GAN and RNN: Crash Course

Alex
Safety Sensing and Control Department
Mechanical and Mechatronics Systems Research Laboratories
Industrial Technology Research Institute

Overview

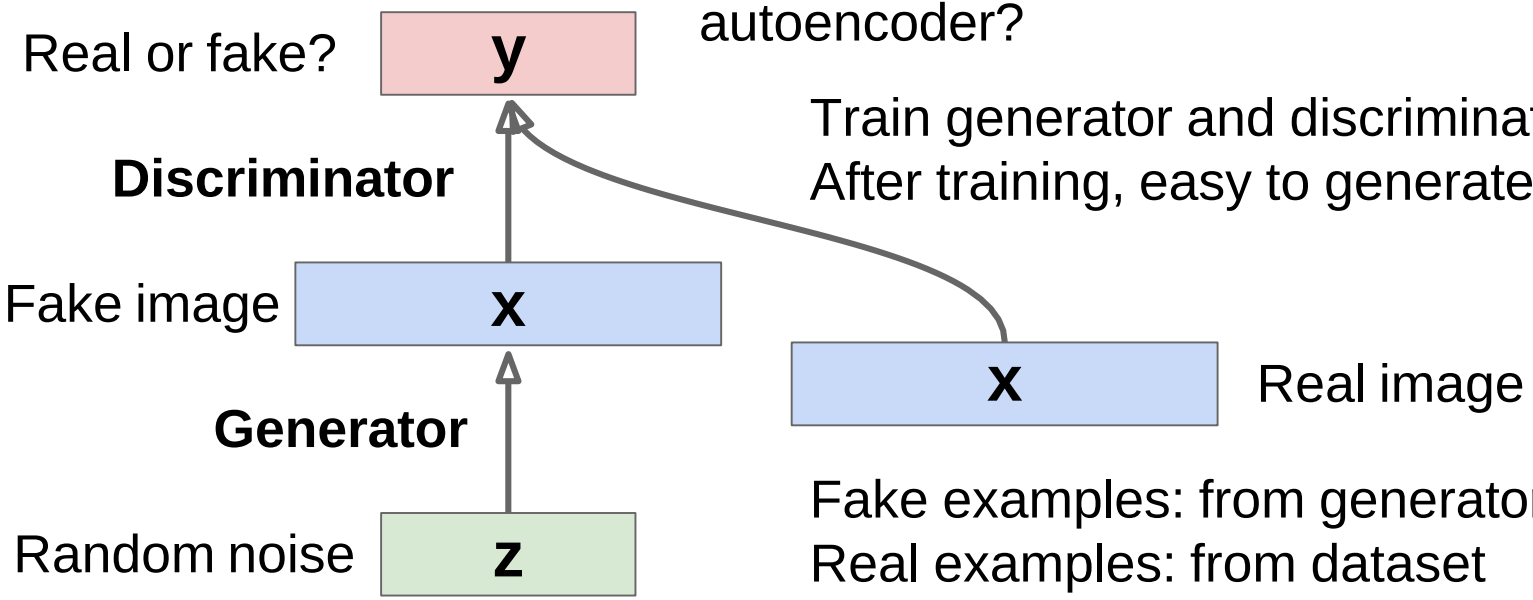
- GAN
- RNN

Generative Adversarial Networks

Adversarial training (also called GAN for Generative Adversarial Networks), and the variations that are now being proposed, is the most interesting idea in the last 10 years in ML, in my opinion.

Yann LeCun, Director of Facebook AI Research

Generative Adversarial Nets



Can we generate images with the idea of autoencoder?

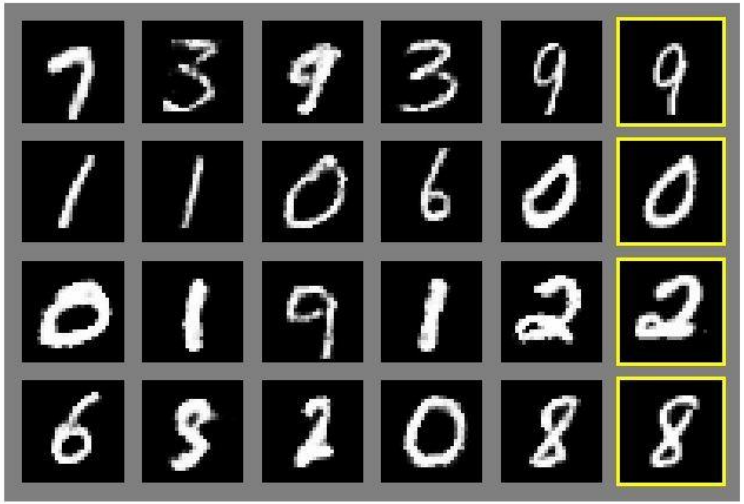
Train generator and discriminator jointly
After training, easy to generate images

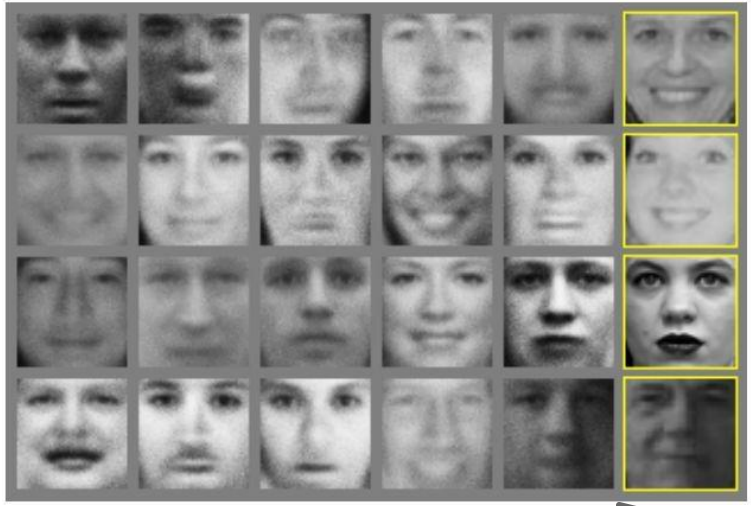
Fake examples: from generator
Real examples: from dataset

Goodfellow et al, “Generative Adversarial Nets”, NIPS 2014

Generative Adversarial Nets

Generated samples



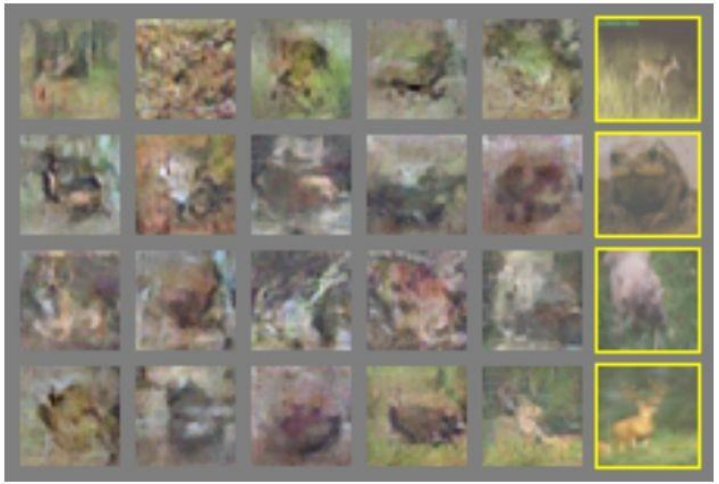


Nearest neighbor from training set

Goodfellow et al, “Generative Adversarial Nets”, NIPS 2014

Generative Adversarial Nets

Generated samples (CIFAR-10)



Nearest neighbor from training set

Goodfellow et al, “Generative Adversarial Nets”, NIPS 2014

Can we expect outputs to be the same “class”?

In supervised learning approach:

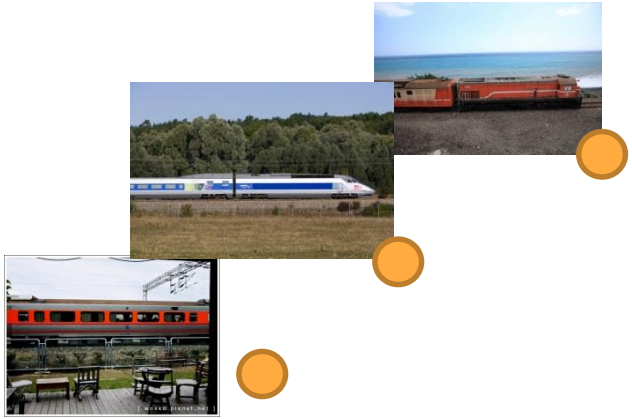
c

Image x

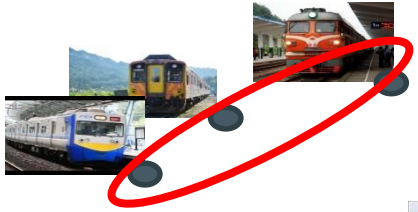
(a point, not a distribution)

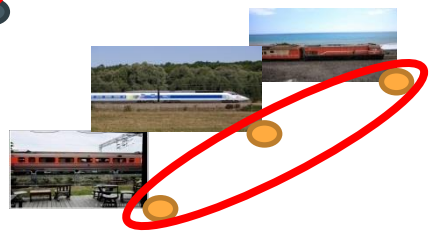
Text: “train”

NN output



Conditional GAN





Learn to approximate $P(x|\hat{c})$

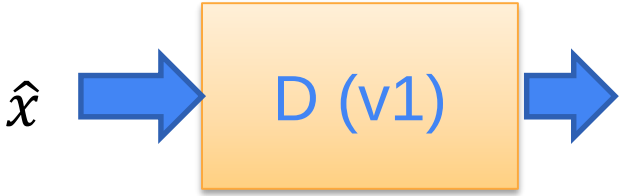
Training data: $(\hat{c}, \hat{x})$

condition $\hat{c}$

Prior distribution z

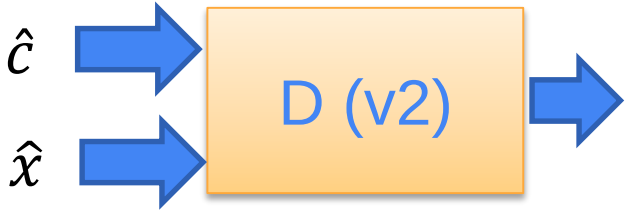
x

$(\hat{c}, x = G(\hat{c}))$ classified as positive



scalar

Can generate x not related to c



scalar

Positive example: $(\hat{c}, \hat{x})$

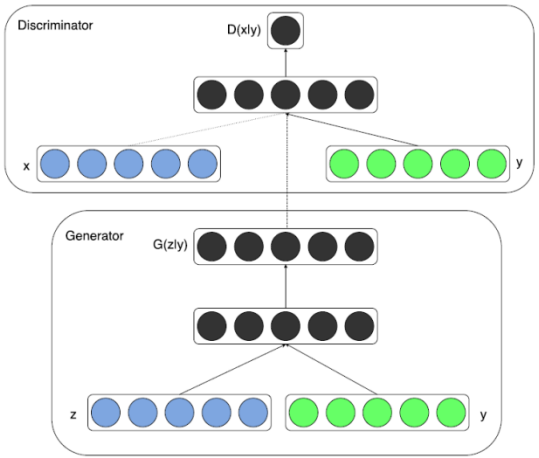
Negative example: $(\hat{c}, G(\hat{c})), (\widehat{c'}, \hat{x})$

Conditional generative adversarial networks (CGAN)

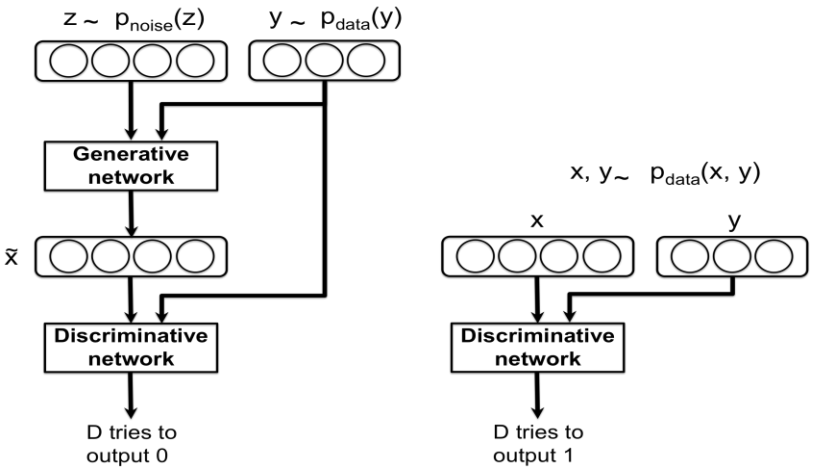
Can we expect outputs to be the same "class"?

Condition generation on additional info y (e.g. class label, another image)

D has to determine if samples are realistic given y



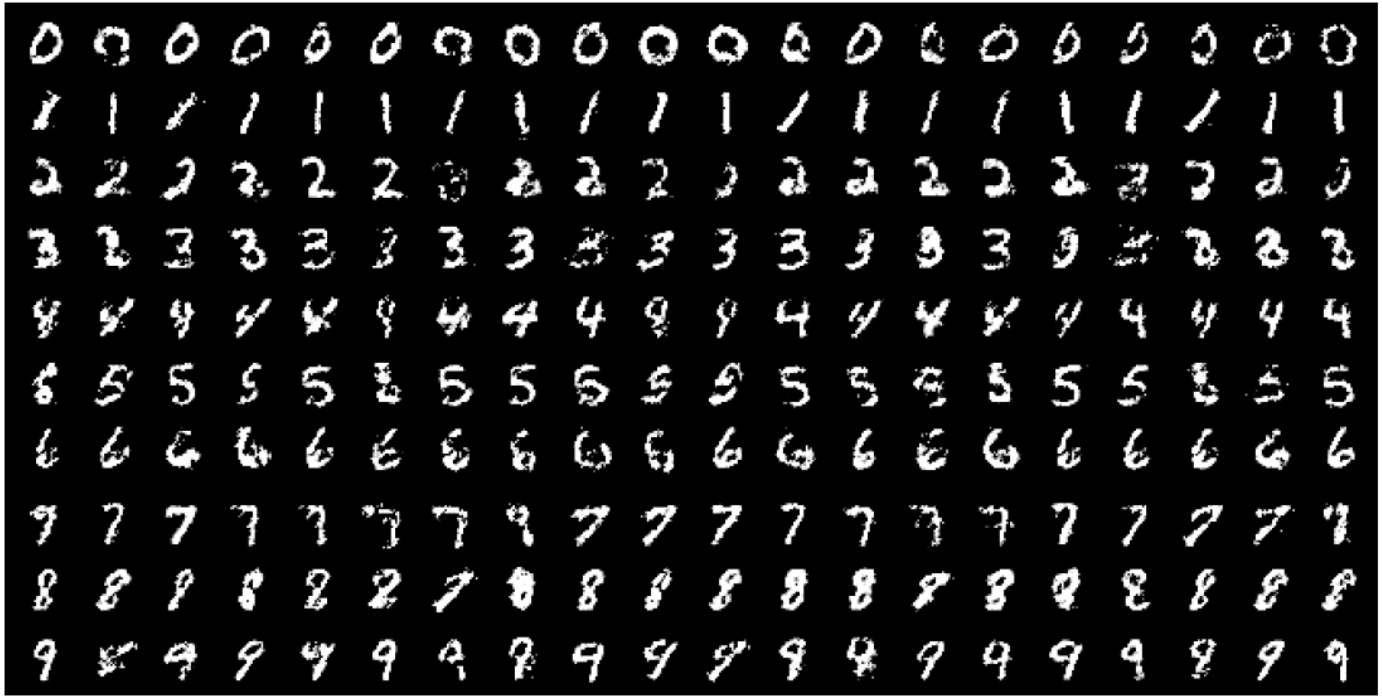
Original block diagram



Another interpretation of CGAN

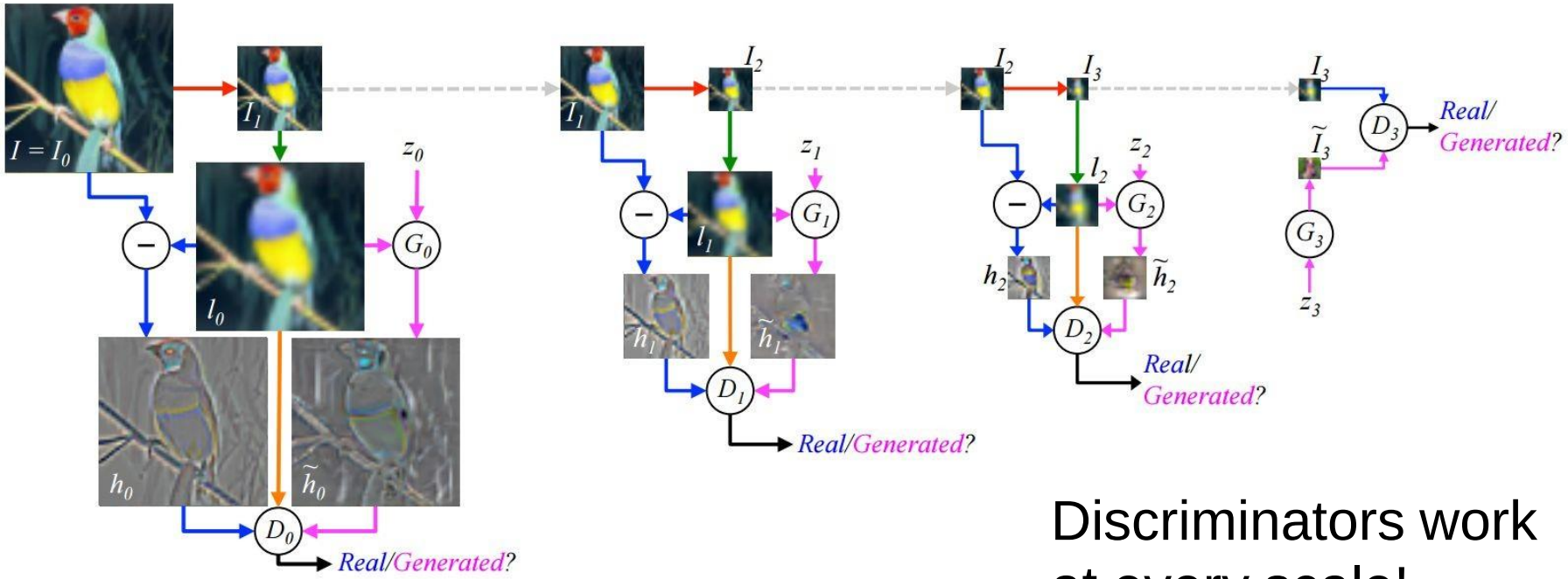
Mirza et al, "Conditional Generative Adversarial Nets", arXiv:1411.1784, 2014

Conditional generative adversarial networks (CGAN)



Generated MNIST digits, each row conditioned on one label

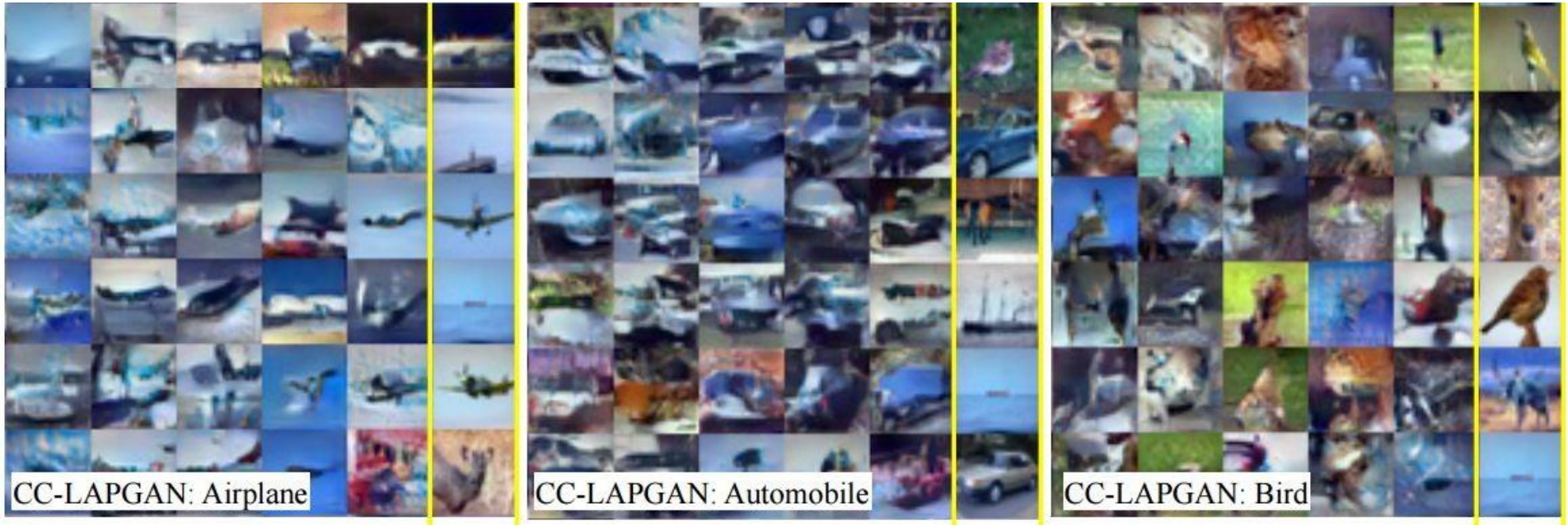
Generative Adversarial Nets: Multiscale



Discriminators work at every scale!

