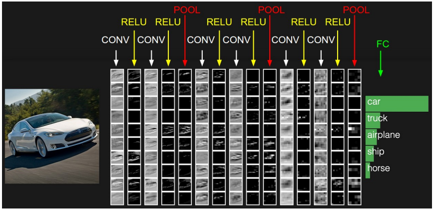
# A Novel Design of Adaptive and Hierarchical Convolutional Neural Networks using Partial Reconfiguration on FPGA

Mohammad Farhadi, Mehdi Ghasemi, Yezhou Yang

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

# Convolutional Neural Networks

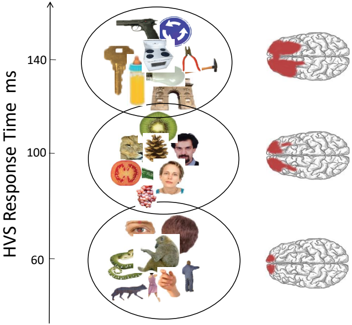
- Convolutional Neural Networks (CNNs)-based methods have become popular in image classification
- Deeper models such as ResNet (152 layers) [He'16] can provide high recognition accuracy
- However, complex models are not suitable for embedded system with confined computing resources
  - Power and performance constraints




http://cs231n.github.io/convolutional-networks

# Adaptive Model

- Human vision system (HVS) has two stages for visual classification [Ritchie'2015]:
  - A shallow primary stage
  - A decision layer to pick a further processing pathway
- A feedback model can be designed to determine the exit from the CNN model



## Adaptive CNNs

- Compressing the structure of CNN models ([Han'2016], [Iandola'2016])
  - Network pruning
- Cascaded classification [Li'2015]
- Branchy-Net: early exit from the model based on the entropy of model output [Teerapittayanon'2016]
- Skip-Net: skip intermediate convolution layers based on the gate decision [Wang'2018]

## CNN acceleration using FPGA

- FINN: A framework for the binarized neural networks (Umuroglu'2017)
- xDNN: Mapping CNNs to Xilinx FPGAs (Sequential) (2018)
- OpenVino: optimize and mapping CNNs to Intel FPGAs (Sequential) (2018)

# Adaptive And Hierarchical CNNs

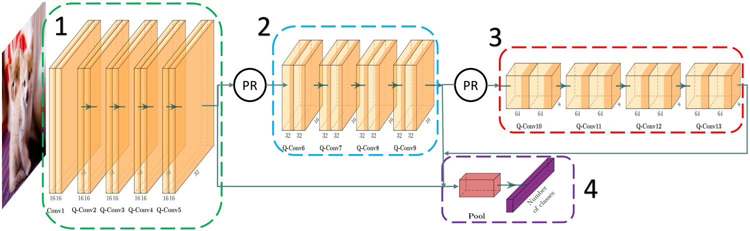
- Start the next convolution as soon as having enough input
  - Connect layers using internal streams
  - Batch processing used for a stream
  - Lower latency
- Quantized CNN model has been used to improve the performance

# Adaptive And Hierarchical CNNs

- Large models cannot fit in the target chip
- Partial reconfiguration utilized to run the whole model
- Shallow part which is a light-weight CNN model
- A decision layer which evaluates shallow part's performance and makes a decision
- A deep part which is a deep CNN with a high inference accuracy

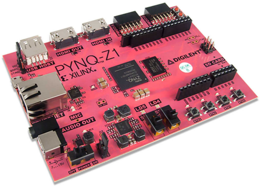
# Implementation on FPGA

- FPGAs are suitable candidates for the low power design of CNN inference
- The adaptive feedback decides to classify the image or apply the next stack of convolutional layers based on the output confidence
- Partial reconfiguration has been used on FPGA to switch between models
  - Dynamic: feature extractor
  - Fixed: data loader, decoder



