Real-Time Image Classification and Transformation Apps on iOS by "Chainer2MPSNNGraph"

Yuki Izumi† Daichi Horita† Ryosuke Tanno†† Keiji Yanai†

† The University of Electro-Communications, †† NTT Communications, Tokyo, Japan

Overview

DNN code generator, "Chainer2MPSNNGraph"

– Convert DNN models trained by Chainer [1] into Swift codes utilizing MPSNNGraph API.

Demo applications

- VGG16
- Multi-Style Transfer ("DeepStyleCam") [2]
- Food Image Transformation ("Magical Rice Bowl") [3]

Chainer2MPSNNGraph

Method

1. We read a model trained by Chainer and feed-forward the model once with Chainer module to create a model graph.
2. Chainer2MPSNNGraph analyze the model graph and generate a Swfit code and parameter files for the MPSNNGraph API.

➡ We can develop GPU-powered DNN application easily.

MPSNNGraph

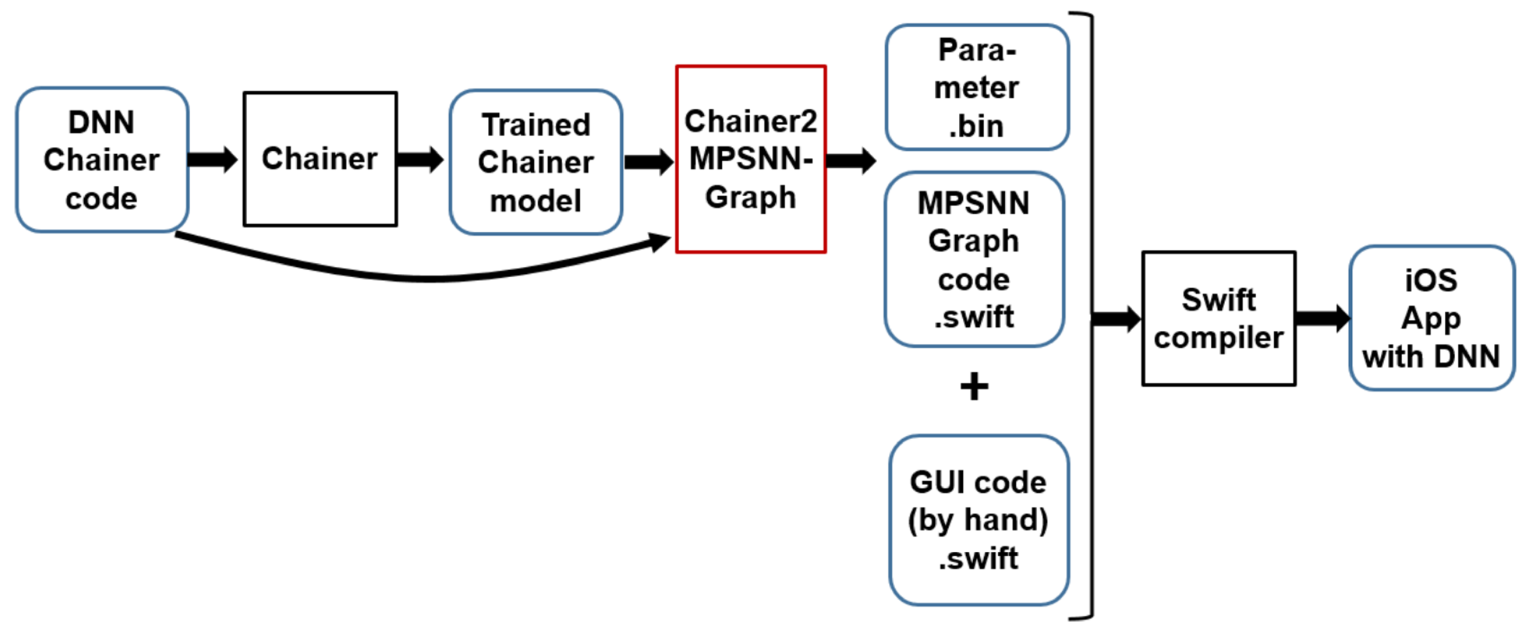
MPSNNGraph is a part of the Metal Performance Shaders library which is a library to utilize a GPU on an iPhone.

