Applications of GANs

- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
- Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
- Generative Adversarial Text to Image Synthesis

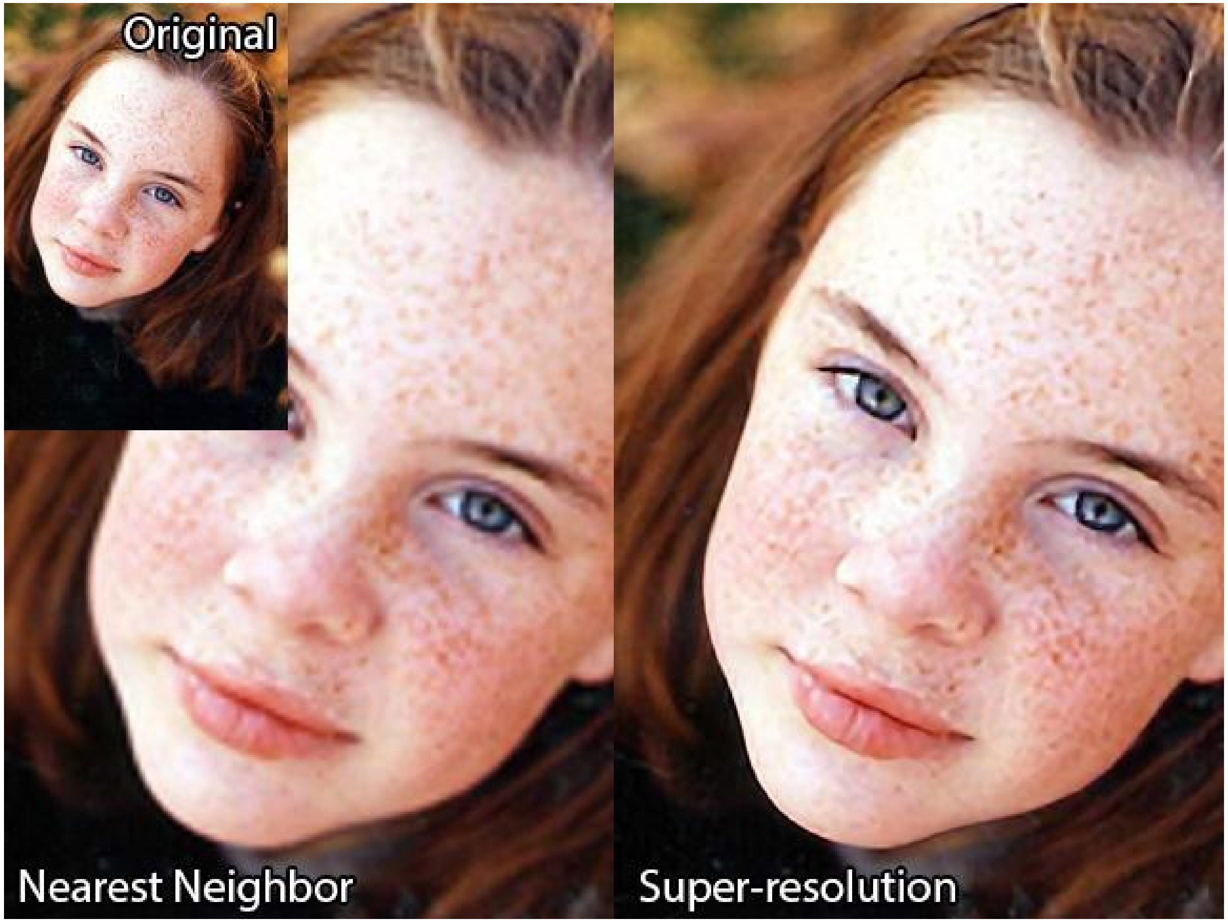
Using GANs for Single Image Super-Resolution

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi

Problem

How do we get a high resolution (HR) image from just one (LR) lower resolution image?

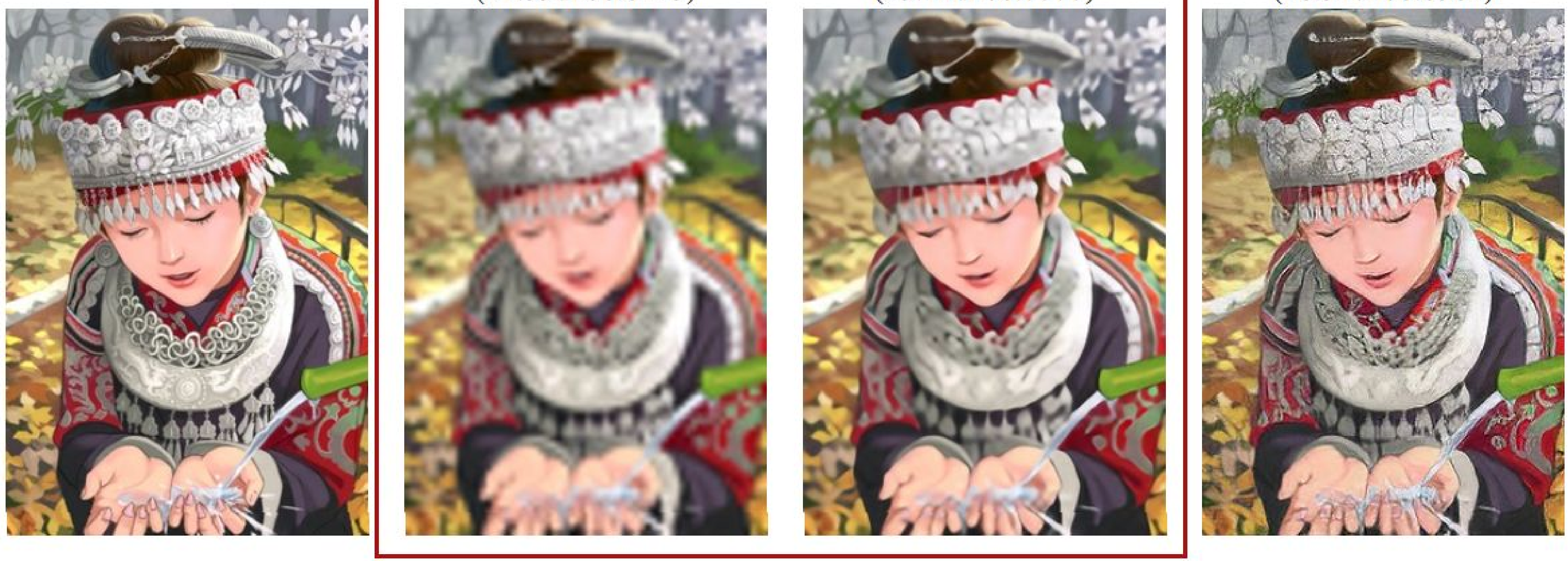
Answer: We use super-resolution (SR) techniques.



http://www.extremetech.com/wp-content/uploads/2012/07/super-resolution-freckles.jpg

Previous Attempts

original | bicubic (21.59dB/0.6423) | SRResNet (23.44dB/0.7777) | SRGAN (20.34dB/0.6562)



SRGAN

original | bicubic (21.59dB/0.6423) | SRResNet (23.44dB/0.7777) | SRGAN (20.34dB/0.6562)

SRGAN - Generator

- G: generator that takes a low-res image I^{LR} and outputs its high-res counterpart I^{SR}
- θ_G: parameters of G, $\{W_{1:L}, b_{1:L}\}$
- l^{SR}: loss function measures the difference between the 2 high-res images

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR})$$

SRGAN - Discriminator

- D: discriminator that classifies whether a high-res image is I^{HR} or I^{SR}
- θ_D: parameters of D

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))]$$

SRGAN - Perceptual Loss Function

Loss is calculated as weighted combination of:

➔ Content loss

➔ Adversarial loss

➔ Regularization loss

SRGAN - Content Loss

Instead of MSE, use loss function based on ReLU layers of pre-trained VGG network. Ensures similarity of content.

