DeepFoodCam: A DCNN-based Real-time Mobile Food Recognition System

Ryosuke Tanno, Koichi Okamoto, Keiji Yanai
The University of Electro-communications, Tokyo

1. Objective

- **Features of DeepFoodCam**
  - all processing complete within iPhone (server not required)
  - speeding up by multi-threading and fast framework
  - recognizing any size of images by multi-scale CNN
  - significant reduction in memory
  - built-in easy to various mobile devices

**Example: 101 class recognition**
- recognition time: 26.2ms (iPhone 7 Plus)
- top-5 accuracy: 91.5%

2. Proposal Contents

Anyone can make high-speed, high-precision object recognition and conversion iOS app

〜Flow of Making Mobile App〜

- Prepare a training image data
- Train a CNN model by Caffe (or DIGITS)
- Generate a C source code by Caffe2C automatically
- Prepare a GUI code of mobile app
- Generate CNN-based image recognition app by compiling the generated C code and the GUI code

Developed in our Lab

3. Recognition Engine

- **Training DCNN**
  - We use Network-In-Network (NIN)[2] considering mobile implementation
  - No parameters than Alexnet

**Network In Network [2]**
- only Conv layers
- no FC layers
- relatively smaller than the other architectures
- any image size correspondence

- **Pre-trained CNNs with ImageNet**
  - 2000 category images (totally 2.1 million images)
  - 4-layer Network-in-network

**Speeding up Conv layers ⇒ Speeding up GEMM**
- Computation of conv layers decomposed into "im2col" operation and matrix multiplications
- BLAS (iOS: Accelerate Framework, Android: OpenBLAS)
- We use the NEON instruction set which can execute four multiplications and accumulating calculations at once.
- iOS: 2Core*4 = 8 calculation, Android: 4Core*4 = 16 calculation

**Features of DeepFoodCam**
- Holding your smartphone to the target!!

4. Accuracy and Recognition Time

- **We use multi-scale network-in-networks (NIN)[2]**
  - Users can adjust the trade-off between recognition time and accuracy.
- **We implemented multi-threaded mobile applications on both iOS and Android**
  - Employing either NEON SIMD instructions or the BLAS library for fast computation of convolutional layers

**DeepFoodCam**
- Top-1: 78.8%
- Top-5: 65.3%

**NIN(NEON)**
- iPad Pro: 66.0[ms]
- iPad SE: 77.6[ms]
- Galaxy Note 3: 1652[ms]

**NIN(BLAS)**
- iPad Pro: 221.4[ms]
- iPad SE: 251.8[ms]
- Galaxy Note 3: 251.1[ms]

5. Characteristic analysis of iOS and Android

- We revealed that BLAS is better for iOS, while NEON is better for Android, and that reducing the size of an input image by resizing is very effective for speedup of DCNN-based recognition.
  - Please refer to [1] about the details.

6. Conclusion

- **Stand-alone DCNN-based mobile image recognition**
  - No need of a recognition server and communication.
  - Built-it trained DCNN model with UECFOOD-100
  - Implemented as iOS/Android app.
  - Released as iOS app on Apple Store (Please search "DeepFoodCam") as Android app (APK) on [http://foodcam.mobi](http://foodcam.mobi)/

- **Excellent performance with reasonable speed and model size**
  - UECFOOD100: 78.8% (top-1) 95.2% (top-5)
  - in 66.6 [ms] with 5.5M weights (22MB)
  - Employing Network-in-Network
  - Adding batch normalization and additional layers

- **Multi-scale recognition**
  - User can choose the balance between speed and accuracy
  - Ex. iPhone 7 Plus:
  - 26.2[ms] for 160x160 images 55.7[ms] for 227x227 images

Multiple Style Transfer and Object Recognition App

- **Food Rec App (both iOS/Android)**
  - Our Project page
  - [http://foodcam.mobi](http://foodcam.mobi/)
  - Please search "DeepFoodCam"

- **Multi Style Transfer (only iOS)**
  - [http://foodcam.mobi/](http://foodcam.mobi/)
  - Please search "RealTimeMultiStyleTransfer"

---

Reference