2021「中技社科技獎學金」
2021 CTCI Foundation Science and Technology Scholarship
境外生研究獎學金
Research Scholarship for International Graduate Students

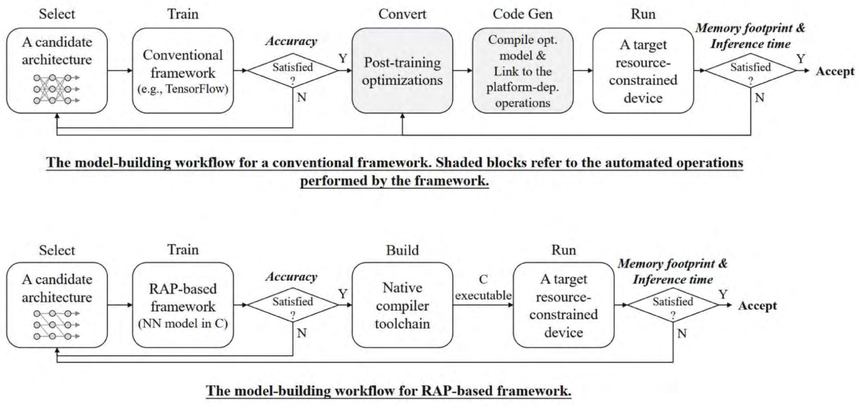
# RAP: A Software Framework of Developing Convolutional Neural Networks for Resource-Constrained Devices Using Environmental Monitoring as a Case Study

4th PhD QiHui Sun, Chia-Heng Tu
Department of Computer Science and Information Engineering
National Cheng Kung University

## Abstract

Monitoring environmental conditions is an important application of cyber-physical systems. Typically, the monitoring is to perceive surrounding environments with battery-powered, tiny devices deployed in the field. While deep learning-based methods, especially the convolutional neural networks (CNNs), are promising approaches to enriching the functionalities offered by the tiny devices, they demand more computation and memory resources, which makes these methods difficult to be adopted on such devices. In this article, we develop a software framework, RAP, that permits the construction of the CNN designs by aggregating the existing, lightweight CNN layers, which are able to fit in the limited memory (e.g., several KBs of SRAM) on the resource-constrained devices satisfying application-specific timing constrains. With the vigorous development of lightweight CNNs, such as binarized neural networks with binary weights and activations, RAP facilitates the model building process for the resource-constrained devices by allowing them to alter, debug, and evaluate the CNN designs over the C/C++ implementation of the lightweight CNN layers.

## Research Focus




The model-building workflow for a conventional framework. Shaded blocks refer to the automated operations performed by the framework.

The model-building workflow for RAP-based framework.

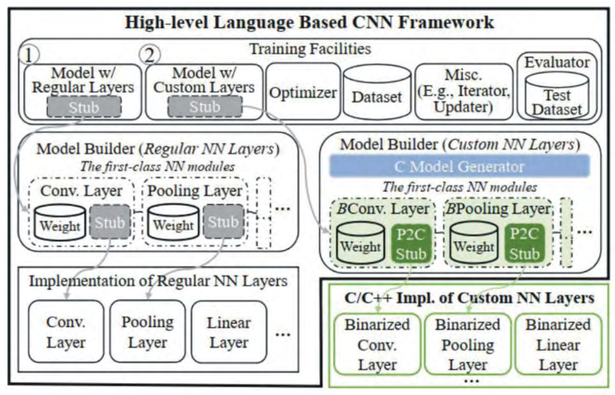



Fig. 2. Software architecture of RAP, where the left half illustrates the organization of the software modules in the CNN framework to represent the regular CNN model (1) and the right half depicts the organization for the custom CNN model (2) .

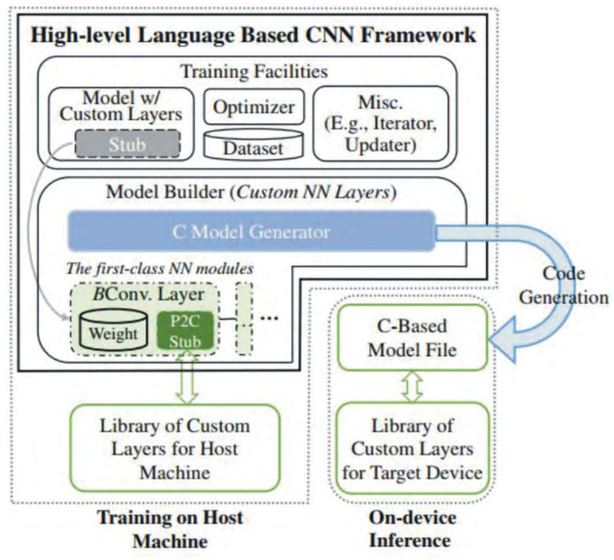



Fig. 3. CNN model training and inference with RAP.

