# Unseen Style Transfer
# Based on a Conditional Fast Style Transfer Network

Keiji Yanai   The University of Electro-Communications, Tokyo, Japan

UEC TOKYO

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

Real-time transfer app is available !

DeepStyleCam

**bit.ly/deepstylecam**

---

## 1. Introduction

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

Pre-trained VGG

content image $I_c$

Stylized image $x$

style image $I_s$

Content $F^{l_c}$

$Loss_{content} = \sum_{i,j}(F^l_{x,i,j} - F^l_{I_c,i,j})^2$

Correlation of the activation
$G^l = F^l(F^l)^T$

$Loss_{style} = \sum_l \sum_{i,j}(G^l_{x,i,j} - G^l_{I_s,i,j})^2$

Style $\{G^l|l\epsilon l_s\}$

$G^l$: Gram matrix of feature map $F^l$
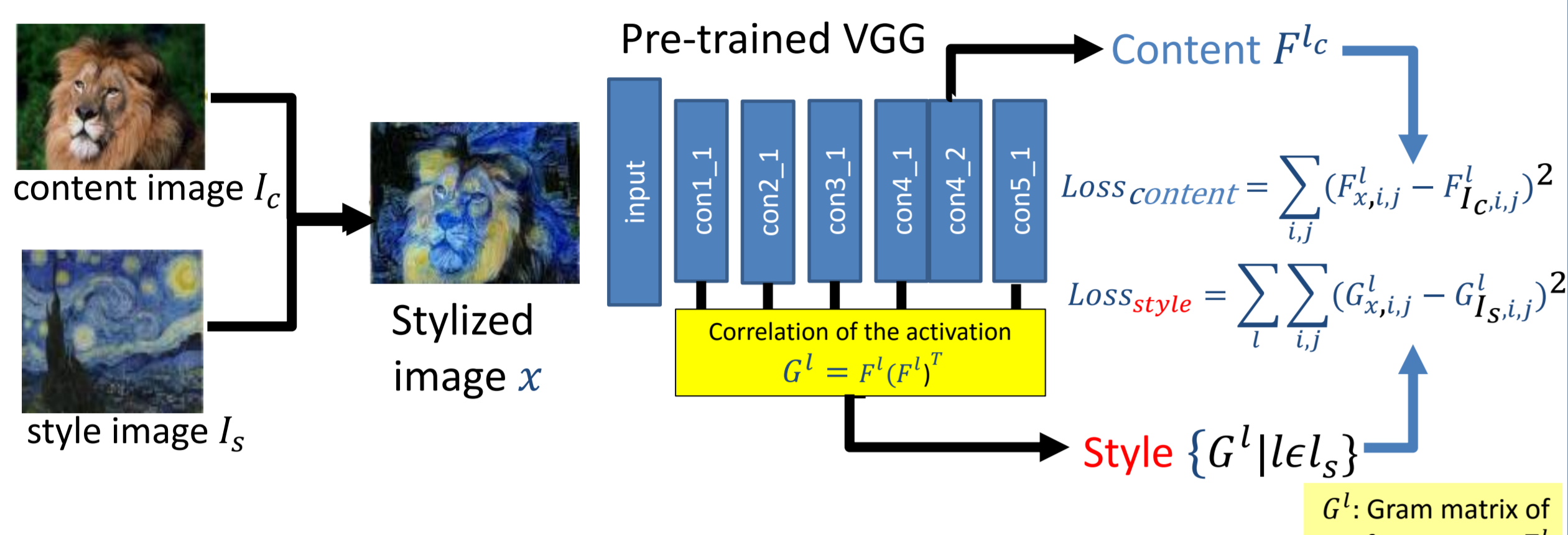
- *Optimization-based method* ➡ **time consuming** (1 min or more)