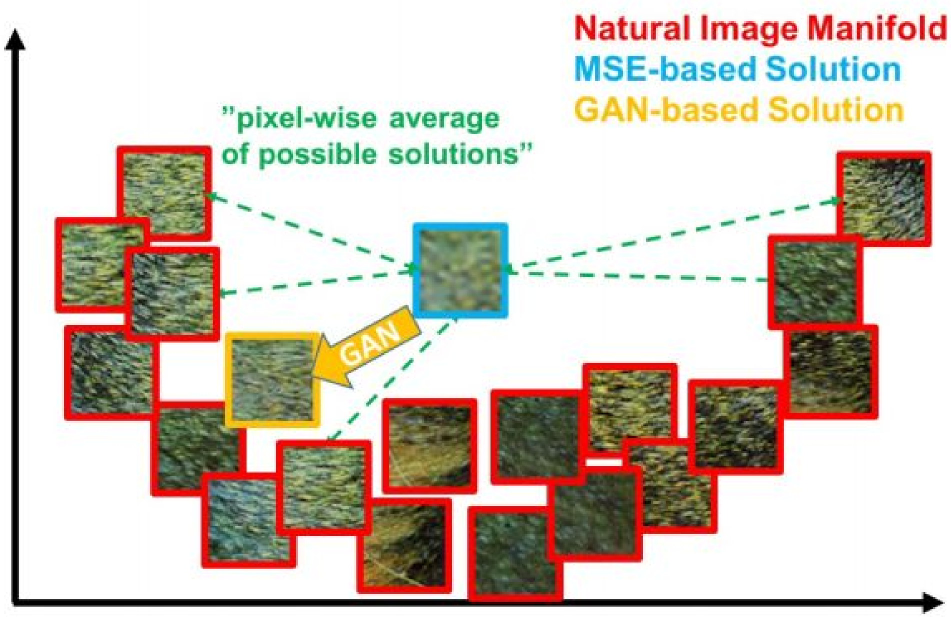
- $\boldsymbol{\Phi}_{i,j}$: feature map of j^{th} convolution before i^{th} maxpooling
- $W_{i,j}$ and $H_{i,j}$: dimensions of feature maps in the VGG

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

SRGAN - Adversarial Loss

Encourages network to favour images that reside in manifold of natural images.

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$



SRGAN - Regularization Loss

Encourages spatially coherent solutions based on total variations.

$$l_{TV}^{SR} = \frac{1}{r^2WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} ||\nabla G_{\theta_G}(I^{LR})_{x,y}||$$

SRGAN - Examples

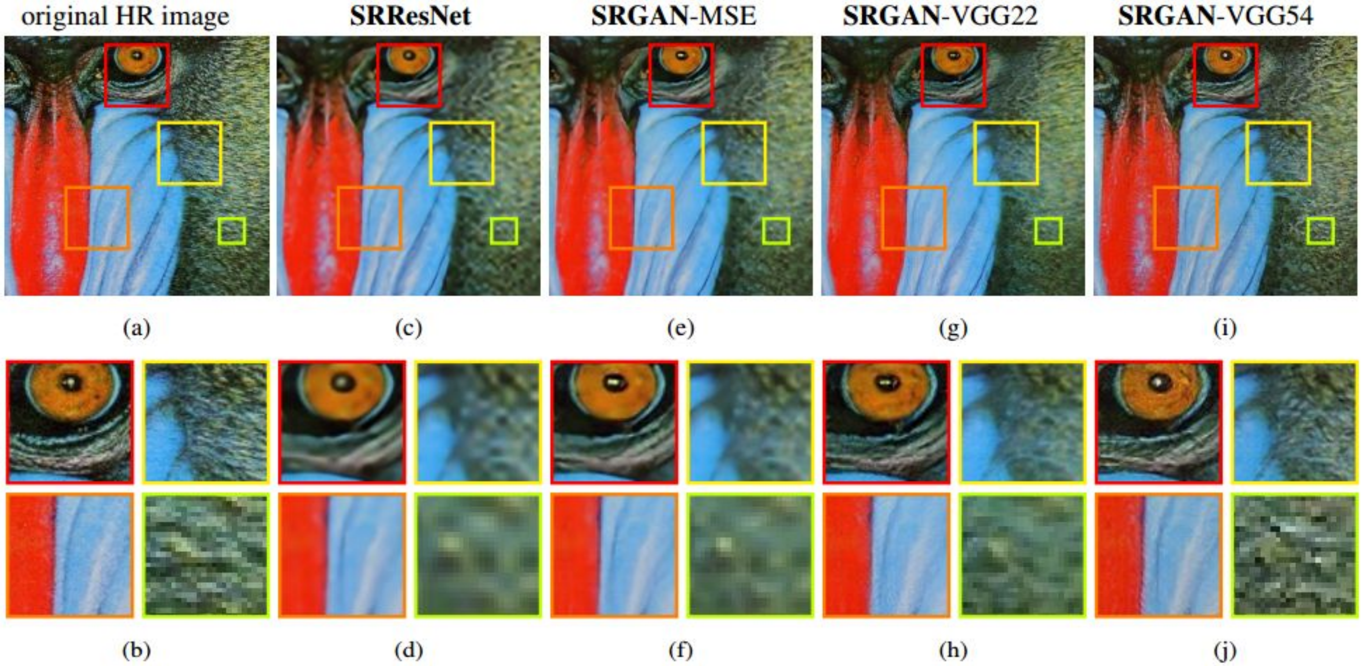
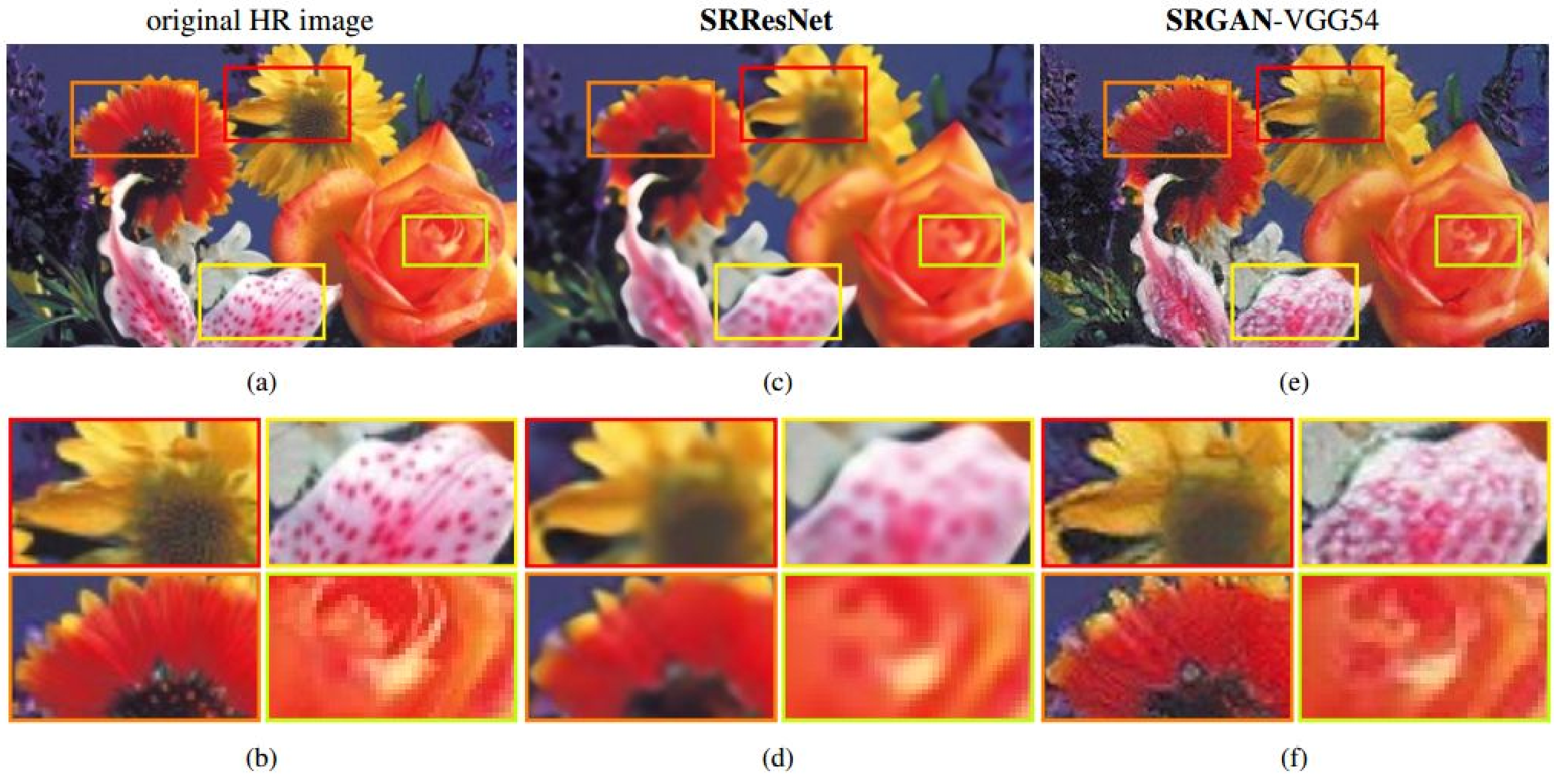


