

Keiji Yanai Ryosuke Tanno

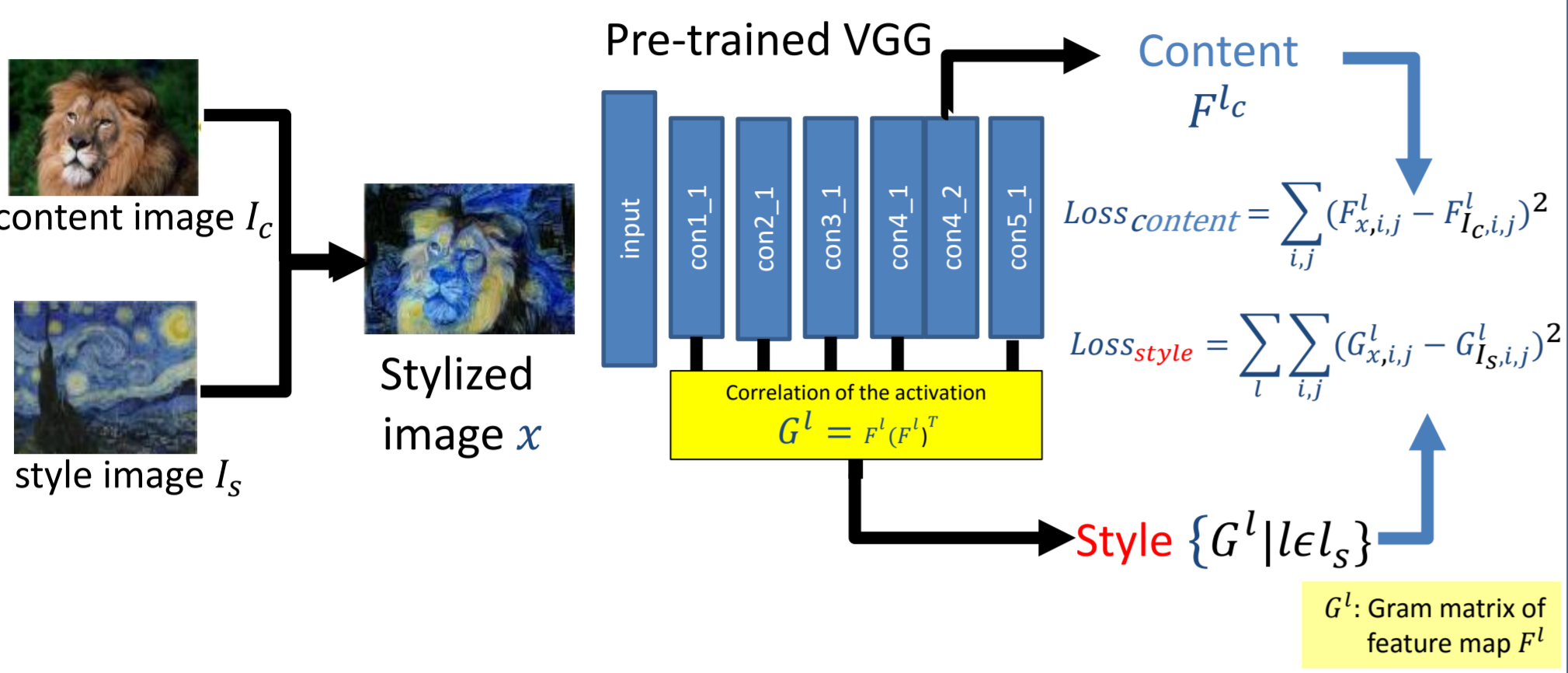
The University of Electro-Communications, Tokyo, Japan



Project HP with online demos:  
<http://foodcam.mobi/deepstylecam/>

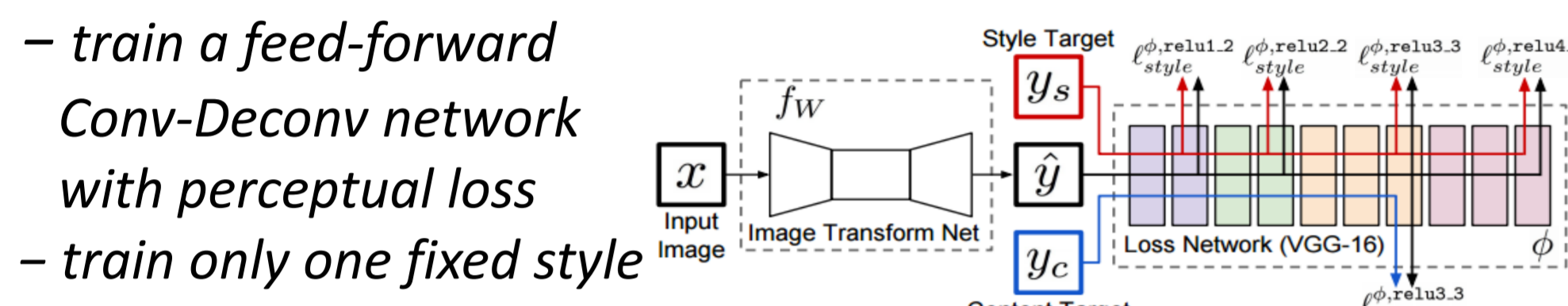
## 1. Introduction

• Neural artistic style transfer by Gatys et al. (2015) [A]



– Optimization-based method → **time consuming** (1 min-)

• Feed-forward fast style transfer network by Johnson et al. (2016) [B]



– train a feed-forward Conv-Deconv network with perceptual loss  
 – train only one fixed style  
 → need to train an individual model for each style image  
 e.g. ten styles → ten models

Multiple style feed-forward network is desirable!

