Slides available at:

https://ppt.cc/ficc8x

AI Summer School 2019

Deep Transfer Learning and Representation Disentanglement for Visual Analysis

Yu-Chiang Frank Wang 王鈺強, Associate Professor

Dept. Electrical Engineering, National Taiwan University
Taipei, Taiwan

2019/08/14

About Myself

- **Research Interests**

  Computer Vision, Machine Learning, Deep Learning, Artificial Intelligence

- **Education**
  - PhD in ECE, Carnegie Mellon University, 2004 – 2009
  - MS in ECE, Carnegie Mellon University, 2002 – 2004
  - BS in EE, National Taiwan University, 1997 – 2001

- **Work Experience**
  - Associate Professor — **2017** – present

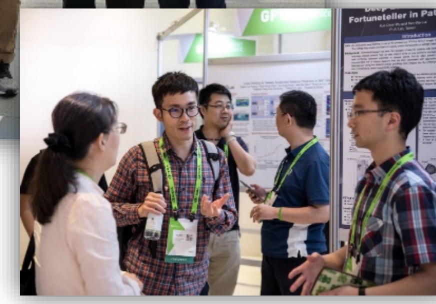
    GICE/EE, National Taiwan University
  - Deputy Director — 2015 – 2017

    Research Center for IT Innovation (CITI), Academia Sinica
  - Associate Research Fellow — **2013** – 2017

    CITI, Academia Sinica
  - Assistant Research Fellow — **2009** – 2013

    CITI, Academia Sinica

About Myself (cont'd)



- **Selected Honors & Awards**
  - **Outstanding Young Researcher**

    Ministry of Science & Technology
    2017-2019, 2013-2015
  - Nominated for **Best Paper Awards**

    IEEE AVSS 2015, IEEE ICME 2013
  - [2017/12] **1st Place Award**
    MOST Workshop on Generative Adversarial Networks & Project Competition
  - [2018/05] **2nd Place Award**
    NVIDIA GTC Taiwan 2018, Research Presentation
  - [2018/06] **2nd Place Award**
    CVPR 2018 Challenge on DeepGlobe (by Facebook & DigitalGlobe)
    - 1st Place Award by *Sensetime*
    - Other teams are from *MIT, Univ. Maryland*, etc.

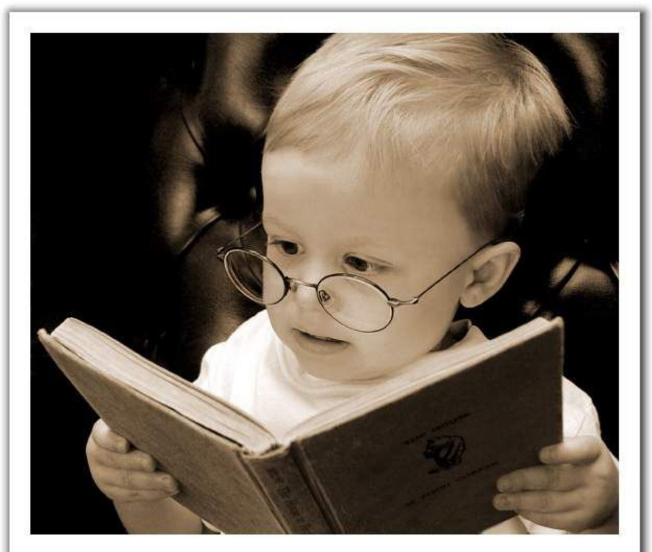
About Myself (cont'd)

- **Industrial Collaboration**
  - **Collaborators**:

  - **Remarks/Recognition**
    - "*9 startups to watch*", Alibaba Entrepreneurs Fund, 2017
    - "*10 coolest tech startups in Taiwan*", MOST, 2018
    - "*Top 9 AI startups in Taiwan*", Crunchbase, 2019

What Will Be Covered in Today's Lecture?

- Brief Review to CV/ML Backgrounds
- Recent Advances in Deep Learning for Computer Vision
- Transfer Learning and Its Applications to Image Analysis and Synthesis



Computer Vision: What, When, and Why

- **Remarks**
  - Give machines *visual perception*
  - Learning for visual data
  - In addition to Machine Learning, computer vision is closely related to Image Processing, Computer Graphics, Computational Photography, etc.

***How many** people are there?*

***What** are people doing?*

***What object** is the guy standing on?*

***Where** is this picture taken?*

*Why is this picture **funny**?*

Learning from Visual Data

- **Computer Vision**
  - Learning from visual data; give machines ***visual perception***
  - In addition to Machine Learning, computer vision is closely related to Image Processing, Computer Graphics, Computational Photography, etc.

***How many** people are there?*

***What** are people doing?*

***What object** is the guy standing on?*

***Where** is this picture taken?*

*Why is this picture **funny**?*

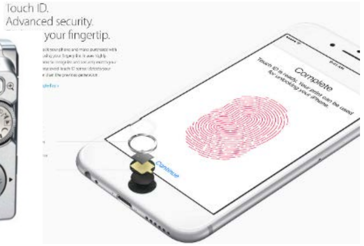

2000
10-year challenge?


2018

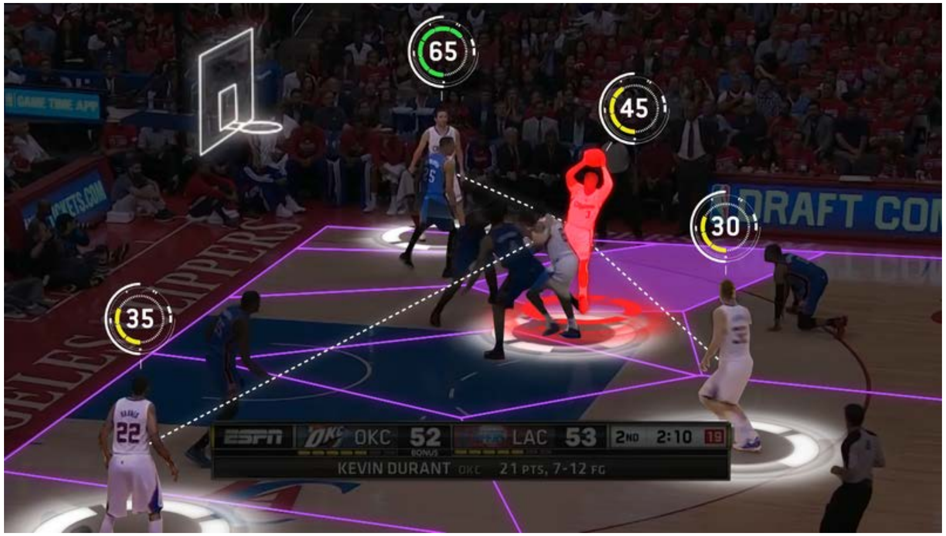
Learning from Visual Data (cont'd)

- Existing CV Applications
  - Biometrics (e.g., face, iris, gait recognition)
  - Optical character recognition (OCR)
  - Sports (tennis, football, basketball, etc.)