- **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
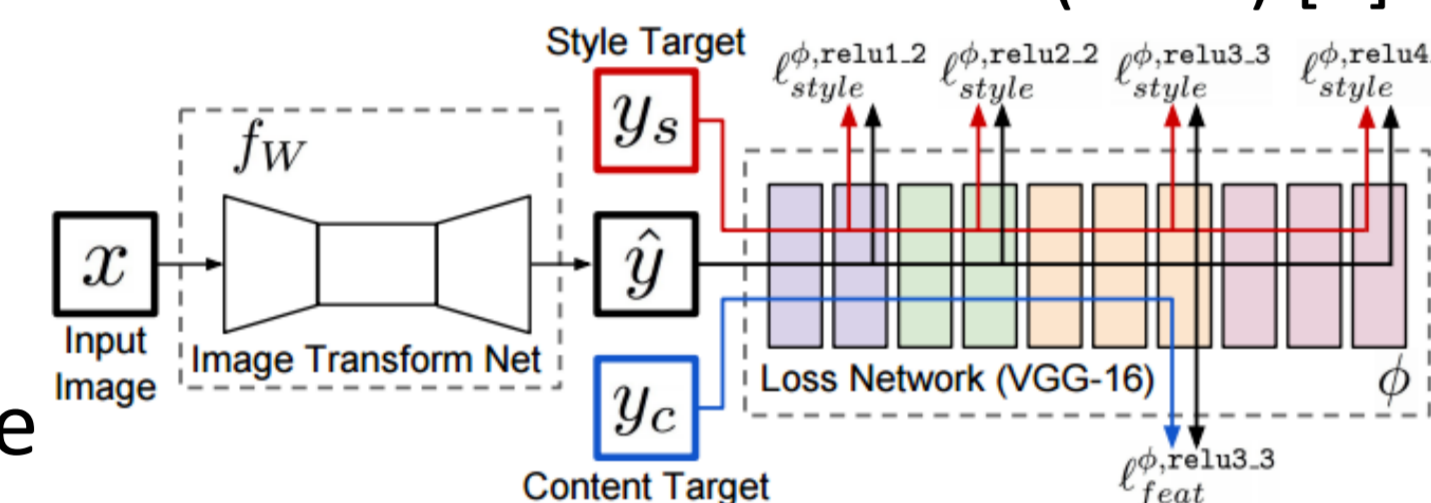
➡ **need to train an individual model for each style image**
e.g. ten styles ➡ ten models

