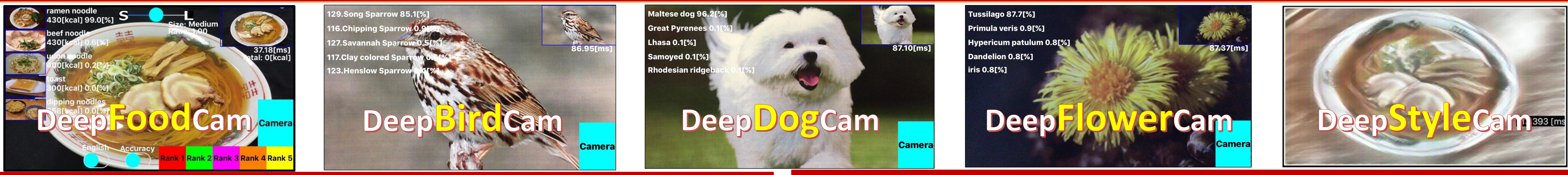
D-3A-48

DeepXCam: Very Fast CNN-based Mobile Applications: Multiple Style Transfer and Object Recognition

Ryosuke Tanno, Wataru Shimoda, Keiji Yanai   The University of the Electro-Communications, Tokyo



1. Objective

Common Features of All the Apps

- Standalone CNN mobile applications (no external server required)
- Speeding up by multi-threading and fast framework
- Recognizing any size of images by multi-scale Fully Convolutional Network
- Significant reduction in memory requirements
- Being applicable to various kinds of mobile devices

Example: 100-class food recognition

- recognition time: 26.2ms(iPhone7Plus), top-5 accuracy: 91.5%

2. Proposal Contents

Anyone can build very fast CNN-based mobile apps including object recognition apps and style transfer apps.

〜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

- **Caffe2C / Chainer2C**
  - convert parameter files to C source codes that run on mobile devices
- **Very fast CNN-based mobile recognition/transfer engine**
  - speeding up by multi-threading and fast framework
- **Adopting NIN architecture for a recognition engine**
  - any size of input images
  - the trade-off between accuracy and processing time by changing input image sizes

If you prepare training data, you can create mobile recognition apps in a day !!

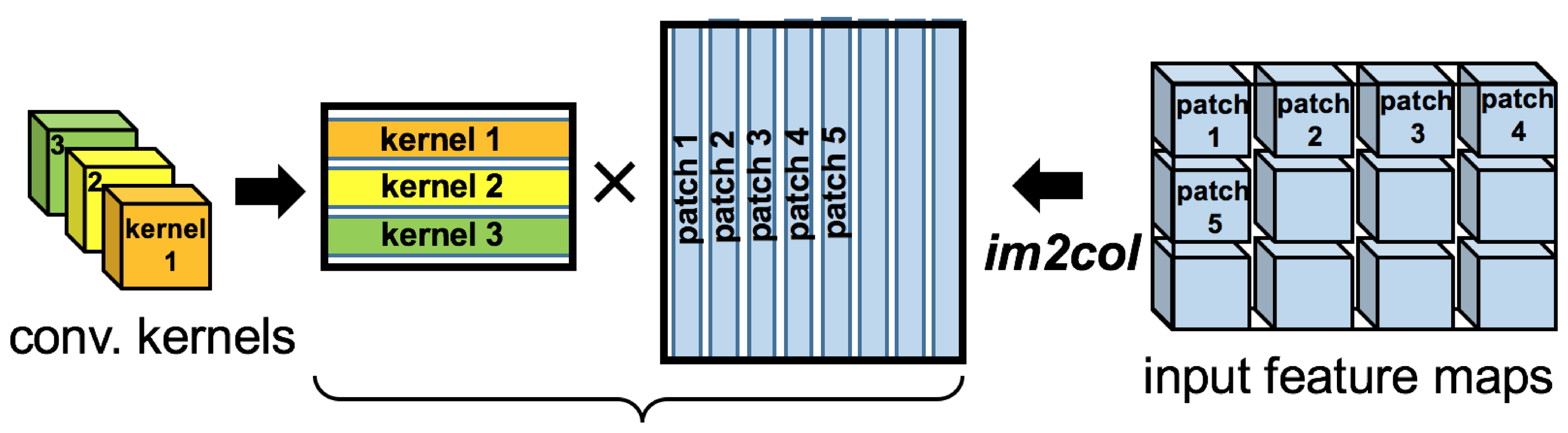
3. DeepXCam for recognition (X = Food, Dog, Bird, Flower)