Figure 5: Reference HR image (left: a,b) with corresponding SRResNet (middle left: c,d), SRGAN-MSE (middle: e,f), SRGAN-VGG2.2 (middle right: g,h) and SRGAN-VGG54 (right: i,j) reconstruction results.

SRGAN - Examples



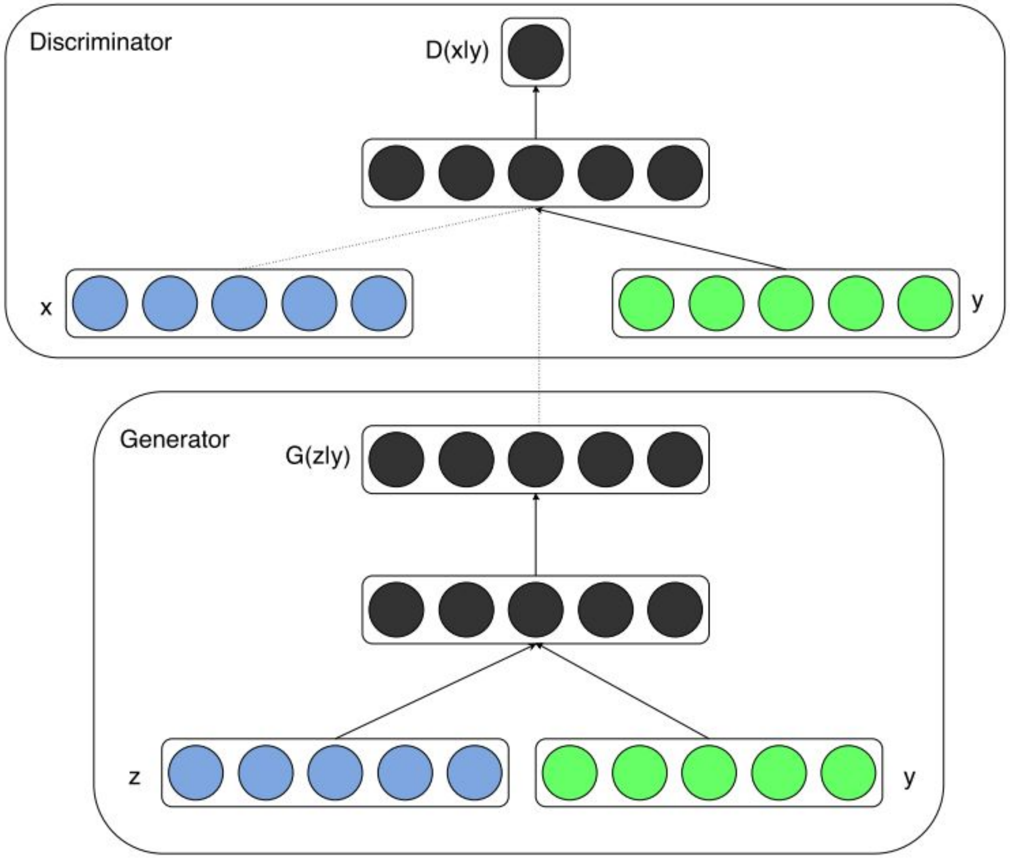
Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

Work by Emily Denton, Soumith Chintala, Arthur Szlam, Rob Fergus

Short Background

Conditional Generative Adversarial Nets (CGAN)