  And many more…



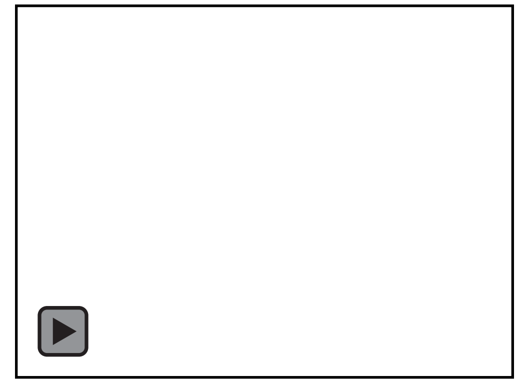
Google Translate

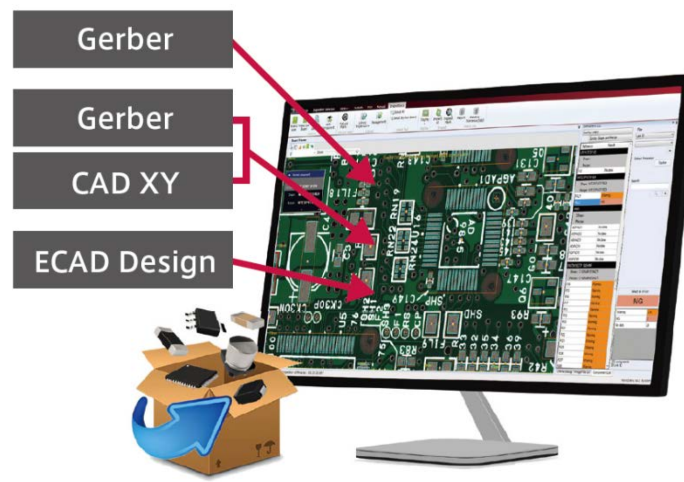


From Computer Vision to Artificial Intelligence

- Coming AI+CV Applications
  - Virtual/augmented reality (VR/AR)
  - Automated optical inspection (AOI)
  - Self-driving car
  - Industrial robots
  - Medical imaging

  And increasingly more than we can imagine!





Style Transfer

[EverFilter](#)

Style Transfer

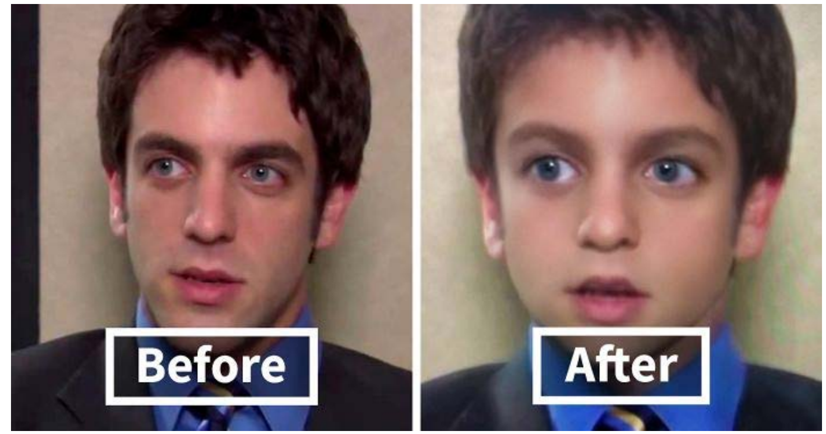
Snapchat

Snap Inc 社交

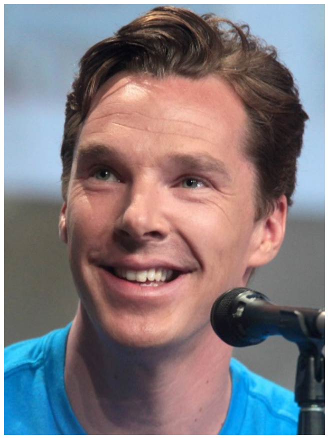
輔導級

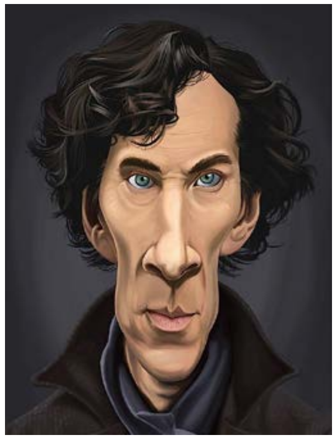
含廣告內容

你沒有任何裝置。



More Examples for Style Transfer

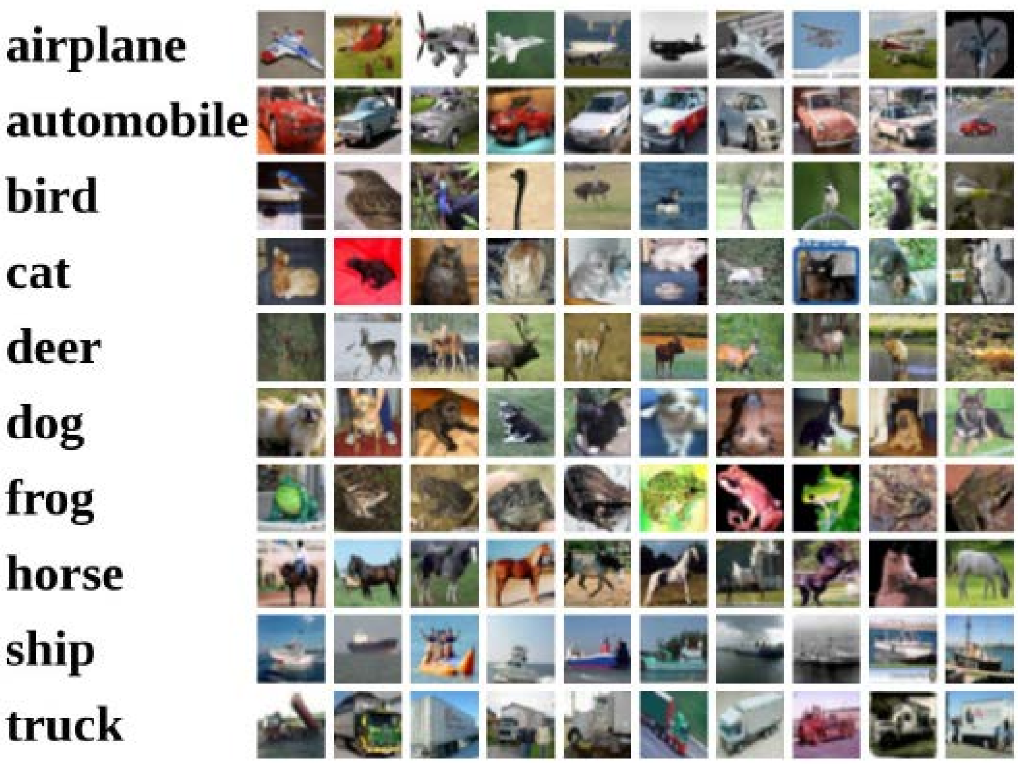




Abduzeedo.com

Take Visual Classification for Example

- Linear Classifier as the Learning Model
  - Can be viewed as a parametric approach. Why?
  - Assuming that we need to recognize 10 object categories of interest
  - E.g., CIFAR10 with 50K training & 10K test images of 10 categories. And, each image is of size 32 x 32 x 3 pixels.



