Demo: CNN Performance Prediction on a CPU-based Edge Platform

D. Velasco-Montero
Inst. Microelectrónica de Sevilla (IMSE-CNM), CSIC-Univ. Sevilla (Spain)
delia@imse-cnm.csic.es

J. Fernández-Berni
Inst. Microelectrónica de Sevilla (IMSE-CNM), CSIC-Univ. Sevilla (Spain)
berni@imse-cnm.csic.es

R. Carmona-Galán
Inst. Microelectrónica de Sevilla (IMSE-CNM), CSIC-Univ. Sevilla (Spain)
rcarmona@imse-cnm.csic.es

Á. Rodríguez-Vázquez
Inst. Microelectrónica de Sevilla (IMSE-CNM), CSIC-Univ. Sevilla (Spain)
angel@imse-cnm.csic.es

ABSTRACT

The implementation of algorithms based on Deep Learning at edge visual systems is currently a challenge. In addition to accuracy, the network architecture also has an impact on inference performance in terms of throughput and power consumption. This demo showcases per-layer inference performance of various convolutional neural networks running at a low-cost edge platform. Furthermore, an empirical model is applied to predict processing time and power consumption prior to actually running the networks. A comparison between the prediction from our model and the actual inference performance is displayed in real time.

CCS CONCEPTS

• Computing methodologies • Artificial intelligence • Computer vision • Computer vision tasks • Scene understanding

KEYWORDS

Embedded vision system, visual inference, deep neural networks, CPU-based hardware, inference performance

1. INTRODUCTION

The adoption of the Deep Learning (DL) paradigm [1] in the field of computer vision is becoming a focus of interest in both industry and academia. The main advantages of DL are boosted accuracy – even outperforming humans – and unification of previous diverse approaches for vision tasks such as image recognition or object localization. This interest has given rise to a number of technological advances, from new Convolutional Neural Networks (CNN) to a variety of hardware accelerators and software tools for visual inference. However, the computational and memory requirements of CNNs hinder their implementation in resource-constrained embedded systems operating at the edge, such as smart cameras. Thus, the challenge is how to efficiently leverage and integrate this variety of components in practical realizations, taking also into account that CNN models keep evolving at a rapid pace.

With this scenario in mind, we have been working on a simplified procedure to predict the performance of CNNs running on embedded platforms. By means of performance measurements on generic CNN layers, our model is able to predict the throughput and energy consumption of any other CNN running on the same platform. The objective is to facilitate the evaluation of CNN models prior to actually implementing them, thereby speeding up the deployment of optimal solutions.

In the proposed demonstrator, we will show an accurate per-layer prediction of execution time and power consumption of up to four state-of-the-art CNNs for 1000-category image recognition – Network in Network [2], RestNet-18 [3], SqueezeNet [4] and MobileNet [5]. Our smart-camera framework consists of a low-cost CPU-based platform, namely Raspberry Pi (RPi) 3 model B [6], making use of Caffe [7] open-source tool. We provide a comparison between predictions and measurements of performance metrics for the aforementioned CNNs running on this system.

2. VISITOR EXPERIENCE

The experimental set-up is depicted in Fig. 1. The embedded platform has Ethernet connectivity with a companion host computer intended to showcase the demonstrator. Visitors will be able to select on a graphical interface both the particular CNN architecture and the input – image – to be processed. Once the network and input are specified, per-layer processing time and power consumption predictions are depicted on the screen. Then, by clicking on a button, the embedded platform will process the selected input and measure the time required for inference. A live comparison between predictions and real-time measurements is depicted on the graphical interface. Corresponding top-3 category labels assigned to the input are also displayed on the screen. By changing the particular CNN model on the same input, the output categories may vary according to the network accuracy. In addition, a continuous frame stream on surrounding objects can be classified in real-time while assessing the inference performance predictions.

3. CONCLUSIONS

Application requirements such as accuracy, throughput or energy budget are crucial for smart cameras operating at the edge. We

ICDSC '19, September 9-11, 2019, Trento, Italy
ACM XXX.
http://dx.doi.org/XXX

demonstrate how an accurate performance prediction procedure can facilitate the selection of the most suitable network architecture to meet prescribed requirements. Our performance prediction model can be experimentally evaluated in this demo.