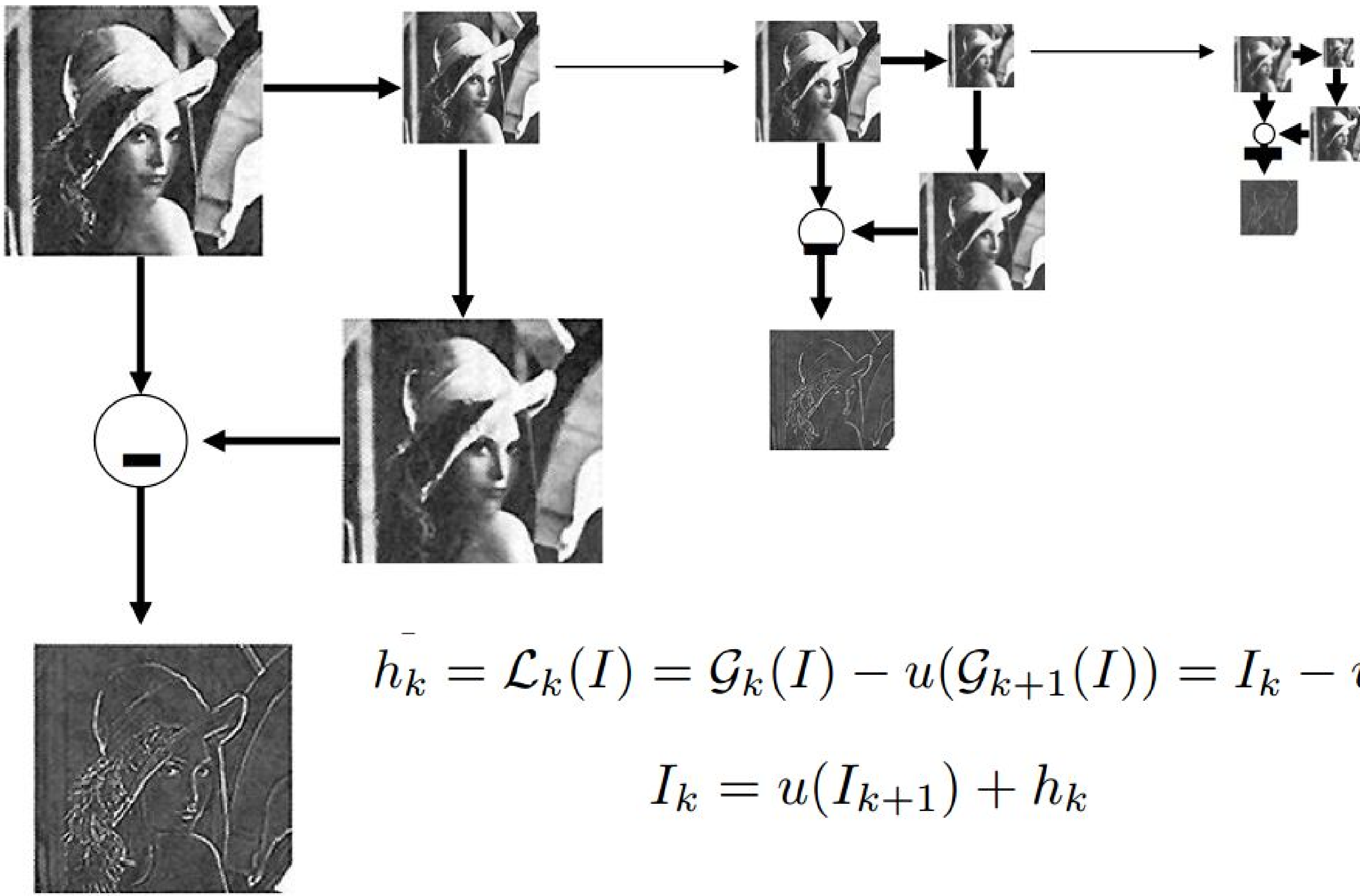
Mirza and Osindero (2014)



GAN $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$

CGAN $\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\underline{\boldsymbol{y}})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\underline{\boldsymbol{y}})))]$

Laplacian pyramid

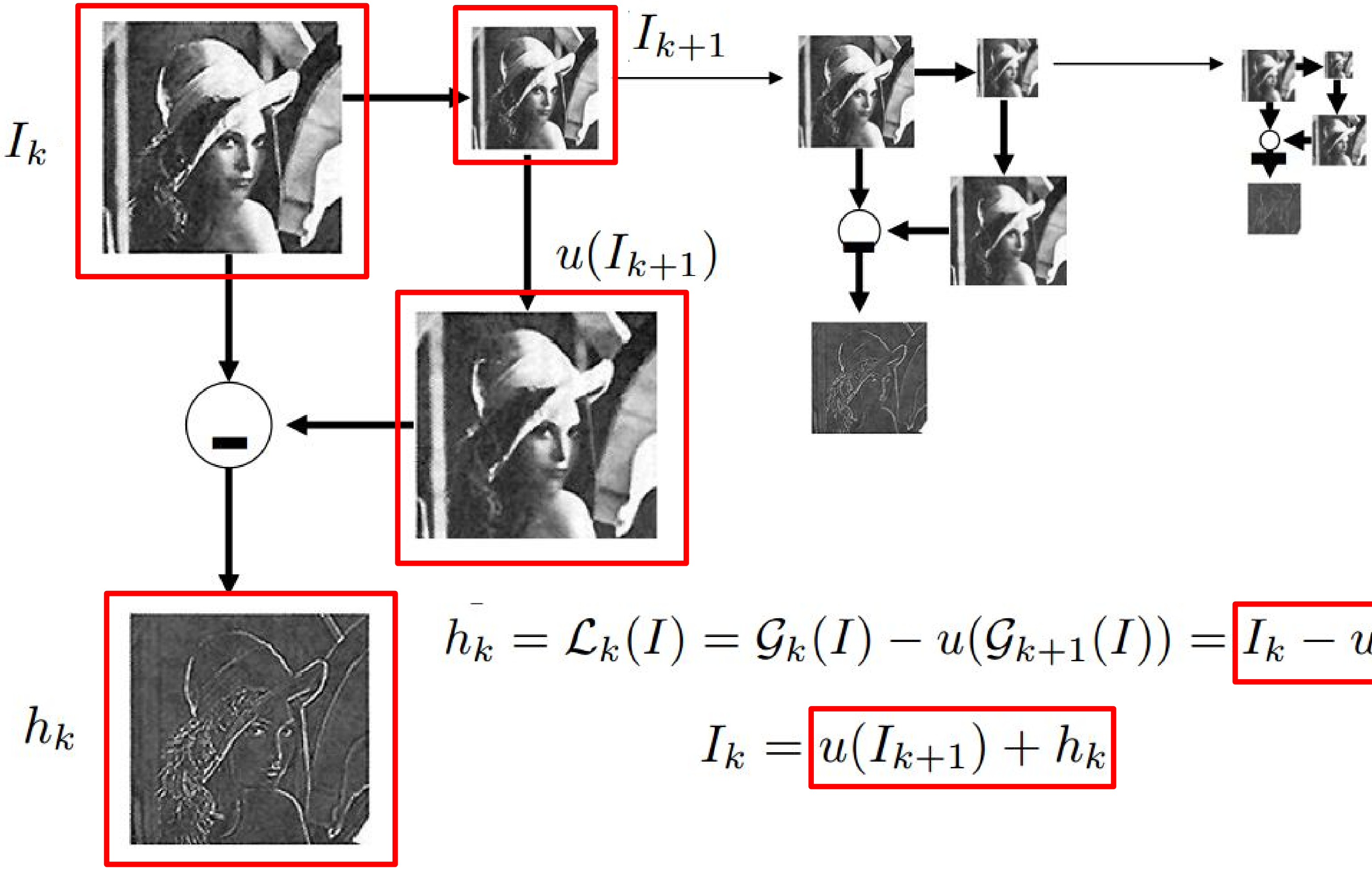


$$h_k = \mathcal{L}_k(I) = \mathcal{G}_k(I) - u(\mathcal{G}_{k+1}(I)) = I_k - u(I_{k+1})$$

$$I_k = u(I_{k+1}) + h_k$$

Burt and Adelson (1983)