Example of generated Swift code

```
let conv1 = MPSCNNConvolutionNode(source: conv01.resultImage,
                                   weights: DataSource("conv1", 9, 9, 3, 32, 1, useBias: true))
let relu1 = MPSCNNNeuronReLUNode(source: conv1.resultImage)
let bn1 = MPSCNNBatchNormalizationNode(source: relu1.resultImage,
                                        dataSource: DataSource2("bn1", 32))
let conv2 = MPSCNNConvolutionNode(source: bn1.resultImage,
                                   weights: DataSource("conv2", 4, 4, 32, 64, 2, useBias: true))
let relu2 = MPSCNNNeuronReLUNode(source: conv2.resultImage)
let bn2 = MPSCNNBatchNormalizationNode(source: relu2.resultImage,
                                        dataSource: DataSource2("bn2", 64))
let conv3 = MPSCNNConvolutionNode(source: bn2.resultImage,
                                   weights: DataSource("conv3", 4, 4, 64, 128, 2, useBias: true))
let relu3 = MPSCNNNeuronReLUNode(source: conv3.resultImage)
let bn3 = MPSCNNBatchNormalizationNode(source: relu3.resultImage,
                                        dataSource: DataSource2("bn3", 128))
let concat1 = MPSNNConcatenationNode(sources: [bn3.resultImage, styleImage1,
                                               styleImage2, styleImage3, styleImage4])
let conv4 = MPSCNNConvolutionNode(source: concat1.resultImage,
                                   weights: DataSource("conv4", 1, 1, 141, 128, 1, useBias: true))
let relu4 = MPSCNNNeuronReLUNode(source: conv4.resultImage)
let bn4 = MPSCNNBatchNormalizationNode(source: relu4.resultImage,
                                        dataSource: DataSource2("bn4", 128))
```

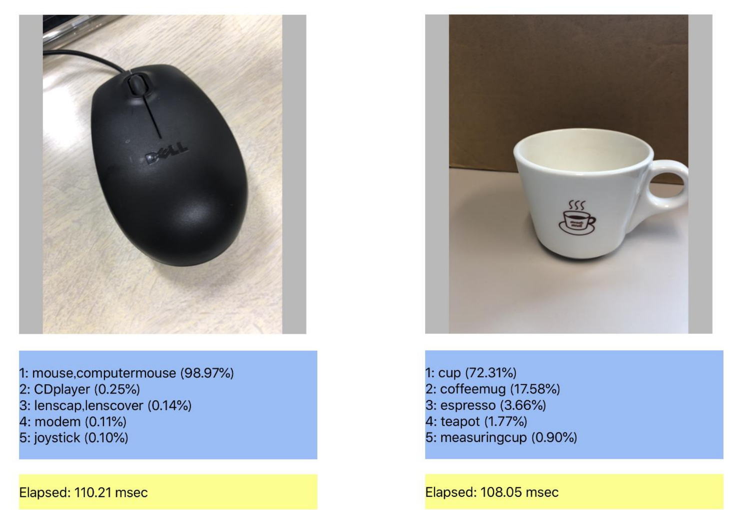
Work Flow

1. We train a model by Chainer.
2. We prepare the trained model and codes written in python
3. "Chainer2MPSNNgraph" converts the model and Chainer codes into parameter files and Swift code for the MPSNNGraph API.
4. We write a GUI code by hand.
5. We develop DNN iOS application from parameter files, a MPSNNGraph code and a GUI code.