- Linear Classifier as the Learning Model (cont'd)
  - Can be viewed as a parametric approach. Why?
  - Assuming that we need to recognize 10 object categories of interest (e.g., CIFAR10).
  - Let's take the input image as **x**, and the linear classifier as **W**.
  - We hope to see that **y** = **Wx** + **b** as a 10-dimensional output, in which each entry indicates the score of the associated class.

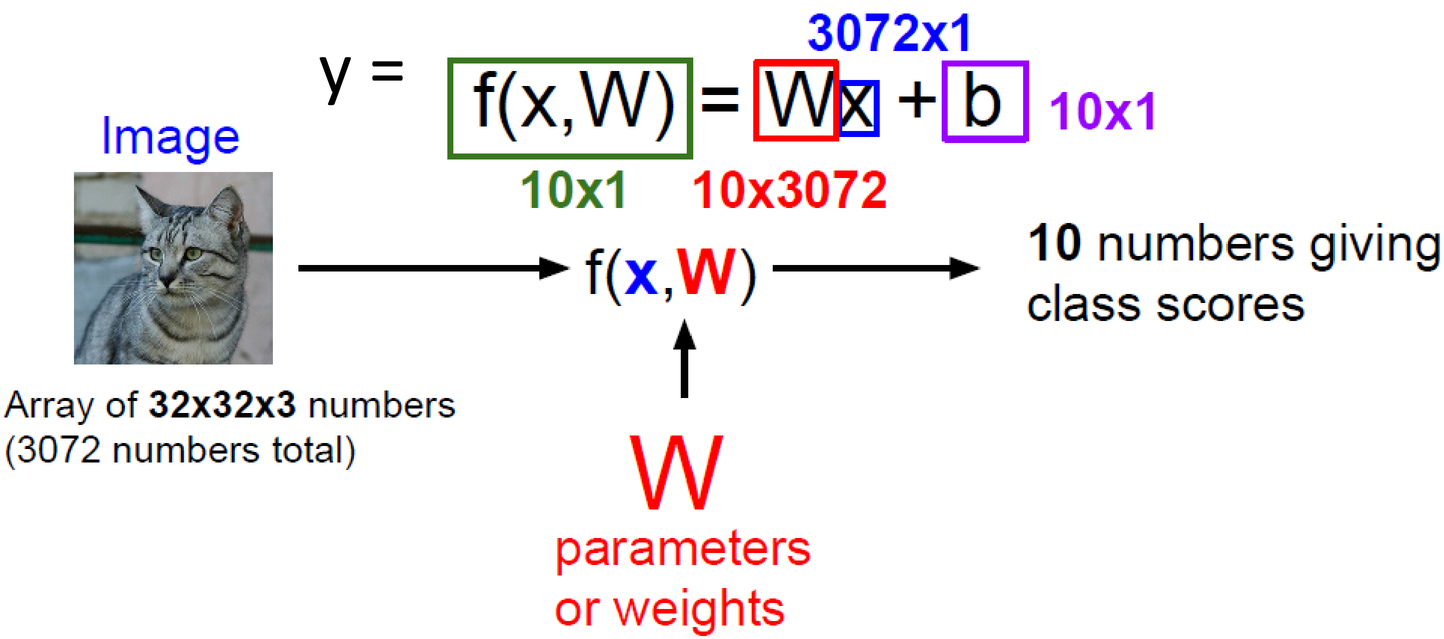


Image credit: Stanford CS231n

- Linear Classifier as the Learning Model (cont'd)
  - Take an image with 2 x 2 pixels & 3 classes of interest as example.
  - We need to learn a linear classifier **W** (with a bias **b**), so that a set of desirable outputs **y** = **Wx** + **b** can be expected.

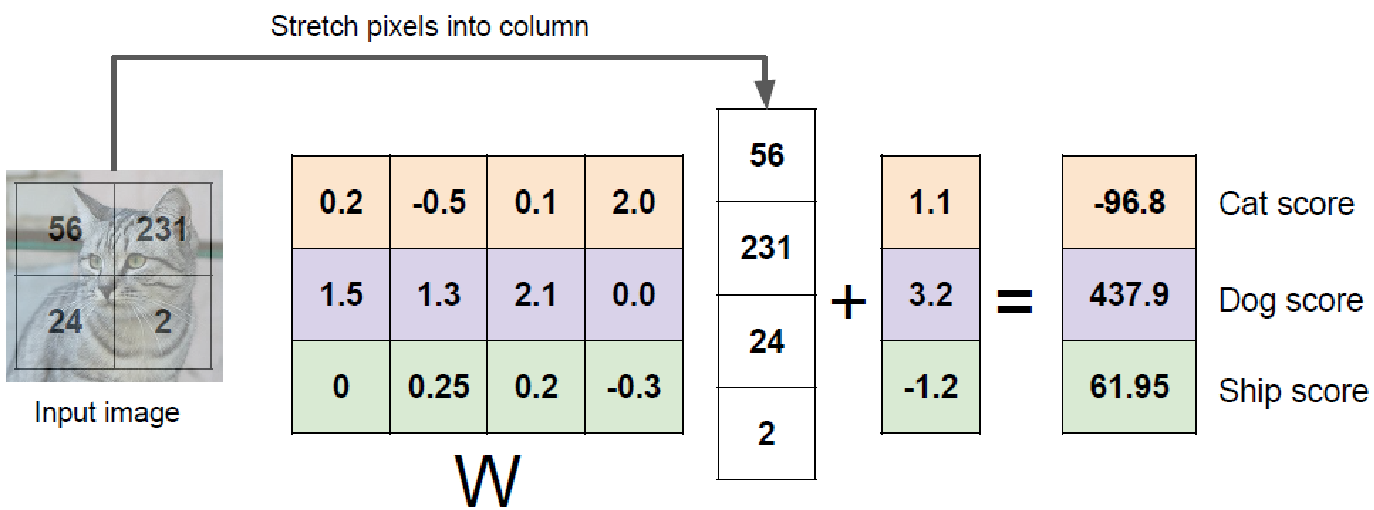
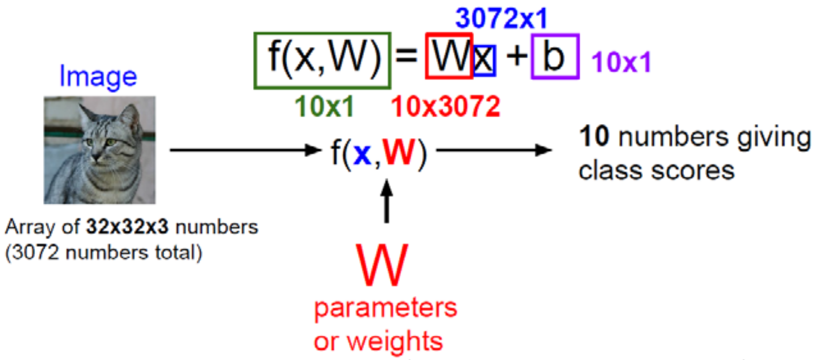
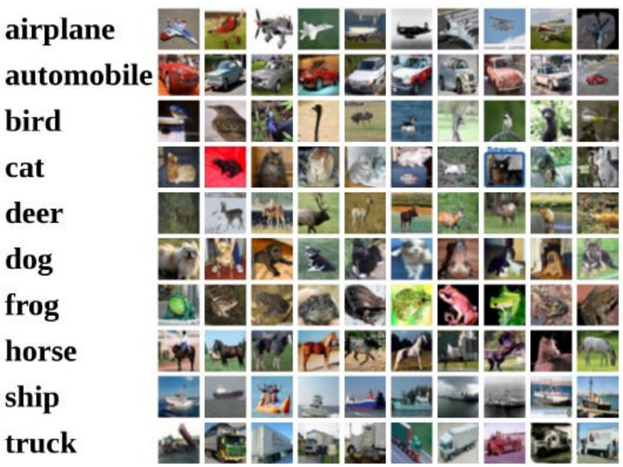