Laplacian pyramid



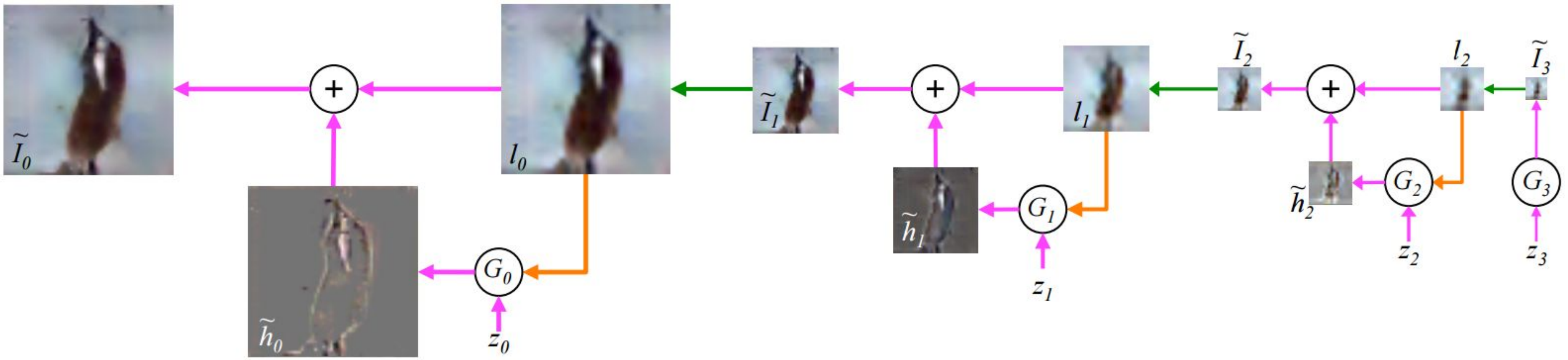
$$h_k = \mathcal{L}_k(I) = \mathcal{G}_k(I) - u(\mathcal{G}_{k+1}(I)) = \boxed{I_k - u(I_{k+1})}$$

$$I_k = \boxed{u(I_{k+1}) + h_k}$$

Burt and Adelson (1983)

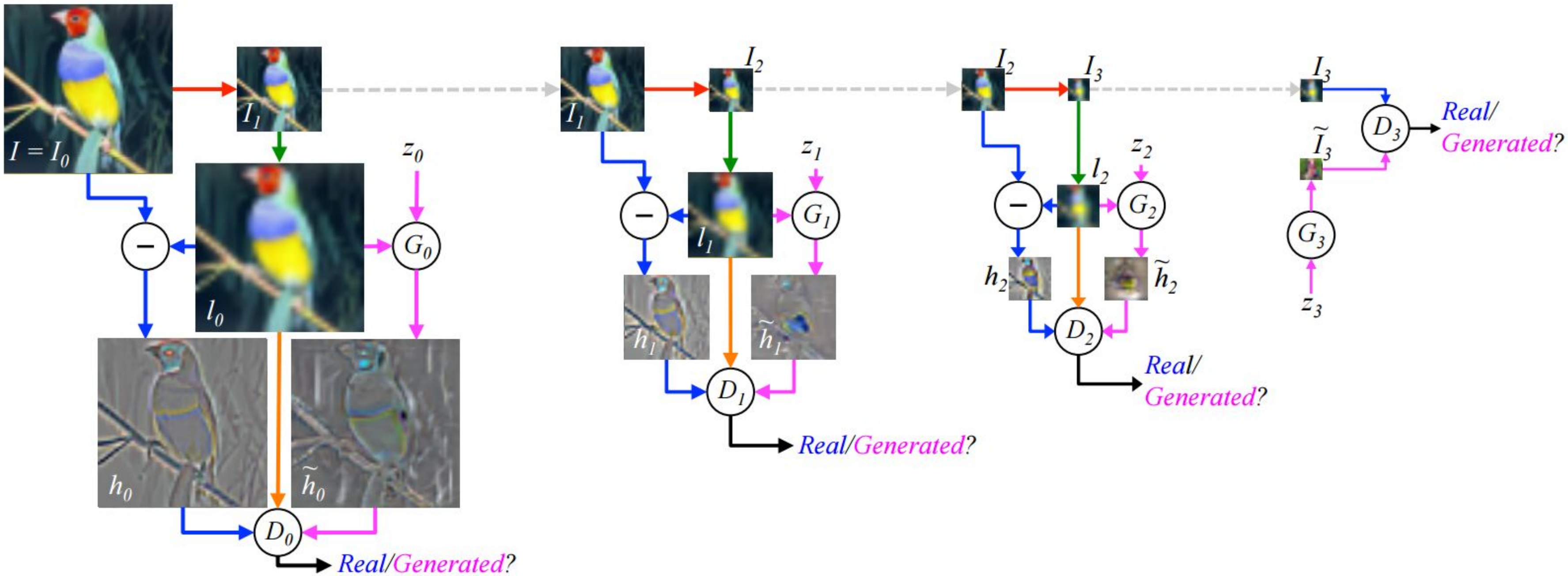
Laplacian Pyramid Generative Adversarial Network (LAPGAN)

Image Generation



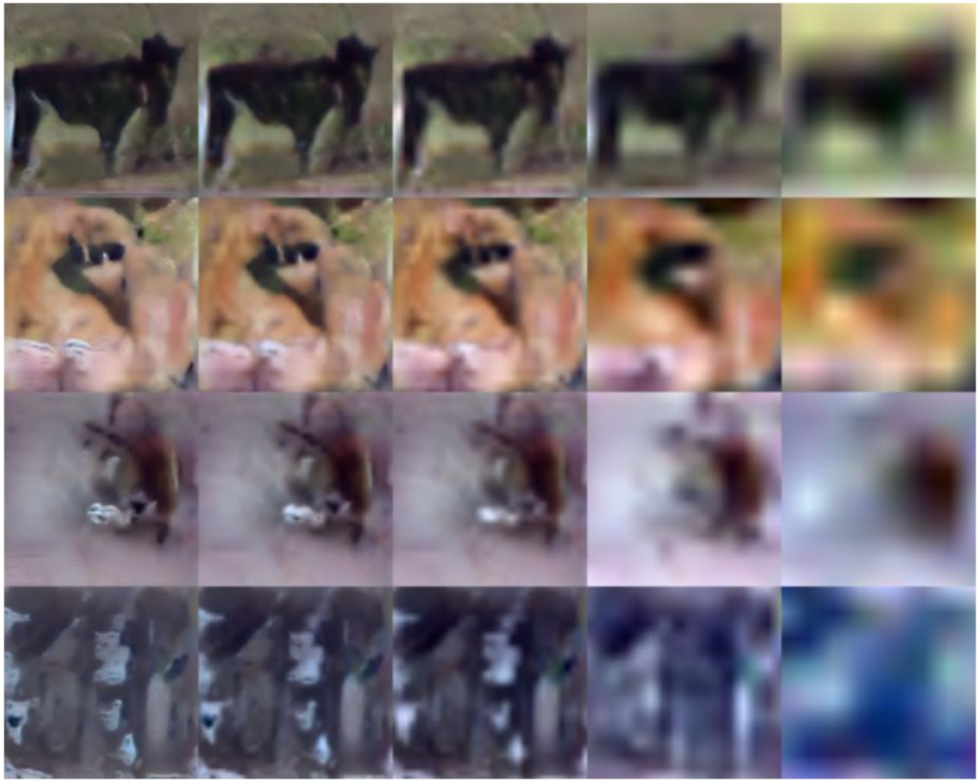
$$\tilde{I}_k = u(\tilde{I}_{k+1}) + \tilde{h}_k = u(\tilde{I}_{k+1}) + G_k(z_k, u(\tilde{I}_{k+1}))$$

