International Workshop on *Intelligent Multi-core Systems*

Amit Kumar Singh

School of Computer Science and Electronic Engineering

University of Essex

United Kingdom

W: http://aksingh.co.uk/ E: a.k.singh@essex.ac.uk

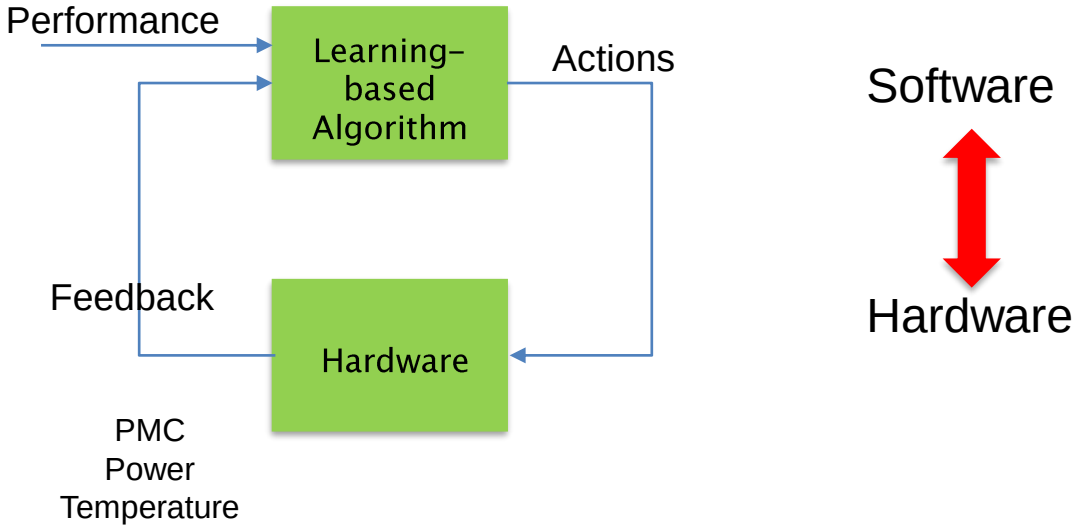
Online ML based Intelligence

Possible Solutions to Address Challenges

- Carefully consider (limited) number of states (and actions) in online learning (RL) process
- Support with offline learning or training
  - Approach that ingests all the data at one time to build a model, e.g. CNN.
- Online training

Carefully consider (limited) number of states (and actions) in online learning (RL) process

RL Model



- States
  - Workload, Power, Temperature, etc. -> very high variation -> many states
- Actions
  - Number of cores needed to process an application
    - cpuquiet API
  - Voltage-frequency of the active cores
    - cpufreq API

What is the problem if there are too many states and actions?

What is the problem if there are very limited number of states?

Reinforcement Learning Q-Table (Exploration - Exploitation)