**[idea 1] Conditional fast style transfer network**

→ However, only trained styles can be transferred.

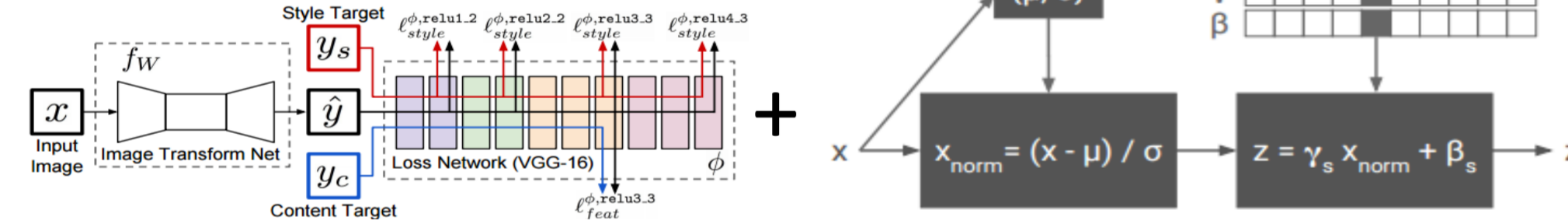
Unseen style transfer feed-forward network is better!

**[idea 2] Unseen style transfer network**  
 (Extension of a conditional fast style transfer net.)

## 2. Concurrent Works (mini survey)

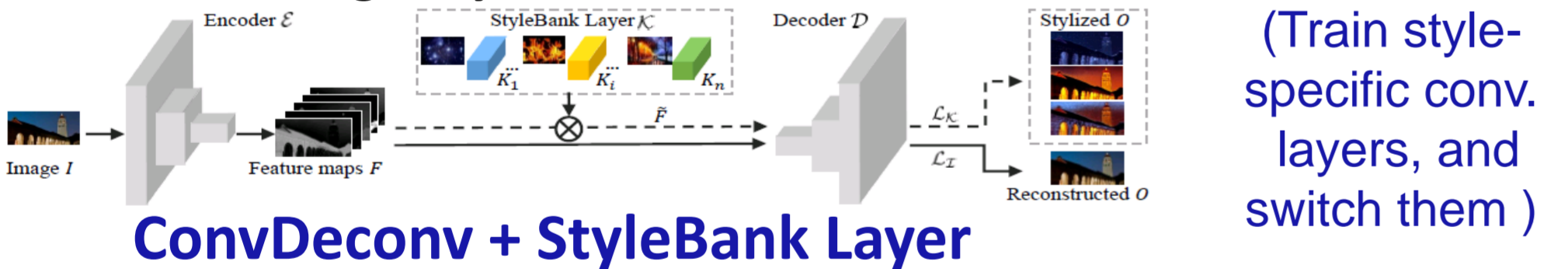
### (A) Multiple Styles

[1] V. Dumoulin et al.: A Learned Representation of Artistic Style, ICLR conf. track, 2017/03



ConvDeconv + Conditional Instance Normalization  
 (Train style-specific scale and shift parameters of all the IN layers)

[2] D. Chen et al.: StyleBank: An Explicit Representation for Neural Image Style Transfer, arXiv: 1703.09210, 2017/03.



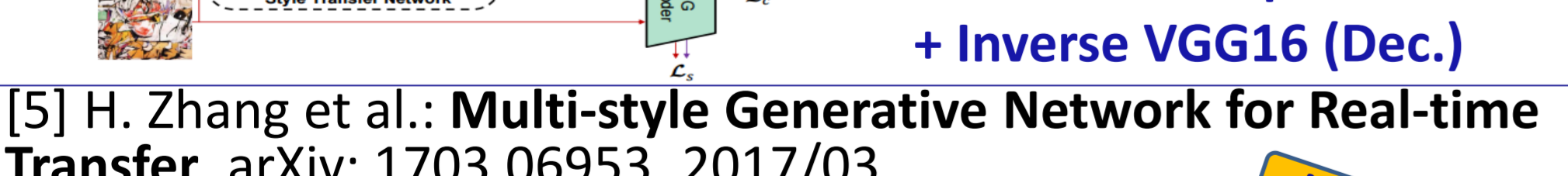
ConvDeconv + StyleBank Layer  
 (Train style-specific conv. layers, and switch them)

[3] T. Q. Chen et al.: Fast Patch-based Style Transfer of Arbitrary Style, arXiv: 1612.04337, 2016/12.