Denton et al, “Deep generative image models using a Laplacian pyramid of adversarial networks”, NIPS 2015

Generative Adversarial Nets: Multiscale



Train separate model per-class on CIFAR-10

Denton et al, NIPS 2015

Generative Adversarial Nets: Simplifying

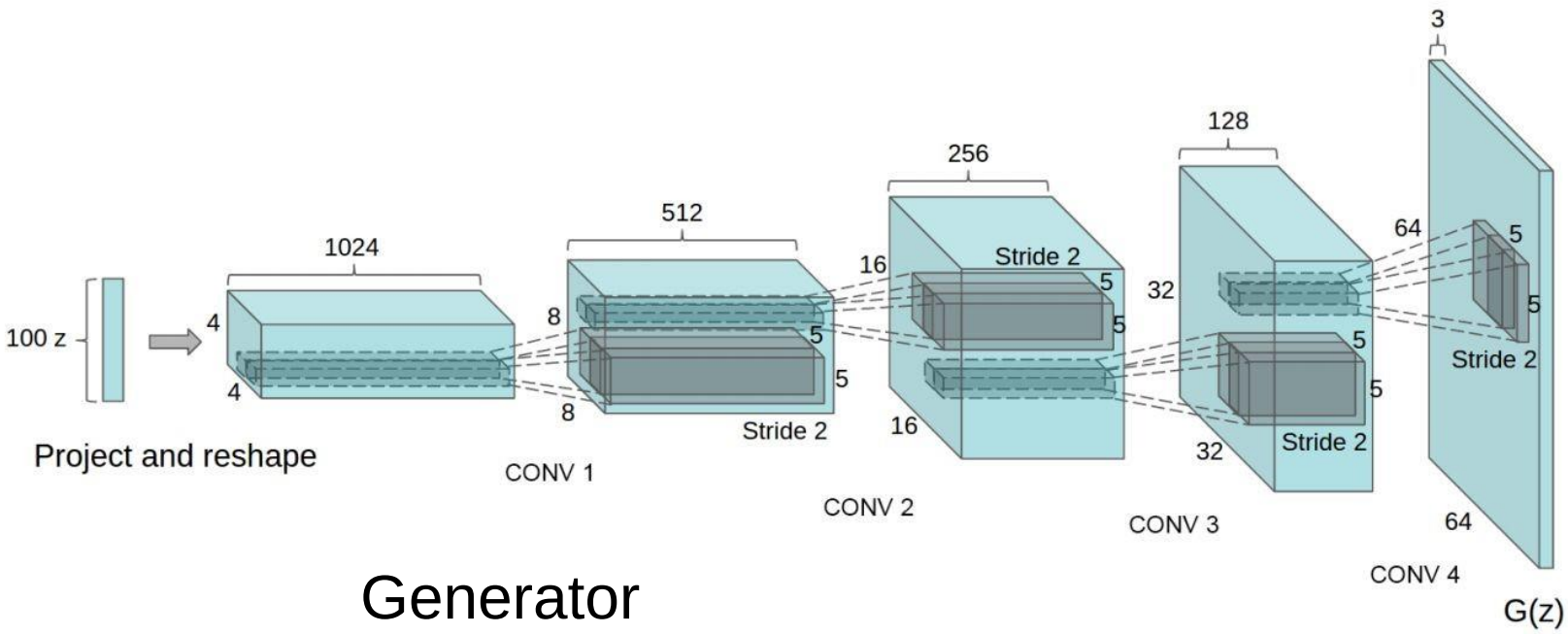
Generator is an upsampling network with fractionally-strided convolutions
Discriminator is a convolutional network

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generative Adversarial Nets: Simplifying



Generator

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generative Adversarial Nets: Simplifying

Samples from the model look amazing!

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generative Adversarial Nets: Simplifying

Interpolating between random points in latent space

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

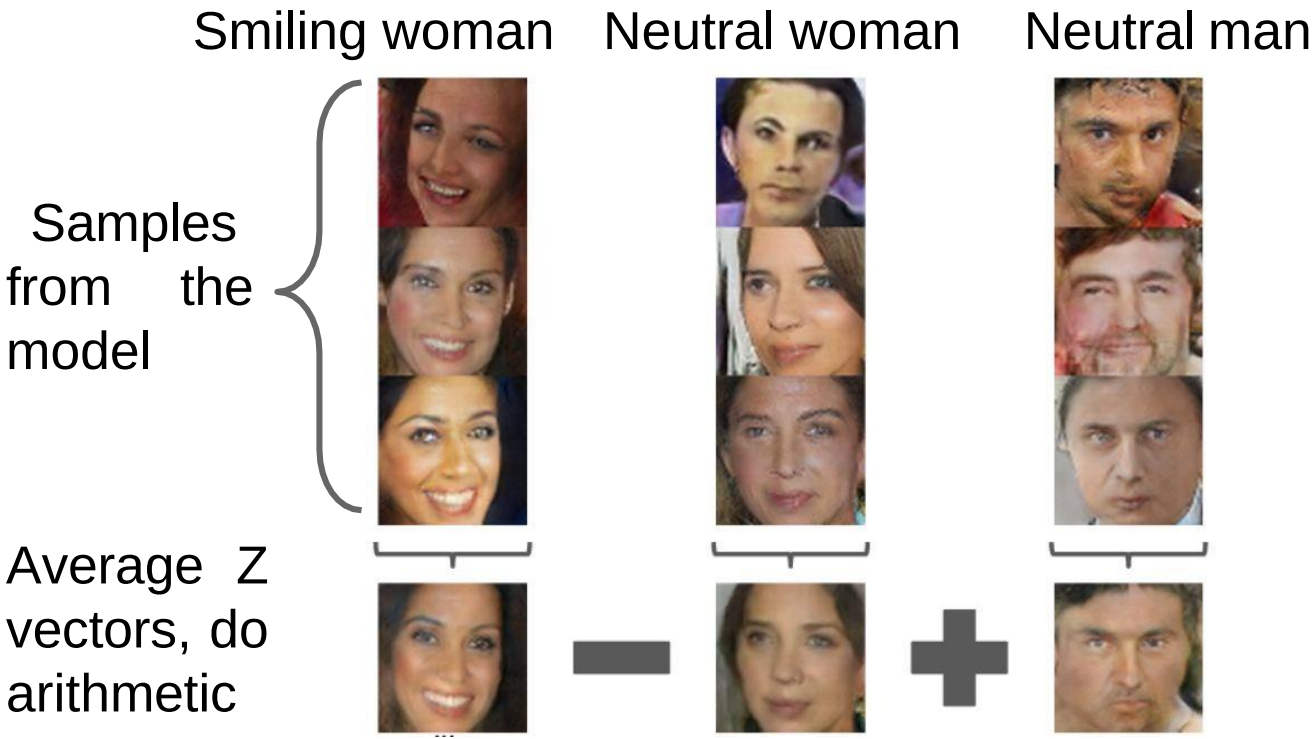
Generative Adversarial Nets: Vector Math

Smiling woman Neutral woman Neutral man

Samples from the model

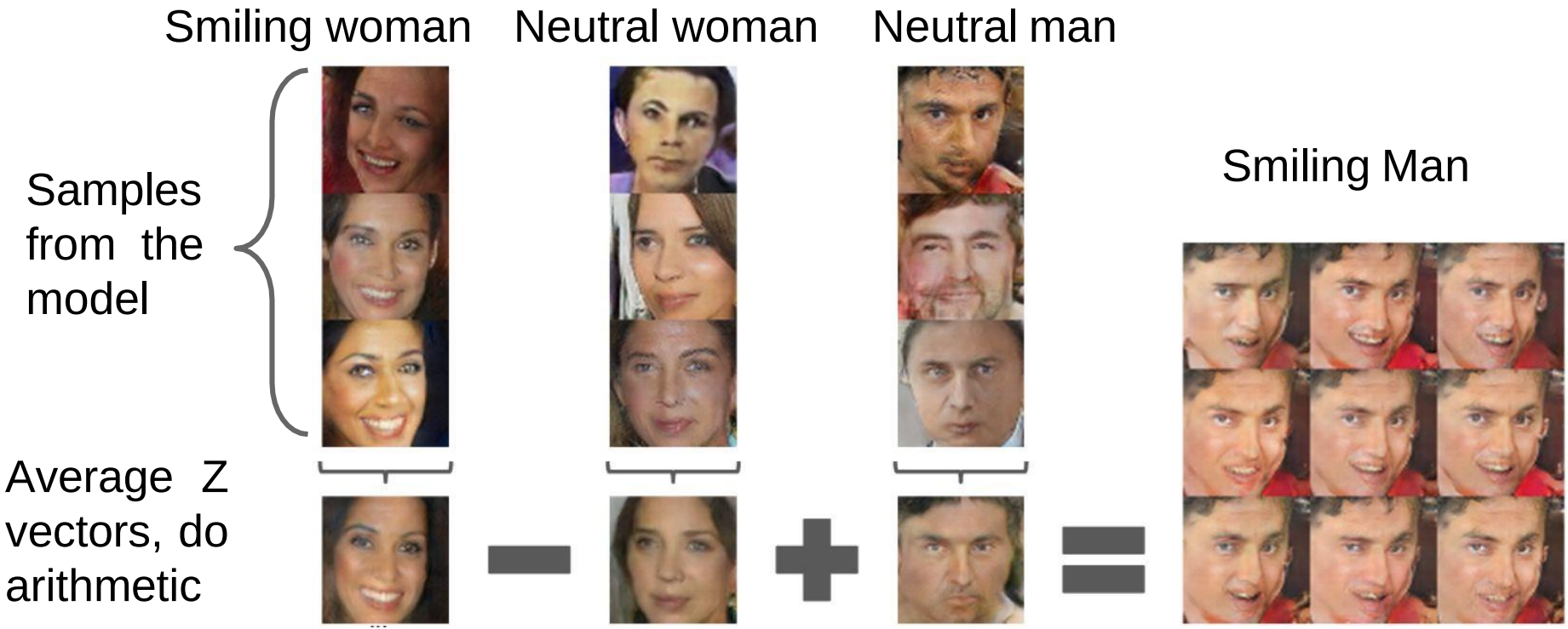
Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generative Adversarial Nets: Vector Math



Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

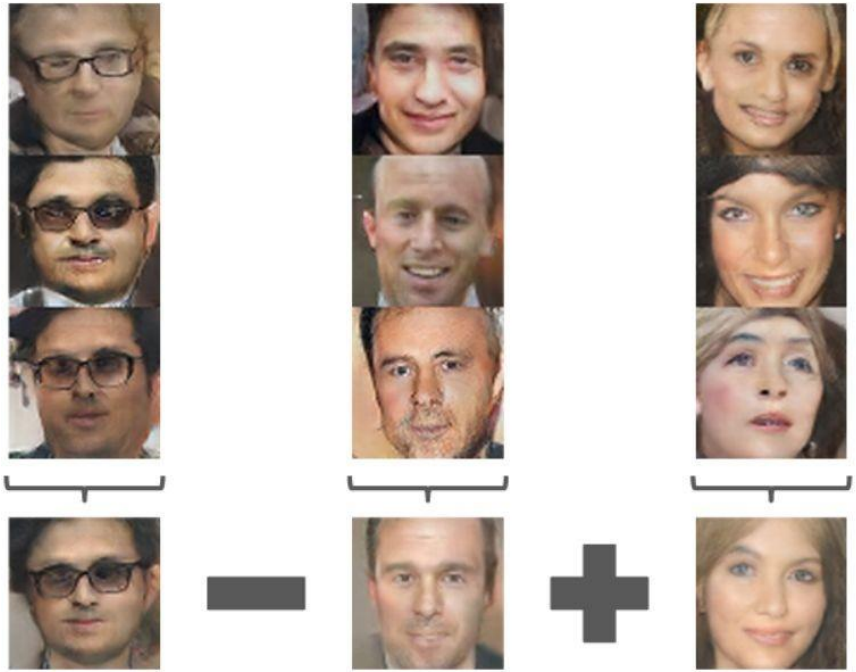
Generative Adversarial Nets: Vector Math



Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generative Adversarial Nets: Vector Math

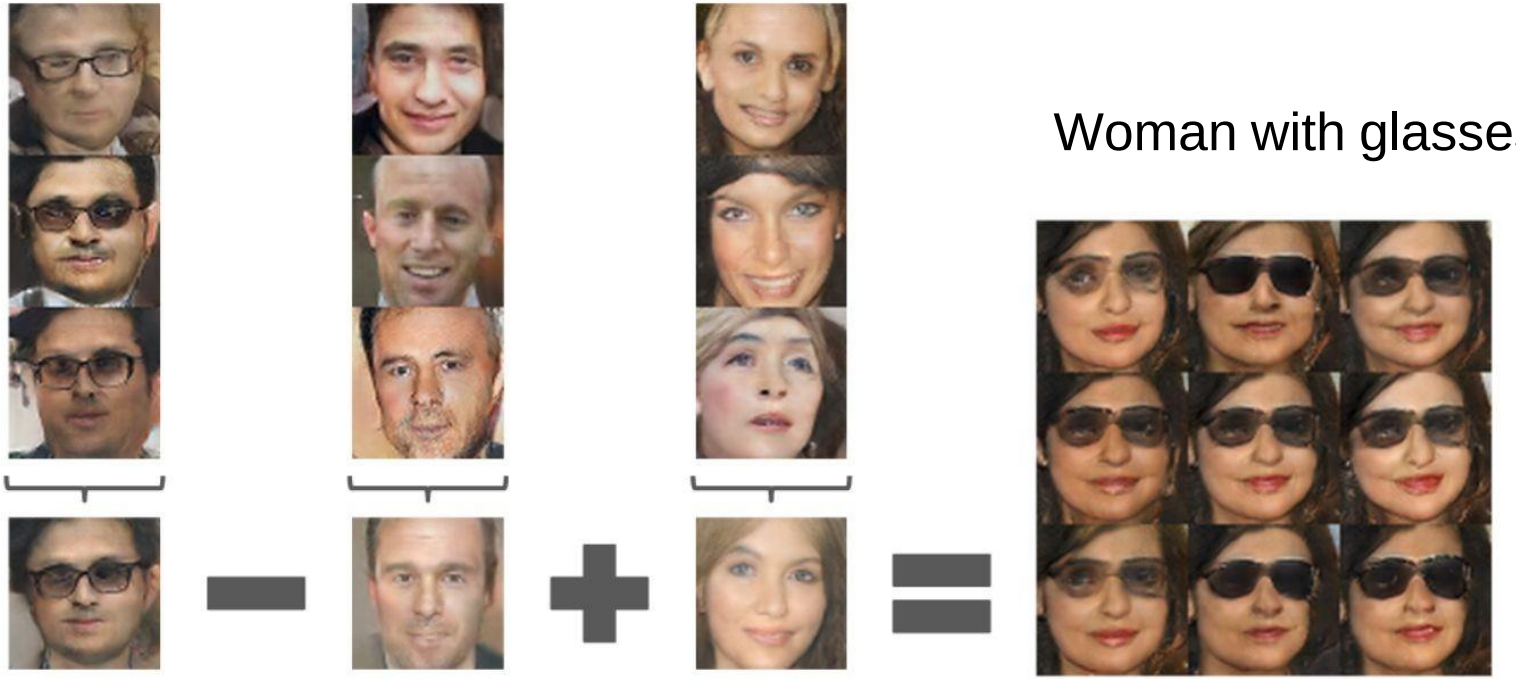
Glasses man No glasses man No glasses woman



Radford et al,
ICLR 2016

Generative Adversarial Nets: Vector Math

Glasses man No glasses man No glasses woman



Woman with glasses

Radford et al,
ICLR 2016

Generative Adversarial Nets: Vector Math

Application in ADAS or Autonomous Vehicle?

Sports Car

−

Car

+

Scooter

=

?

Generative Adversarial Nets: Vector Math

Application in ADAS or Autonomous Vehicle?