Training

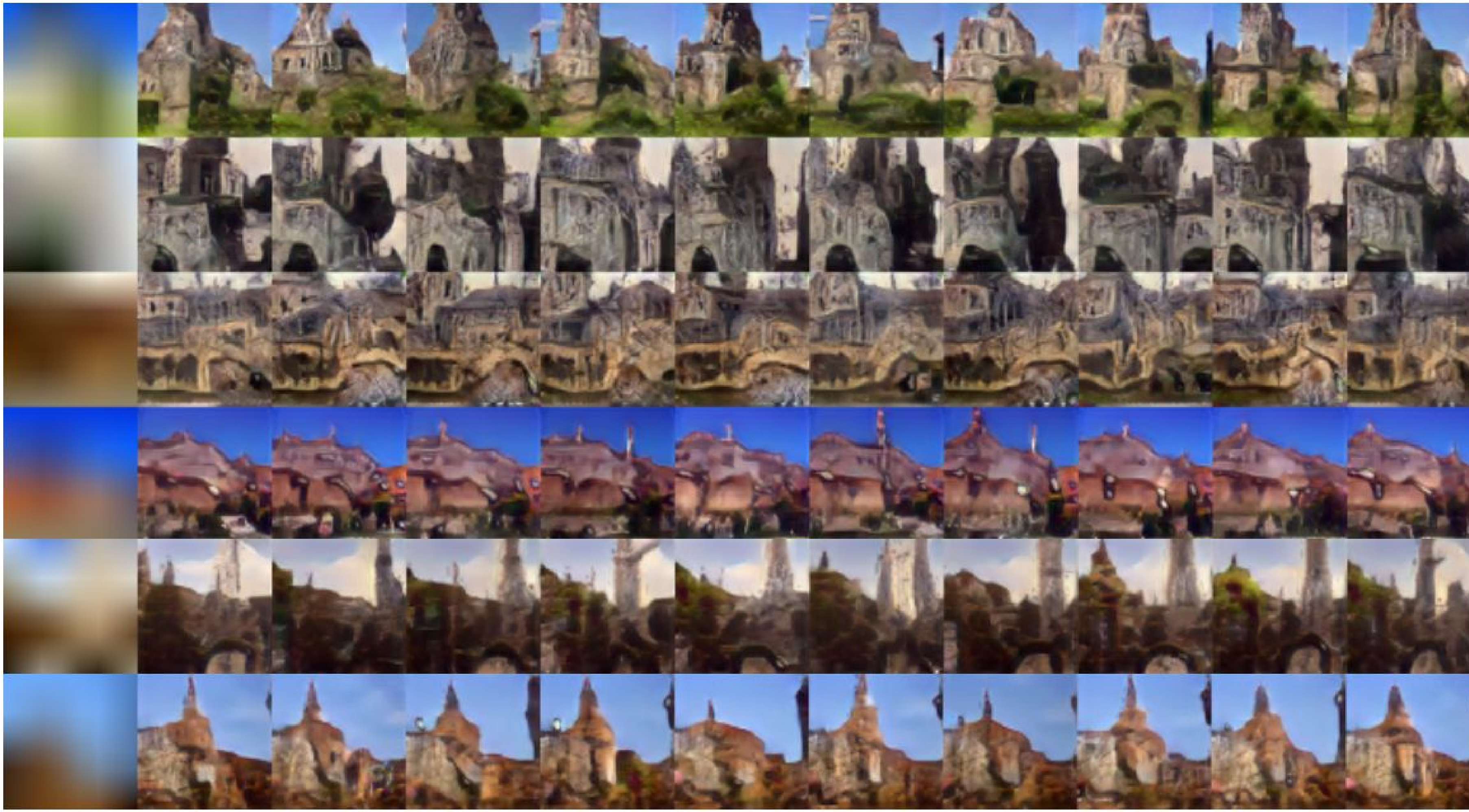


$I = I_0$
I_1
l_0
z_0
G_0
h_0
$\tilde{h}_0$
D_0
Real/Generated?
I_1
I_2
l_1
z_1
G_1
h_1
$\tilde{h}_1$
D_1
Real/Generated?
I_2
I_3
l_2
z_2
G_2
h_2
$\tilde{h}_2$
D_2
Real/
Generated?
I_3
$\tilde{I}_3$
D_3
G_3
z_3
Real/
Generated?