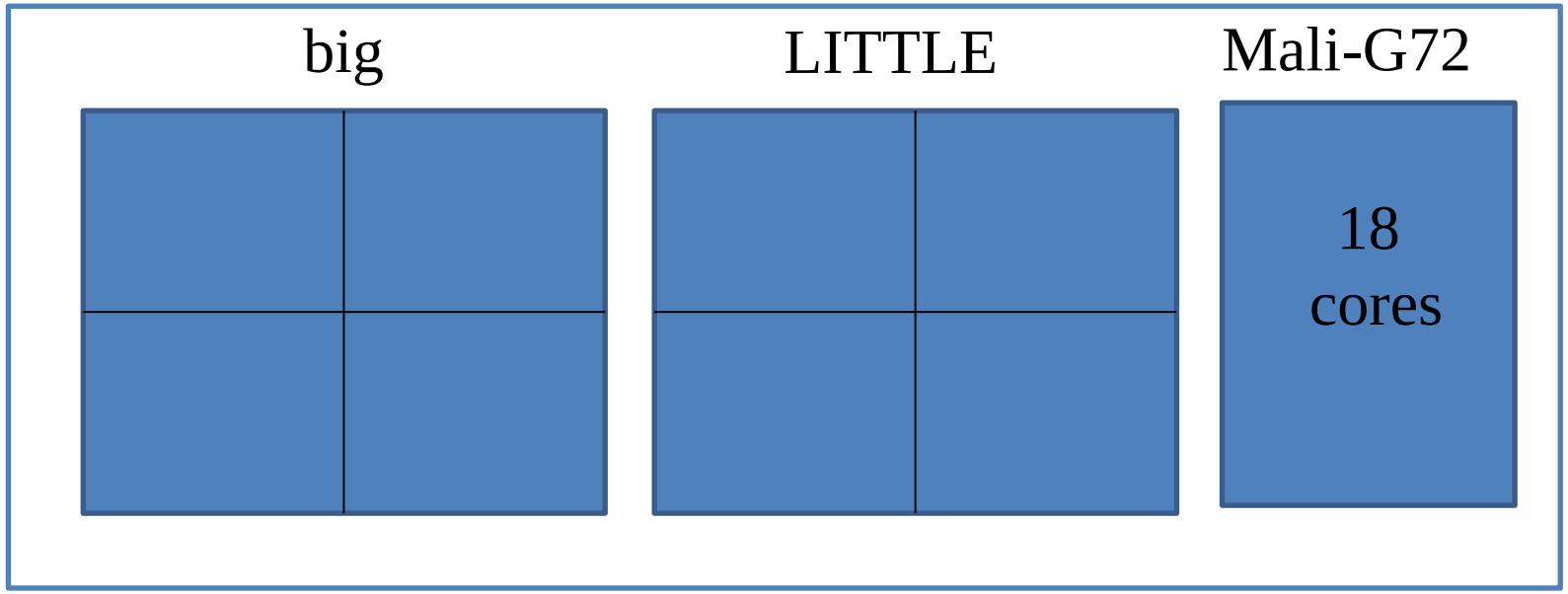
A balanced number of states is needed
-> need to apply right level of Intelligence

	ACTIONS			
STATES	P0	P1	P2	P3
WD0	0	0	0	0
WD1	0	0	0	0
WD2	0	0	0	0
WD3	0	0	0	0
WD4	0	0	0	0
WD5	0	0	0	0
WD6	0	0	0	0

	ACTIONS (Power Modes)			
STATES	P0	P1	P2	P3
WD0	219	224	230	235
WD1	222	230	238	246
WD2	224	235	245	125
WD3	204	220	236	252
WD4	210	232	253	106
WD5	210	232	127	105
WD6	195	225	127	97

States and Actions for Modern Multi-core Chips

- Exynos 9810 MPSoC – Used in Samsung Galaxy Note 9



- big: 18 frequency levels
- LITTLE: 10 frequency levels
- GPU: 6 frequency levels

Too many actions!

Will you consider all the actions/states and why?

Balancing between States and Actions for Exynos 9810 MPSoC

To apply appropriate level of Intelligence

States
- big_CPUfreq
- LITTLE_CPUfreq
- GPUfreq
- FPScurrent
- TargetFPS
- Power_current
- Temperature_big
- Temperature_device

Actions
- big frequency up
- big frequency down
- do not change big frequency
- LITTLE frequency up
- LITTLE frequency down
- do not change LITTLE frequency
- GPU frequency up
- GPU frequency down
- do not change GPU frequency

User Interaction Aware Reinforcement Learning for Power and Thermal Efficiency of CPU-GPU Mobile MPSoCs