Sports Car

−

Car

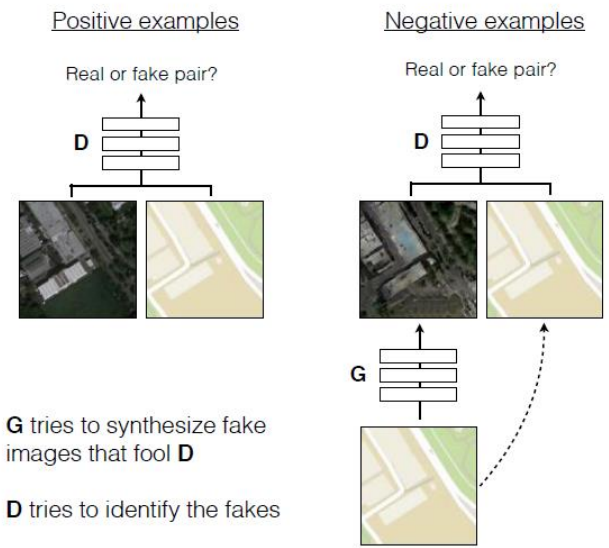
+

Scooter

=

Heavy Motorcycle

A variation of CGAN:pix2pix



Framework

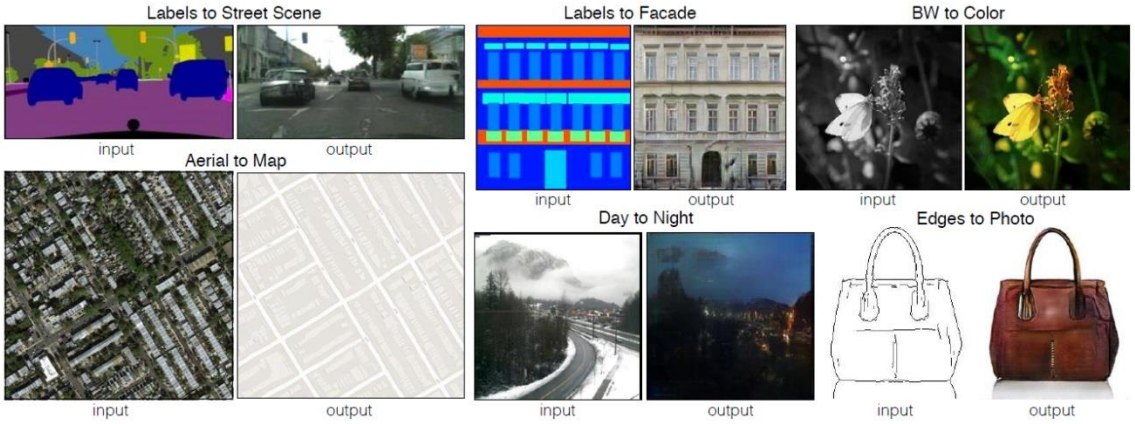
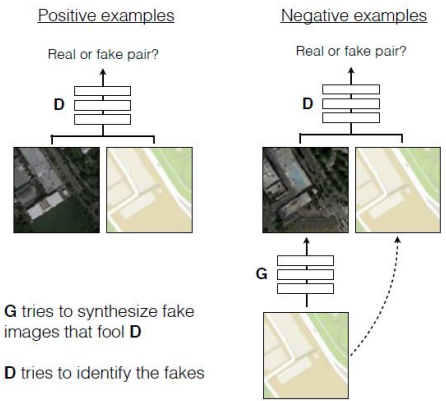


Image-to-image translation

Isola et al, “Image-to-Image Translation with Conditional Adversarial Networks”, CVPR 2017

Pix2pix: a variation of CGAN



Framework
(Condition without L1 loss)

Unconditional GAN:

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{y\sim p_{data}(y)}[\log D(y)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(G(x,z))]. \quad (2)$$

conditional GAN:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(x,G(x,z))], \quad (1)$$

Conditional GAN+L1 loss:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[\|y-G(x,z)\|_1]. \quad (3)$$

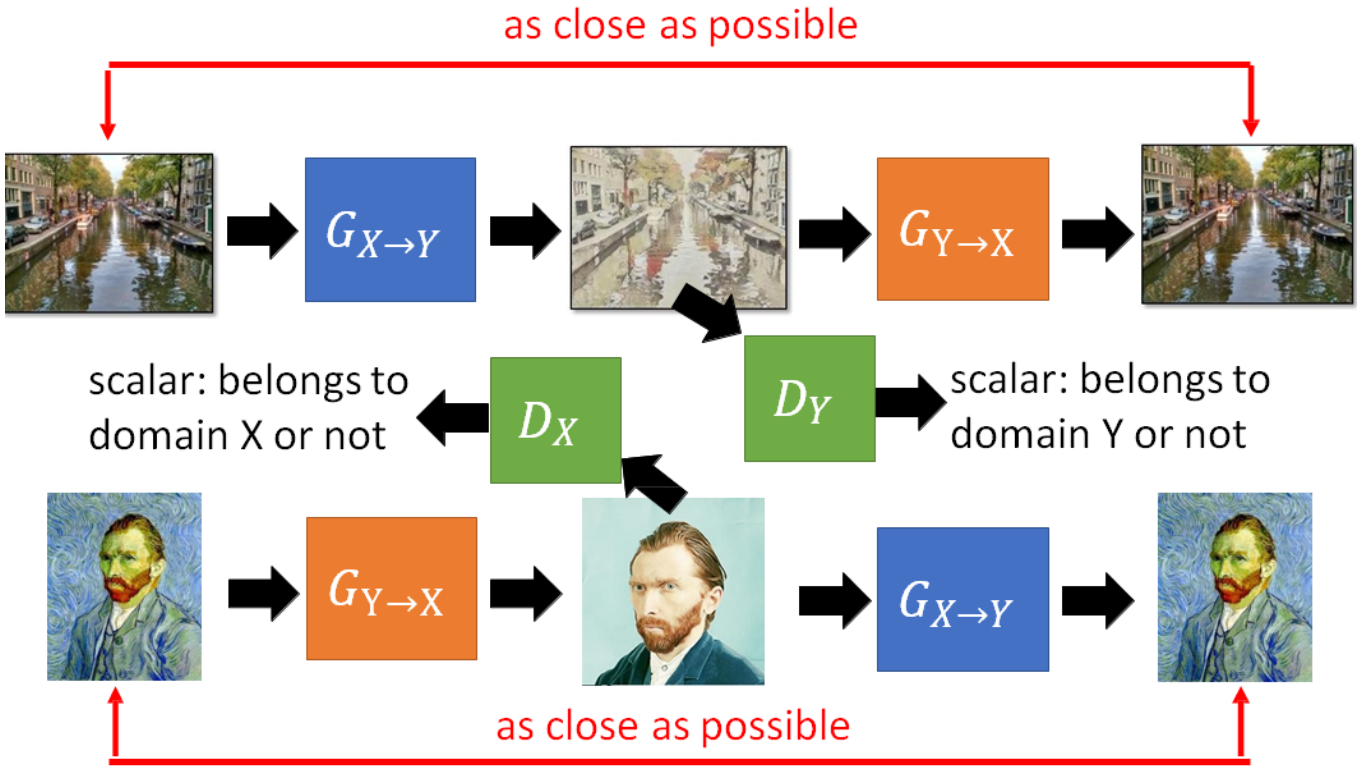
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G). \quad (4)$$

Isola et al, “Image-to-Image Translation with Conditional Adversarial Networks”, CVPR 2017

Pix2Pix HD

High-Resolution Image Synthesis and
Semantic Manipulation with Conditional GANs
Ting-Chun Wang[1], Ming-Yu Liu[1], Jun-Yan Zhu[2], Andrew Tao[1],
Jan Kautz[1], Bryan Catanzaro[1]
[1]NVIDIA Corporation [2]University of California, Berkeley

CycleGAN



Zhu et al, “Unpaired image-to-image translation using cycle-consistent adversarial networks”, ICCV 2017

Unpaired Transformation- Cycle GAN, Disco GAN

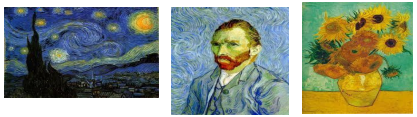
paired data

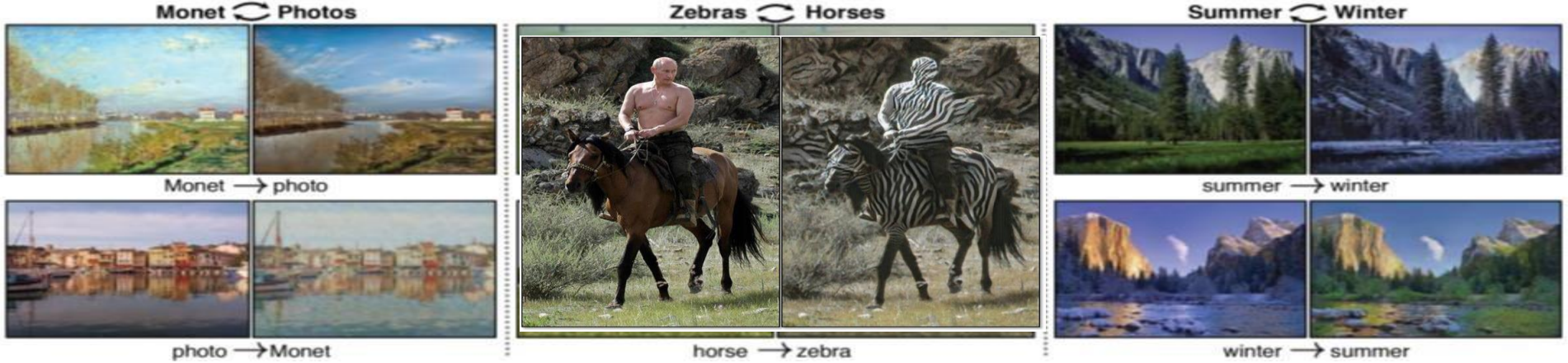
Transform an object from one domain to another ***without paired data***

photo van Gogh

Domain X

Domain Y

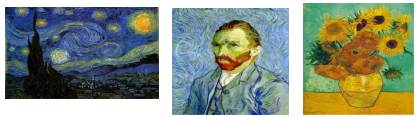




Cycle GAN

Domain X

Domain Y

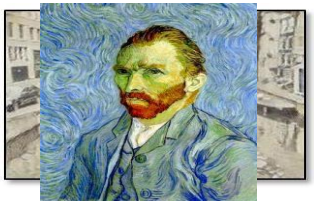


Domain X

ignore input

$G_{X \to Y}$

Become similar to domain Y



Not what we want

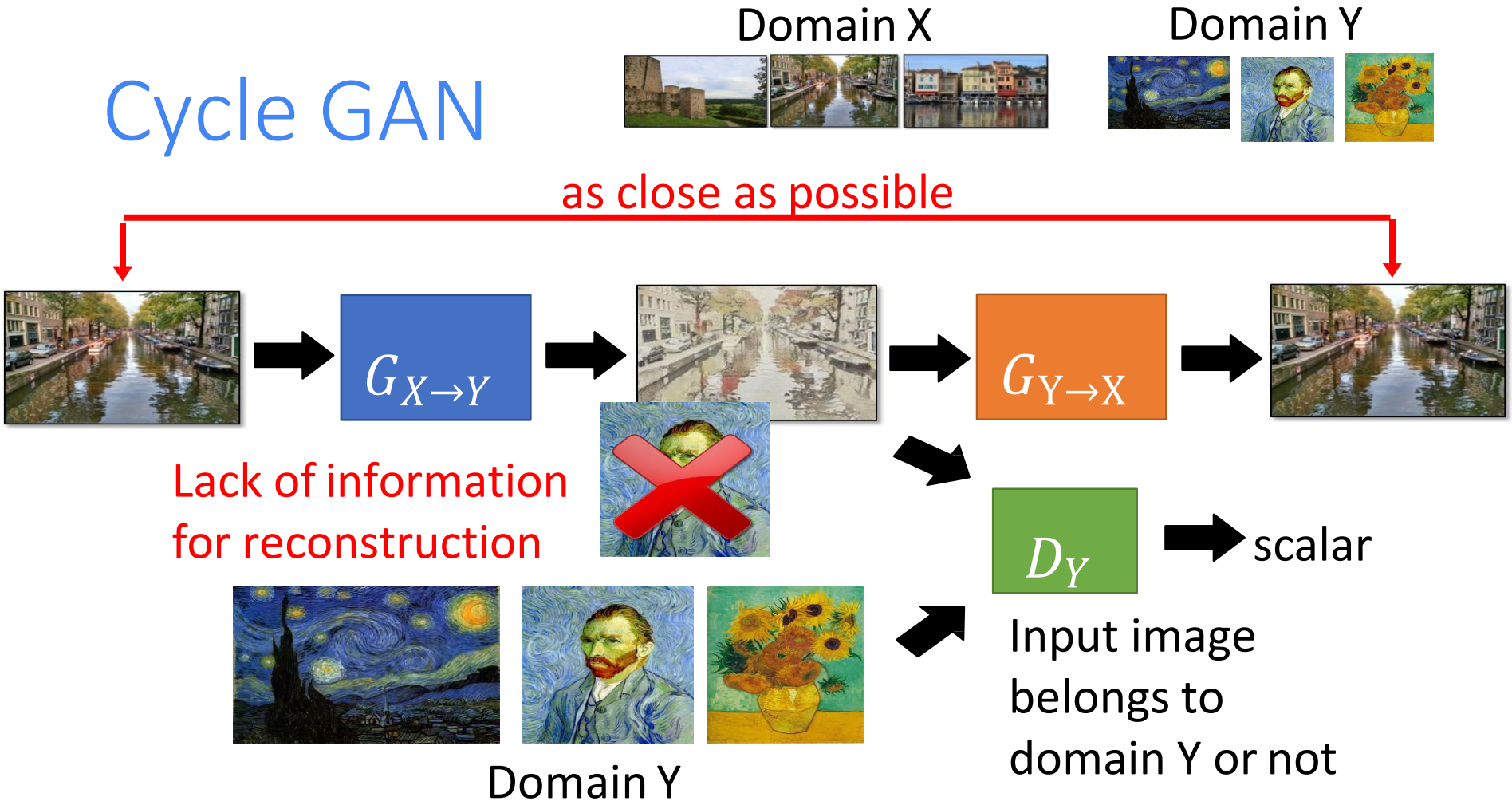
D_Y

scalar

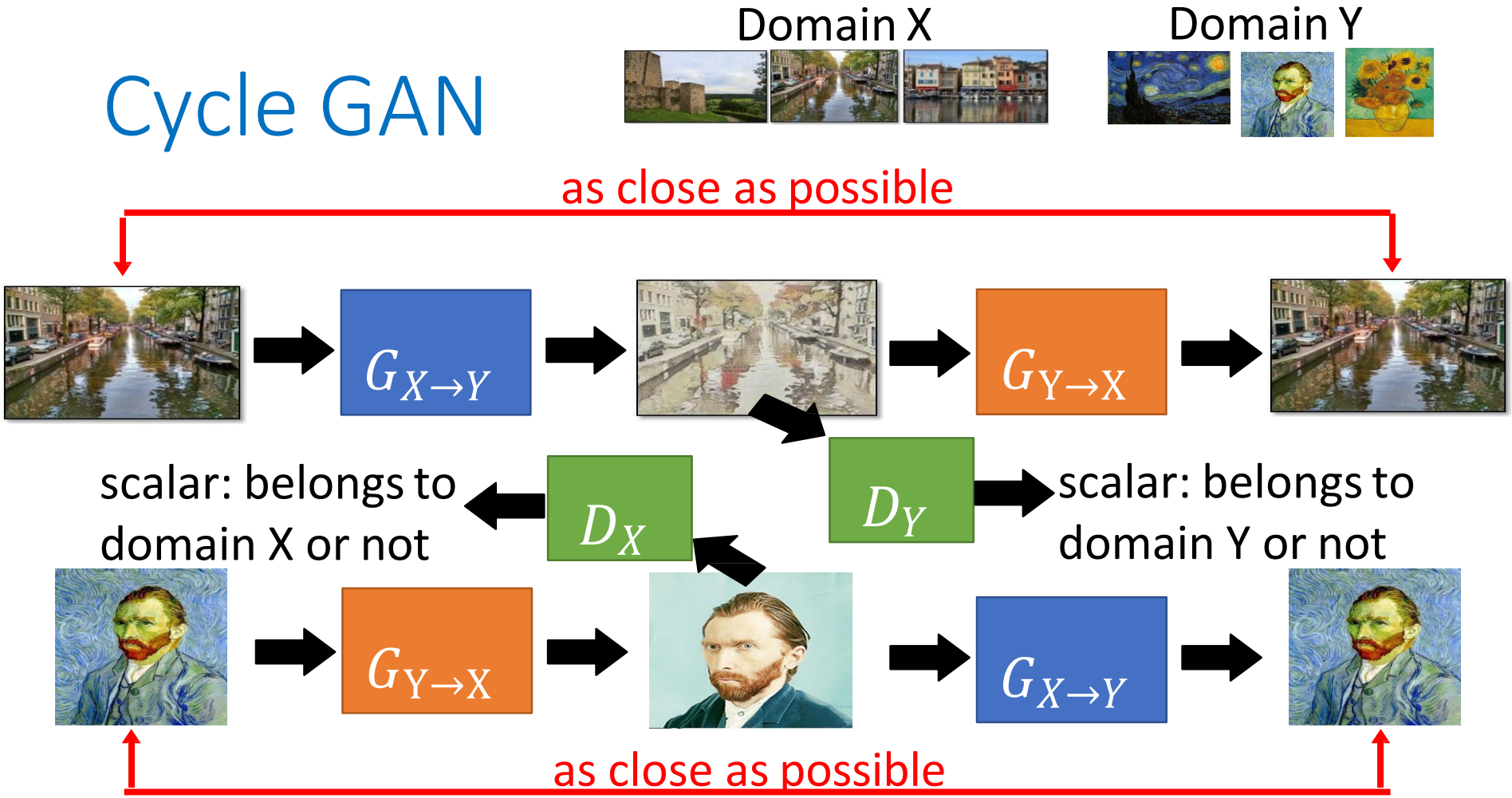
Input image belongs to domain Y or not

Domain Y

Cycle GAN


Domain X
Domain Y
as close as possible
$G_{X\to Y}$
$G_{Y\to X}$
Lack of information
for reconstruction
D_Y
scalar
Input image
belongs to
domain Y or not
Domain Y

Cycle GAN


Domain X
Domain Y
as close as possible
$G_{X\to Y}$
$G_{Y\to X}$
D_X
D_Y
scalar: belongs to domain X or not
scalar: belongs to domain Y or not
$G_{Y\to X}$
$G_{X\to Y}$
as close as possible

Cycle GAN

GAN loss:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))], \quad (1)$$

Cycle consistency loss:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]. \quad (2)$$

Full objective:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F), \quad (3)$$

Optimization goal:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y). \quad (4)$$

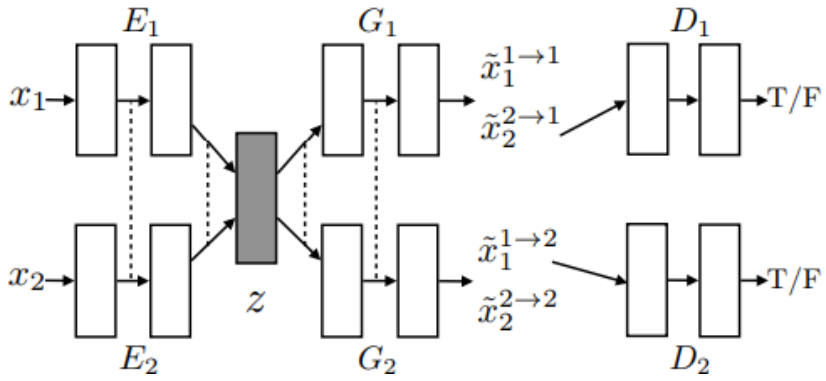
CycleGAN, a Master of Steganography

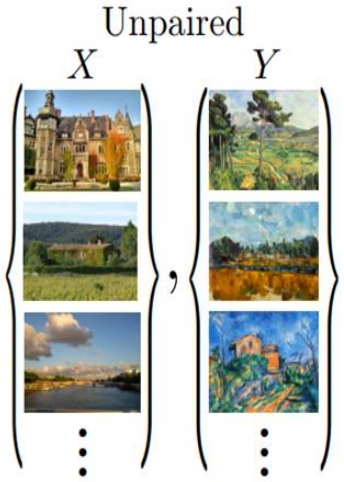
(a) Aerial photograph: x. (b) Generated map: Fx. (c) Aerial reconstruction: GFx.

Figure 1: Details in x are reconstructed in GFx, despite not appearing in the intermediate map Fx.

Chu, Casey, Andrey Zhmoginov, and Mark Sandler. "CycleGAN: a Master of Steganography." *arXiv preprint arXiv:1712.02950* (2017).