Image credit: Stanford CS231n

Some Remarks



- Interpreting the classifier **W**
  - The weights in **W** are trained by observing training data **X** and their ground truth **Y**.
  - Each column in **W** can be viewed as an "exemplar" of the corresponding class.
  - Thus, **Wx** basically performs inner product (or correlation) between the "input **x**" and the "exemplar of each class".



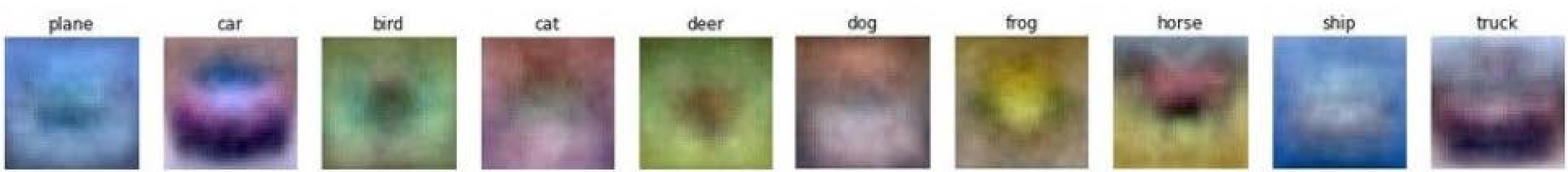
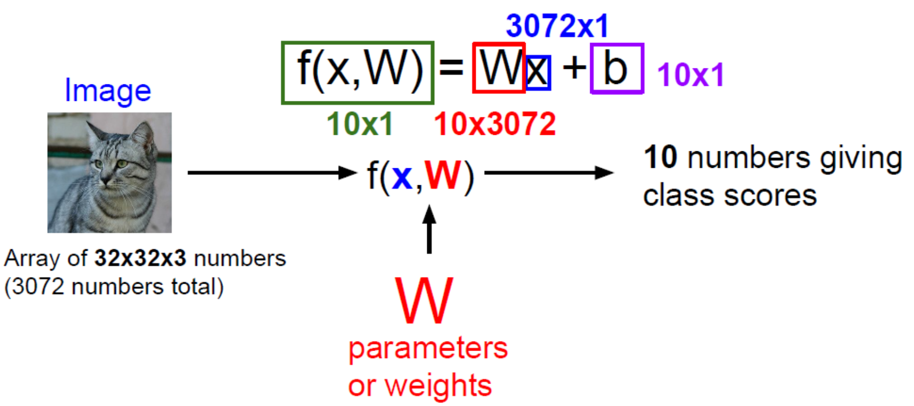


Image credit: Stanford CS231n

Some Remarks (cont'd)

- From Linear to Non-Linear Classifiers
  - Starting points for complex/nonlinear classifiers
  - How to determine a proper loss function for matching **y** and **Wx+b**, followed by the learning of **W** (including **b**), are the keys to the success of ML models.



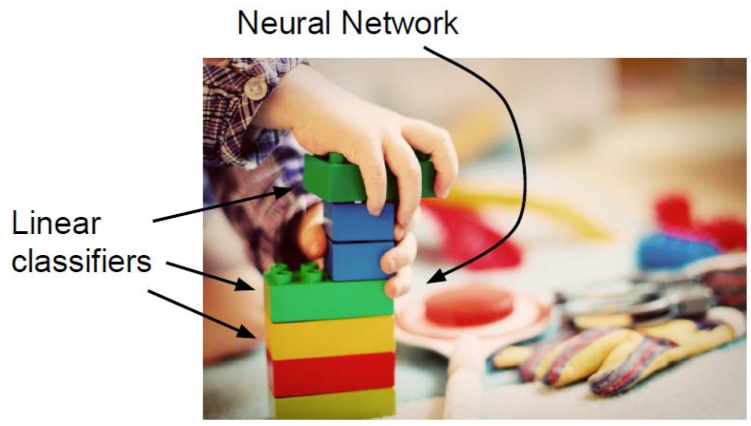
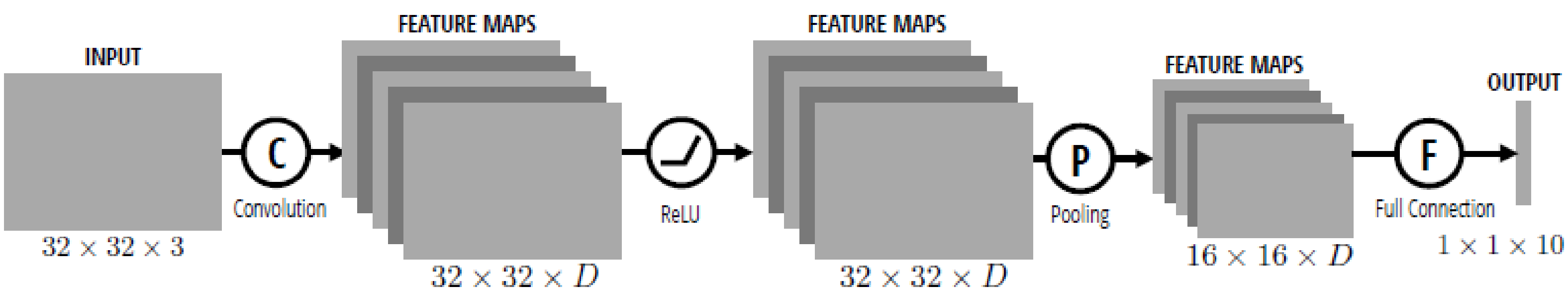