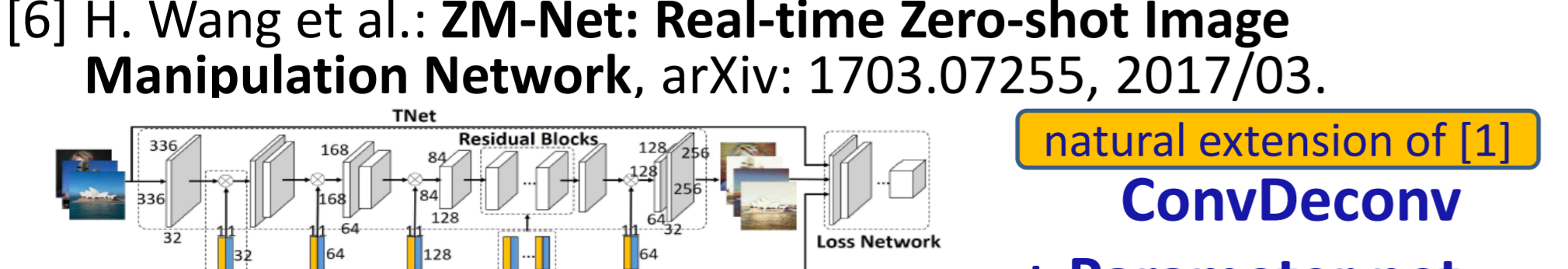
VGG16 (Enc.) + Style Swap + Inverse VGG16 (Dec.)  
 (modifying f-maps)

[4] X. Huang et al.: Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization, ICLR WS, 2017/03.  
 (inspired by Y. Li et al.: Demystifying Neural Style Transfer, arXiv: 1701.01036, 2017.)



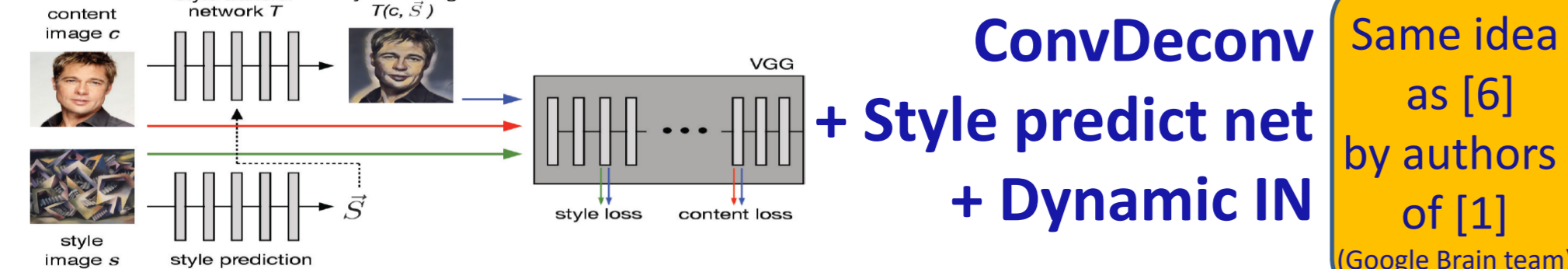
VGG16 (Enc.) + Adaptive IN + Inverse VGG16 (Dec.)  
 + ConvDeconv + Style feature net (VGG16) + Integration (inspiration) Layer

[5] H. Zhang et al.: Multi-style Generative Network for Real-time Transfer, arXiv: 1703.06953, 2017/03.



ConvDeconv + Parameter net + Dynamic IN  
 (natural extension of [1])

[6] H. Wang et al.: ZM-Net: Real-time Zero-shot Image Manipulation Network, arXiv: 1703.07255, 2017/03.

