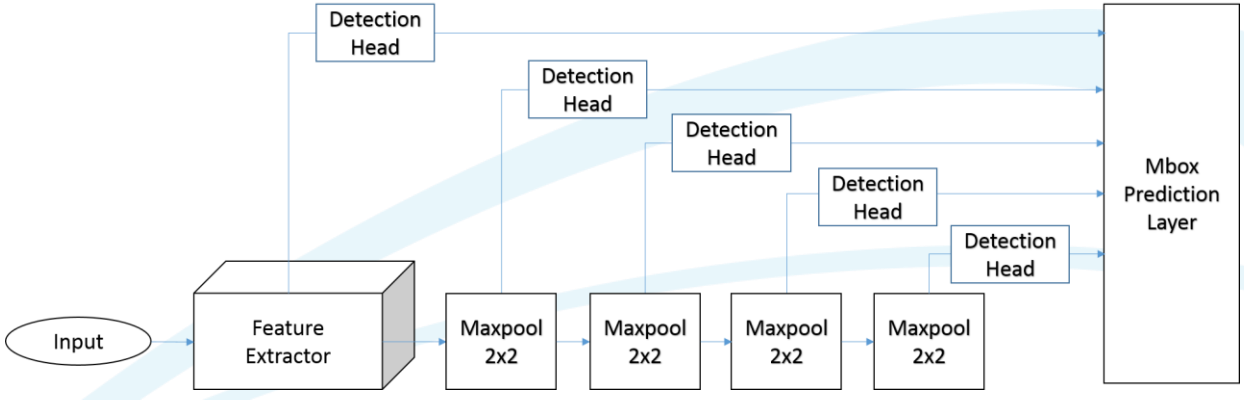
Layer Type	Kernel Size	# O/p Channel	Stride	Groups	MMAC
Conv,ReLU	5	8	2	1	78.64
Conv,ReLU	3	32	2	1	75.50
MaxPool	2	32	2		
Conv,ReLU	3	32	1	2	37.75
Conv,ReLU	3	64	1	4	37.75
MaxPool	2	64	2		
Conv,ReLU	3	64	1	2	37.75
Conv,ReLU	3	64	1	2	37.75
Conv,ReLU	3	128	1	2	75.50
Conv,ReLU	3	128	1	2	75.50
Conv,ReLU	1	256	1	1	67.11
Conv,ReLU	3	128	3	8	75.50

Design Strategy For TDA2PX

HARMAN

(III) Possible hardware Optimizations: Quantization	(IV) Network Compression Techniques: Sparsification
• Converts: 32-bit floating point -> 8-bit fixed point • Reduces energy consumption • Improves inference speed	• Reduces redundant filter coefficients to zero • 3 stage training procedure : i. With l2 regularization ii. With l1 regularization iii. Thresholding and retraining • Improves inference speed

Proposed Detection Architecture



HARMAN

1. Objectives
2. Network Design Challenges On Embedded Platforms
3. Object Detection : An Overview
4. Real Time Object Detection Neural Networks : Survey
5. Embedded Platforms : Insights
6. Proposed Network Design and Verification on TDA2PX
7. **Results & Conclusion**

Results

HARMAN

Evaluation on BDD100k Dataset for 3 classes on 1024x512 images

- Comparison while running on 2 EVEs and single C66x DSP core

Model	FPS	Latency	GMACs	mAP
MobileNetv1-0.5	6.90	0.70s	2.20	60
JDetNet (S)	9.19	0.50s	4.49*	63
HX-LPNet (S)	22.47	0.20s	0.74*	53

- Effect of l1 regularization, sparsification, on model mAPs

Training Stage	mAP	FPS	Latency
Initial with l2 reg	53.79	16.79	0.28s
Effect of l1 reg	54.00	16.90	0.28s
Effect of sparsity (53.55%)	52.51	22.47	0.20s

* Specified GMACs are for dense models

Summary

- Following strategy can be used to design optimal networks on a target hardware:
  1. Understand the hardware, and find the optimal layers/operations
  2. Maintain a balance between spatial resolution and channel depth
  3. Make use of the hardware optimizations present in the platform
  4. Explore different model compression strategies
- Proposed model HX-LPNet performs object detection on TDA2PX with :
  - High FPS Low Latency Low Power

Future Work:

- Explore how this design strategy extends to other embedded platforms
- Neural Architecture Search (NASNets) with hardware in loop

References

- A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko,W.Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.
- F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J.Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv:1602.07360, 2016.
- N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In European Conference on Computer Vision, 2018.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C.Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In IEEE Conference on Computer Vision and Pattern Recognition, 2018.
- M. Mathew, K. Desappan, P. Kumar Swami, and S. Nagori. Sparse, quantized, full frame cnn for low power embedded devices. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.
- B. Wu, F. Iandola, P. H. Jin, and K. Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.
- X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In IEEE Conference on Computer Vision and Pattern Recognition, 2018.


HARMAN
A SAMSUNG COMPANY
THANK YOU