UNIT: Unsupervised Image-to-Image Translation Networks





Optimization goal:

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{\text{VAE}_1}(E_1,G_1) + \mathcal{L}_{\text{GAN}_1}(E_1,G_1,D_1) + \mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2)$$
$$\mathcal{L}_{\text{VAE}_2}(E_2,G_2) + \mathcal{L}_{\text{GAN}_2}(E_2,G_2,D_2) + \mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1).$$

VAE Loss

GAN Loss

Cycle Consistency Loss

Liu et al, “Unsupervised Image-to-Image Translation Networks”, NIPS 2017

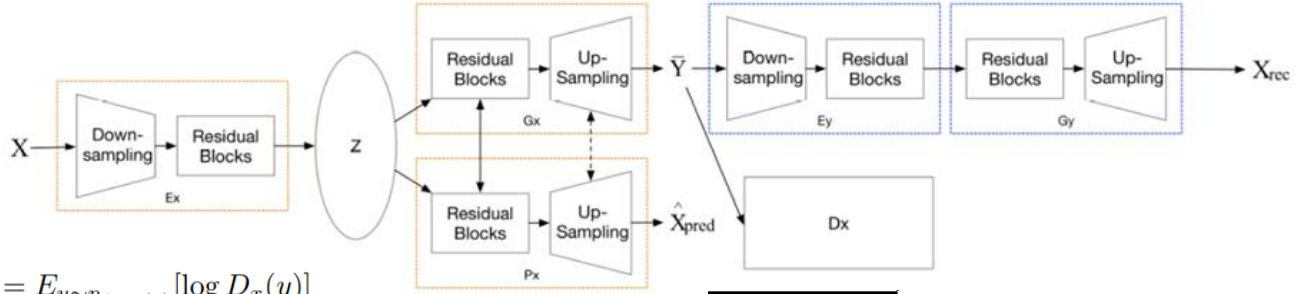
Day-to-Night translation: Pix2Pix and CycleGAN

Generator (Pix2Pix)

Generator (CycleGAN)

UNIT: official results

AugGAN: Forward Cycle



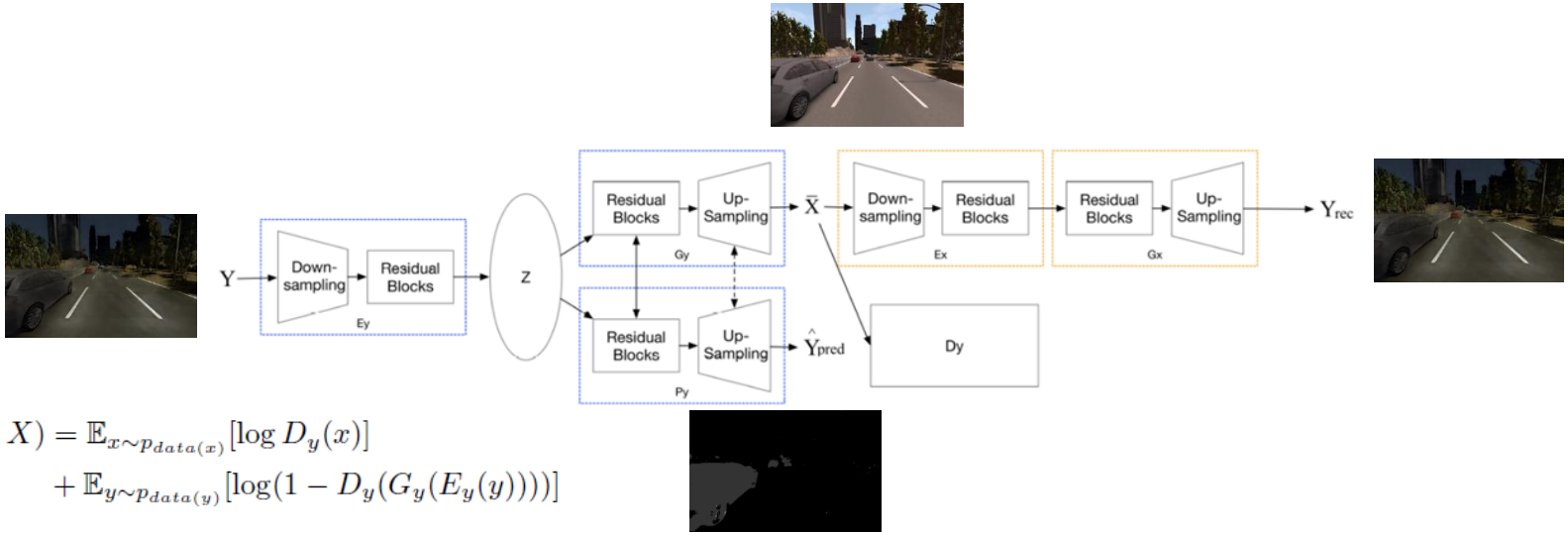
$$L_{GAN_1}(E_x, G_x, D_x, X, Y) = E_{y\sim p_{data(y)}}[\log D_x(y)] + E_{x\sim p_{data(x)}}[\log(1 - D_x(G_x(E_x(x))))]$$

$$L_{seg-x}(P_x, E_x, X, \hat{X}) = \mathbb{E}_{x\sim p_{data(x)}}[\|P_x(E_x(x)) - \hat{x}\|_1]$$

$$\mathcal{L}_\omega(\omega_{G_x}, \omega_{P_x}) = -\log((\omega_{G_x} \cdot \omega_{P_x} / \|\omega_{G_x}\|_2 \|\omega_{P_x}\|_2)^2)$$

$$L_{cyc1}(E_x, G_x, E_y, G_y, X) = \mathbb{E}_{x\sim p_{data(x)}}[\|G_y(E_y(G_x(E_x(x)))) - \mathrm{x}\|_1].$$

AugGAN: Backward Cycle



$$\mathcal{L}_{GAN_2}(E_y, G_y, D_y, Y, X) = \mathbb{E}_{x \sim p_{data(x)}}[\log D_y(x)] + \mathbb{E}_{y \sim p_{data(y)}}[\log(1 - D_y(G_y(E_y(y))))]$$

$$L_{seg-y}(P_y, E_y, Y, \hat{Y}) = \mathbb{E}_{y \sim p_{data(y)}}[\|(P_y(E_y(y)) - \hat{y}\|_1]$$

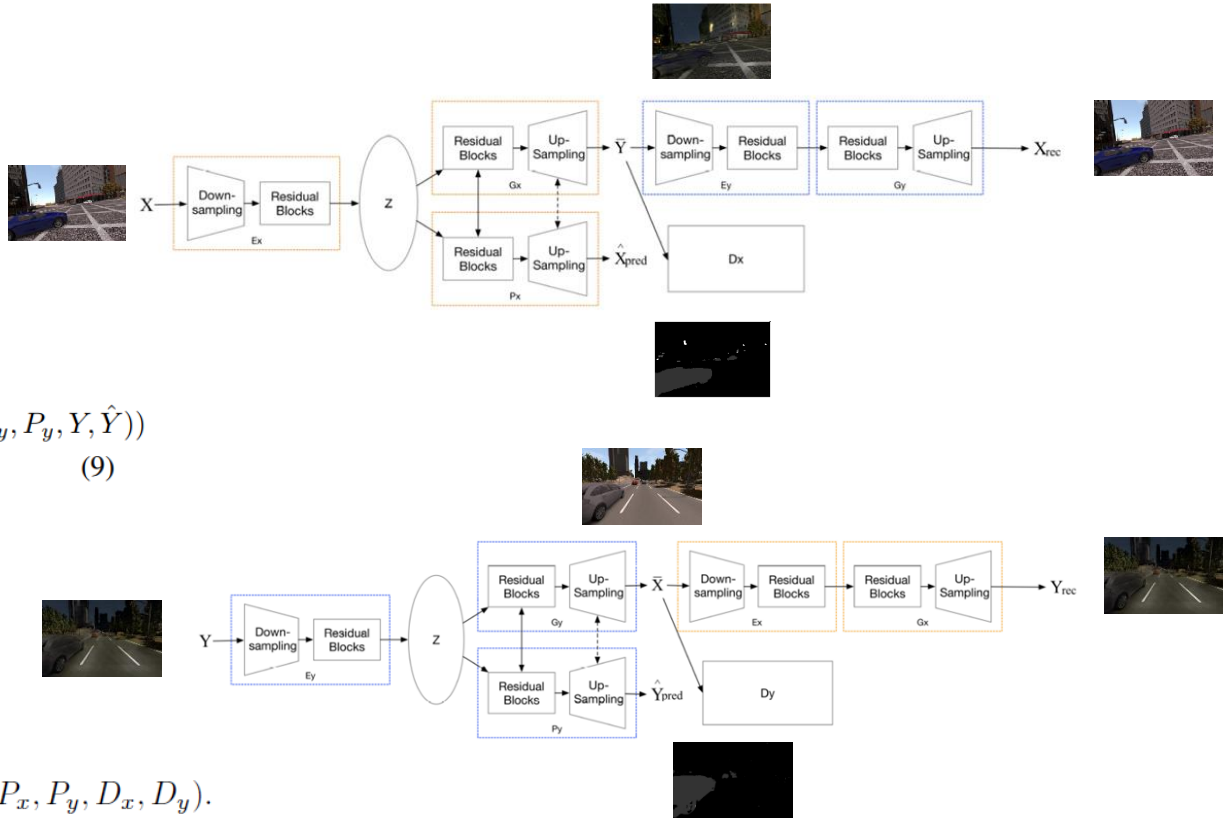
$$\mathcal{L}_{\omega}(\omega_{G_y}, \omega_{P_y}) = -\log((\omega_{G_y} \cdot \omega_{P_y} / \|\omega_{G_y}\|_2 \|\omega_{P_y}\|_2)^2)$$

$$L_{cyc2}(E_x, G_x, E_y, G_y, Y) = \mathbb{E}_{y \sim p_{data(y)}}[\|G_x(E_x(G_y(E_y(y)))) - y\|_1].$$

AugGAN: full objective

Full Objective:

$$
\begin{aligned}
\mathcal{L}_{full} = \; & \mathcal{L}_{GAN_1}(E_x, G_x, D_x, X, Y) \\
& + \mathcal{L}_{GAN_2}(E_y, G_y, D_y, Y, X) \\
& + \lambda_{cyc} * (\mathcal{L}_{cyc1}(E_x, G_x, E_y, G_y, X) \\
& + \mathcal{L}_{cyc2}(E_x, G_x, E_y, G_y, Y)) \\
& + \lambda_{seg} * (\mathcal{L}_{seg-x}(E_x, P_x, X, \hat{X}) + \mathcal{L}_{seg-y}(E_y, P_y, Y, \hat{Y})) \\
& + \lambda_{\omega} * (\mathcal{L}_{\omega_x}(\omega_{G_x}, \omega_{P_x}) + \mathcal{L}_{\omega_y}(\omega_{G_y}, \omega_{P_y})) \qquad (9)
\end{aligned}
$$



Optimization goal:

$$
\min_{E_x, G_x, E_y, G_y, P_x, P_y} \; \max_{D_x, D_y} \; \mathcal{L}_{full}(E_x, G_x, E_y, G_y, P_x, P_y, D_x, D_y).
$$

Typical Results in Day-to-Night Transformation: SYNTHIA

Input

CycleGAN

UNIT

Ours

Typical Results in Day-to-Night Transformation: GTA

Input

CycleGAN

UNIT

Ours

Typical Results in Day-to-Night Transformation: ITRI-Day

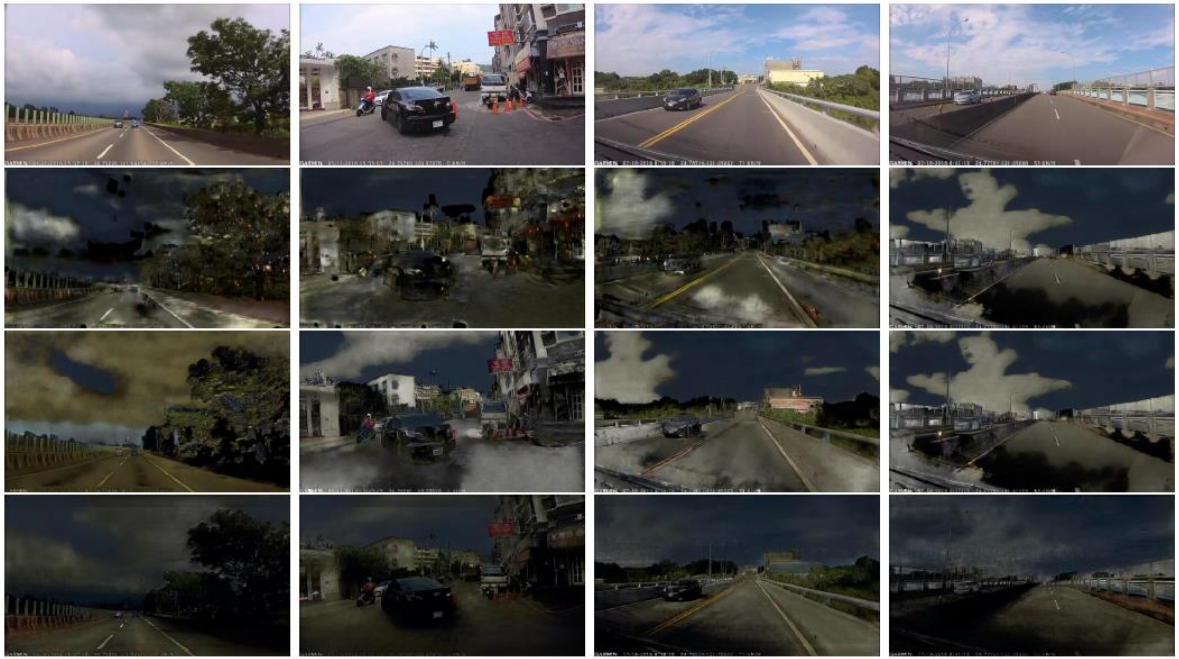
- day-to-night transformation results for ITRI-Day dataset by using different GAN models learnt from SYNTHIA:

Input

CycleGAN

UNIT

Ours



Typical Results in Day-to-Night Transformation: ITRI-Day

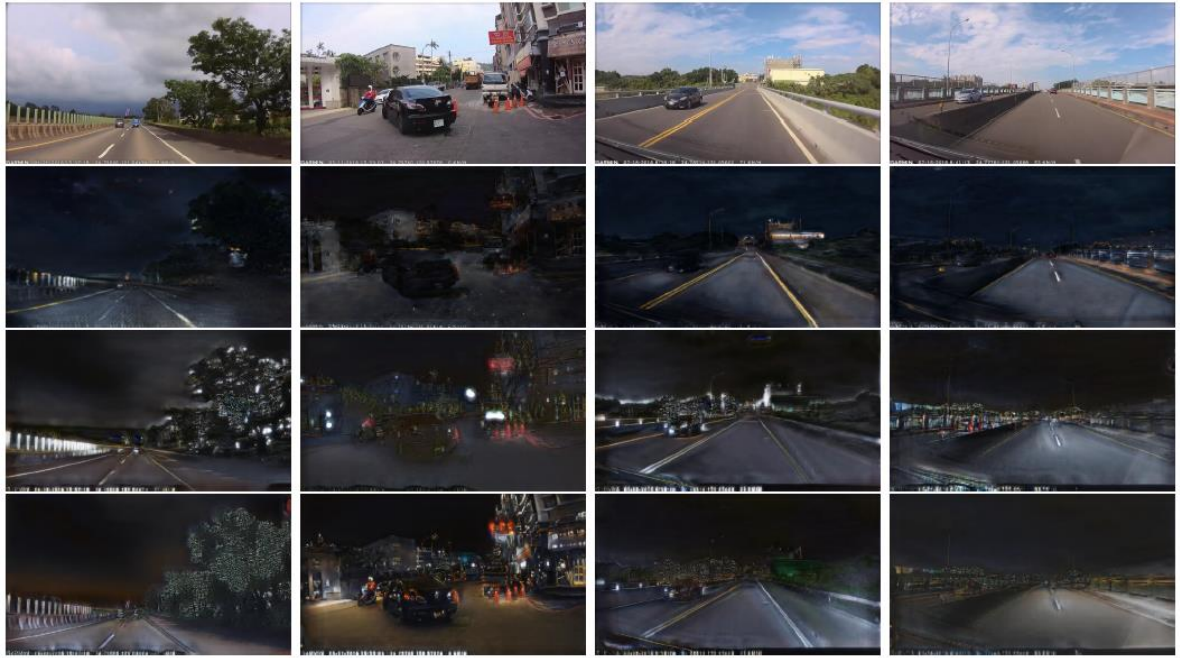
- day-to-night transformation results for ITRI-Day dataset by using different GAN models learnt from GTA:

Input

CycleGAN

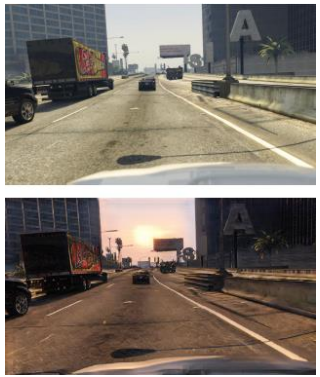
UNIT

Ours



Other Transformations:

GTA-Day
to
SYNTHIA



GTA-Day
To
GTA-Sunset

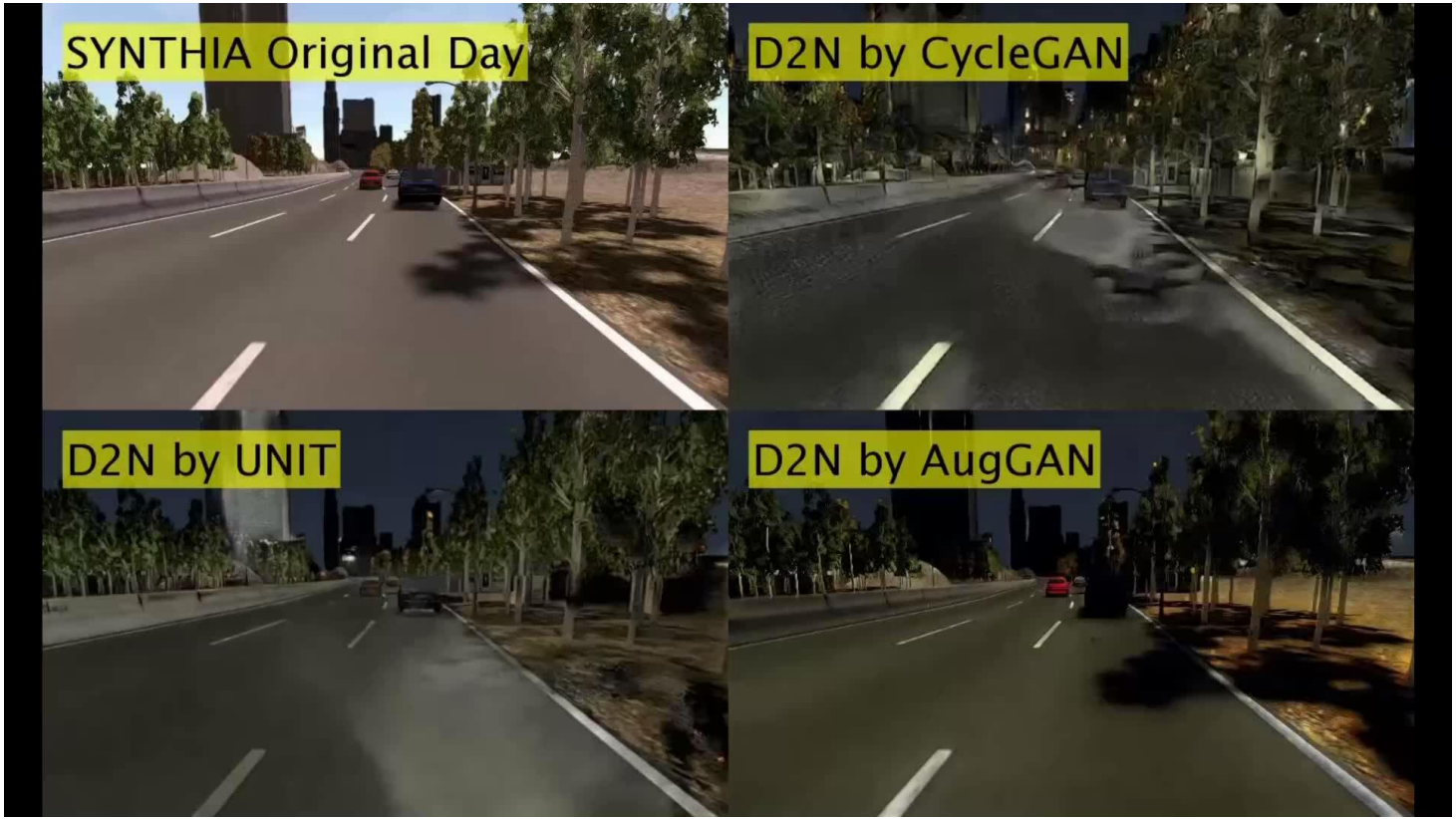
GTA-Day
To
GTA-Rain

Cityscape
to
SYNTHIA

SYNTHIA
To
ITRI-Night

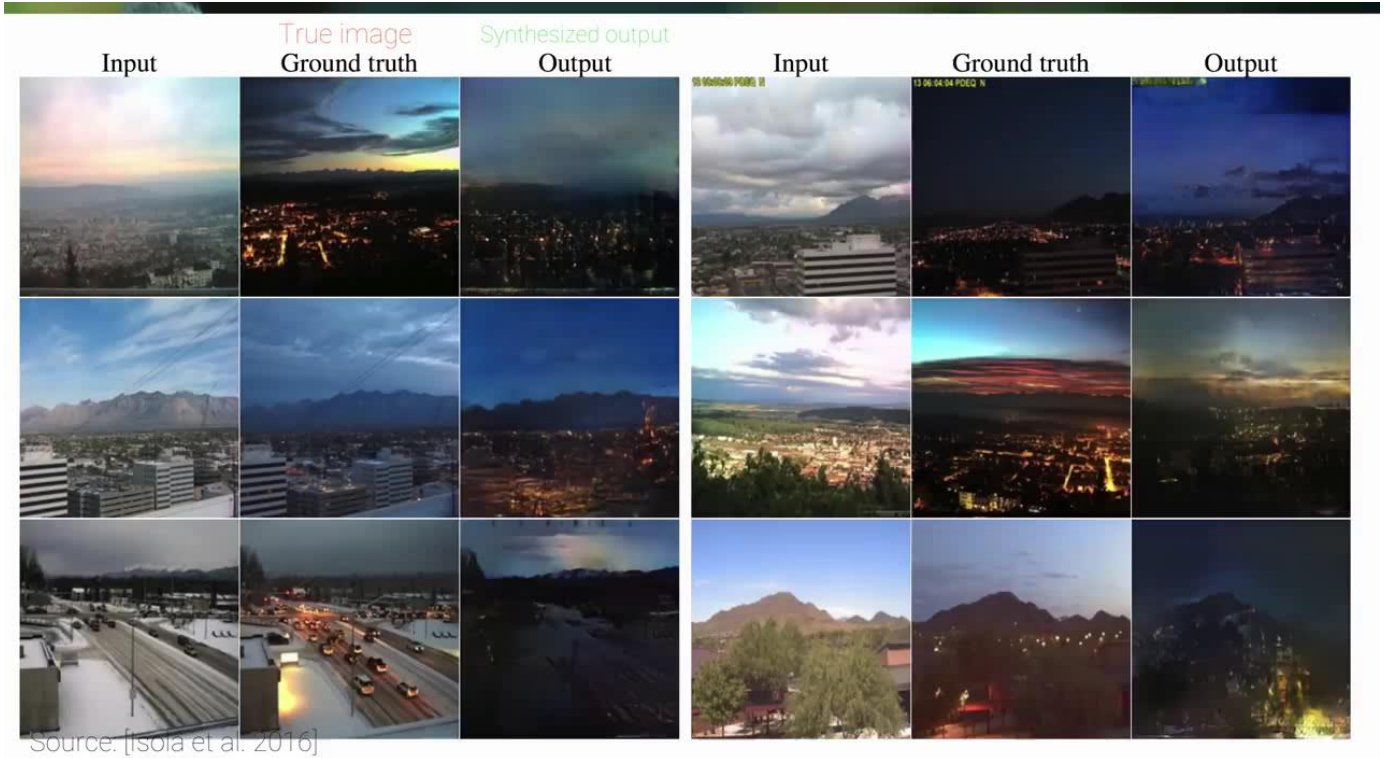
Cityscape
To
ITRI-Night

Video Demonstration



Detection comparison

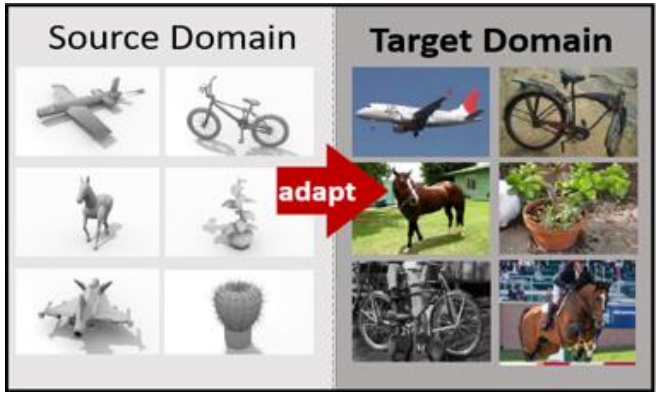
Video-to-Video



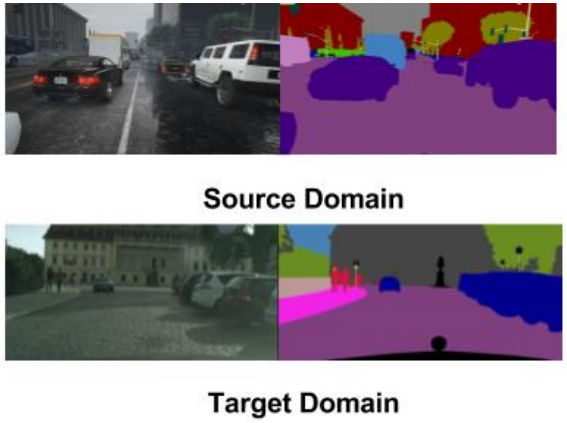
Wang, Ting-Chun, et al. "Video-to-Video Synthesis." *arXiv preprint arXiv:1808.06601* (2018).

GAN is powerful in domain adaptation!

Classification



Segmentation



Detection

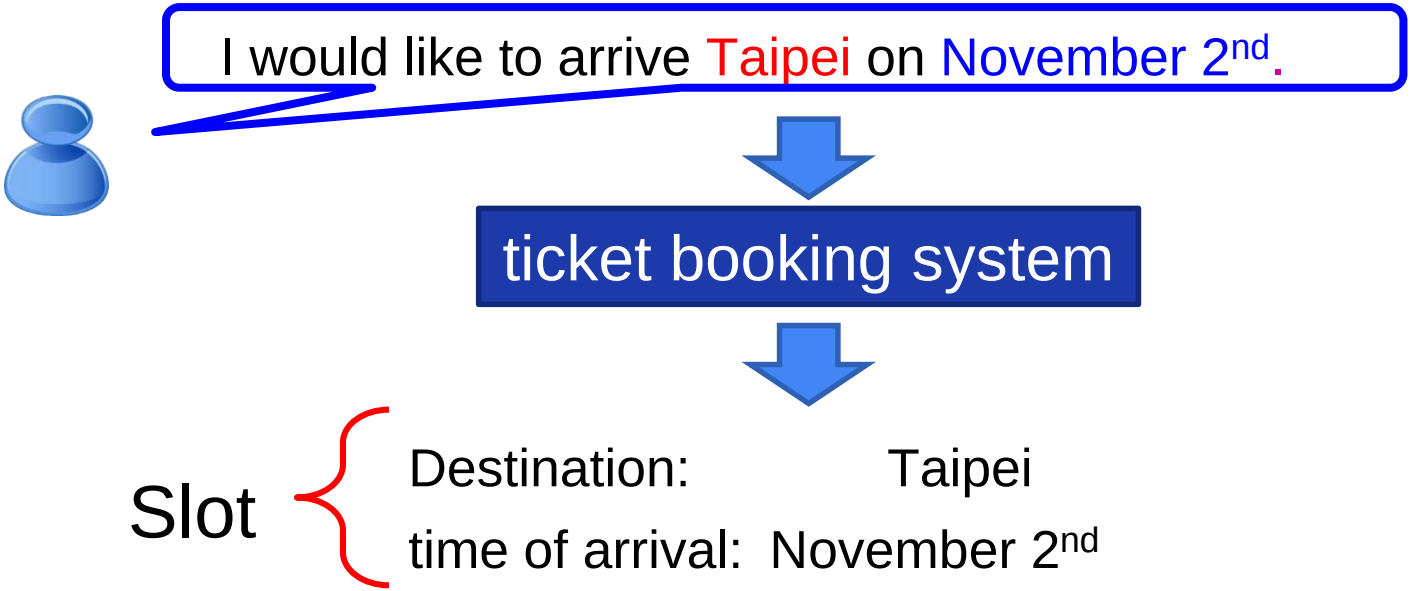
Source Domain

Target Domain

Recurrent Neural Networks

Example Application

- Slot Filling

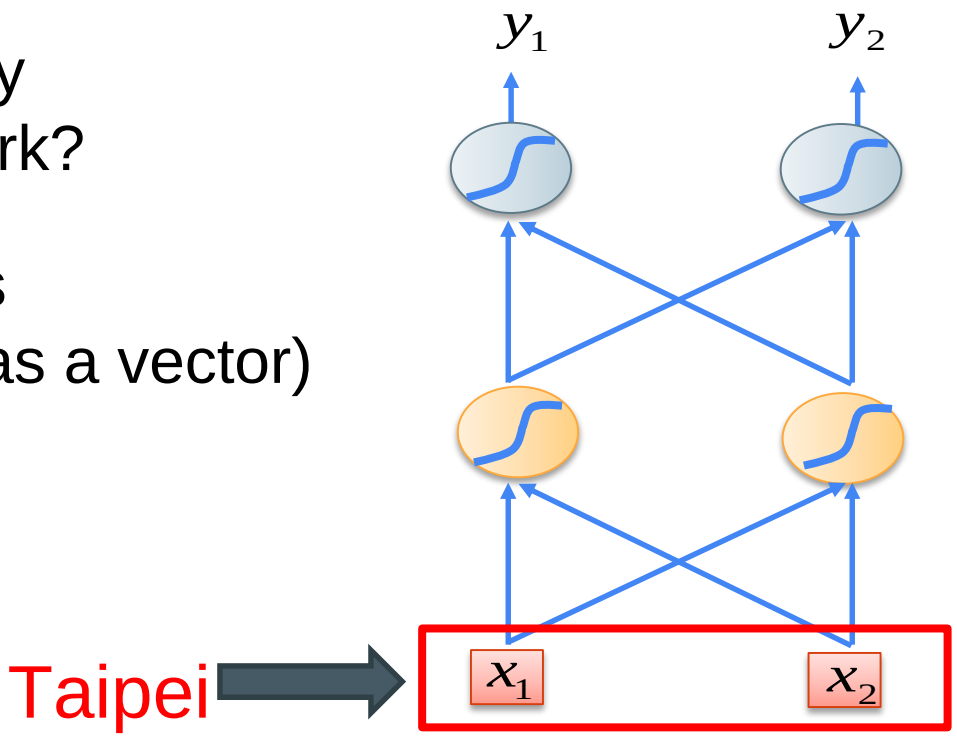


Example Application

Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)



1-of-N encoding

How to represent each word as a vector?

1-of-N Encoding

lexicon = {apple, bag, cat, dog, elephant}

The vector is lexicon size.

Each dimension corresponds to a word in the lexicon

The dimension for the word is 1, and others are 0

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

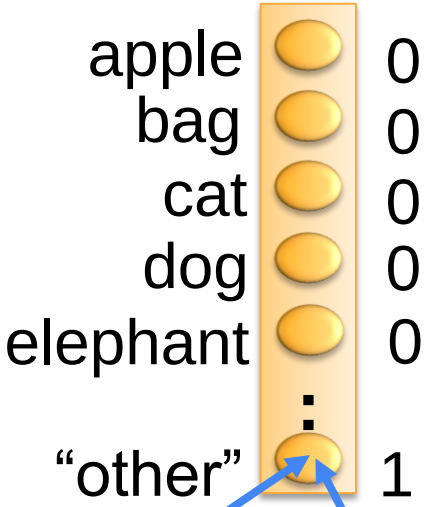
cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

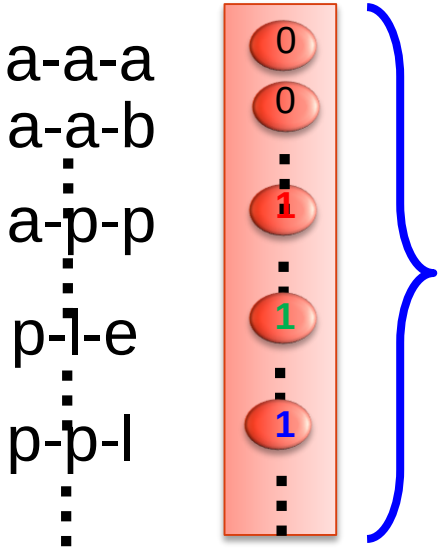
Beyond 1-of-N encoding

Dimension for "Other"



w = "Gandalf" w = "Sauron"

Word hashing



26 X 26 X 26

w = "apple"

Example Application

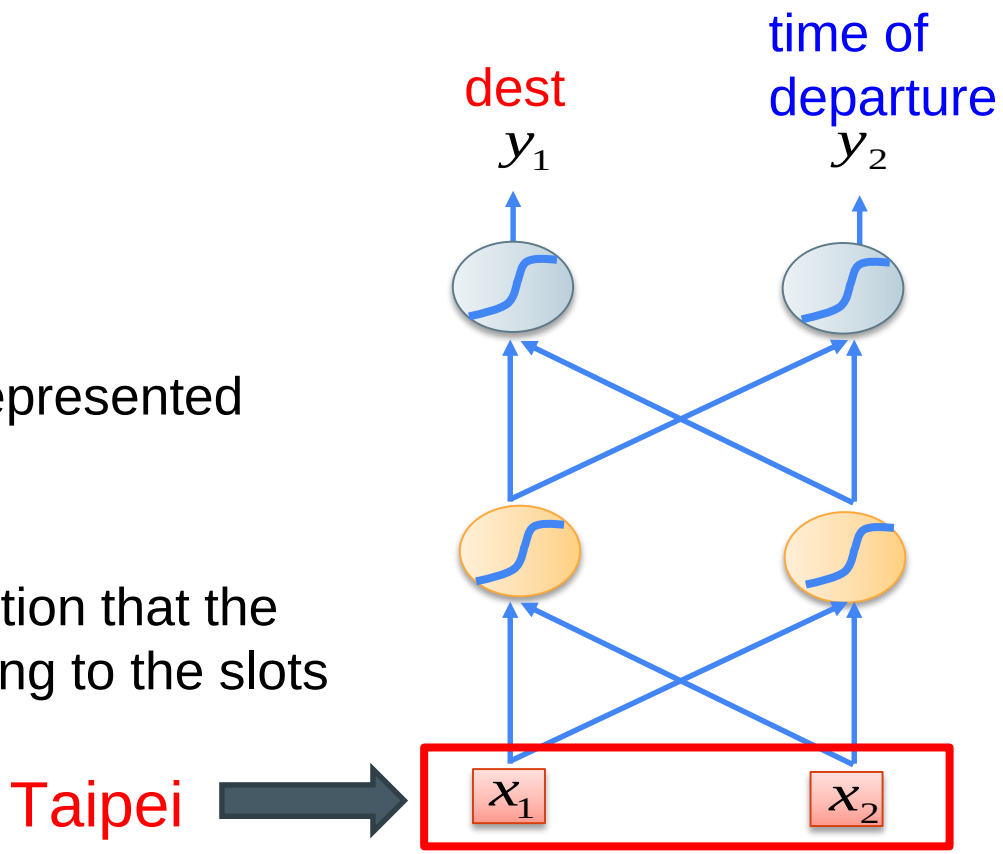
Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)

Output:

Probability distribution that the input word belonging to the slots



Example Application

arrive Taipei on November 2nd

other dest other time time

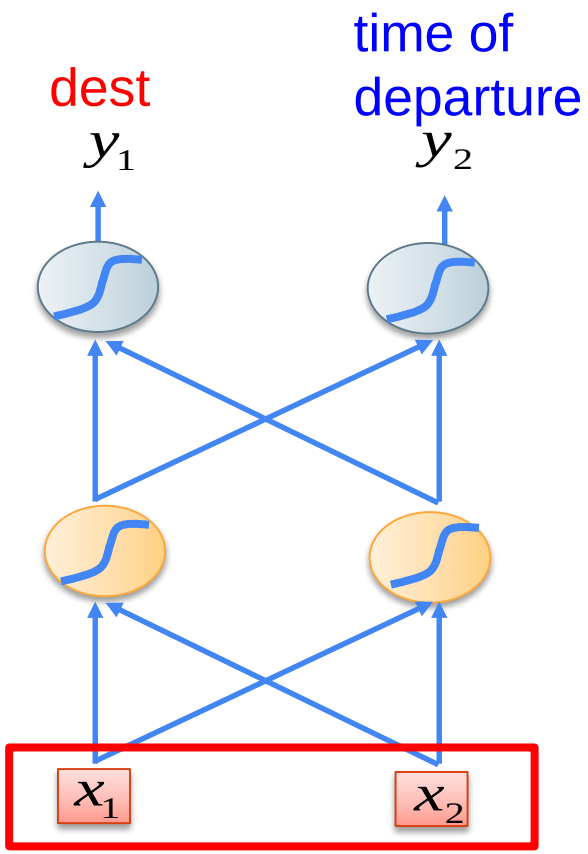
Problem?

leave Taipei on November 2nd

place of departure

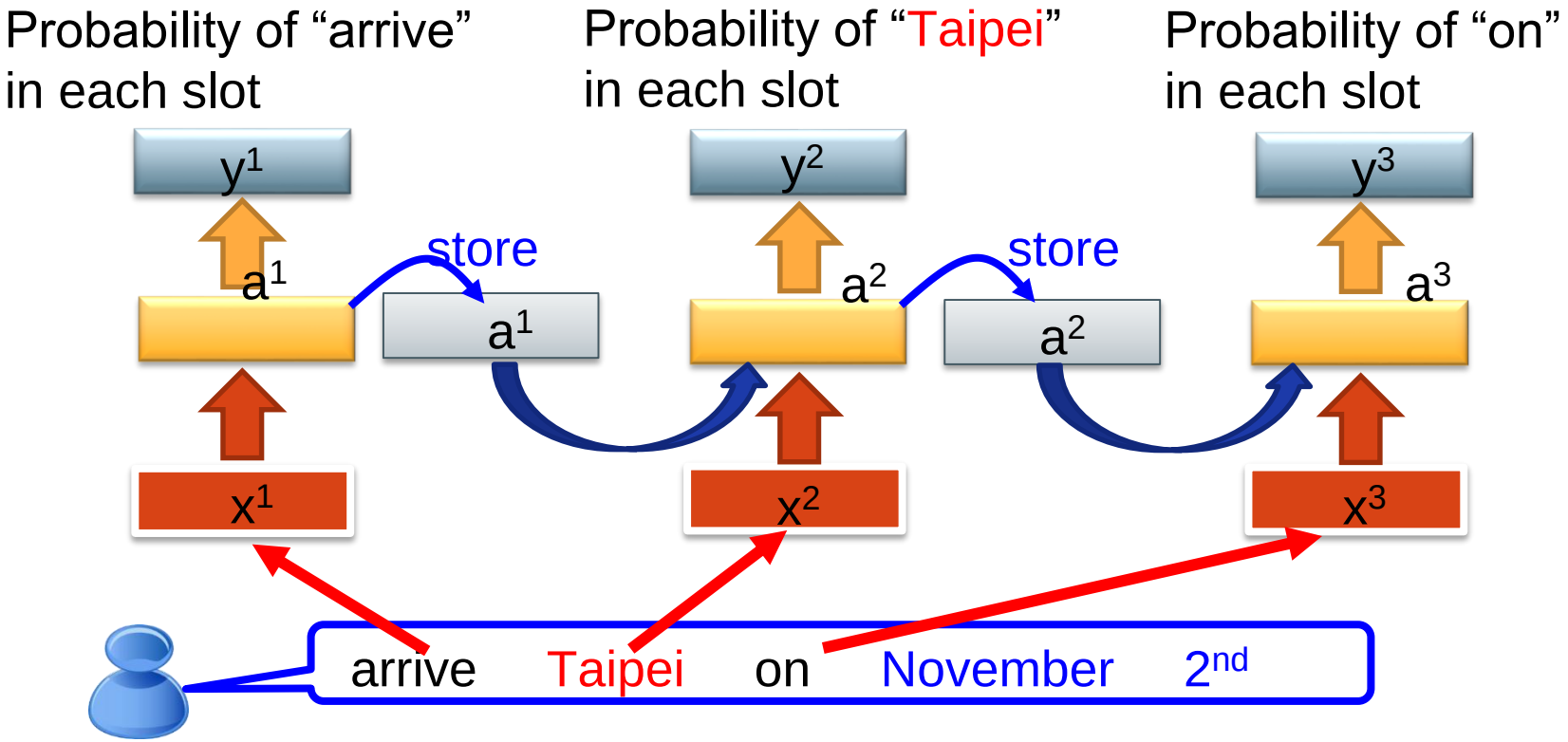
Neural network needs memory!