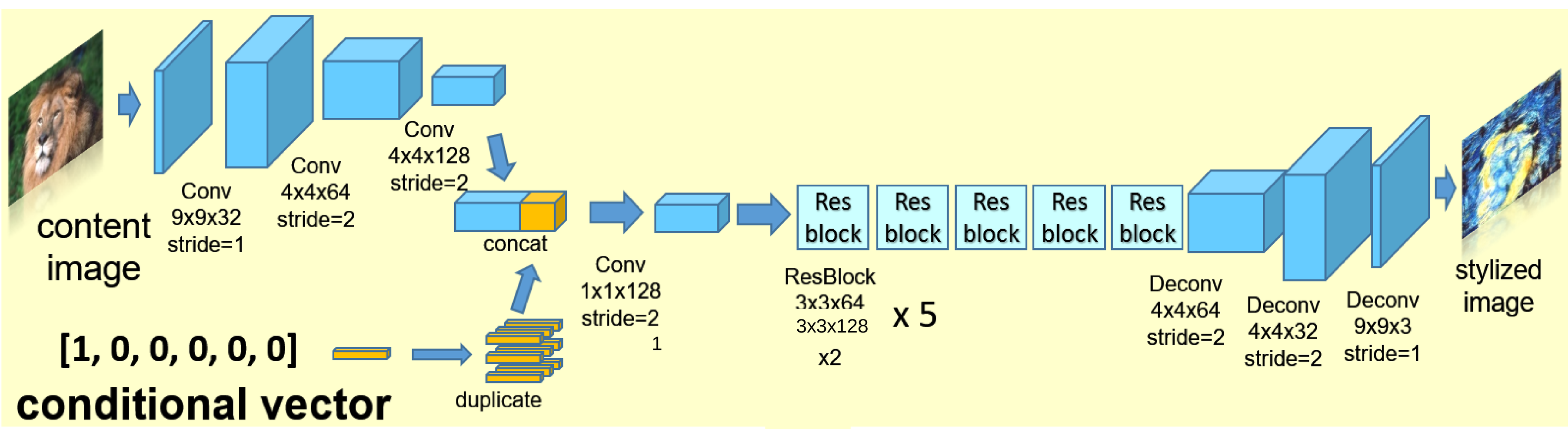
VGG16

method (iPhone8Plus)	time (ms)
MPS	155.2
CoreML	144.9
Chainer2MPSNNGraph	**109.0**