- ***Training DCNN***
  - Use **Network-In-Network(NIN)**[3] considering mobile implementation
  - Save the size of the network parameters

Network In Network [3]
- only conv layers
- no FC layers
- relatively smaller than the other architectures
- any image size correspondence

	Param	Memory	Top-5
AlexNet	62Million	248MB	95.1%
NIN(4L+BN)	5.5Million	22MB	95.2%
NIN(5L+BN)	15.8Million	63MB	96.2%

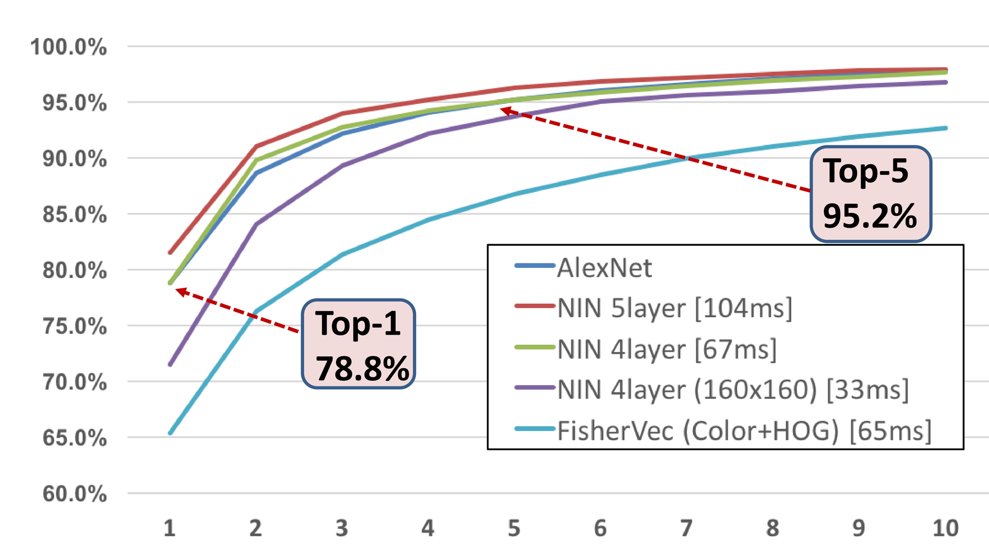
- ***Pre-trained CNNs with ImageNet 2000 category images*** (totally 2.1 million images)
- ***Speeding up Conv layers ⇒ Speeding up GEMM***
  - computation of conv layers is decomposed into “im2col” operation and matrix multiplications
  - BLAS(iOS: Accelerate Framework, Android: OpenBLAS)
  - we use the NEON instruction set which can execute four multiplications and accumulating calculations at once.
  - iOS: 2Core*4 = 8 calculation, Android: 4Core*4 = 16 calculation



GEMM: generic matrix multiplication (=conv. layer computation)