Taipei

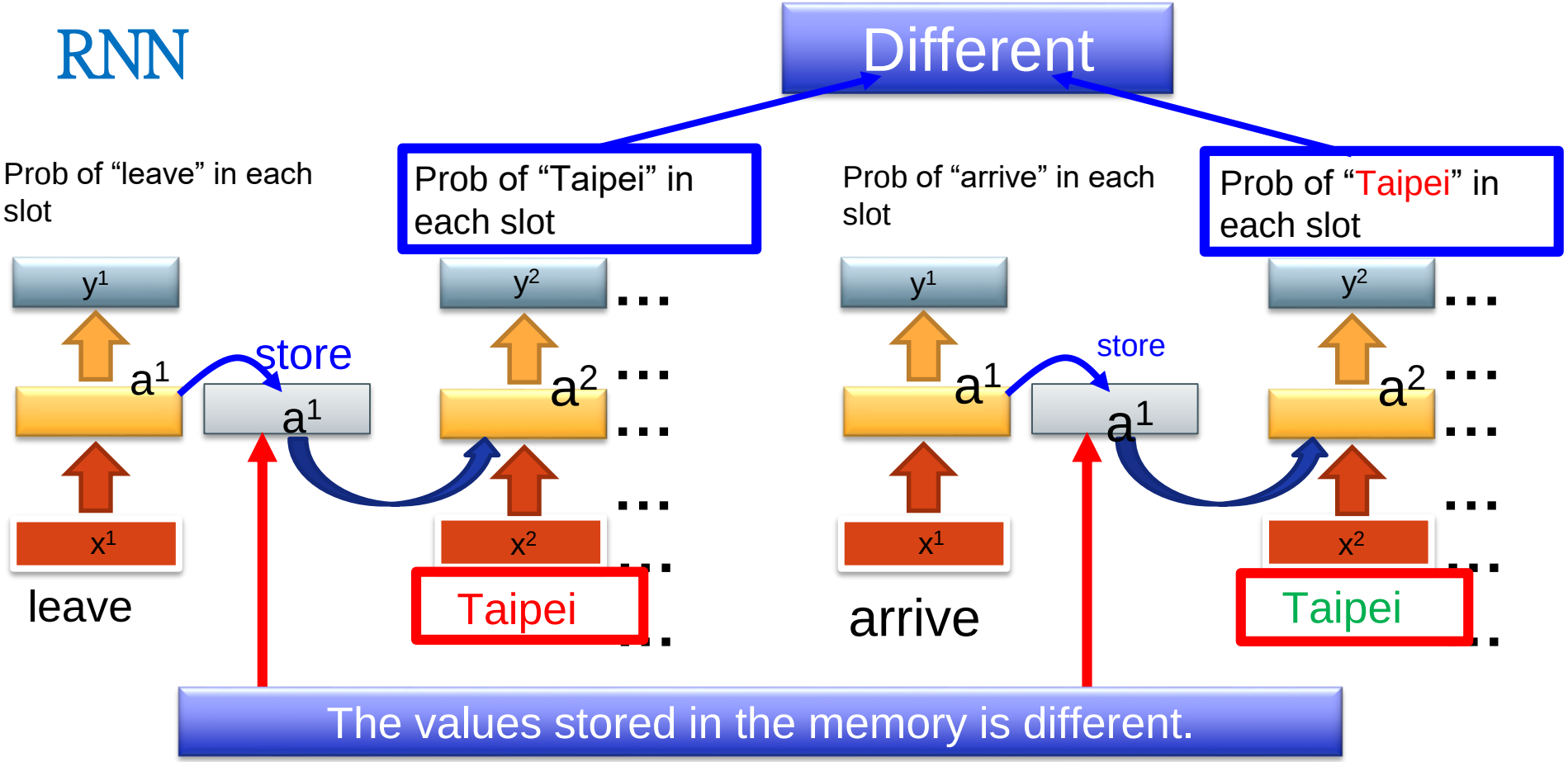


RNN

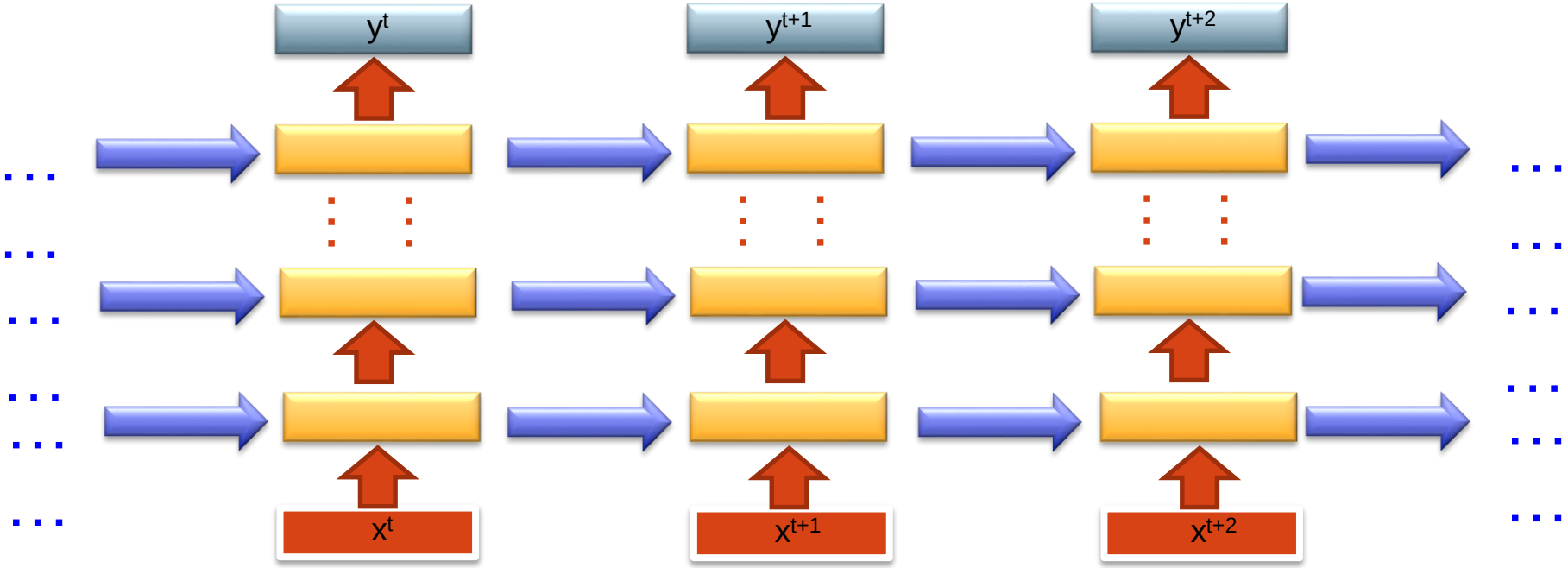
The same network is used again and again.



RNN


Different
Prob of "leave" in each slot
Prob of "Taipei" in each slot
Prob of "arrive" in each slot
Prob of "Taipei" in each slot
y1
y2
a1
a2
store
x1
x2
leave
Taipei
arrive
Taipei
The values stored in the memory is different.

Of course it can be deep …



Helpful Techniques

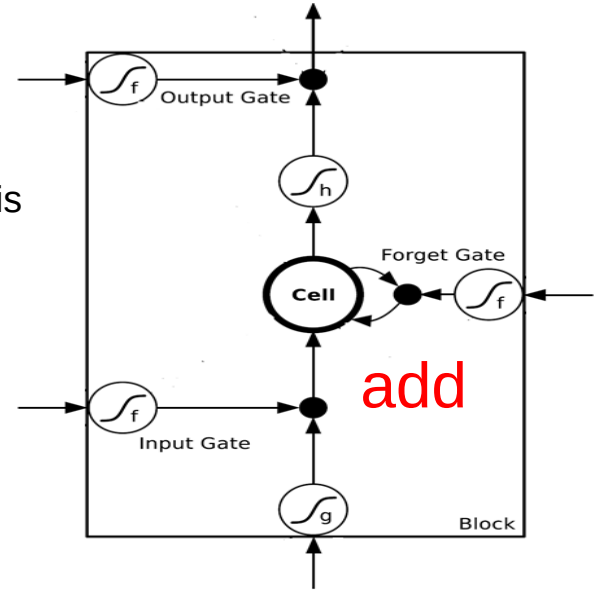
- Long Short-term Memory (LSTM)

  - **Can deal with gradient vanishing (not gradient explode)**

    - Memory and input are ***added***

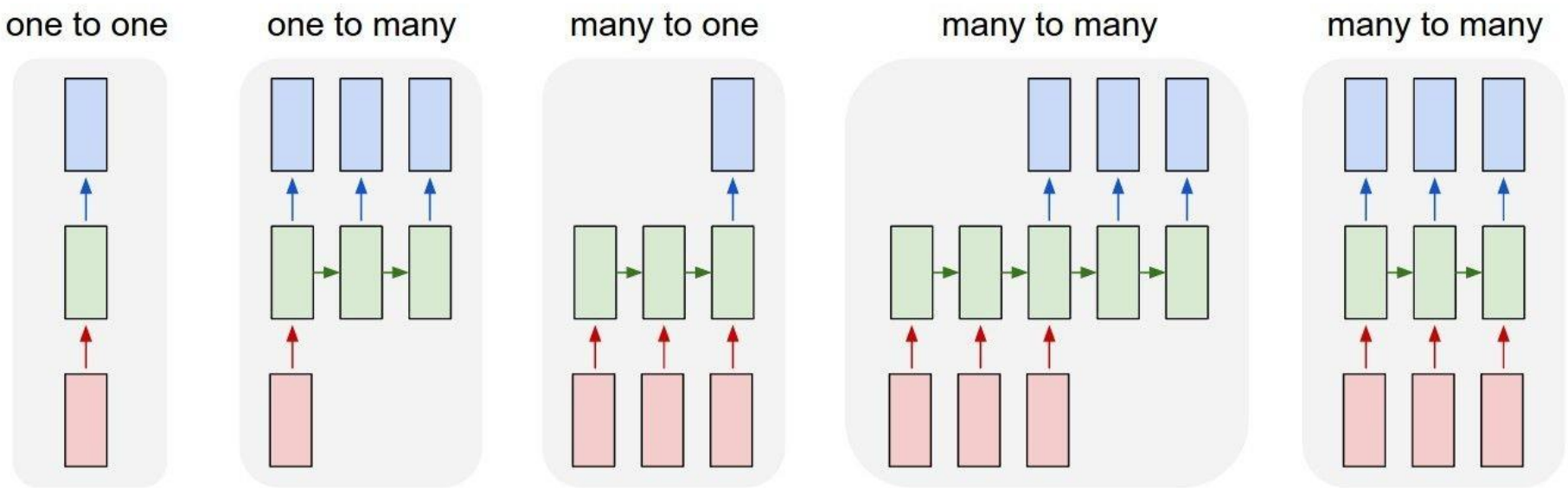
    - The influence never disappears unless forget gate is closed

    No Gradient vanishing
    (unless forget gate is opened.)



RNN = LSTM and LSTM=RNN!

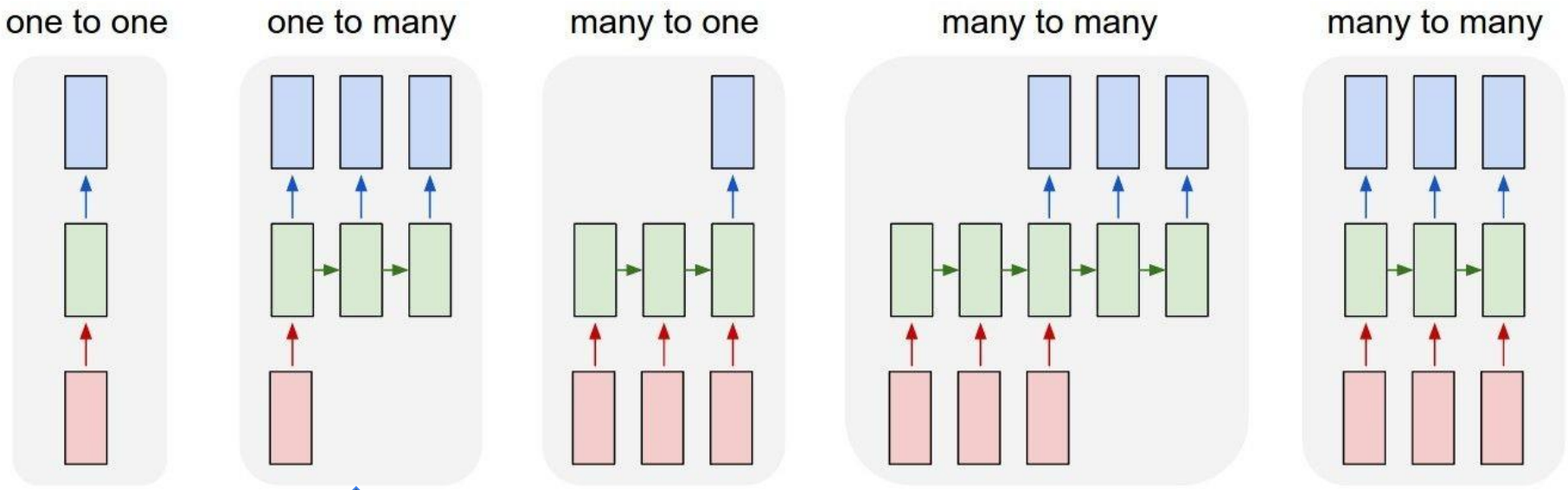
- LSTM is supported by caffe, Tensorflow, Keras



Vanilla Neural Networks

RNN = LSTM and LSTM=RNN!

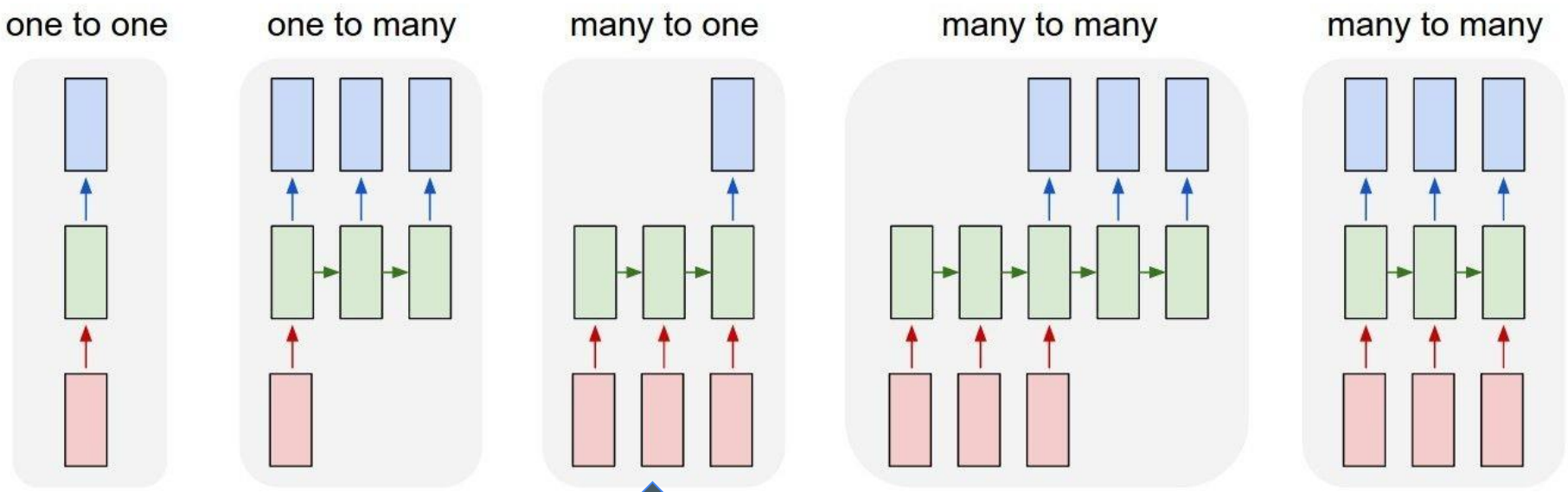
- LSTM is supported by caffe, Tensorflow, Keras



e.g. **Image Captioning**
image -> sequence of words

RNN = LSTM and LSTM=RNN!

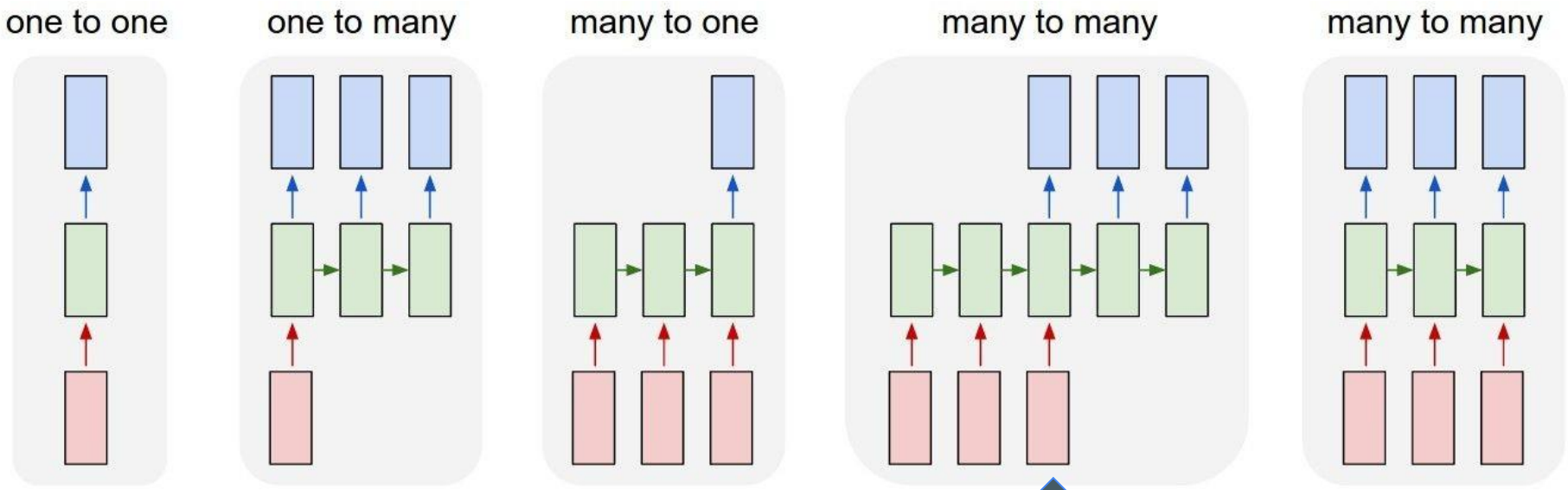
- LSTM is supported by caffe, Tensorflow, Keras



e.g. **Sentiment Classification**
sequence of words -> sentiment

RNN = LSTM and LSTM=RNN!