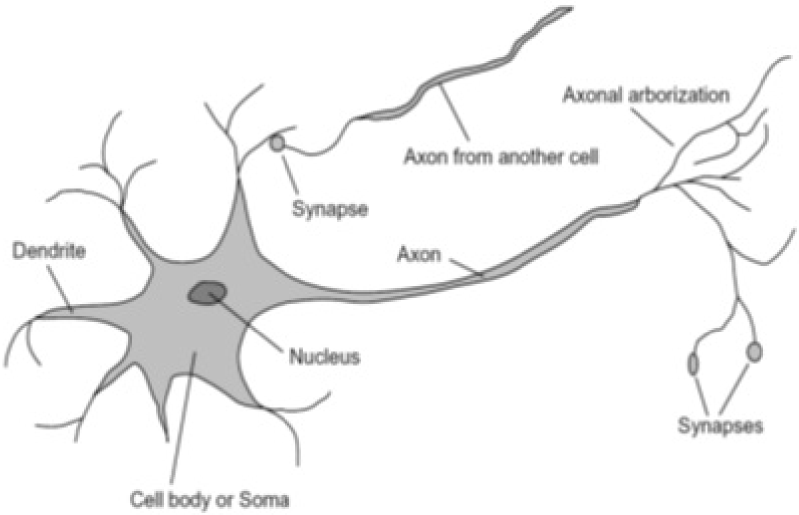
Image credit: Stanford CS231n

(A Very Quick) Intro to Neural Networks & CNN

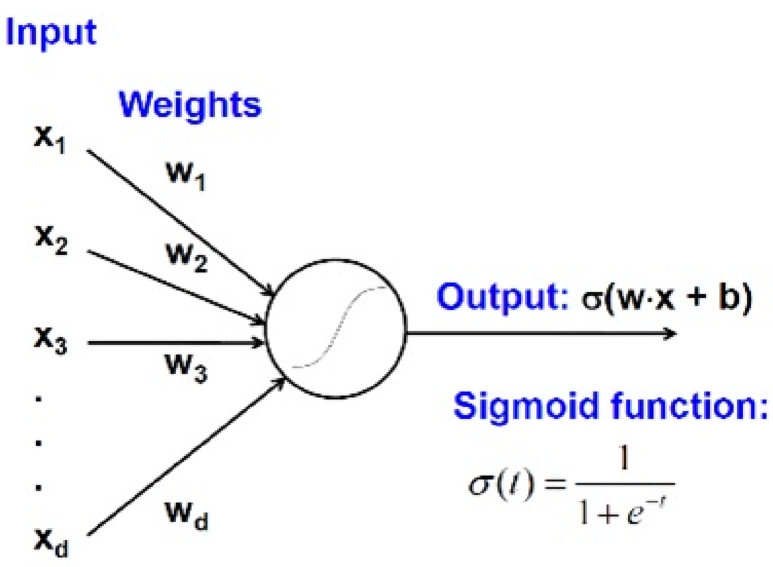
- Neural Network & Multilayer Perceptron
- Convolutional Neural Networks



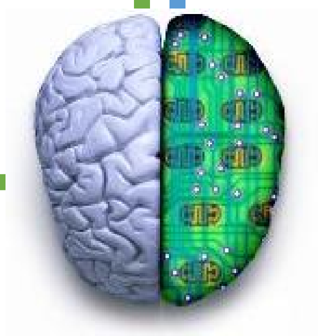
Biological neuron and Perceptrons



A biological neuron

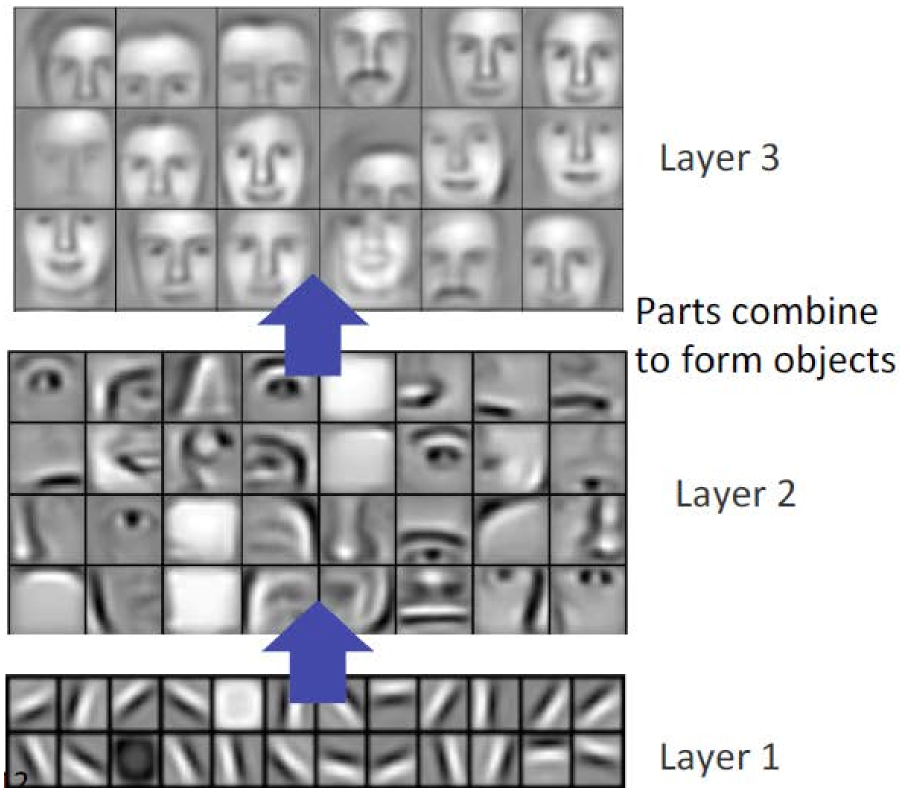


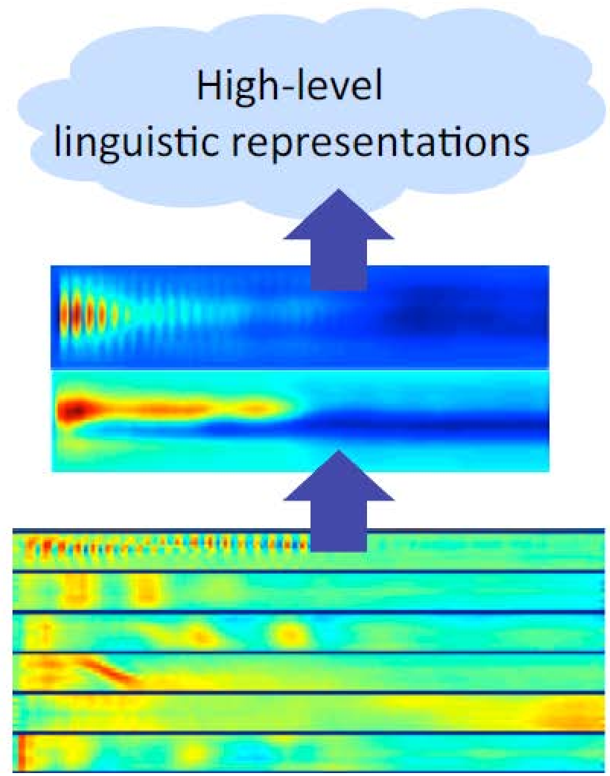
An artificial neuron (Perceptron)
- a linear classifier