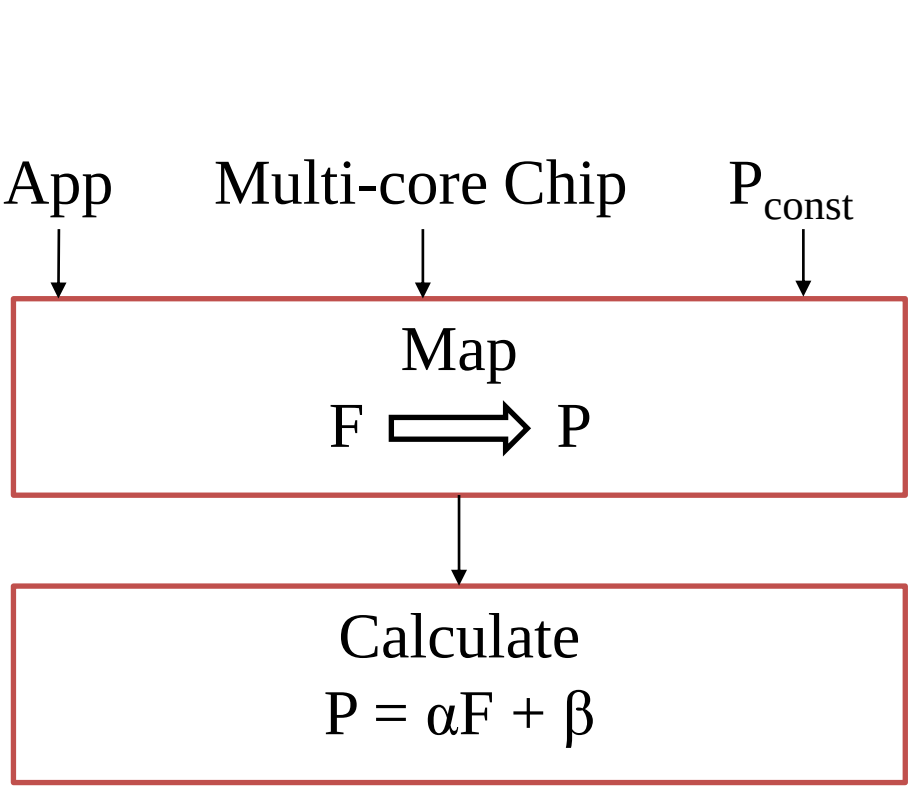
Somdip Dey, Amit Kumar Singh, Xiaohang Wang and Klaus McDonald-Maier

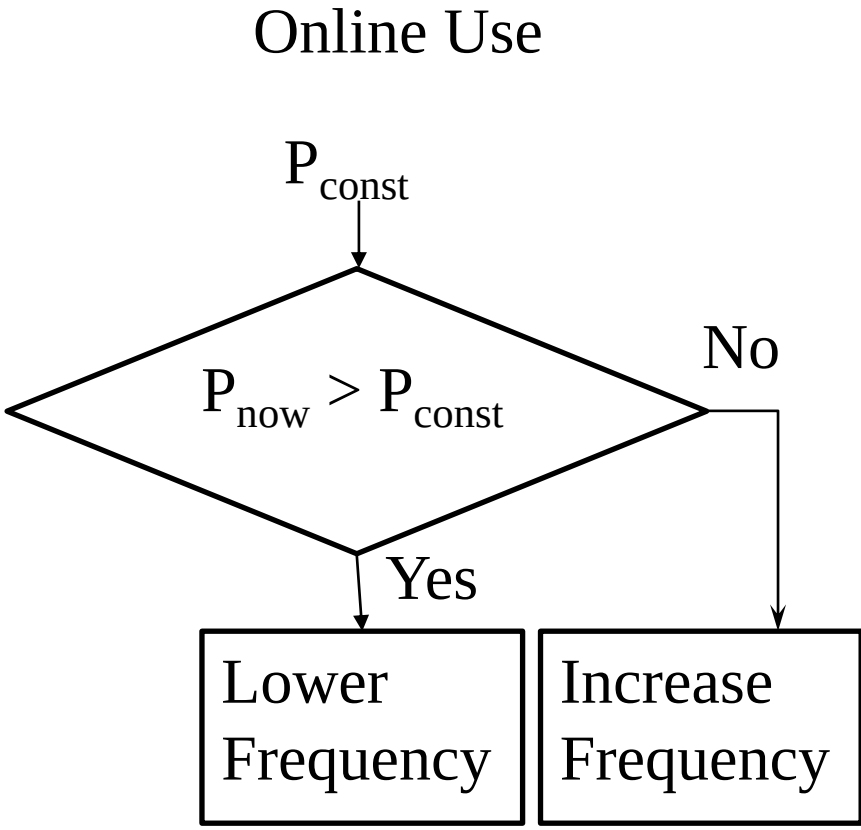
IEEE Design, Automation & Test in Europe (DATE), Grenoble, France, March 2020.