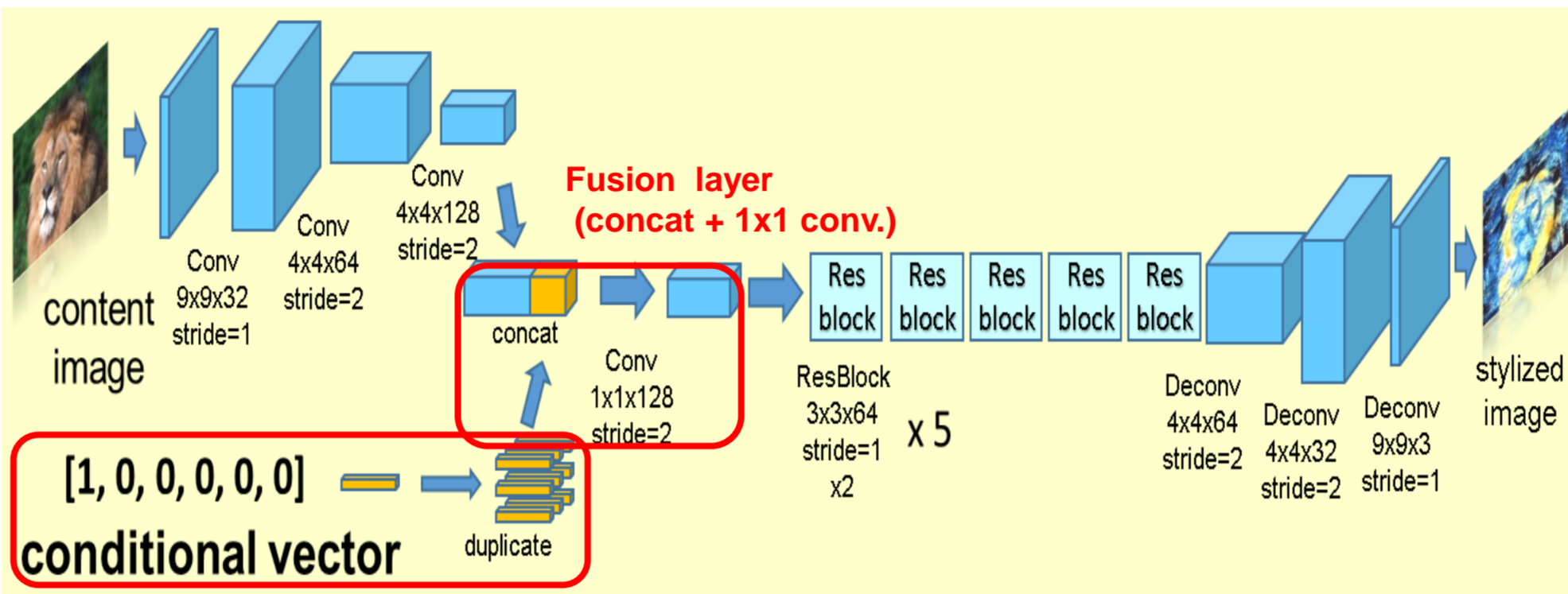


ConvDeconv + Style predict net + Dynamic IN  
 (Same idea as [6] by authors of [1] (Google Brain team))

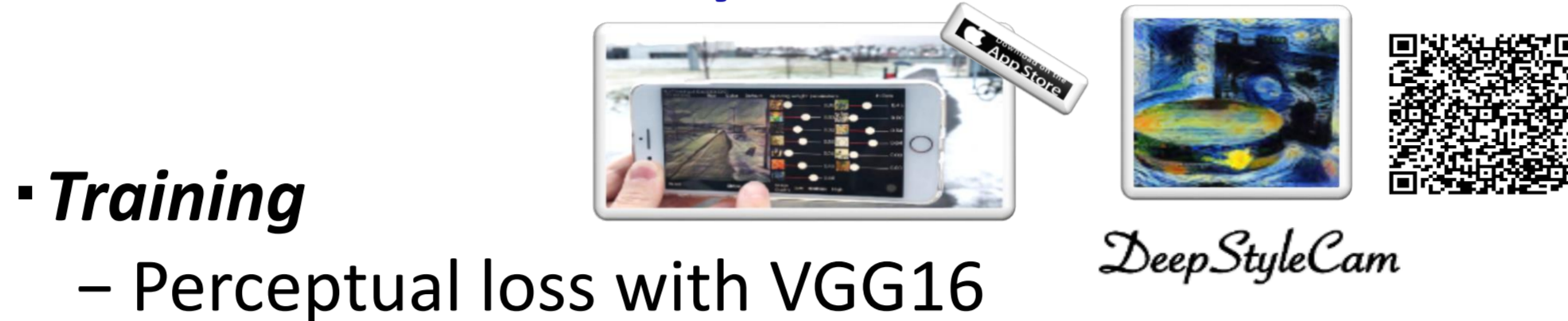
## 3. Conditional Fast Style Transfer

• Add a style condition input to ConvDeconv net.

- Add a fusion layer and a style input
- Style input: one-hot conditional vector e.g. style1 [1,0,0,0,...], style2: [0,1,0,0,...], style3: [0,0,1,0,...]...
- Base network: Johnson's ConvDeconv net [B] Each layer has BN and ReLU except last one.