**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**
*(an extension of a conditional fast style transfer network)*

Qualitative comparison

style content

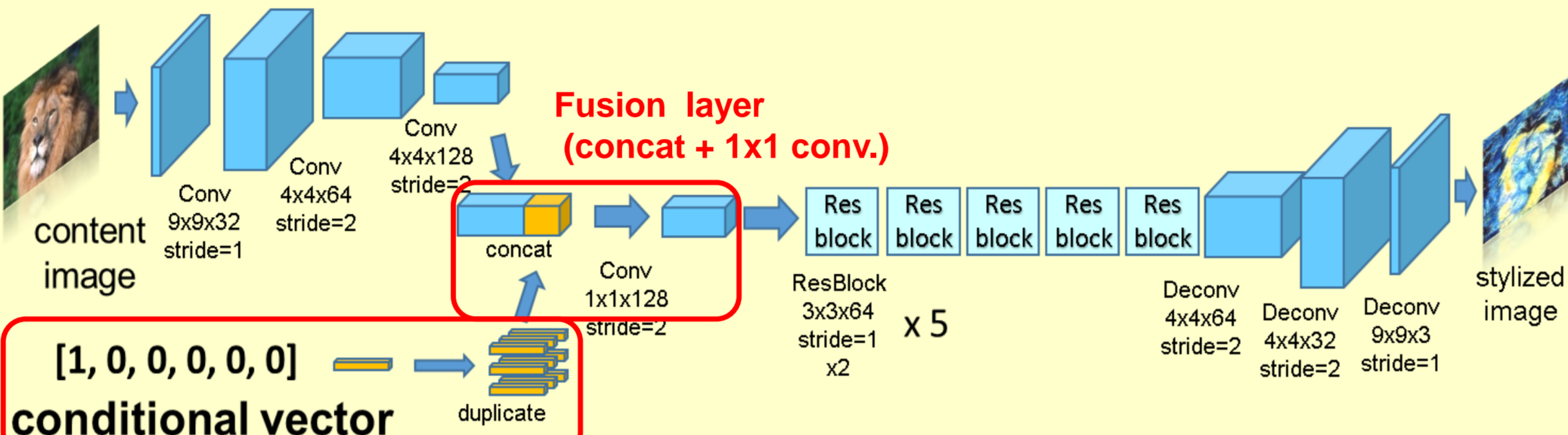Johnson   Conditional   Unseen

### 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.

---

## 2. Conditional Fast Style Transfer

- *Add a style condition input to ConvDeconv network.*
  - 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 network [B]
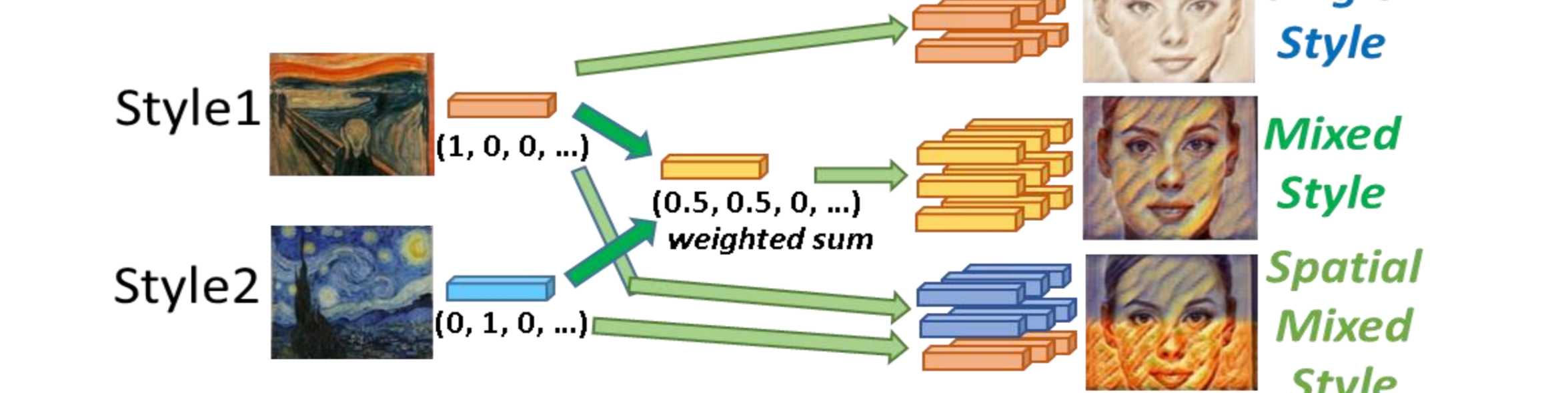    Each layer has BN and ReLU except last one.

Fusion layer (concat + 1x1 conv.)

content image

concat

[1, 0, 0, 0, 0, 0]
**conditional vector**

duplicate

Res block x 5

stylized image

**Conditional Fast Style Transfer Network**

- *Training*
  - Perceptual loss with VGG16 (same 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 :

Style1 (1, 0, 0, ...) → **Single Style**

(0.5, 0.5, 0, ...) weighted sum → **Mixed Style**

Style2 (0, 1, 0, ...) → **Spatial Mixed Style**

- *Results of a cond. style transfer with trained styles*

style content

---

## 3. Unseen Style Transfer Network

- *Add a style condition network to the cond. FST network.*
  - Confirmed that a real-value condition vector is possible.
    *Style cond. vec. can be generated by a CNN (not only hand)*
  - Propose **a style condition network** which generates a style condition vectos from a given style image directly.
  - As a result, **"Unseen Style Transfer Net"** got possible !!