4. Accuracy and Recognition Time

UEC-FOOD100 class recognition performance



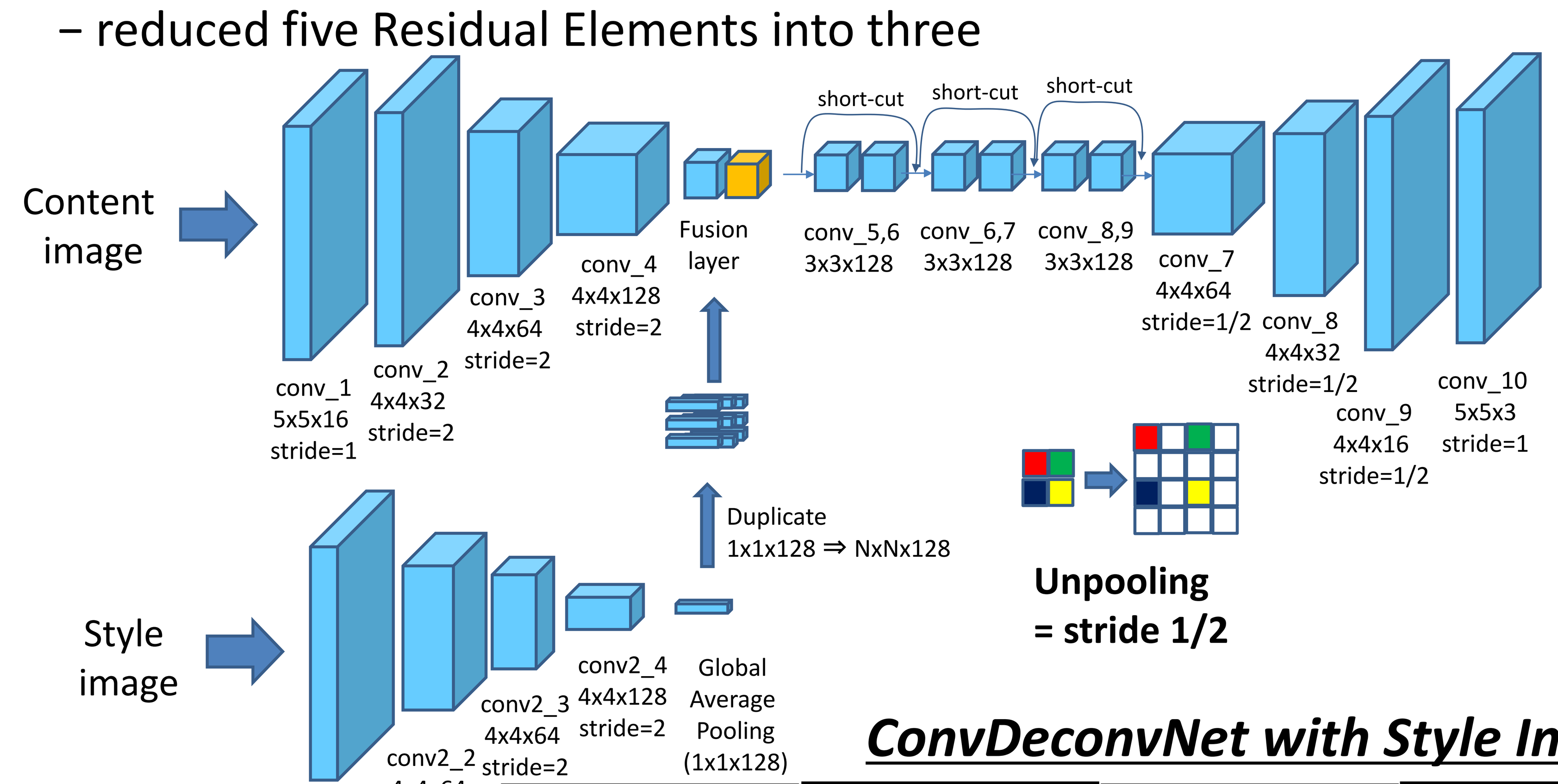
Trade-Off between Accuracy and Recognition Time

