An Efficient CNN Framework from Caffe to ARM

Background
Convolutional Neural Networks (CNNs) are being widely used and deployed in practical applications in domains such as computer vision, natural language processing, and recommendation systems. In particular, CNNs achieve outstanding image object recognition accuracy, setting new records for object detection and classification competitions every year. However, the high accuracy of CNNs comes with a high computational cost, and a high-power consumption as well. A hand optimization to a specific CNN is possible. However, it is usually tedious, time consuming, and case by case. Because of this there is a great need for an efficient and easy-to-use framework to “automatically” optimize CNNs to a given target platform.

NCNN [1] is an open-source neural network inference computing framework optimized for mobile platforms. NCNN does not have third party dependencies. It is a cross-platform framework, and it claims that it runs faster than all known open source frameworks on mobile phone CPUs. In the current release, it provides optimization for X86 and ARM cores. The optimization is provided with fixed C++ code for each CNN component, e.g., convolution and FC. Only floating-point is supported.

Assignment
Only limited optimization approaches are applied in the current NCNN framework. Many advanced loop-optimization approaches are still missing. Our on-going research work, such as Luc Waeijen’s layer fusion, DSE, are also expected to be useful to further improve NCNN’s efficiency. In this assignment, you are asked to:

1. Profile the current NCNN framework in ARM cores, e.g., R16. Both single core results and multiple-core results are required.
2. Further improve the performance of NCNN by introducing different loop and data optimization approaches.
3. (Optional) Upgrade the framework to support 4-bit/8-bit/16-bit fixed point. As more and more practical applications are using narrow-bit fixed point implementations, instead of more time/memory demanding floating point implementations.

Contacts
Luc Waeijen l.j.w.waeijen@tue.nl ES Group, Flux 4.076
Yifan He y.he@tue.nl ES Group

Note: if you find this assignment interesting, we may also make it a master-project assignment by adding extra tasks.

[1] https://github.com/Tencent/ncnn