content image

concat

duplicate

ResBlock x 5

stylized image

style image

**style condition network**
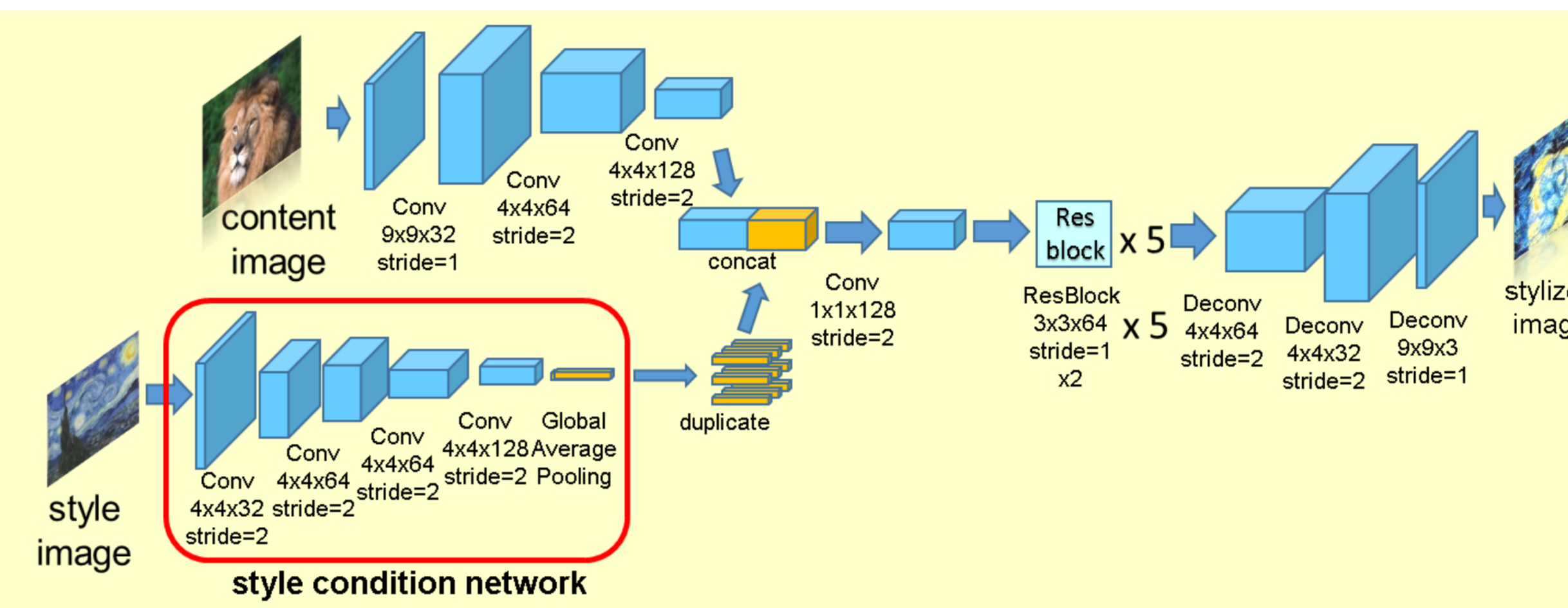
**Conditional Fast Style Transfer Network with a Style Condition Network**
↓
**Unseen Style Transfer Network**

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

- *Generating stylized images in three ways*
  - the same way as a Conditional Fast Style Transfer Network

- *Results of unseen style transfer with NOT-trained styles*
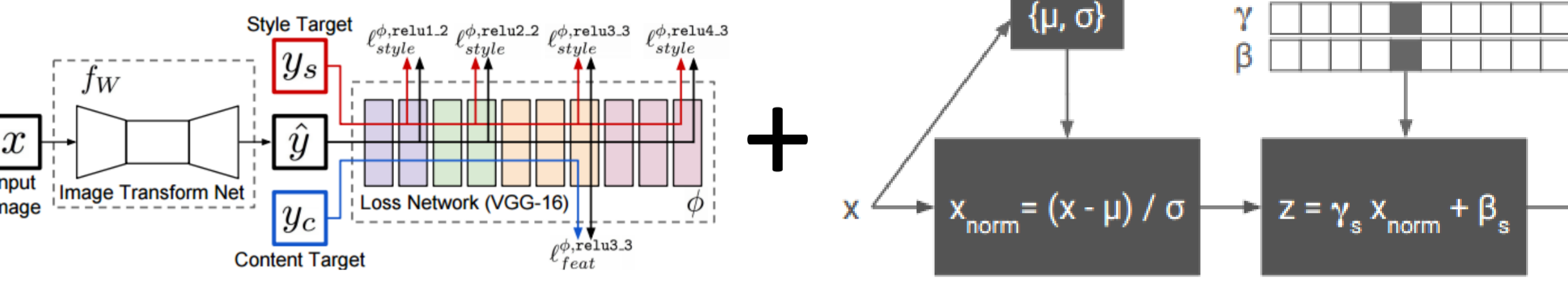
style content

**Online Demo**

Conditional style transfer
http://bit.ly/mixstyle

Unseen style transfer
http://bit.ly/unseenstyle

---

## 4. Concurrent Works *(mini survey)*

### (A) Multiple Styles

[1] V. Dumoulin et al.: **A Learned Representation of Artistic Style**, ICLR conf. tack, 2017.