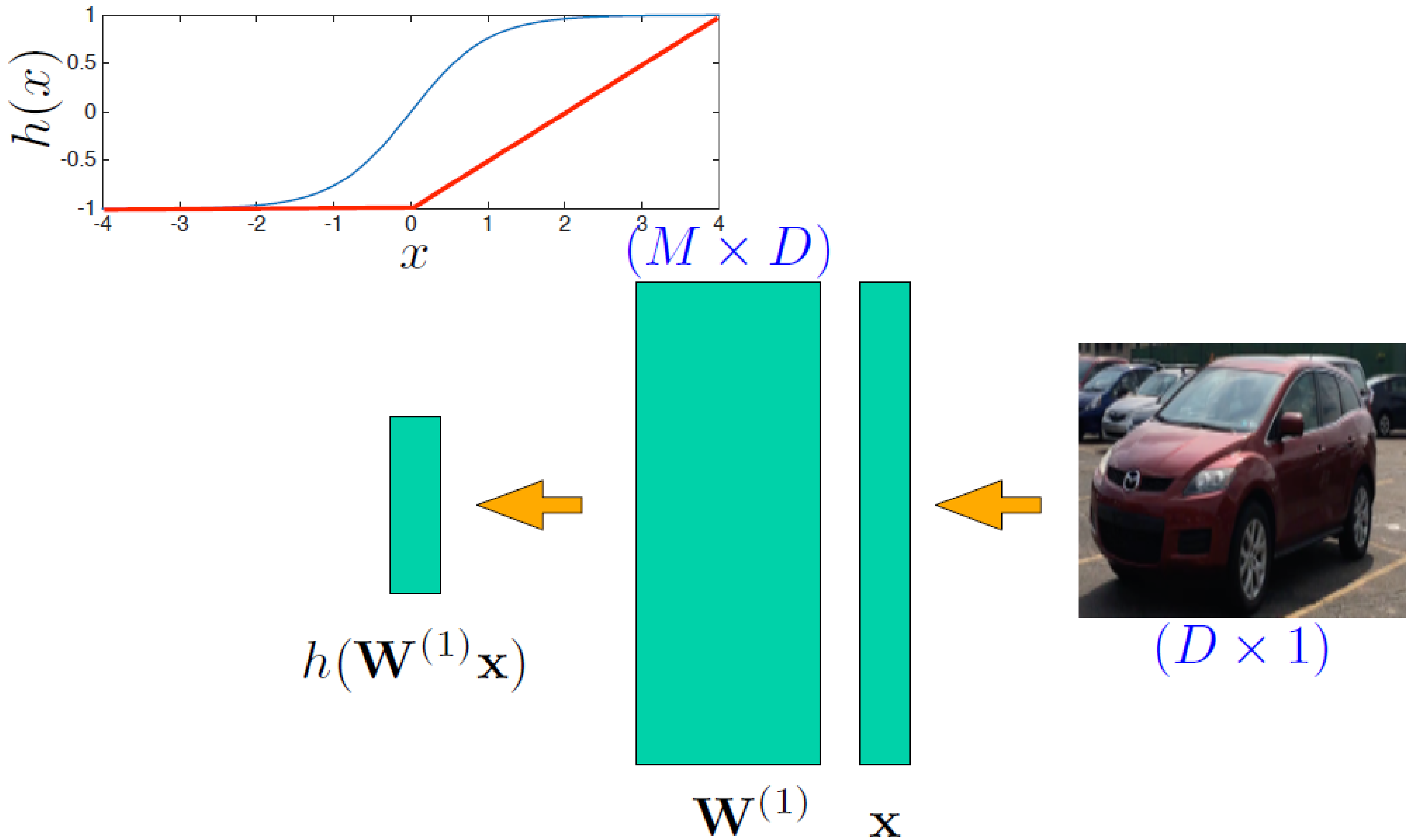
Hierarchical Learning

- Successive model layers learn deeper intermediate representations.