Generation: Coarse to fine

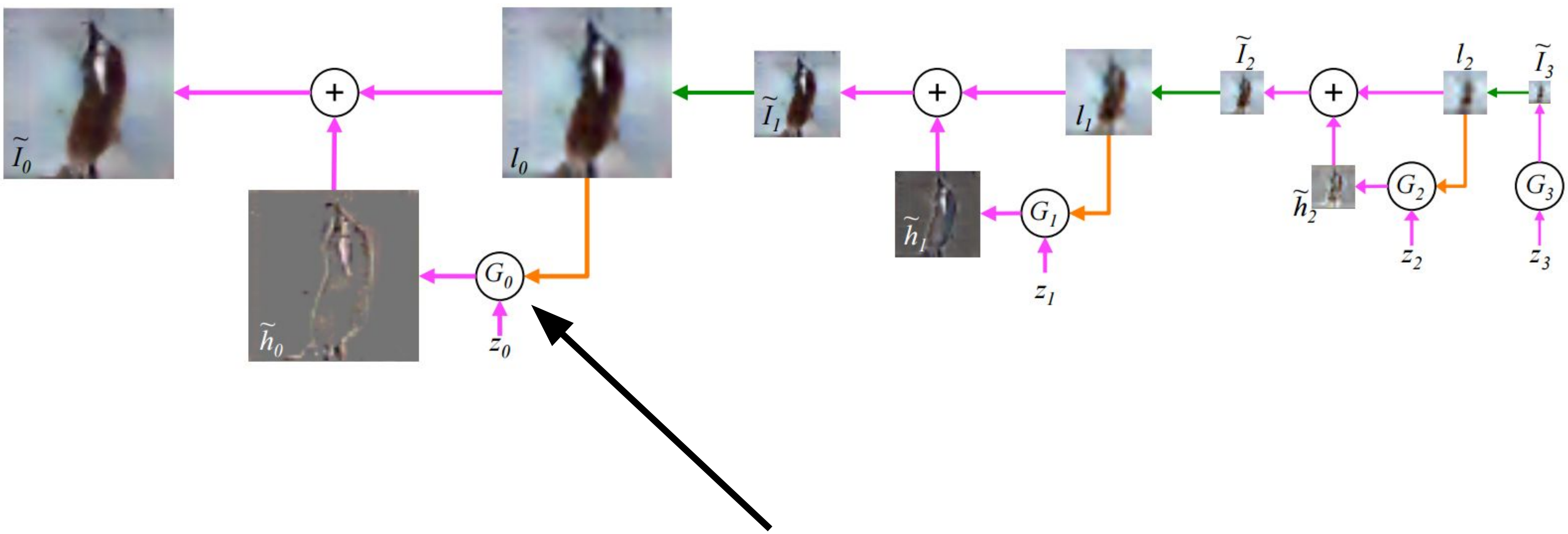


Different draws, starting from the same initial 4x4 image



Some thoughts on the method

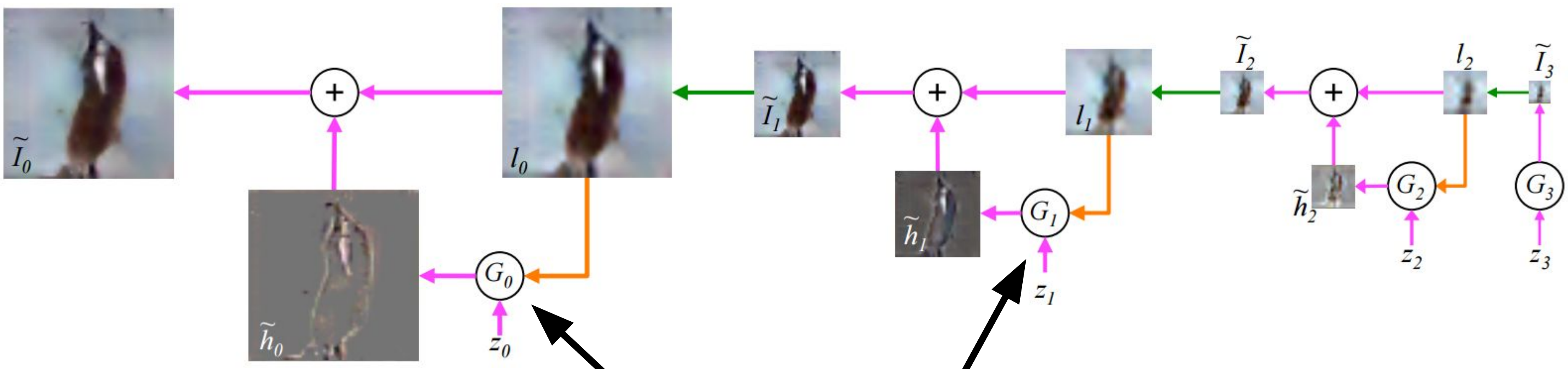
- The Laplacian Pyramid Framework is independent of the Generative Model



Possible to use a completely different model like Pixel RNN

Some thoughts on the method

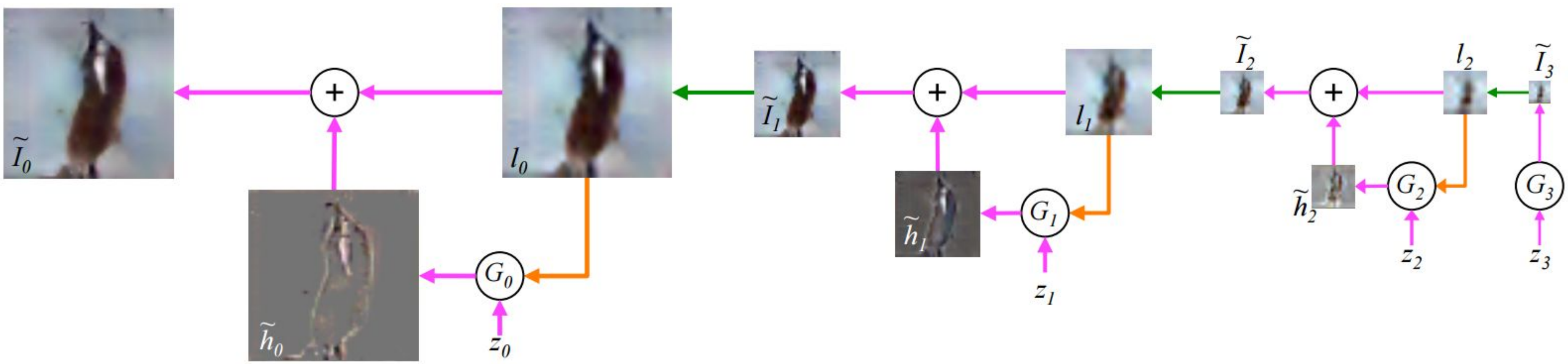
- The Generative Models at each step can be totally different!



These can also be different models!

Some thoughts on the method

- The Generative Models at each step can be totally different!



High resolution architecture

Low resolution architecture

Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

Author's code available at: https://github.com/reedscot/icml2016

Motivation

Current deep learning models enable us to...

- ➢ Learn feature representations of images & text
- ➢ Generate realistic images & text

☑ pull out images based on captions

☑ generate descriptions based on images

☑ answer questions about image content

"Two pizzas sitting on top of a stove top oven"

Problem - Multimodal distribution

- Many plausible image can be associated with one single text description
- Previous attempt uses Variational Recurrent Autoencoders to generate image from text caption but the images were not realistic enough. (Mansimov et al. 2016)

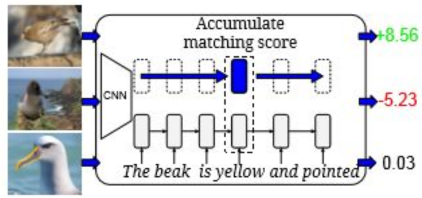
What GANs can do

- CGAN: Use side information (eg. classes) to guide the learning process
- Minimax game: Adaptive loss function

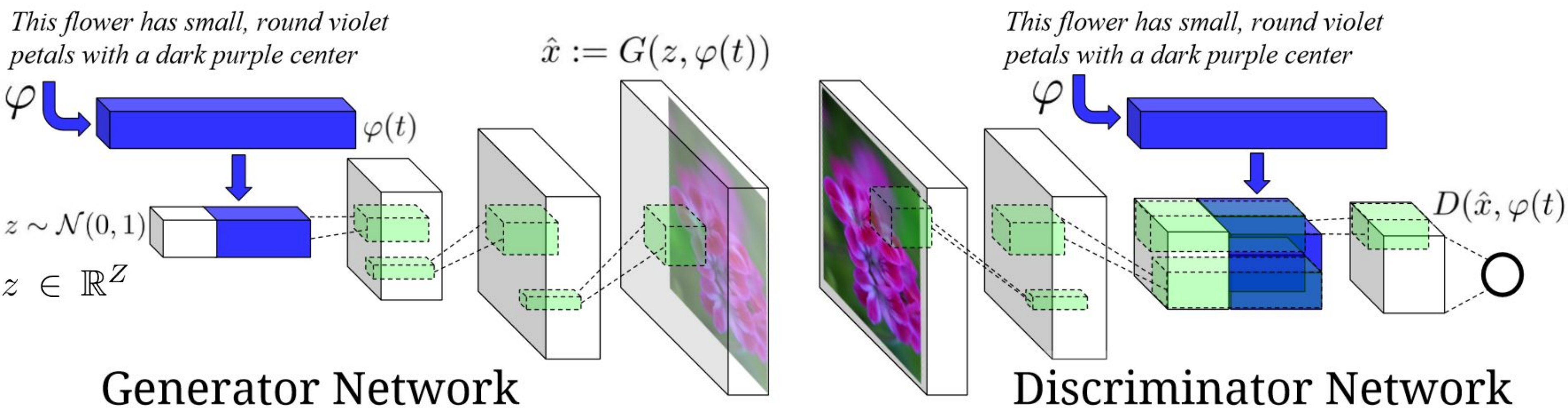
➢ Multi-modality is a very well suited property for GANs to learn.