$\{\mu, \sigma\}$

$x_{norm} = (x - \mu) / \sigma$     $z = \gamma_s x_{norm} + \beta_s$

**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.

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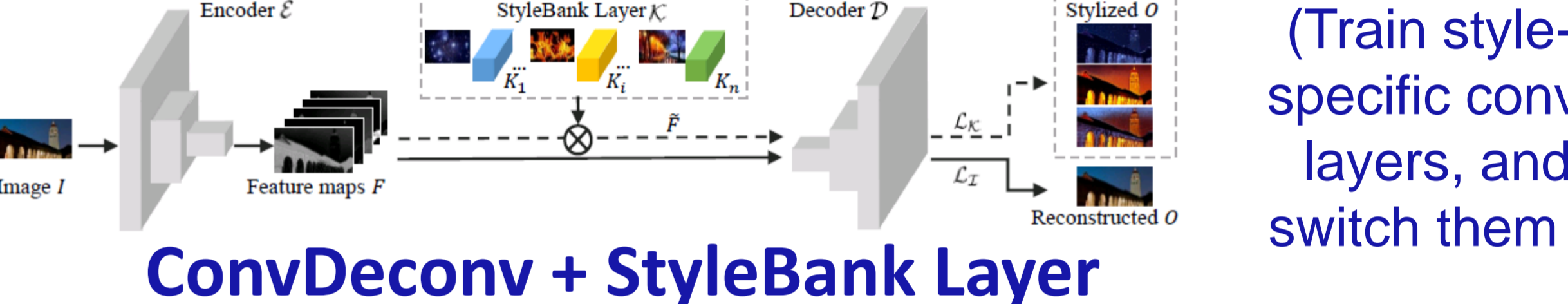
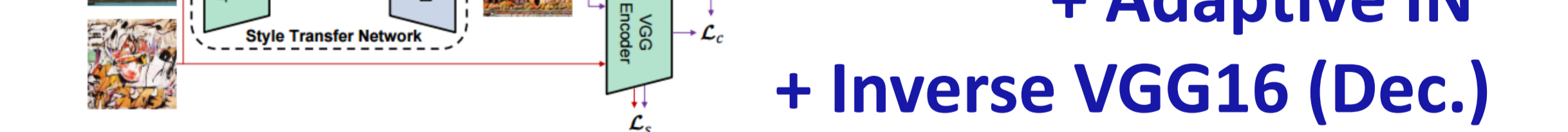**ConvDeconv + StyleBank Layer**

### (B) Arbitrary Styles

[3] T. Q. Chen et al.: **Fast Patch-based Style Transfer of Arbitrary Style**, arXiv: 1612.04337, 2017.

**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.
(inspired by Y. Li et al.: **Demystifying Neural Style Transfer**, arXiv: 1701.01036, 2017.)

**VGG16 (Enc.) + Adaptive IN + Inverse VGG16 (Dec.)**

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

similar to ours
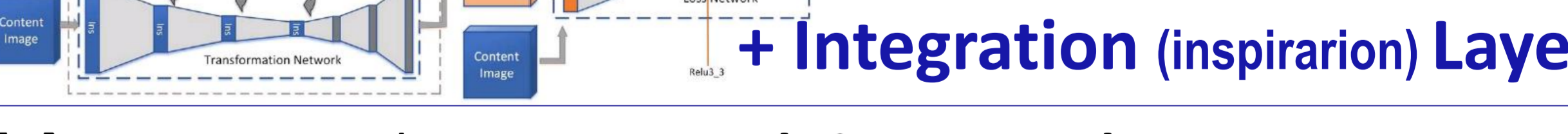
**ConvDeconv + Style feature net (VGG16) + Integration (inspirarion) Layer**

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

natural extension of [1]

**ConvDeconv + Parameter net + Dynamic IN**