Support with offline learning or training

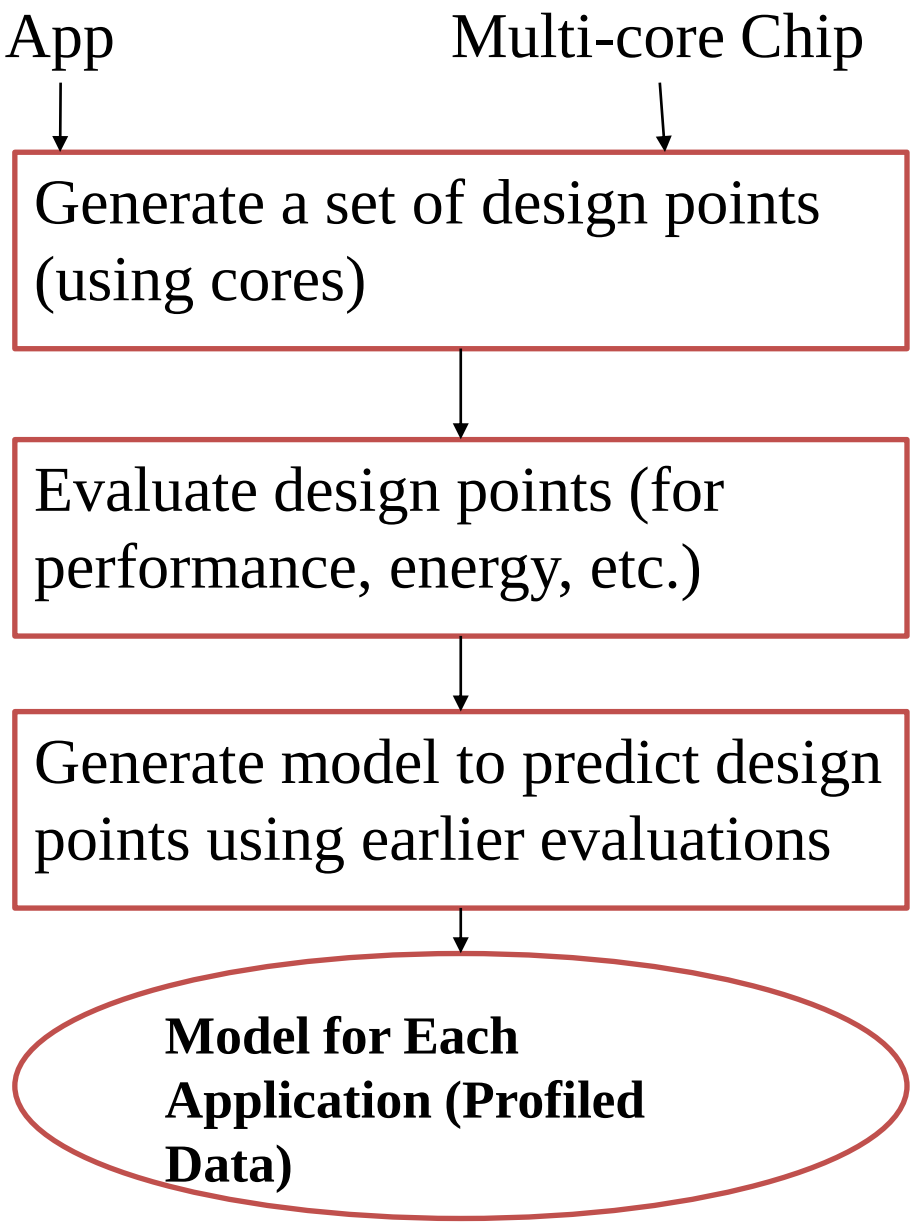
- **Regression**
- **CNN**

A Typical Regression





Regression for Design Points



What are problems with Regression based learning?

Using CNN
for program intensity classification

SoCodeCNN: Classification of Program Source Code Using CNN Based Computer Vision Methodology

Somdip Dey, Amit Kumar Singh, Dilip Kumar Prasad, Klaus McDonald-Maier

IEEE Access, 2019.