### Conditional Fast Style Transfer Network

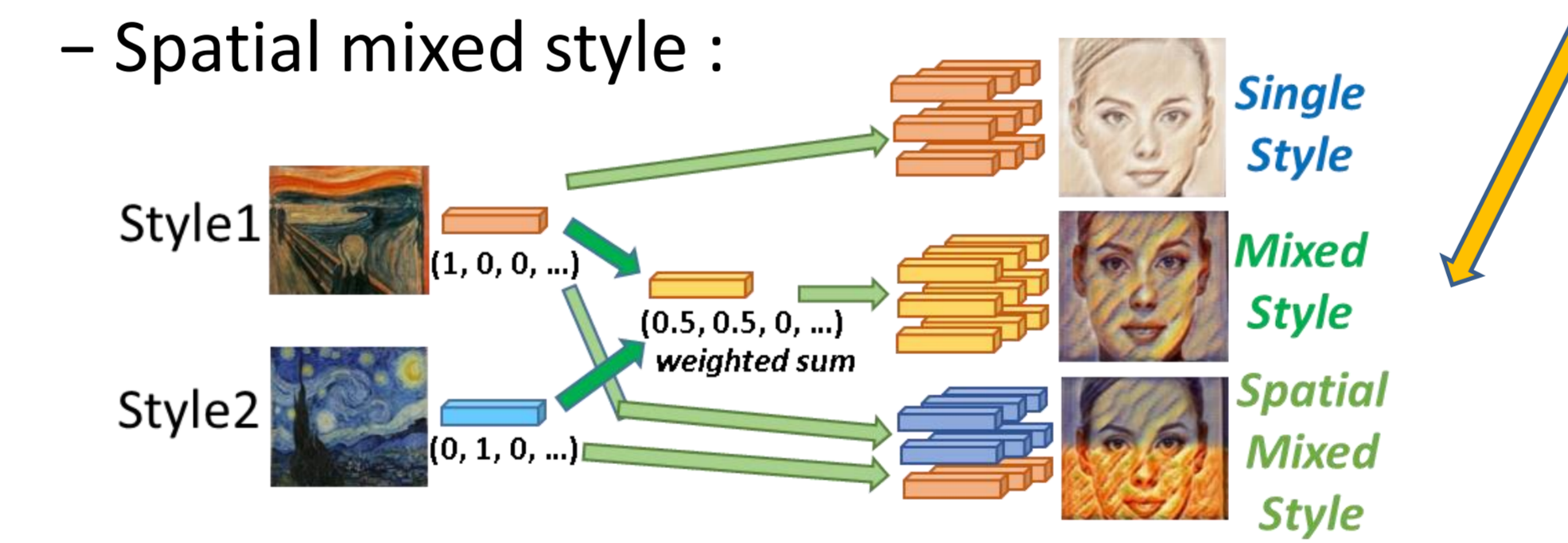


• Training

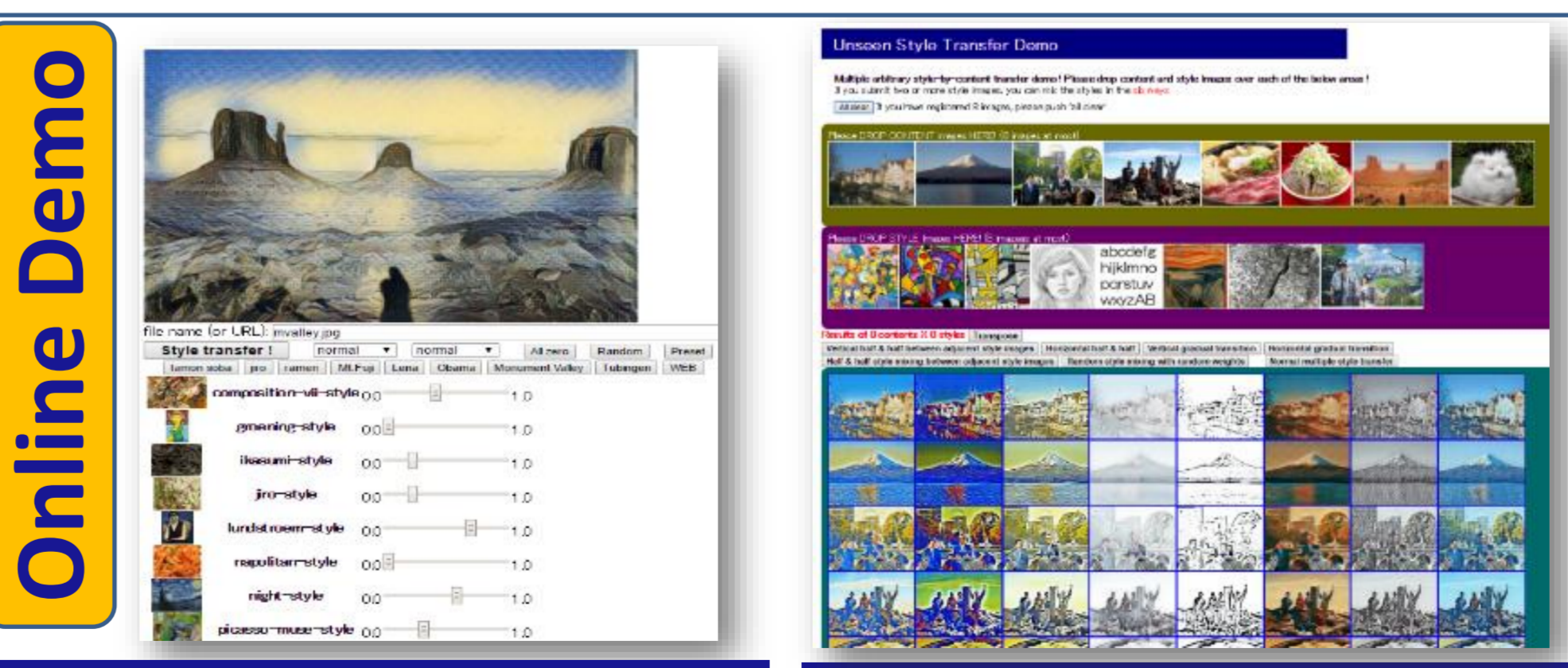
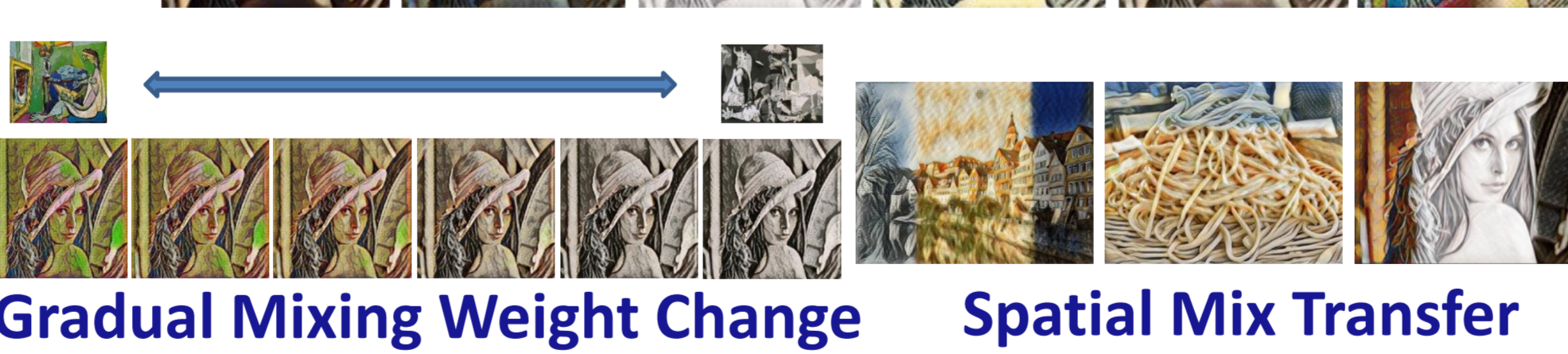
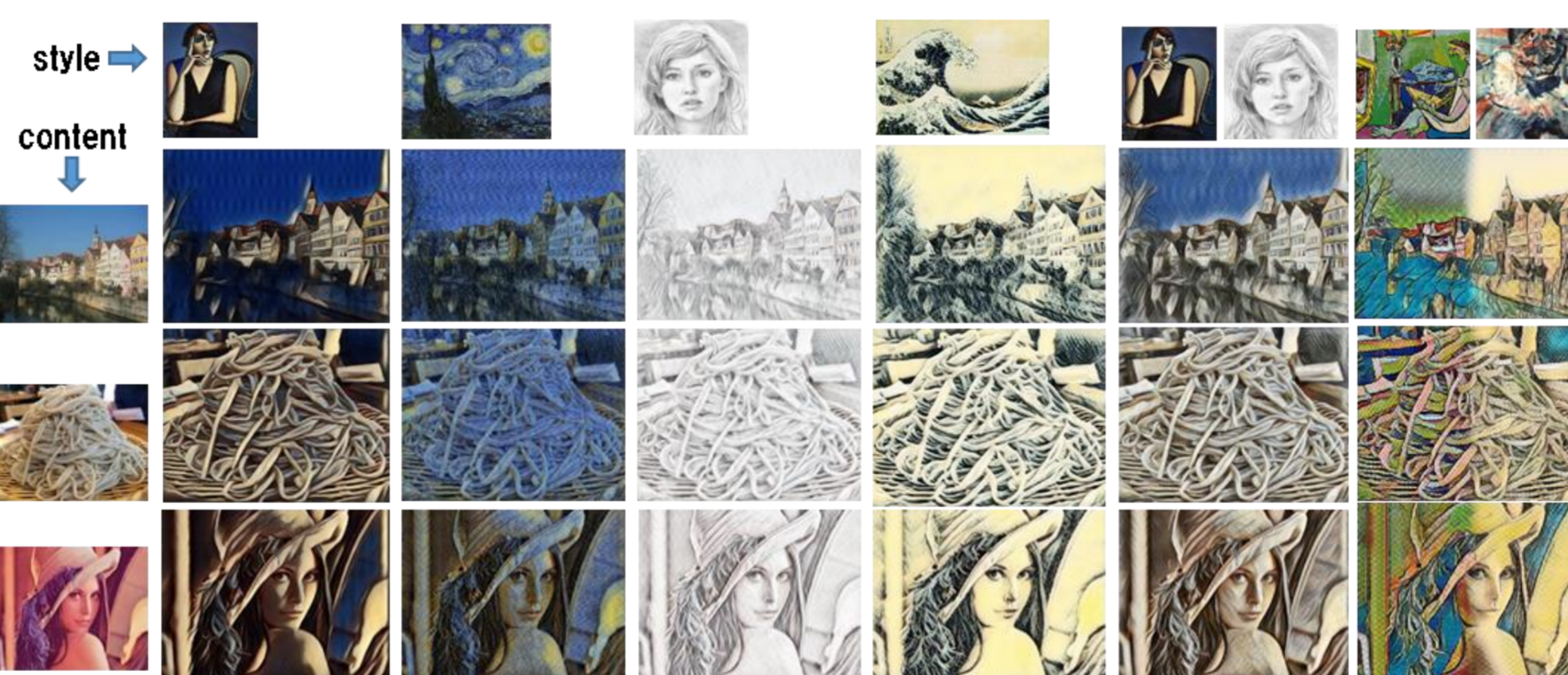
- Perceptual loss with VGG16 (the same way as Johnson's work [B])  
 content: conv3\_3  
 style: conv1\_2, c2\_2, c3\_3, c4\_3
- Each mini batch : one content image + all the style images (= multi-style version of Instance Normalization)

• Generating stylized images in three ways

- Input: content image + style condition vector
- Single style: one-hot vector [1,0,0,0,...], [0,1,0,0,...]
- Mixed style: multiple-style-weighting [1,1,1,...], [0.2, 0.1, 0.8,...]
- Spatial mixed style :



• Results of cond. style transfer with trained styles



Conditional style transfer <http://bit.ly/mixstyle>  
 Unseen style transfer <http://bit.ly/unseenstyle>