The Model - Basic CGAN

Pre-trained char-CNN-RNN



Learns a compatibility function of images and text -> joint embedding



$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_z(z)}[\log(1 - D(G(z)))]$$

The Model - Variations

<u>GAN-CLS</u>

In order to distinguish different error sources:

Present to the discriminator network **<u>3</u>** different types of input. (instead of 2)

Algorithm

Input: minibatch images x, matching text t, mis-matching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

The Model - Variations cont.

GAN-INT

In order to generalize the output of G:

Interpolate between training set embeddings to generate new text and hence fill the gaps on the image data manifold.

Updated Equation

$$\min_G \max_D V(D,G) =$$

$$= \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)]$$

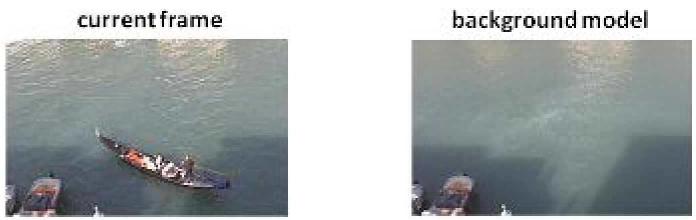
$$+ \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z)))] +$$

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1 - D(G(z, \beta t_1 + (1-\beta)t_2)))]$$

{fake image, fake text}

GAN-INT-CLS: Combination of both previous variations

Disentangling



- Style is background, position & orientation of the object, etc.
- Content is shape, size & colour of the object, etc.

- Introduce S(x), a style encoder with a squared loss function:

$$\mathcal{L}_{style} = \mathbb{E}_{t,z\sim\mathcal{N}(0,1)}||z - S(G(z, \varphi(t)))||_2^2$$

- Useful in generalization: encoding style and content separately allows for different new combinations

Training - Data (separated into <u>class-disjoint</u> train and test sets)

Caltech-UCSD Birds

Caption	Image
this vibrant red bird has a pointed black beak	
this bird is yellowish orange with black wings	
the bright blue bird has a white colored belly	

MS COCO

Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

Oxford Flowers

Caption	Image
this flower has white petals and a yellow stamen	
the center is yellow surrounded by wavy dark purple petals	
this flower has lots of small round pink petals	

Training – Results: Flower & Bird

Training – Results: MS COCO

a large blue octopus kite flies above the people having fun at the beach.

a toilet in a small room with a window and unfinished walls.

a man in a wet suit riding a surfboard on a wave.

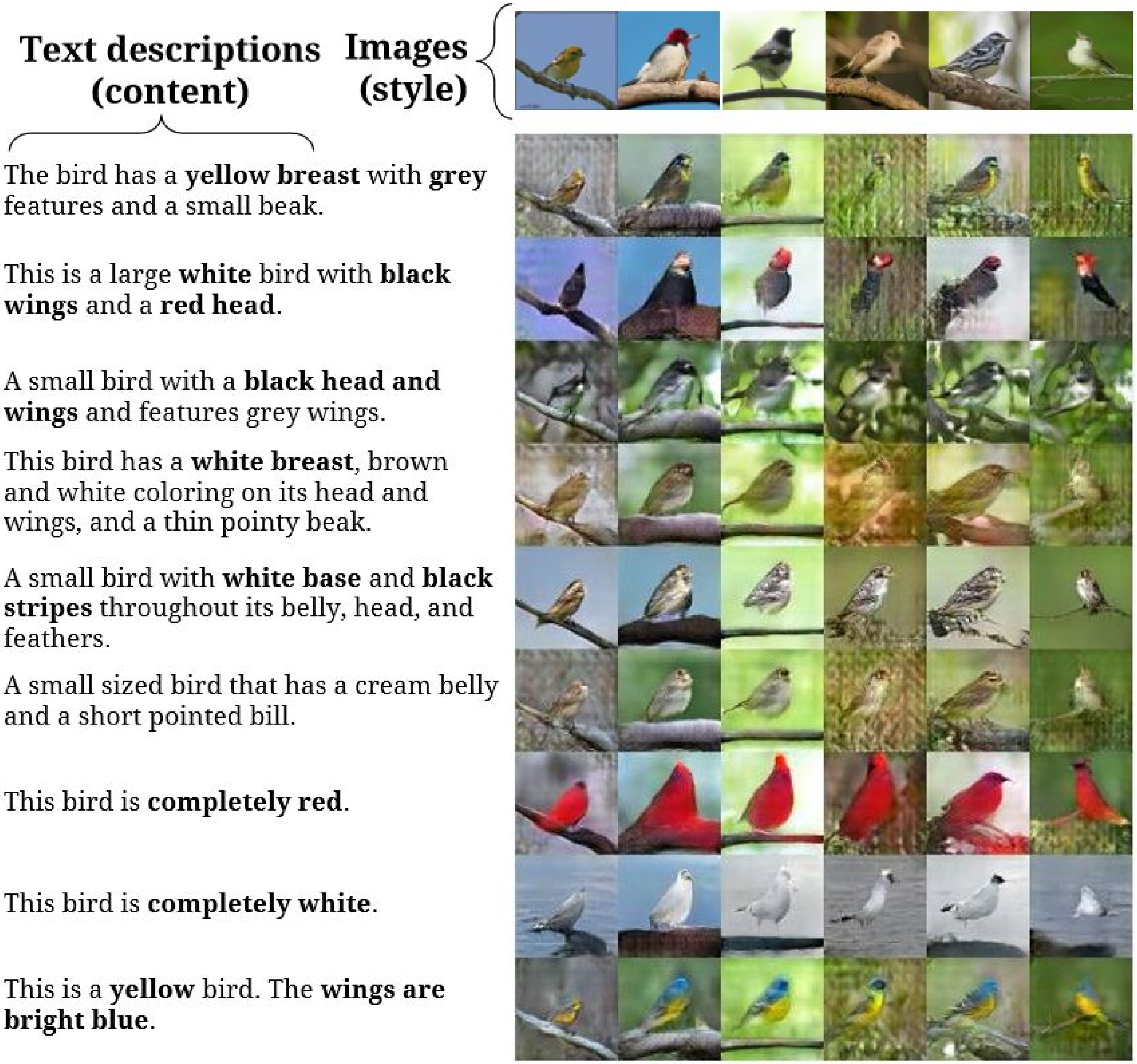
Mansimov et al.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.

Training – Results Style disentangling



$$s \leftarrow S(x)$$

$$\hat{x} \leftarrow G(s, \varphi(t))$$

Thoughts on the paper

- Image quality

- Generalization

- Future work