Most Popular Article

Motivation

Learning through visual perception

Ball

Dog

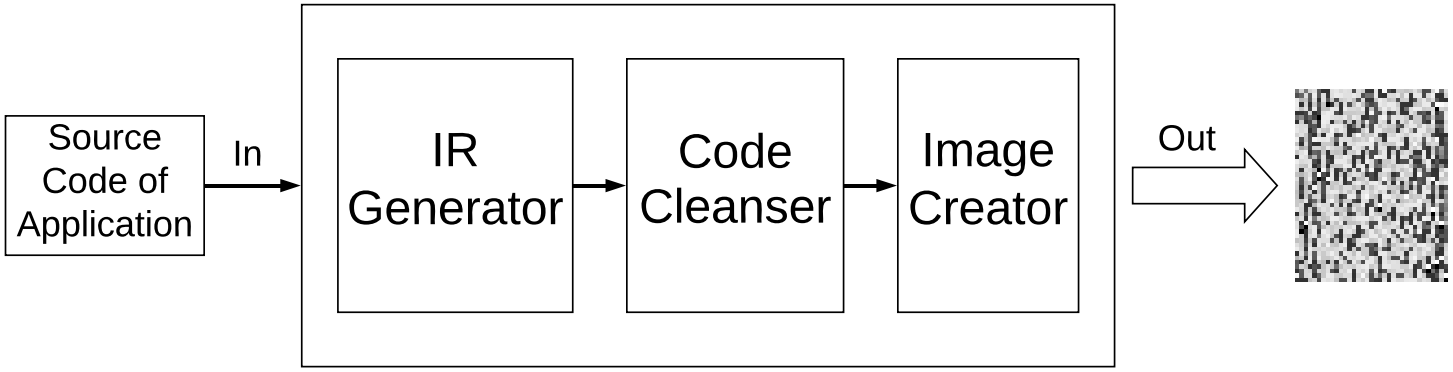
Cat

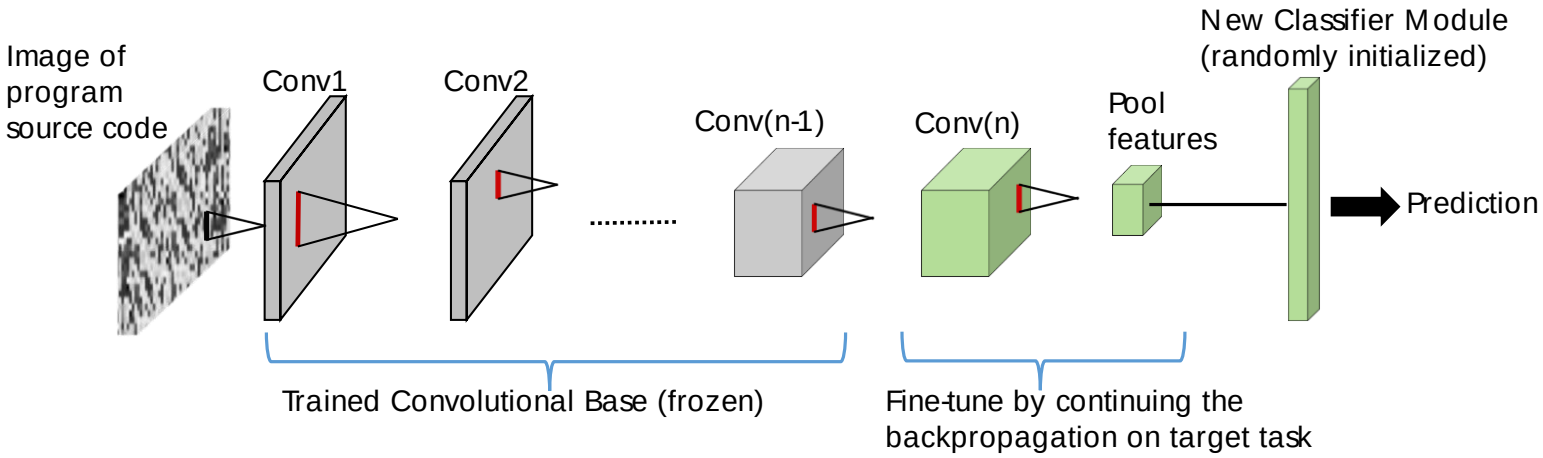
Car

References:

- D. Marr, “Vision: A computational investigation into the human representation and processing of visual information. mit press,” Cambridge, Massachusetts, 1982.
- P. Messaris, Visual" literacy": Image, mind, and reality. Westview Press, 1994.