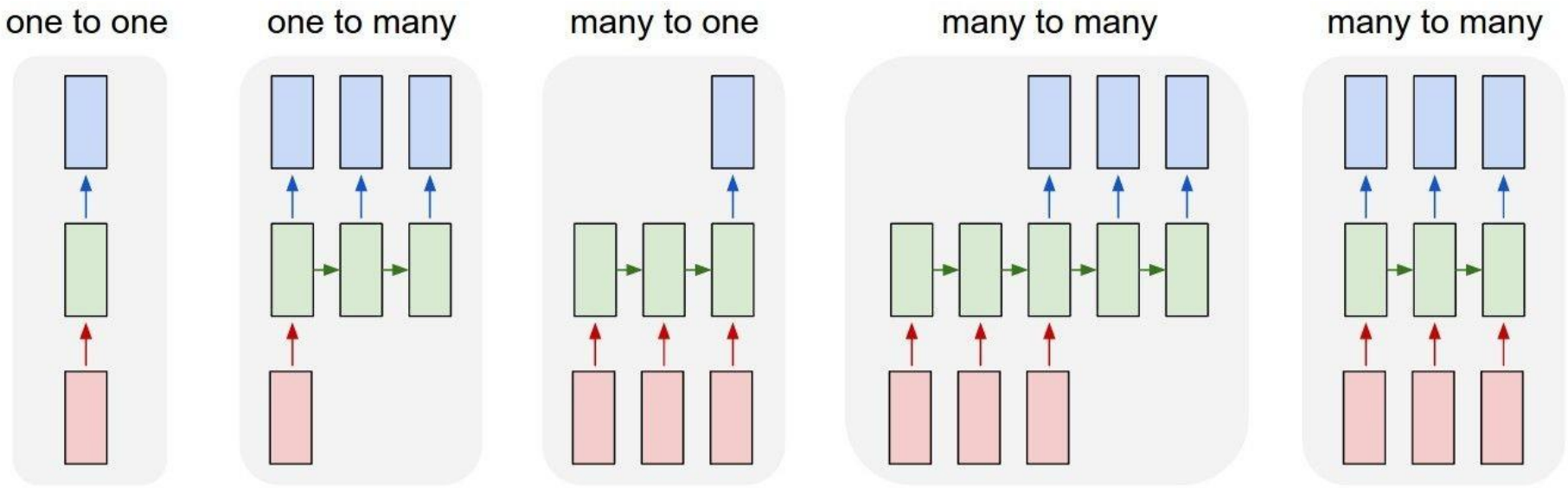
- LSTM is supported by caffe, Tensorflow, Keras



e.g. **Machine Translation**
seq of words -> seq of words

RNN = LSTM and LSTM=RNN!

- LSTM is supported by caffe, Tensorflow, Keras



e.g. **Video classification on frame level**

test image

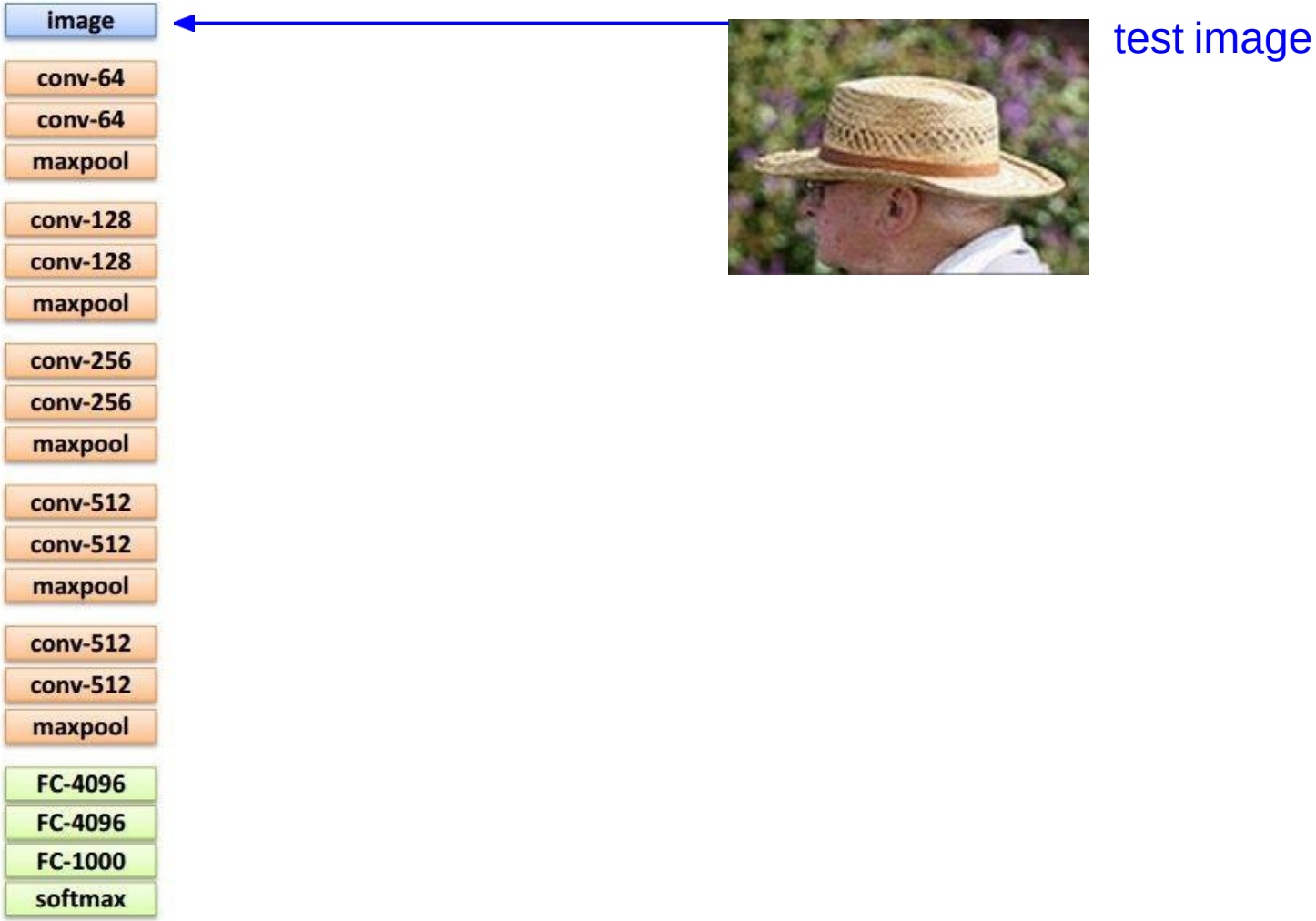


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
test image

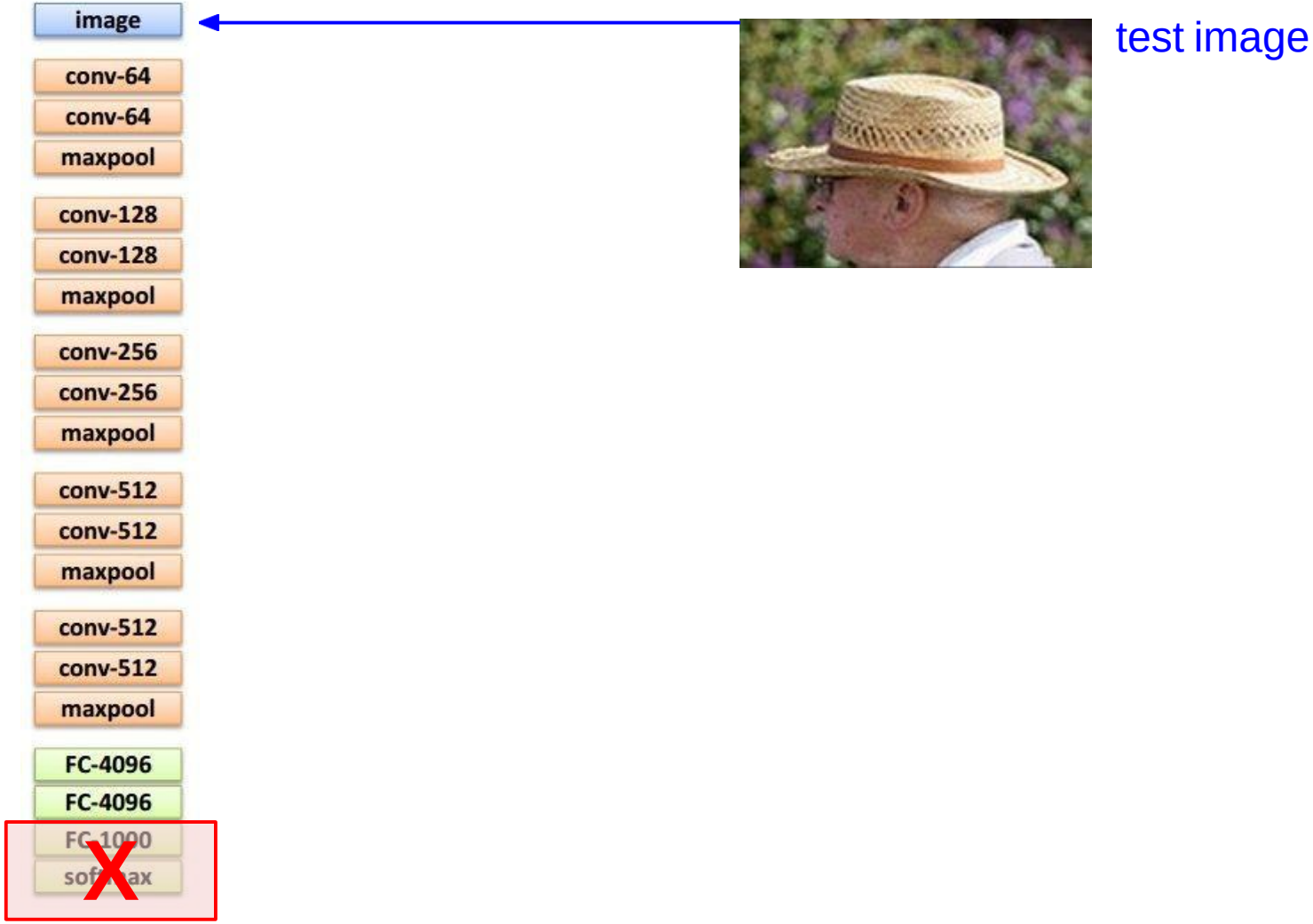


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
test image

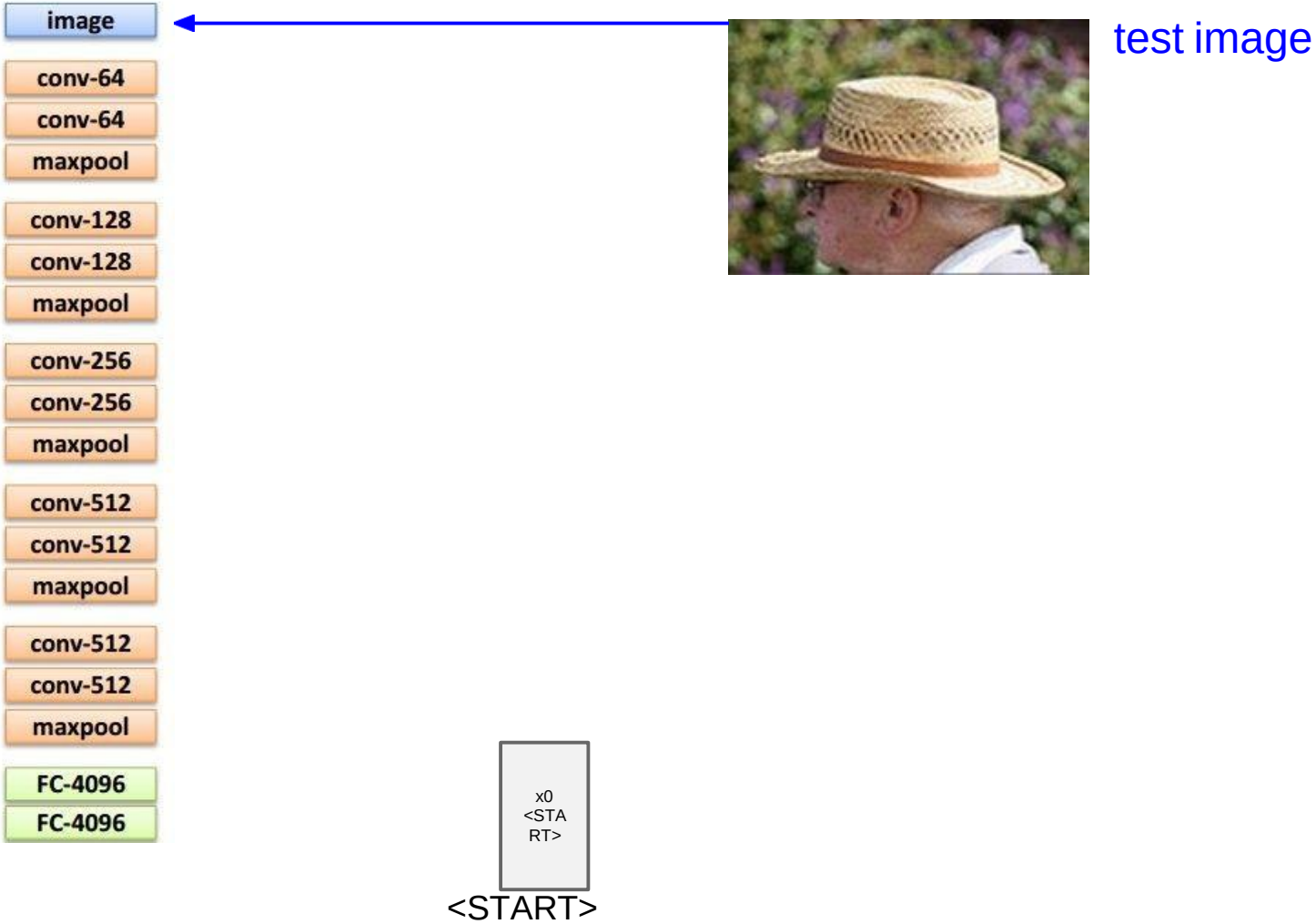
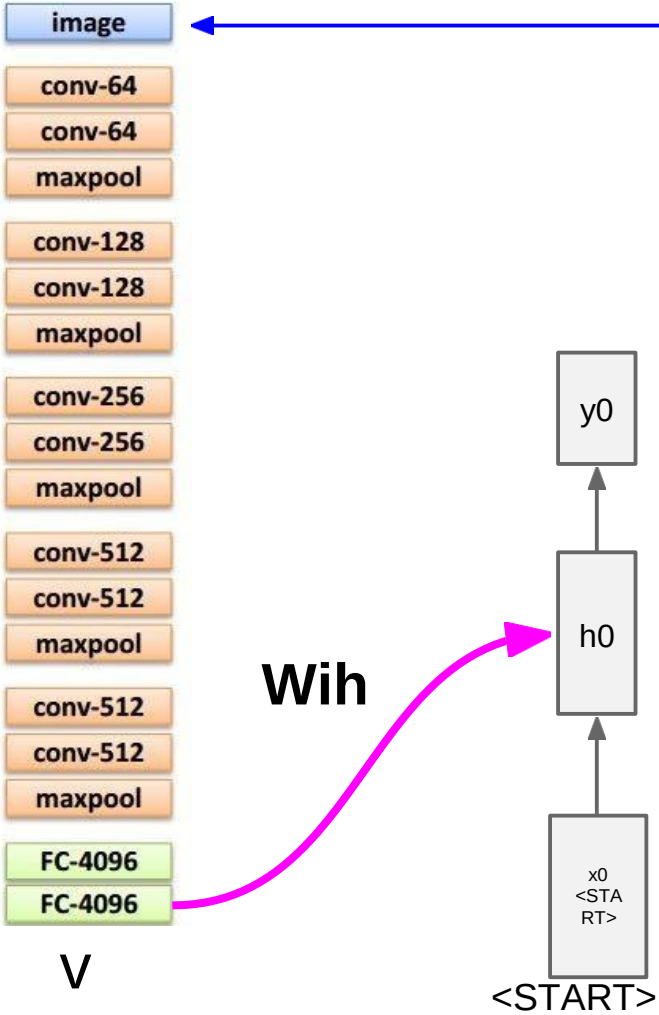


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
x0
<STA
RT>
<START>



test image

before:

h = tanh(Wxh * x + Whh * h)

now:

h = tanh(Wxh * x + Whh * h **+ Wih * v**)

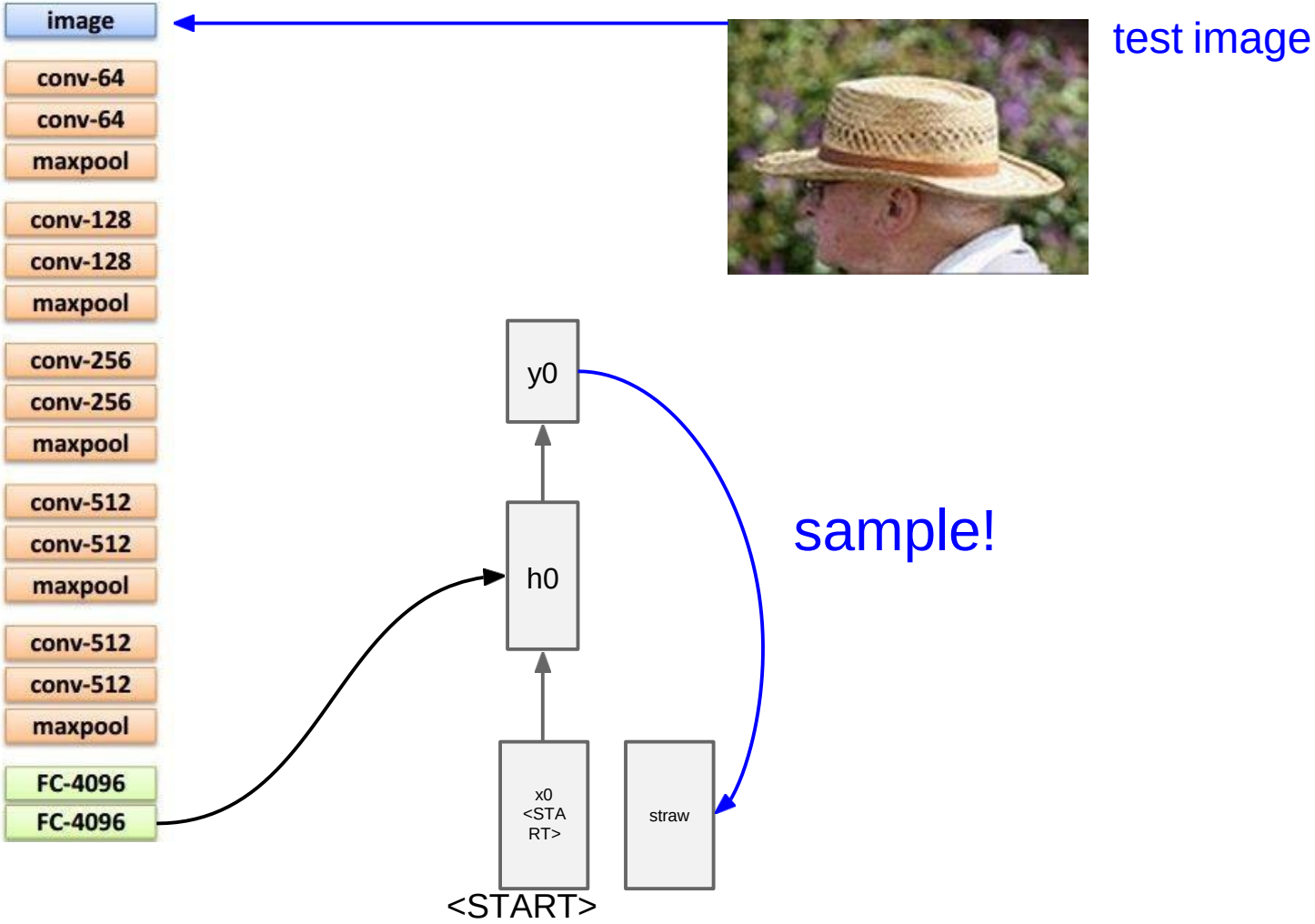


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
h0
sample!
x0
<STA
RT>
straw
<START>