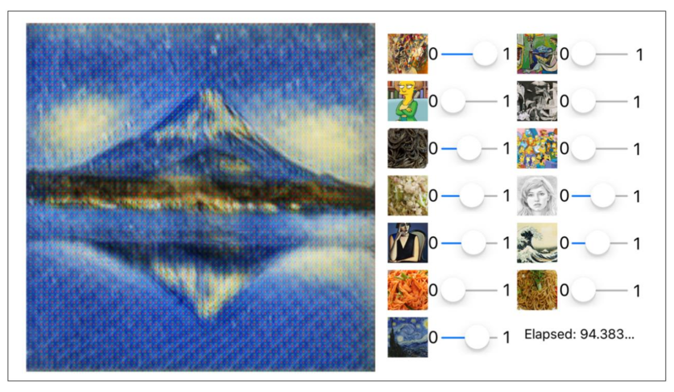


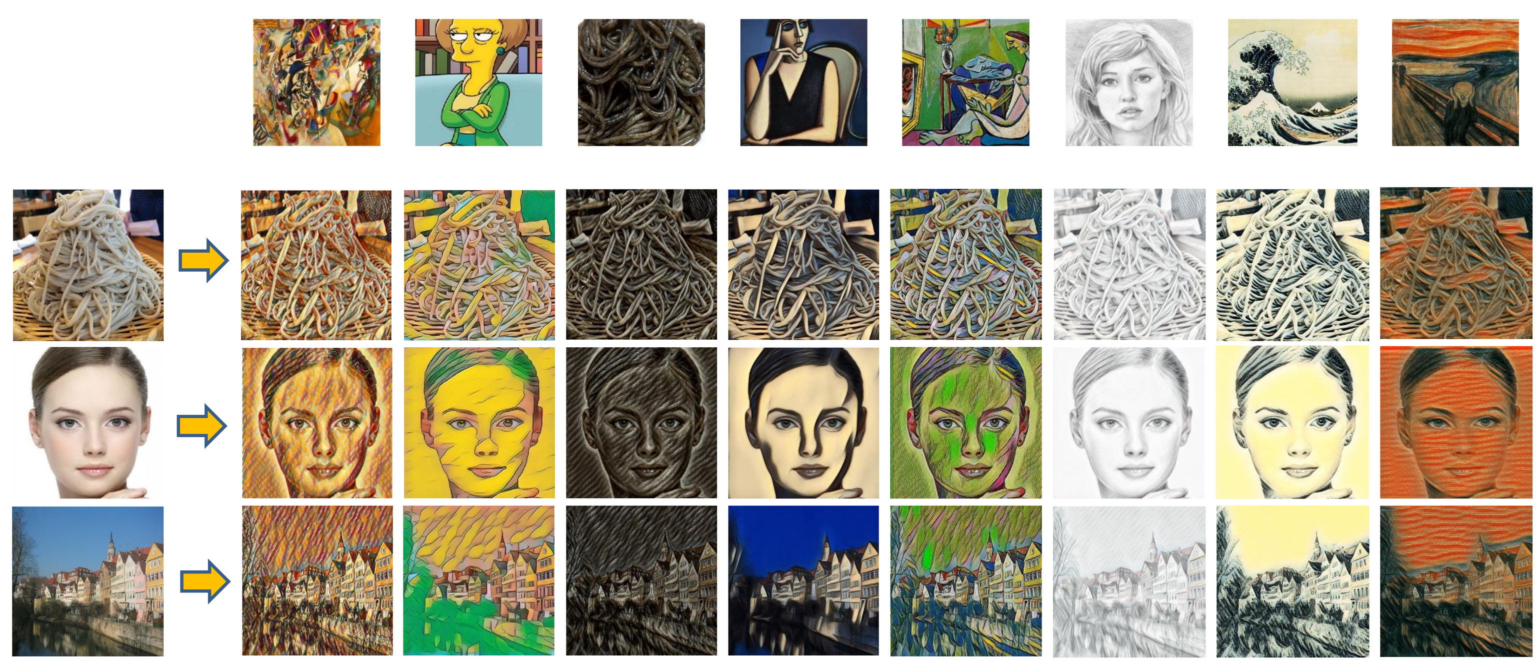
Multi-Style Transfer: "DeepStyleCam" [2]