SoCodeCNN Methodology





Pretrained VGG16 with custom classifier module (compute, memory and mixed)

Step 1: IR Generator

```
; ModuleID = 'hello.c'
source_filename = "hello.c"
target datalayout = "e-m:o-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-apple-macosx10.13.0"

@.str = private unnamed_addr constant [14 x i8] c"Hello, World!\00", align 1

; Function Attrs: noinline nounwind optnone ssp uwtable
define i32 @main() #0 {
  %1 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([14 x i8], [14 x i8]* @.str, i32 0, i32 0))
  ret i32 0
}

declare i32 @printf(i8*, ...) #1

attributes #0 = { noinline nounwind optnone ssp uwtable "correctly-rounded-divide-sqrt-fp-math"="false" "disable-t
"unsafe-fp-math"="false" "use-soft-float"="false" }
attributes #1 = { "correctly-rounded-divide-sqrt-fp-math"="false" "disable-tail-calls"="false" "less-precise-fpmad

!llvm.module.flags = !{!0, !1}
!llvm.ident = !{!2}

!0 = !{i32 1, !"wchar_size", i32 4}
!1 = !{i32 7, !"PIC Level", i32 2}
!2 = !{!"clang version 6.0.1 (tags/RELEASE_601/final)"}
```

LLVM IR code of ”Hello, World!” program

Step 2: Code Cleanser

```
@.str = private unnamed_addr constant [14 x i8] c"Hello, World!\00", align 1
define i32 @main() local_unnamed_addr #0 {
  %1 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([14 x i8], [14 x i8]* @.str, i32 0, i32 0))
  ret i32 0
}
declare i32 @printf(i8* nocapture readonly, ...) local_unnamed_addr #1
attributes #0 = { noinline nounwind optnone ssp uwtable "correctly-rounded-divide-sqrt-fp-math"="false" "disable-t
"unsafe-fp-math"="false" "use-soft-float"="false" }
attributes #1 = { nounwind "correctly-rounded-divide-sqrt-fp-math"="false" "disable-tail-calls"="false" "less-prec
!llvm.module.flags = !{!0, !1}
!llvm.ident = !{!2}
!0 = !{i32 1, !"wchar_size", i32 4}
!1 = !{i32 7, !"PIC Level", i32 2}
!2 = !{!"clang version 6.0.1 (tags/RELEASE_601/final)"}
```

Getting rid of non-relevant information from IR code such as line breaks, system information, etc.

Step 3: Image Creator

- Find the total size (totalSize) of the IR code.
- Using the equation *totalSize = height x width*, height and width is computed such that | height - width | is the least.
- Create an empty image matrix consisting of the computed height and width.
- Take each character (ASCII) of IR code and populate the corresponding cell in the image matrix.