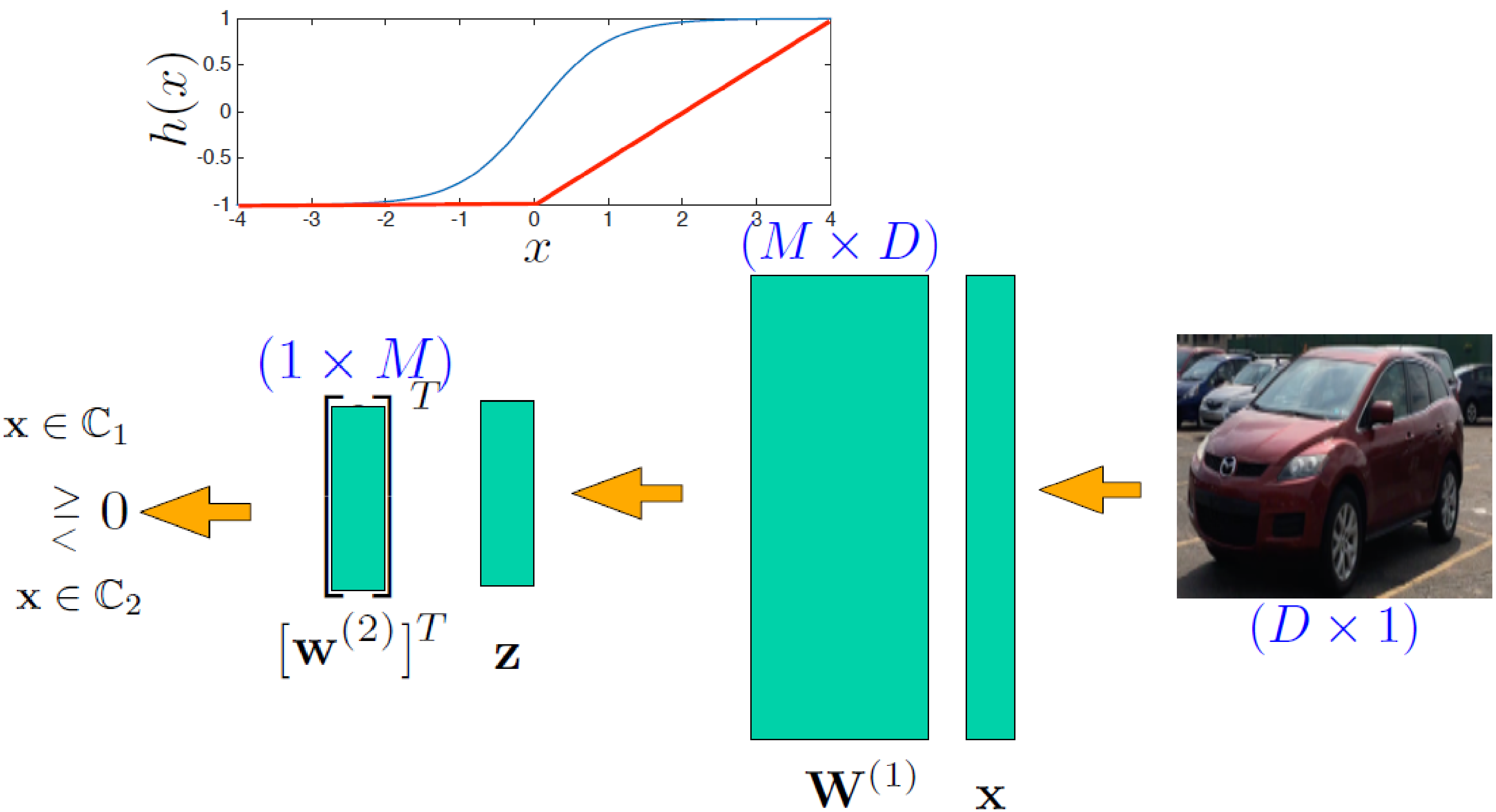




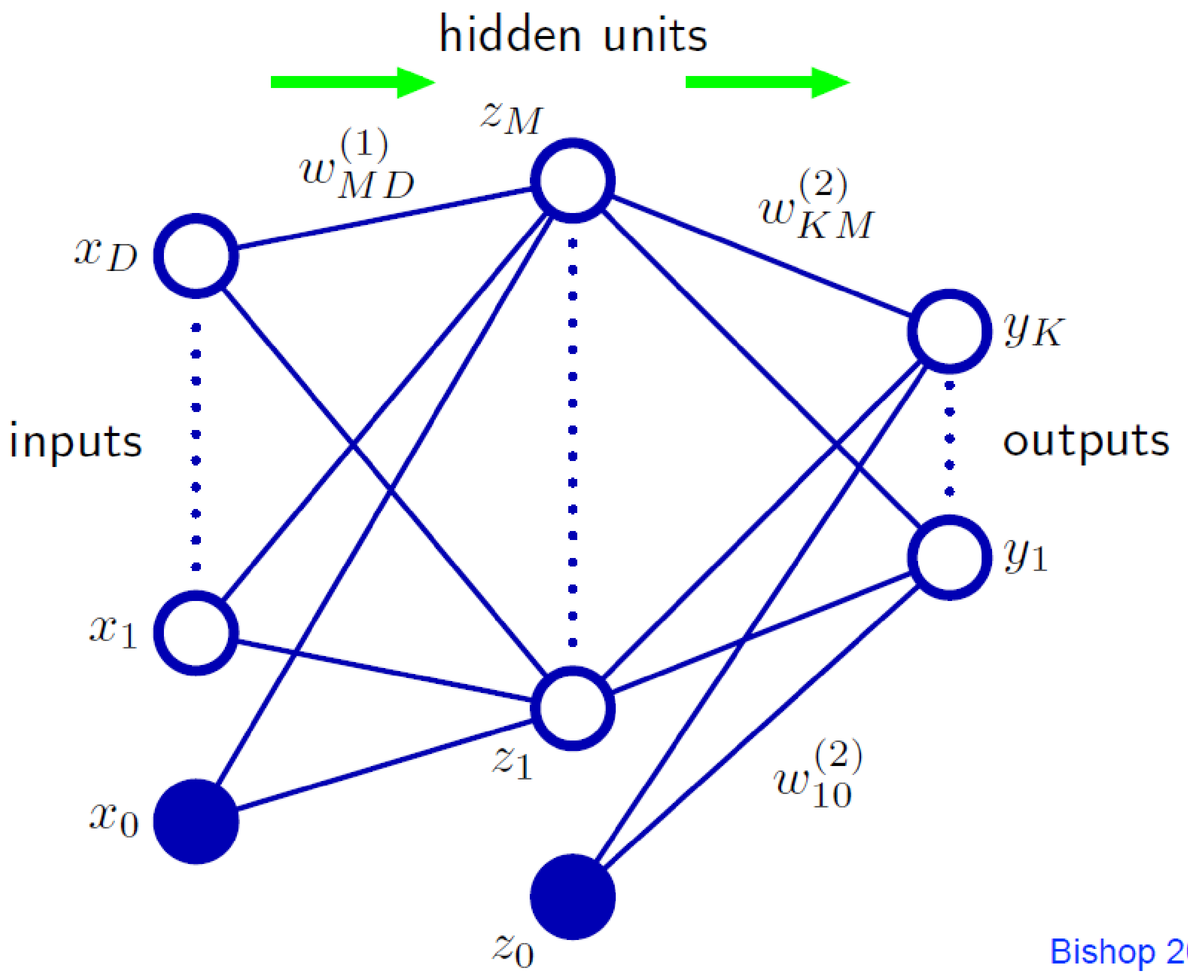
Multi-Layer Perceptron: A Nonlinear Classifier



Multi-Layer Perceptron: A Nonlinear Classifier (cont'd)



Multi-Layer Perceptron: A Nonlinear Classifier (cont'd)