DeepStyleCam

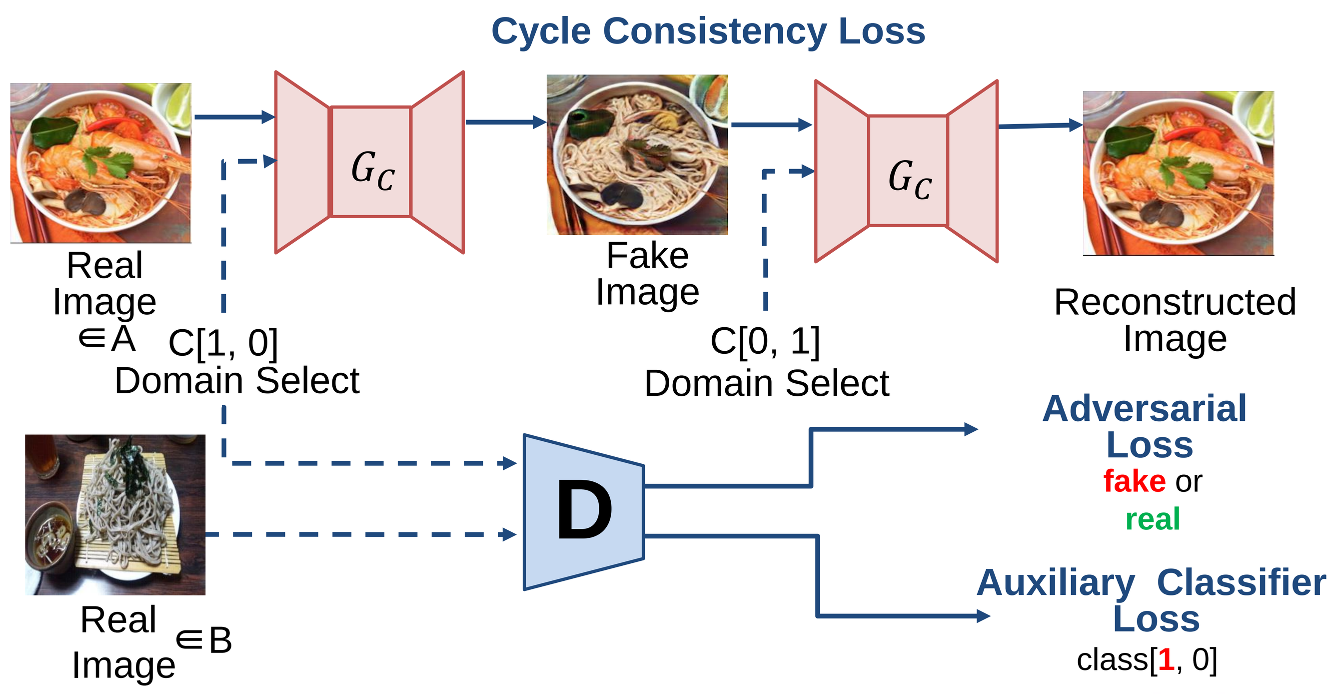


method (iPhone8Plus)	time (ms)
Chainer2C (CPU)	138.6
CoreML (GPU)	101.0
Chainer2MPSNNGraph (GPU)	**96.3**





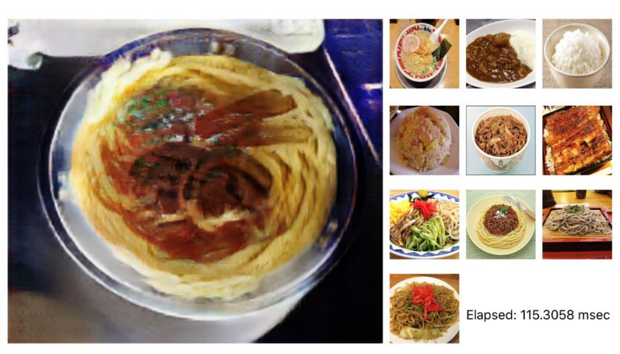
Food Image Transformation: "Magical Rice Bowl" [3,4]



Based on "StarGAN" [5]

G: (trans. net) 3Conv-5ResBlock-3Deconv

device	time (ms)
iPhone 8 Plus	82.8
iPhone Xs Max	71.0



Reference

[1] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In Proc. of NIPS Workshop on Machine Learning Systems (LearningSys), 2015.
[2] K. Yanai. Unseen style transfer based on a conditional fast style transfer network. In Proc. of International Conference on Learning Representation Workshop Track (ICLR WS), 2017.
[3] D. Horita, R. Tanno, W. Shimoda, and K. Yanai. Food category transfer with conditional cycle gan and a large-scale food image dataset. In Proc. of International Workshop on Multimedia Assisted Dietary Management (MADIMA), 2018.
[4] R. Tanno, D. Horita, W. Shimoda and K. Yanai: Magical Rice Bowl: Real-time Food Category Changer, ACM Multimedia, (demo) (2018).
[5] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In Proc. of IEEE Computer Vision and Pattern Recognition, 2018.