Input Image Size	227x227	200x200	180x180	160x160
iPhone 7 Plus	55.7[ms]	42.1[ms]	35.5[ms]	**26.2**[ms]
iPad Pro	66.6[ms]	49.7[ms]	44.0[ms]	**32.6**[ms]
iPhone SE	77.6[ms]	56.0[ms]	50.2[ms]	37.2[ms]
Accuracy (top-5)	**95.2%**	95.1%	94.1%	91.5%

We achieve ***real-time !!***

4. DeepStyleCam (Image Style Transfer)

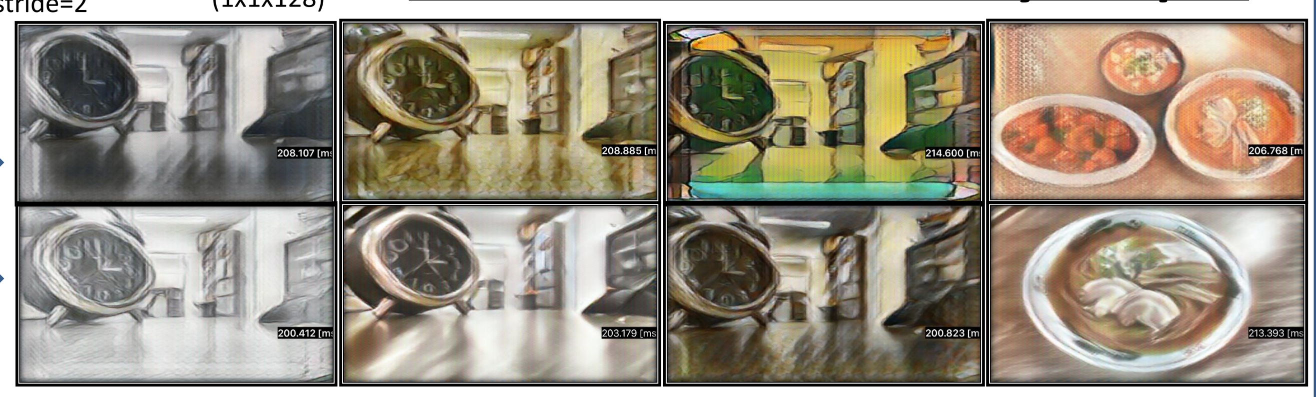
- ***ConvDeconvNetwork[2] can treat only one fixed style.***
  - If transferring ten kinds of styles, we have to train ten different ConvDeconvNetwork independently.
  - This isn’t good for mobile implementation(required memory size)
- ***We modified [2] can train multiple styles at the same time***
  - adding a fusion layer and a style input stream(inspired by [1])
- ***Training***
  - We input sample images to the content stream and style images to the style stream.(The training method is the same as [2])
- ***We shrunk the ConvDeconvNetwork compared to [2]***
  - added one down-sampling layer and up-sampling layer
  - replaced 9x9 kernels with smaller 5x5 kernels in the first and last convolutional layers
  - reduced five Residual Elements into three



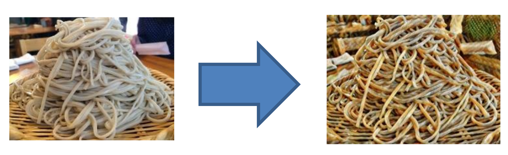
ConvDeconvNet with Style Input

Nomal mode

Color Preserving mode



Ex. Image Size: 250x250 ,
Computation: 1,303,800,800 times(13billion)
Parameter num: 1,250,835



175ms (iPhone7+)
180ms (iPad Pro)
200ms (iPhone SE)

Multiple Style Transfer and Object Recognition App

·Food Rec App (both iOS/Android)

Our Project page

http://foodcam.mobi

Please search “***DeepFoodCam***”

·Multi Style Transfer (only iOS)

Please search “***RealTimeMultiStyleTransfer***”

Reference

[1] S. Iizuka et al.: Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification, SIGGRAPH, 2016.
[2] J. Johnson et al.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution, ECCV, 2016.
[3] M. Lin et al. Network In Network, ICLR, 2014.