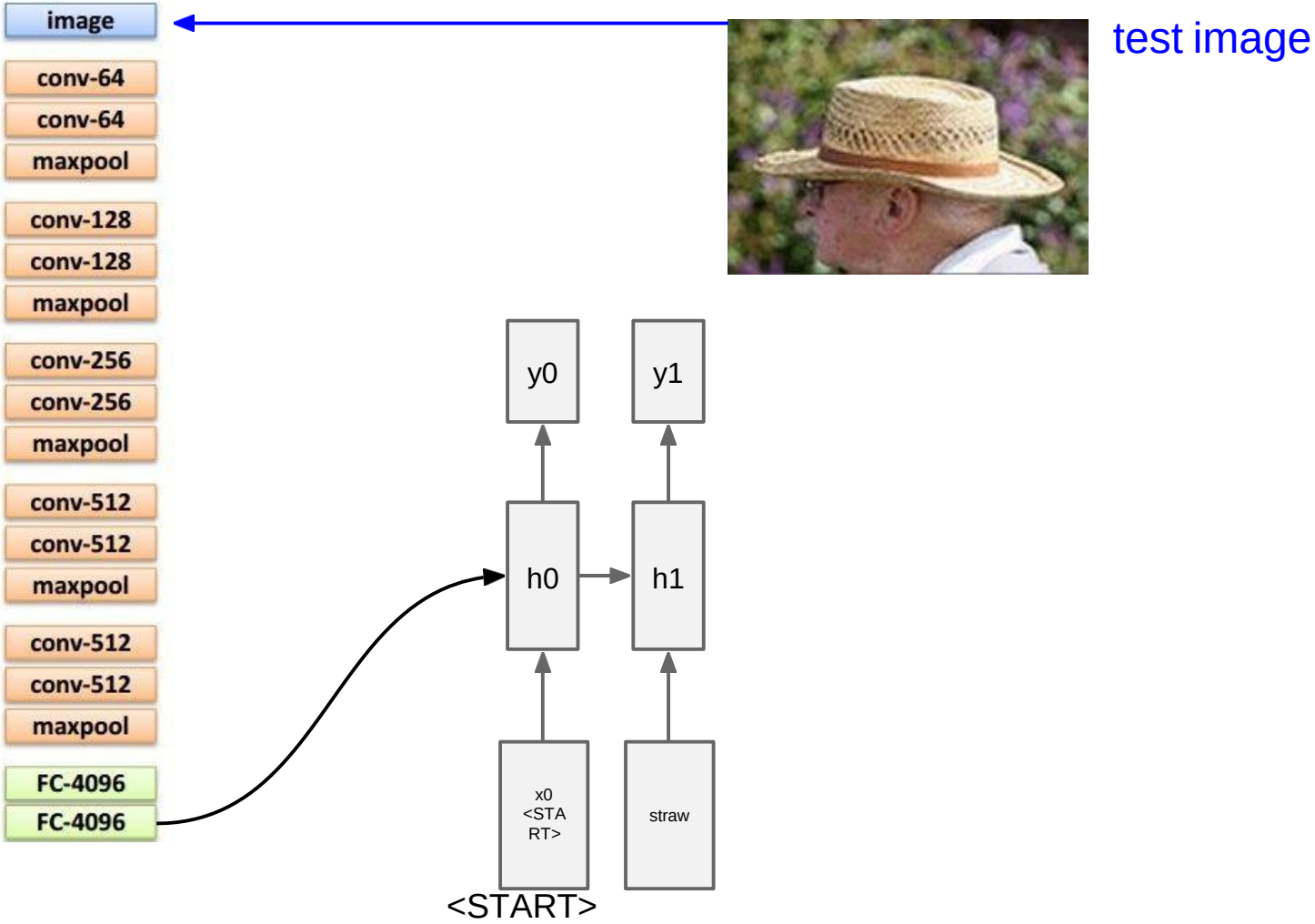


image
test image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
y0
y1
h0
h1
x0
<STA
RT>
straw
<START>

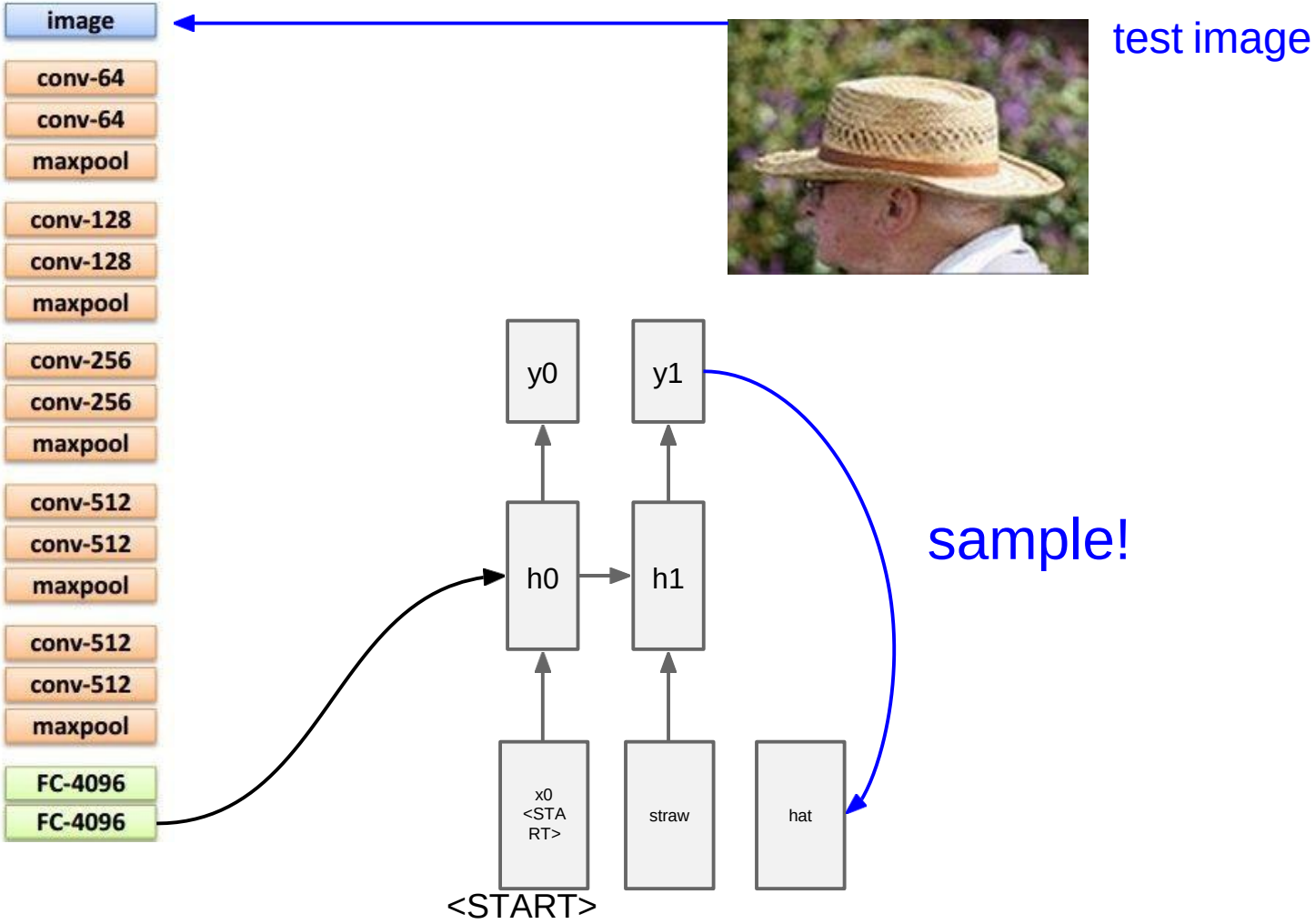


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
sample!
x0
<STA
RT>
straw
hat
<START>

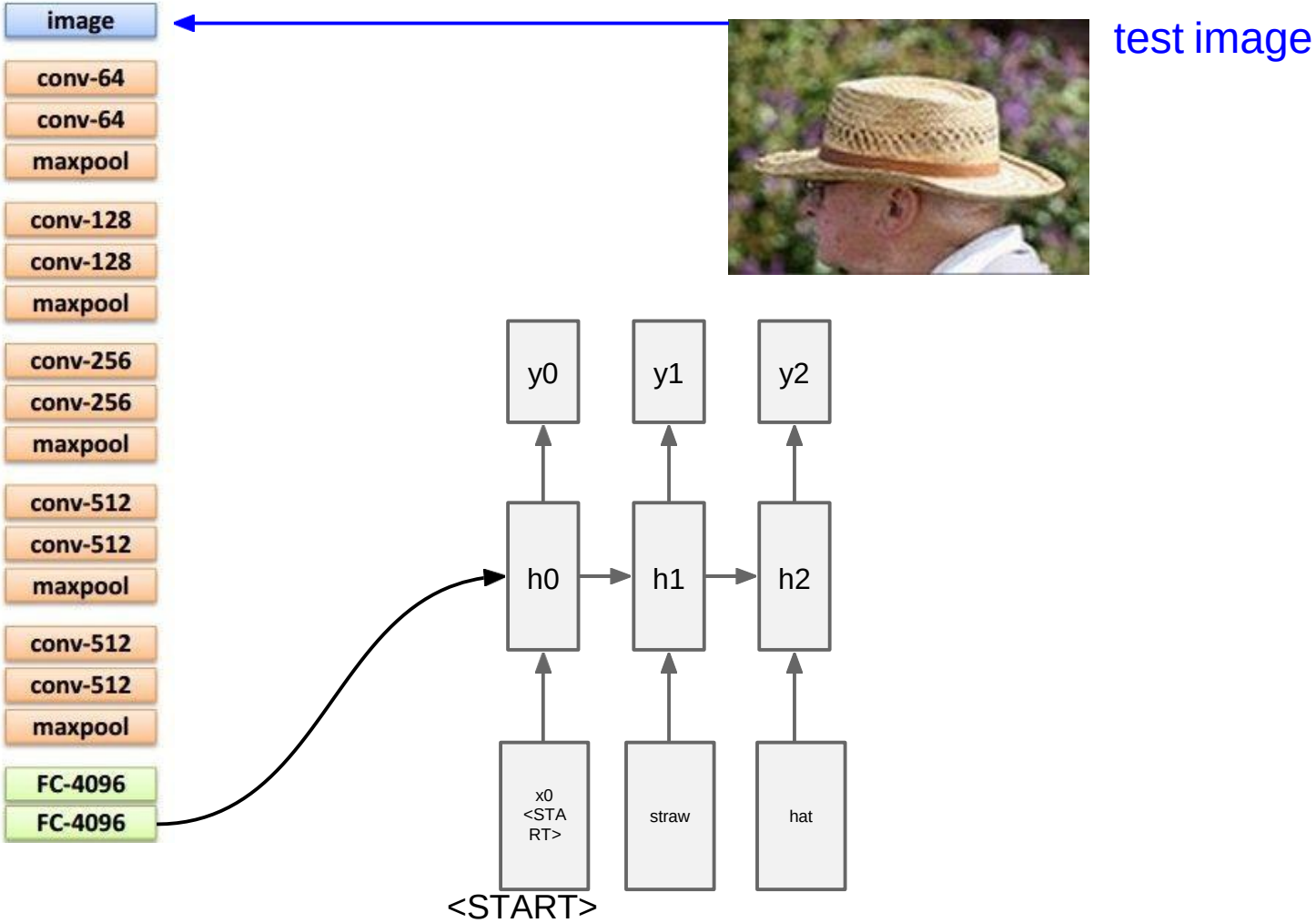


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0
<STA
RT>
straw
hat
<START>

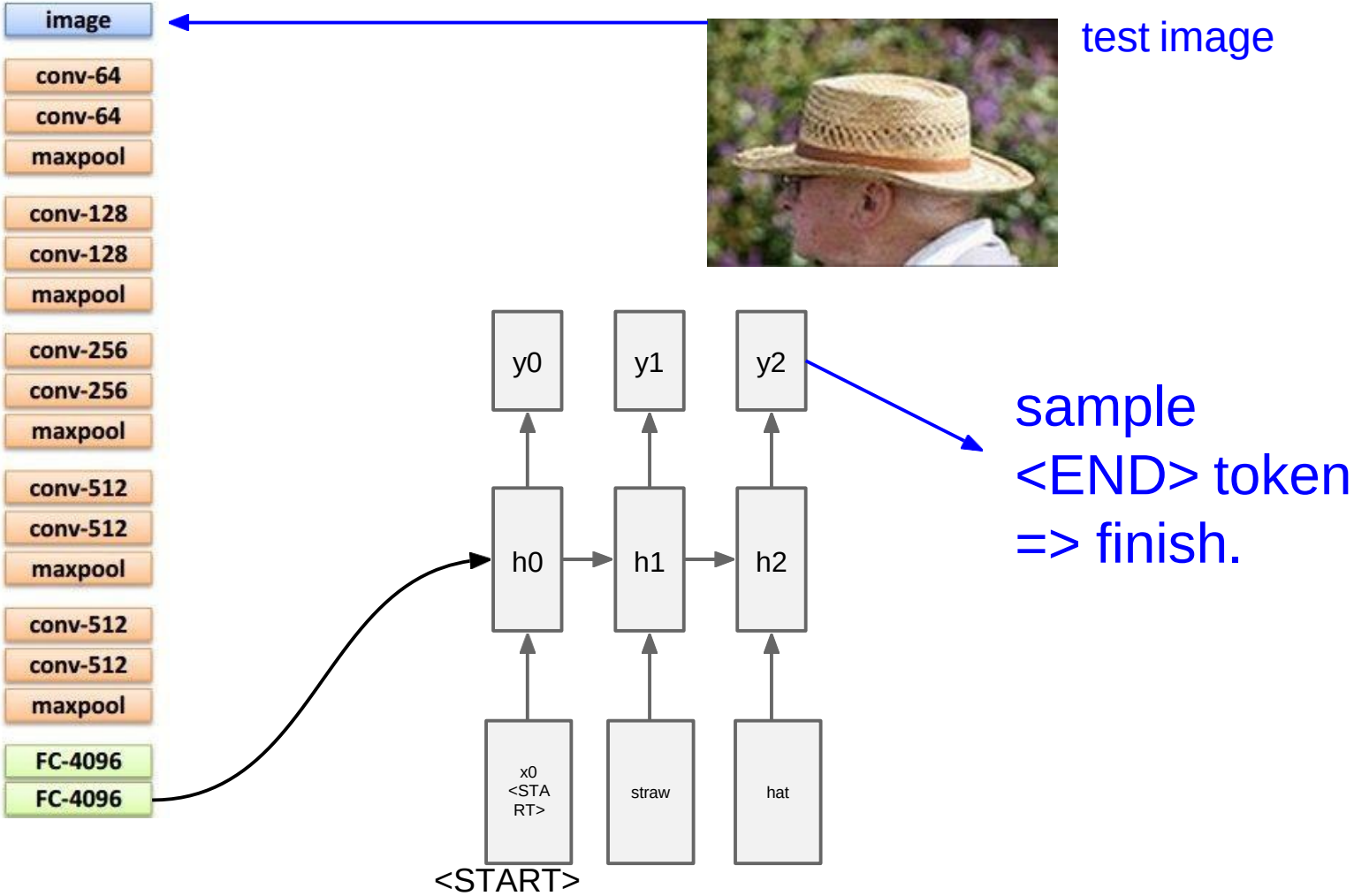


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0
<STA
RT>
straw
hat
<START>
sample
<END> token
=> finish.

Image Sentence Datasets

a man riding a bike on a dirt path through a forest.
bicyclist raises his fist as he rides on desert dirt trail.
this dirt bike rider is smiling and raising his fist in triumph.
a man riding a bicycle while pumping his fist in the air.
a mountain biker pumps his fist in celebration.

Microsoft COCO

[Tsung-Yi Lin et al. 2014]

mscoco.org

currently:

~120K images

~5 sentences each

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

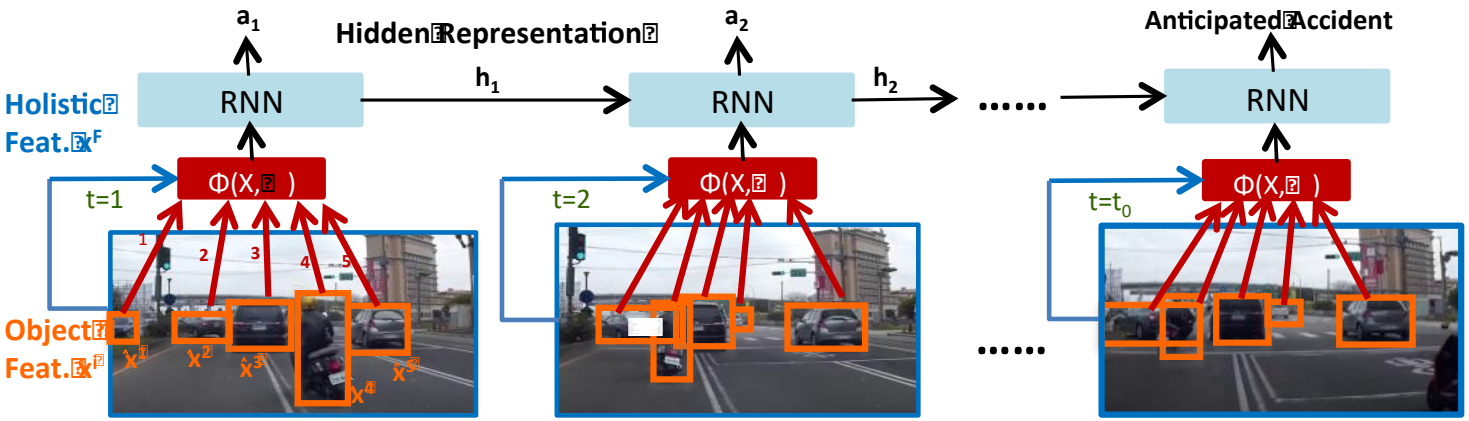
"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

"a horse is standing in the middle of a road."

ADAS application: Accident Prediction



Video Demo

2
ELEVEN
2015/08/22 13:40:45 DOD RX7W
Box attention
high
low
Focus on
the box
weight > 0.4
Probability
1.0
0.8
0.6
0.4
0.2
0.0
Threshold
0
10
20
30
40
50
60
70
80
90
frame

Thank you!