Towards Building Deep Neural Network Models for Mobile Continuous Vision Sensing Applications

Loc N. Huynh
Singapore Management University
nlhuynh.2014@smu.edu.sg

Keywords

Mobile GPU; Mobile Sensing; Deep Learning; Continuous Vision

1. INTRODUCTION

Deep learning has revolutionized vision sensing applications in terms of accuracy comparing to other techniques. Its breakthrough comes from the ability to extract complex high-level features directly from sensor data. However, deep learning models are still yet to be natively supported on mobile devices due to high computational requirements. In this paper, we presents *DeepMon*, a framework to enable deep learning models on conventional mobile devices (e.g. Samsung Galaxy S7) for continuous vision sensing applications. *DeepMon* leverages several techniques to achieve fast inferences on mobile devices. Firstly, offloading computation onto integrated mobile GPU significantly reduces execution time of large models. Secondly, approximating large model into smaller one using decomposition techniques such as Tucker-2 lowers enormous amount of computational operations. Lastly, exploiting the similarity between consecutive video frames for intermediate data caching within model's processing pipeline also enhances inference latency for continuous vision sensing.

2. SYSTEM OVERVIEW

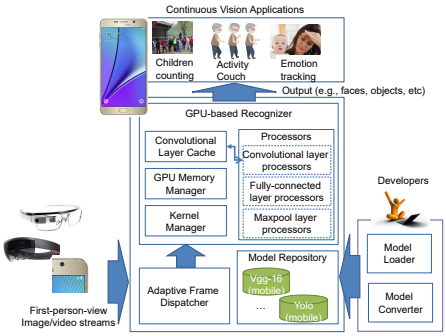

Figure 1: *DeepMon* System Architecture

DeepMon consists of 3 main components (Figure 1).

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

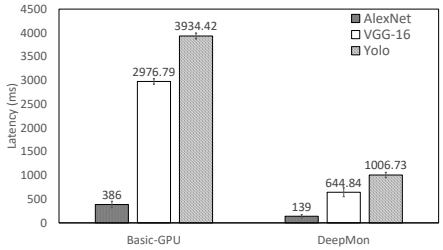

Figure 2: Overall Processing Latency

1) Adaptive frame dispatcher: takes responsibility for choosing important frames and submits them to recognizer.

2) Model repository: stores pre-trained models for various tasks such as image recognition, object detection. *DeepMon's* models are not limited to those we provided but can be converted from other framework such as Caffe [1] via our external *model converter*.

3) GPU-based recognizer: processes interesting frames and sends the output back to applications of interests. At first, *DeepMon* compares the current frame with the previous one to see if any regions within a frame should be recomputed to reduce the total computation. After that, caching kernels will be launched to compute only specific regions of the frame, followed by precision reduction if configured by applications.

3. EVALUATIONS

We used processing latency for a single image as our key evaluation metrics. We used the *UCF101* dataset [4] comprising 13,421 short videos (less than a minute long) created for activity recognition to evaluate *DeepMon* . Figure 2 shows our latency results of image recognition and object detection tasks using VGG-16 [3] and Yolo [2] models respectively.

4. REFERENCES

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.