## References

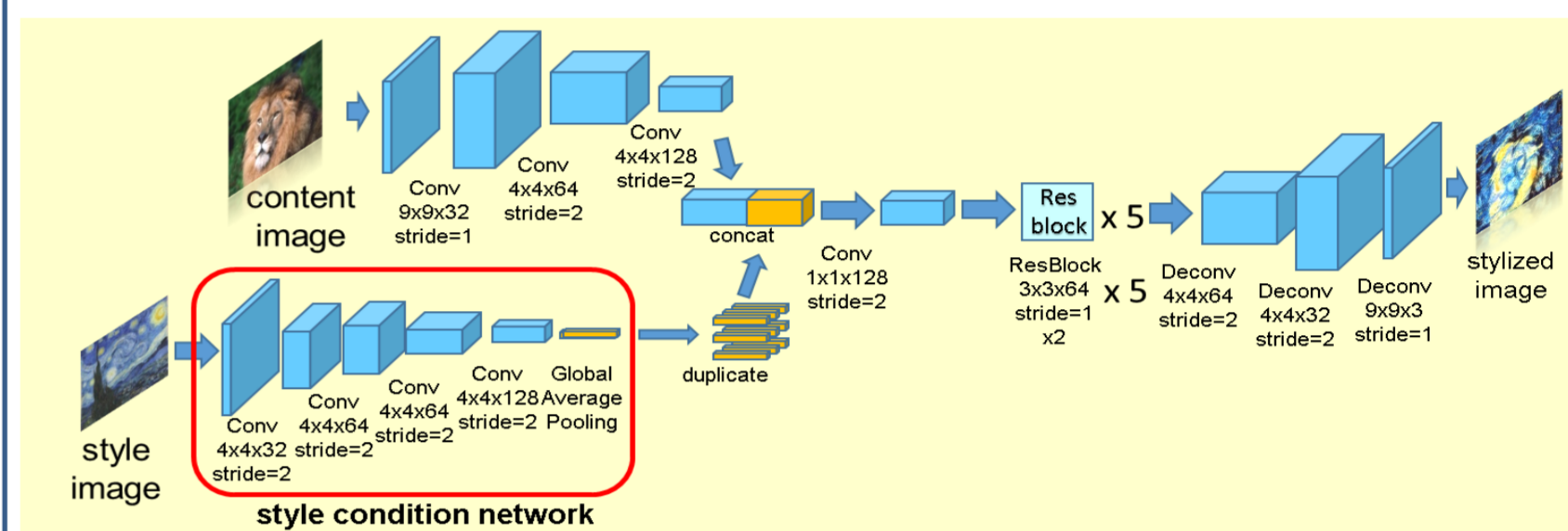
- [A] L. A. Gatys et al.: Image style transfer using convolutional neural networks, CVPR, 2016. (A Neural algorithm of artistic style, arXiv: 1508.06576, 2015)
- [B] J. Johnson et al.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution, ECCV, 2016.

## 4. Unseen Style Transfer

• Add a style condition network to the conditional FST network.

- Confirmed that a real-value cond. is OK.
- Style condition vector can be generated by a CNN (not by hand)
- Propose a style condition network which generates a style condition vector from a given style image directly.

### Conditional Style Transfer + Style Network → "Unseen style transfer network"



Unseen Style Transfer Network = Conditional Fast Style Transfer Network + Style Condition Network

• Training + Style Condition Network

- End-to-end training with perceptual loss
- Each mini batch : one content image + randomly selected style images from 50,000 style images (WikiArt)

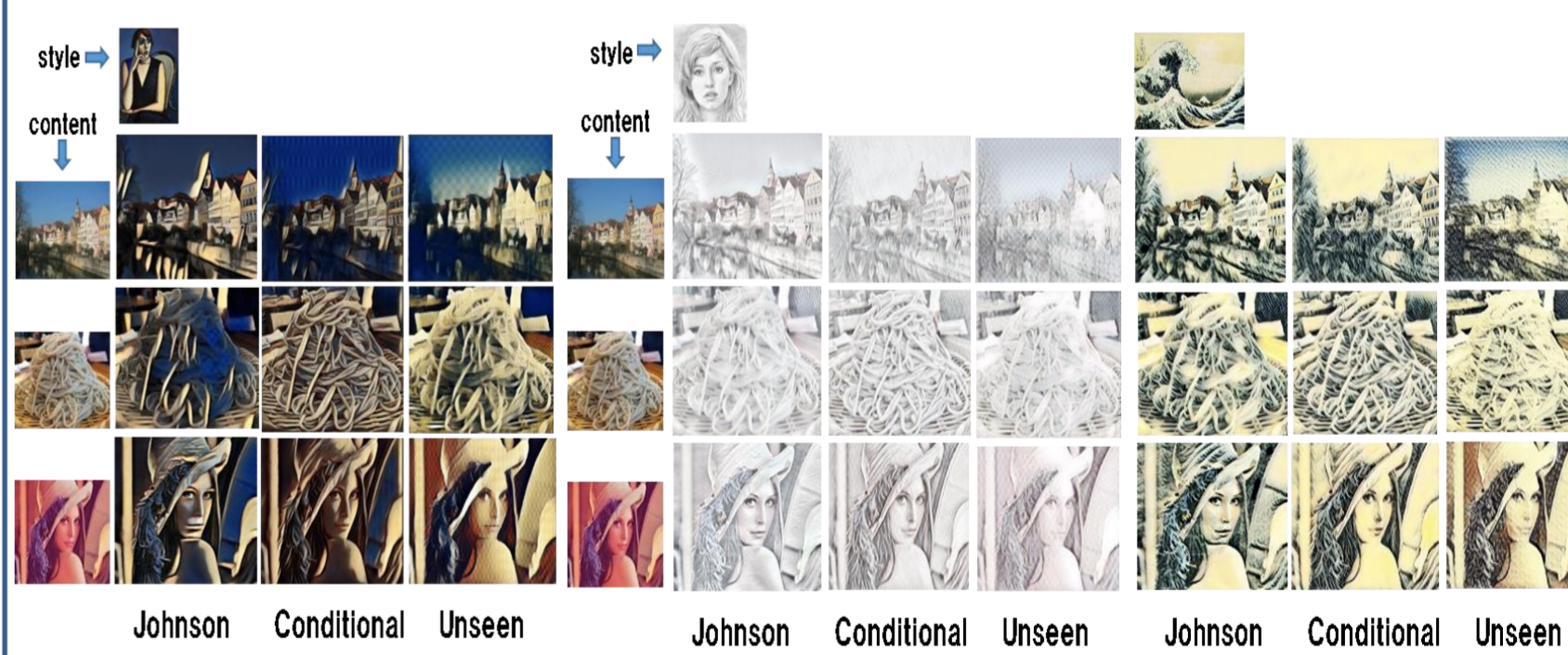
• Generating stylized images in three ways