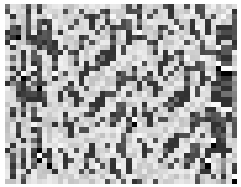
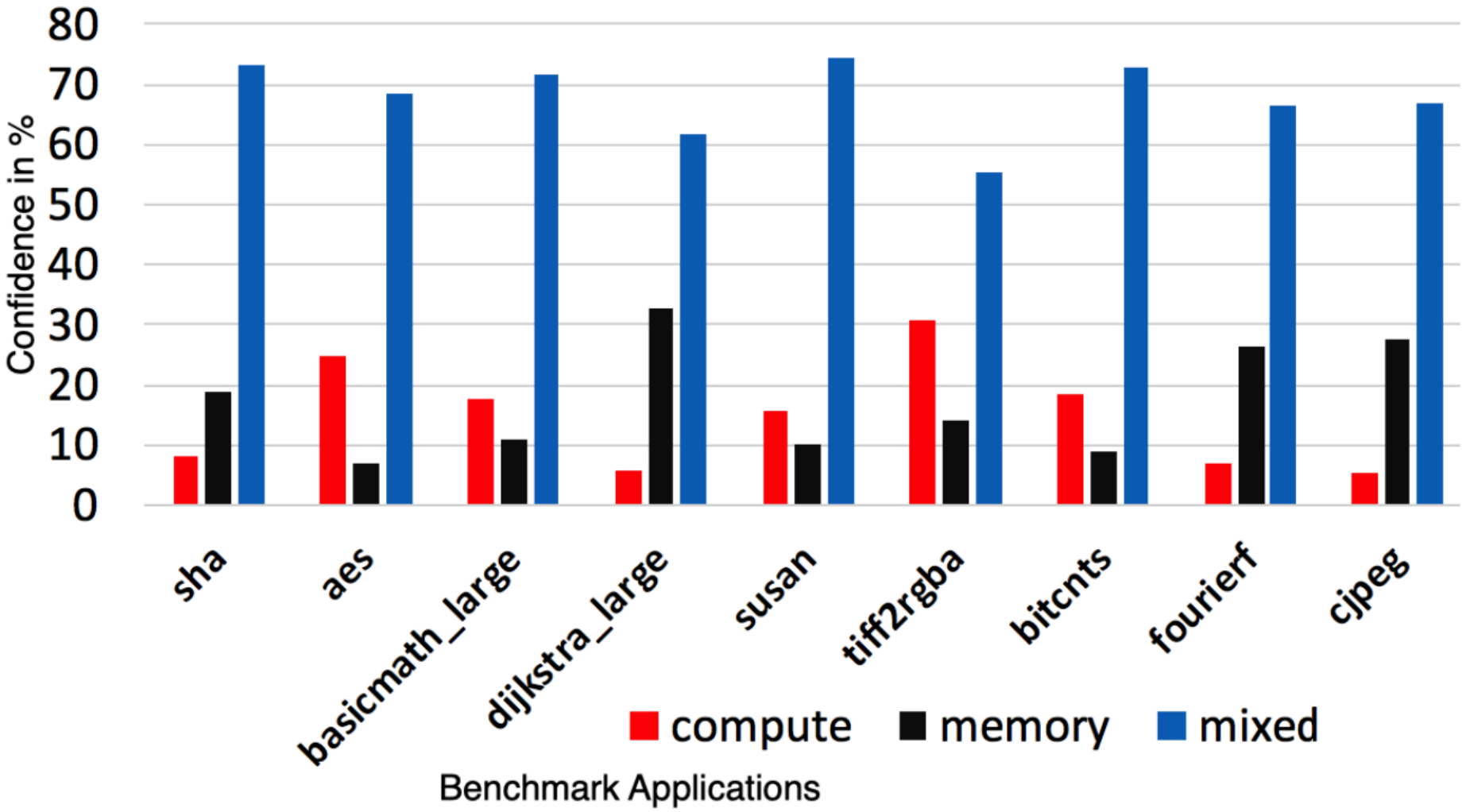