TABLE 5. The experiment result of poaching detection.

| Classifier Type | 2-class FP | 2-class B | |
|---|---|---|---|
| HW Platform | Ameba | Ameba | TI |
| SW Framework | TF Lite | RAP w/ eBNN | RAP w/ eBNN |
| Input Dataset | Cars/Cats | Cars/Cats | Cars/Cats |
| Preprocessing[a] | GS and ED | GS and ED | GS, ED, and F2I |
| Input Channel | 1 | 1 | 1 |
| Input Size | 64 * 64 | 64 * 64 | 50 * 50 |
| Input Type | u_int8 | float | uint8 |
| Network Architecture | conv+fc | conv_b + fc_b | conv_b + fc_b |
| # of Filters | 16 | 16 | 16 |
| Conv. Stride | 3 | 3 | 3 |
| Inference Time (s) / pic | 0.16 | 0.07 | 0.64 |
| Accuracy (%) | 91 | 89 | 87 |
| Memory Footprint (KB) | 76.1 | 17.11 | 3.24 |
| Input Size (KB) | 4 | 16 | 2.44 |
| Parameter Size (KB) | 8.52 | 0.63 | 0.52 |
| Buffer Size (KB) | 63.58 | 0.48 | 0.28 |
| Code Size (KB) | 74.91 | 23.89 | 8.66 |

[a] GS refers to converting the original, three-channel image data into the one-channel version; ED means the input image preprocessed by canny edge detection algorithm; F2I converts the 4-byte floating-point image data into the 1-byte integers.

TABLE 7. The experiment result of cicada detection using RAP.

| Classifier Type | 2-class B | | | |
|---|---|---|---|---|
| HW Platform | TI | TI | TI | TI |
| SW Framework | RAP w/ eBNN | RAP w/ eBNN | RAP w/ eBNN | RAP w/ eBNN |
| Input Dataset | Cicada/Raining | Cicada/Raining | Cicada/Raining | Cicada/Raining |
| Preprocessing[a] | GS | GS | GS, ED and F2I | GS, ED and F2I |
| Input Channel | 1 | 1 | 1 | 1 |
| Input Size | 64 * 64 | 50 * 50 | 64 * 64 | 50 * 50 |
| Input Type[b] | float | float | uint8 | uint8 |
| Network Architecture | conv_b + fc_b | conv_b + fc_b | conv_b + fc_b | conv_b + fc_b |
| # of Filters | 16 | 16 | 16 | 16 |
| Conv. Stride | 3 | 3 | 3 | 3 |
| Inference Time (s) / pic | N/A | N/A | 1.03 | 0.64 |
| Accuracy (%) | 97 | 96 | 95 | 93 |
| Memory Footprint (KB) | 17.11 | 10.56 | 5.11 | 3.24 |
| Input Size (KB) | 16 | 9.76 | 4 | 2.44 |
| Parameter Size (KB) | 0.63 | 0.52 | 0.63 | 0.52 |
| Buffer Size (KB) | 0.48 | 0.28 | 0.48 | 0.28 |
| Code Size (KB) | 10.4 | 10.4 | 8.66 | 8.66 |

[a] GS refers to converting the original, three-channel image data into the one-channel version; ED means the input image preprocessed by canny edge detection algorithm; F2I converts the 4-byte floating-point image data into the 1-byte integers.
[b] By default, the eBNN convolution layer (*conv_b*) manipulates the floating-point numbers. To further cut down the memory footprint, we have implemented the integer version of the convolution layer to accept the input data stored as integers.

## Summary

The major advantage of RAP is that it permits the direct control of CNN layer designs (e.g., eBNN design that cuts down the memory footprint and reduces the computation complexity) to facilitate the development of CNNs on the devices. Besides, RAP helps expand the neural architecture search space, which is limited by the code generation tools of the trained model (e.g., uTensor and TensorFlow Lite). We show that RAP is capable of mounting the regular and lightweight layers implemented in C/C++ to build CNN models.

## Results and Discussion

We have prototyped the RAP framework and built two environmental monitoring applications for protecting endangered species using image- and acoustic-based monitoring methods. Our results show that the built model consumes less than 0.5 KB of SRAM for buffering the runtime data required by the model inference while achieving up to 93% of accuracy for the acoustic monitoring with less than one second of inference time on the TI 16-bit microcontroller platform.

## Publications

1. Cheng, Mu-Hsuan, Qihui Sun, and Chia-Heng Tu. "An adaptive computation framework of distributed deep learning models for internet-of-things applications." 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). IEEE, 2018.

2. Tu, Chia-Heng, QiHui Sun, and Mu-Hsuan Cheng. "On designing the adaptive computation framework of distributed deep learning models for Internet-of-Things applications." The Journal of Supercomputing (2021): 1-33.

3.Tu, Chia-Heng, Qihui Sun, and Hsiao-Hsuan Chang. "RAP: A Software Framework of Developing Convolutional Neural Networks for Resource-constrained Devices Using Environmental Monitoring as a Case Study." ACM Transactions on Cyber-Physical Systems (TCPS) 5.4 (2021): 1-28.