- the same way as a Conditional Fast Style Transfer Network

• Results of unseen style transfer

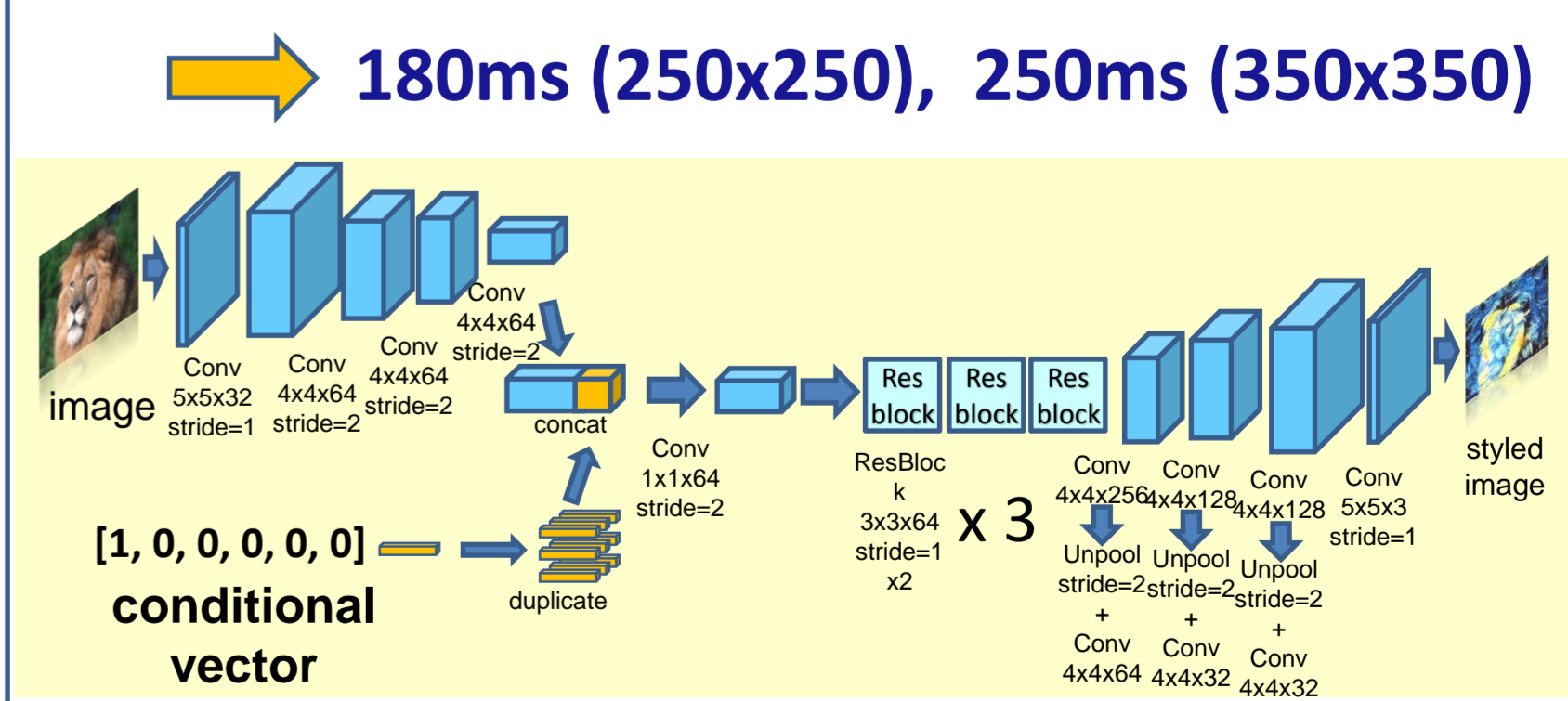


### Qualitative comparison (single/multi/unseen)



## 5. Mobile Implementation

- Shrink the network for mobile devices.
- Add one down-conv. and one up-conv.
- Reduce the num of ResBlock from 5 to 3



Conditional Style Transfer Network for Mobile