ACKNOWLEDGMENTS

This work was supported by EU H2020 MSCA through Project ACHIEVE-ITN (Grant No 765866), by the Spanish MINECO and European Region Development Fund (ERDF/FEDER) through Project RTI2018-097088-B-C31, by the Spanish Government through FPU Grant FPU17/02804, and by the US Office of Naval Research through Grant No. N00014-19-1-2156

REFERENCES

[1] LeCun, Y., Bengio, Y., Hinton, G.: "Deep Learning". Nature 521(7553), 436-444 (2015)

[2] Lin, M., Chen, Q., and Yan, S., "Network in network," arXiv 1312.4400 (2013)

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv 1512.03385 (2015).

[4] Iandola, F. et al., "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1MB model size," arXiv 1602.07360 (2016).

[5] Howard, A. et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv 1704.04861 (2017).

[6] "Raspberry Pi 3 Model B." (2019), https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[7] Y. Jia, E. Shelhamer, et al., "Caffe: Convolutional architecture for fast feature embedding," arXiv 1408.5093, (2014)

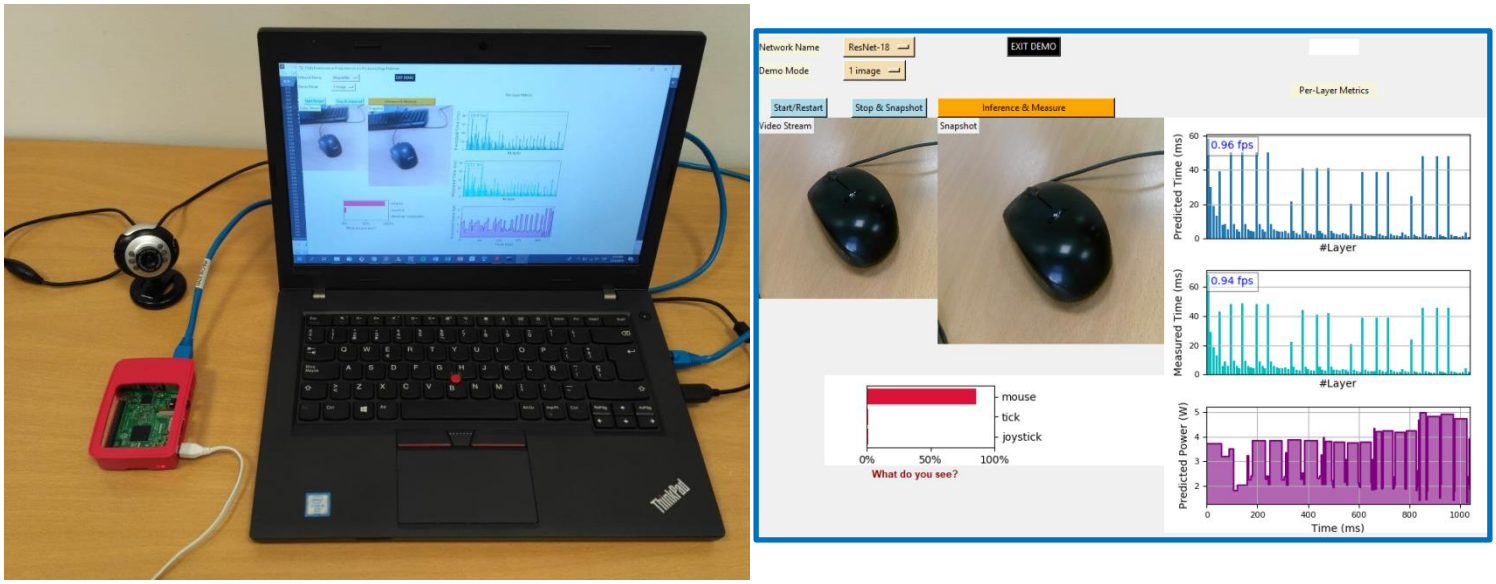

Figure 1. General set-up of the demo (left) and close-up of the graphical interface (right). CNN inference is performed on the embedded device (Raspberry Pi model 3B). Predictions on inference performance – time and power – are shown on the graphical interface, along with actual measurements from the Raspberry Pi.