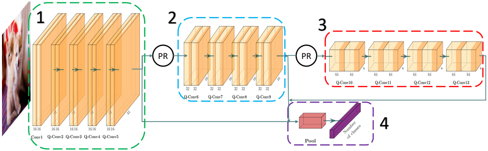
# Device Setup

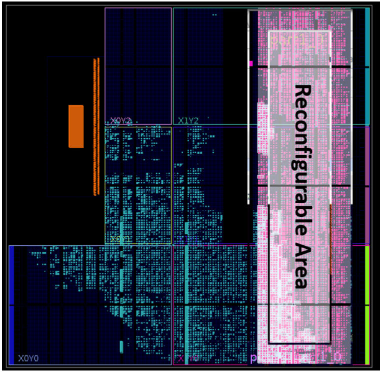
- PYNQ-Z1 board
  - Xilinx Zynq-7000 ZC7020
  - Dual-core ARM A9 processor
- ResNet-18 CNN model
  - Parameters precision: 1-bit weight and 5-bit activation data
- Dataset
  - CIFAR-10
  - CIFA-100
  - SVHN




# Implementation of FPGA

- Layout of the reconfigurable design
- Images are loaded to IP cores using DMA
- Available resources on the Pynq board in comparison with the used resources by the convolution parts




| | Part 1 | Part 2 | Part 3 | Part 4 | Total |
|---|---|---|---|---|---|
| **BRAM** | 81 | 91 | 96 | 31 | 280 |
| **DSP** | 120 | 96 | 96 | 24 | 220 |
| **FF** | 15672 | 16647 | 34069 | 9908 | 106400 |

# Results

- Performance evaluation on the different parts of the design

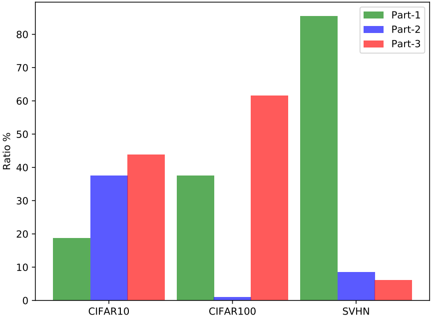
| Bitstream | FPGA Config Time | FPGA Execution Time | CPU Execution Time | FLOPS |
|---|---|---|---|---|
| **Part 1** | 38-42 ms | 2 ms | 98 ms | 10.24M |
| **Part 2** | 38-42 ms | 2 ms | 57 ms | 8.6M |
| **Part 3** | 38-42 ms | 2 ms | 49 ms | 8.5M |

- Top-1 accuracy of the HLS optimized IP cores

| | CIFAR10 | CIFAR100 | | SVHN |
|---|---|---|---|---|
| | | Top1 | Top5 | |
| **Part 1** | 70.95 | 42.26 | 72.14 | 80.35 |
| **Part 2** | 80.57 | 52.23 | 80.25 | 91.24 |
| **Part 3** | 86.27 | 56.60 | 83.46 | 94.62 |

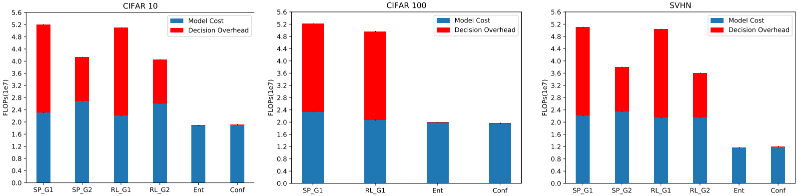
# Results

- The stop ratio for each part of CIFAR-10, CIFAR-100, and SVHN dataset has been shown
- The base model accuracy is preserved
- Switching between IP cores is costly
- By batch processing, the switching cost will be eliminated



# Performance Comparison

- Computation reduction for different methods: Entropy (Ent), Confidence (Conf), SkipNet+SP (SP), SkipNet+HRL+SP (RL)
- Reducing the computation costs by≈30%,≈27%and≈57% on the CIFAR-10, CIFAR-100, and SVHN data using confidence



# Acknowledgement


- The National Science Foundation under the Robust Intelligence Program (1750082), and the IoT Innovation (I-square) fund provided by ASU Fulton Schools of Engineering are gratefully acknowledged.

- We also acknowledge NVIDIA and Xilinx for the donation of GPUs and FPGAs.

# Question