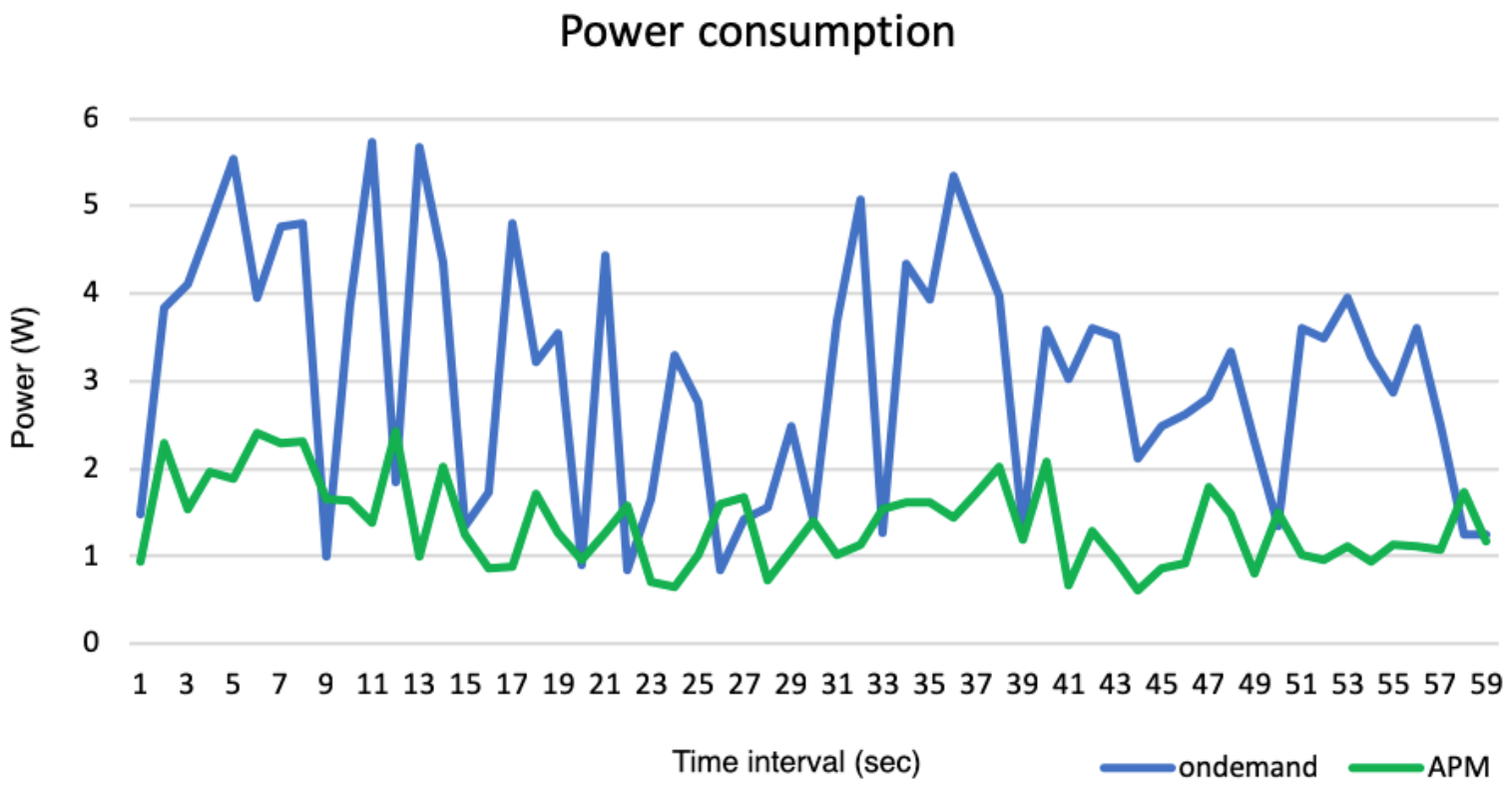
Image of “Hello, World!” program source code

Prediction Confidence



Classification Used for Mapping

A custom program on Samsung Exynos 5422 MPSoC



APM: Power manager using SoCodeCNN
Ondemand: Default governor of Linux

SoCodeCNN

- A new methodology and hence requires more researchers to collaborate and explore the possibilities.
- Initial code availability:

https://github.com/somdipdey/SoCodeCNN

Additional Usage of CNN

- **Temporal Motionless Analysis of Video using CNN in MPSoCs**
  Somdip Dey, Amit Kumar Singh, Dilip Kumar Prasad, Klaus McDonald-Maier
  IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), Manchester, UK, July 2020.
- **ThermalAttackNet: Are CNNs Making It Easy To Perform Temperature Side-Channel Attack In Mobile Edge Devices?**
  Somdip Dey, Amit Kumar Singh, Klaus McDonald-Maier
  MDPI Future Internet, 2021.
  Most Viewed (#1) Article
- **I2UTS: An IoT Based Intelligent Urban Traffic System**
  Vejey Pradeep Suresh Achari, Zeba Khanam, Amit Kumar Singh, Anish Jindal, Alok Prakash and Neeraj Kumar
  IEEE International Conference on High Performance Switching and Routing (HPSR), Paris, France, June 2021.
- **…**

Envisioned Future

- Online training
  - Continual or incremental training
- Hardware-Software Codesign for Performance and Cost balance
- Reliable and Secure Learning

Summarising

- Online ML based Intelligence
  - Carefully considering (limited) number of states (and actions) in online learning (RL) process
  - Regression based learning
  - Usage of CNN
- Envisioned Future

Further Questions?