Bishop 2006

Supervised Learning for Visual Classification

- General framework

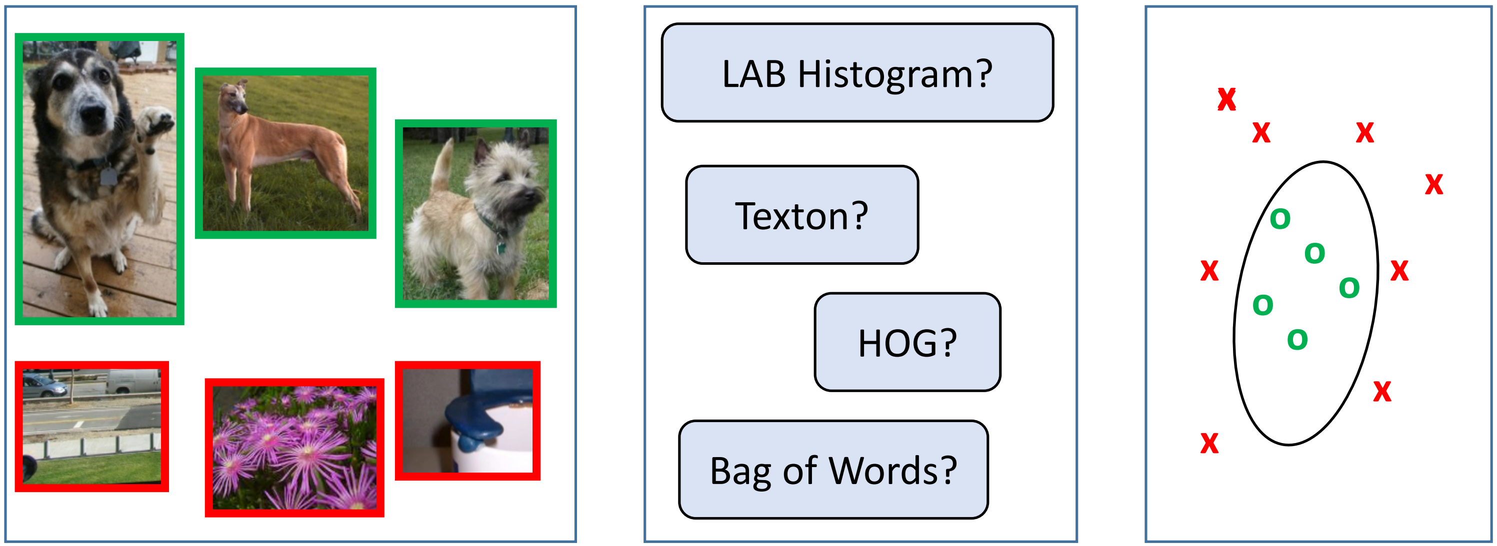
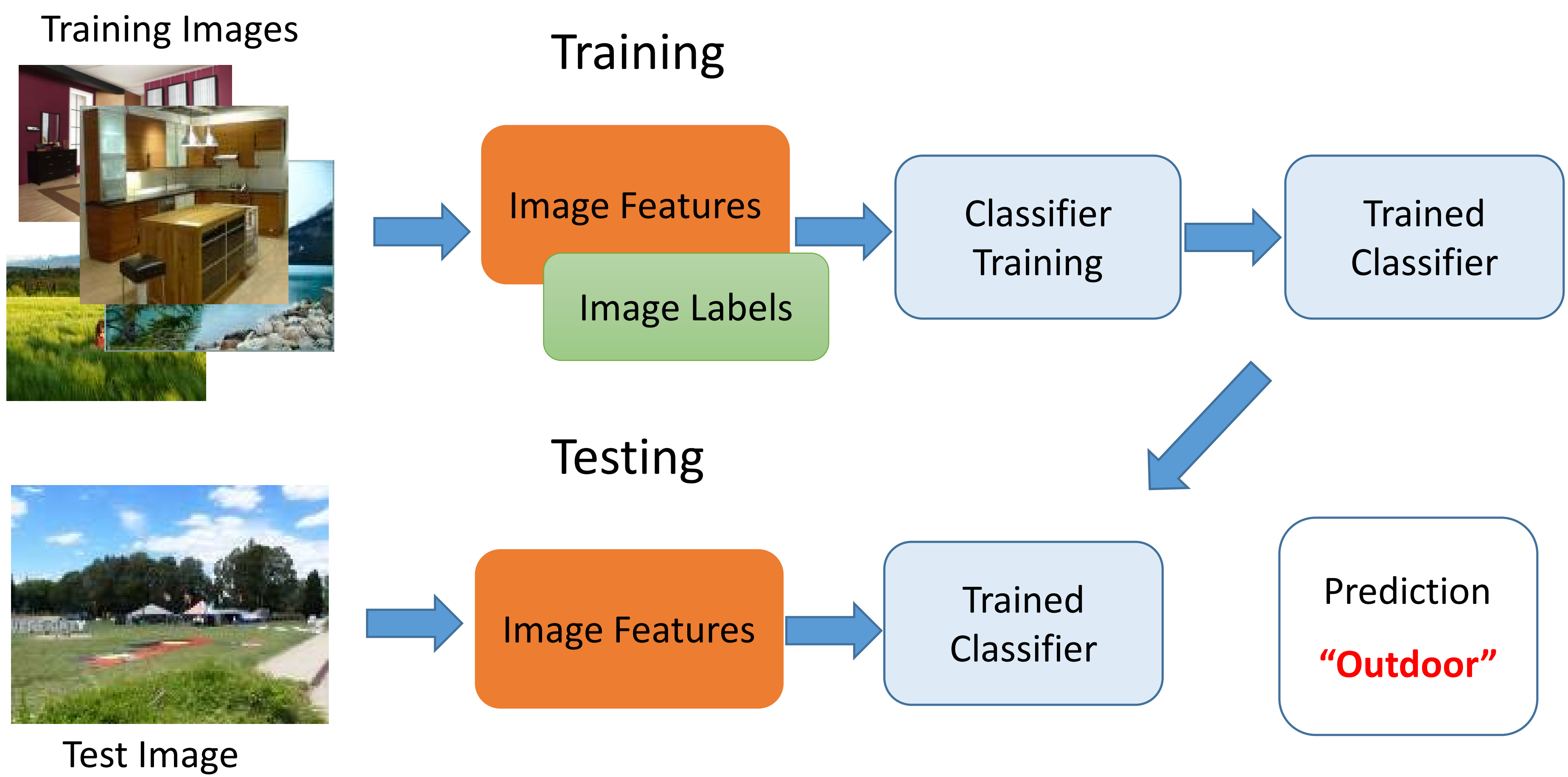


Image Data + Image Features + Classifier = Category label

Supervised Learning for Visual Classification

- Training vs. Testing Phases

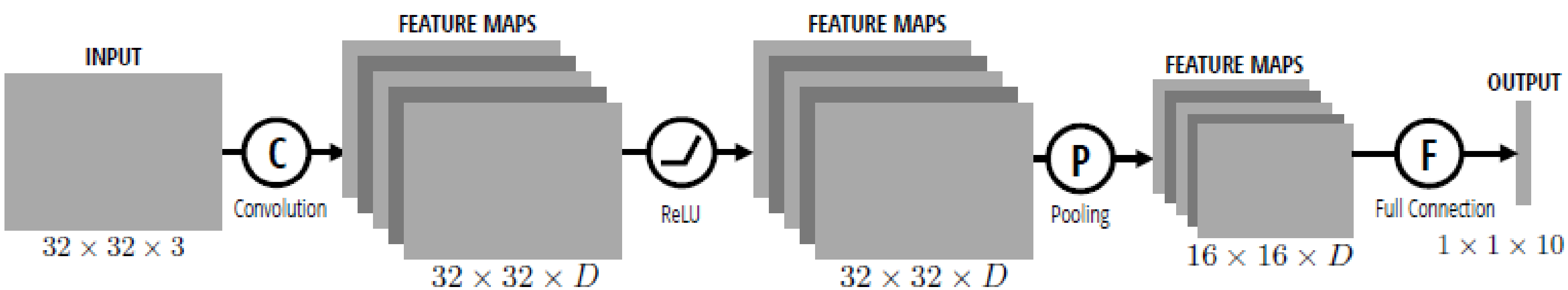


What Are the Right Features?

- Depending on the task of interest!
- Possible choices
  - Object: shape
    - Local shape info, shading, shadows, texture
  - Scene : geometric layout
    - linear perspective, gradients, line segments
  - Material properties: albedo, feel, hardness
    - Color, texture
  - Action: motion
    - Optical flow, tracked points

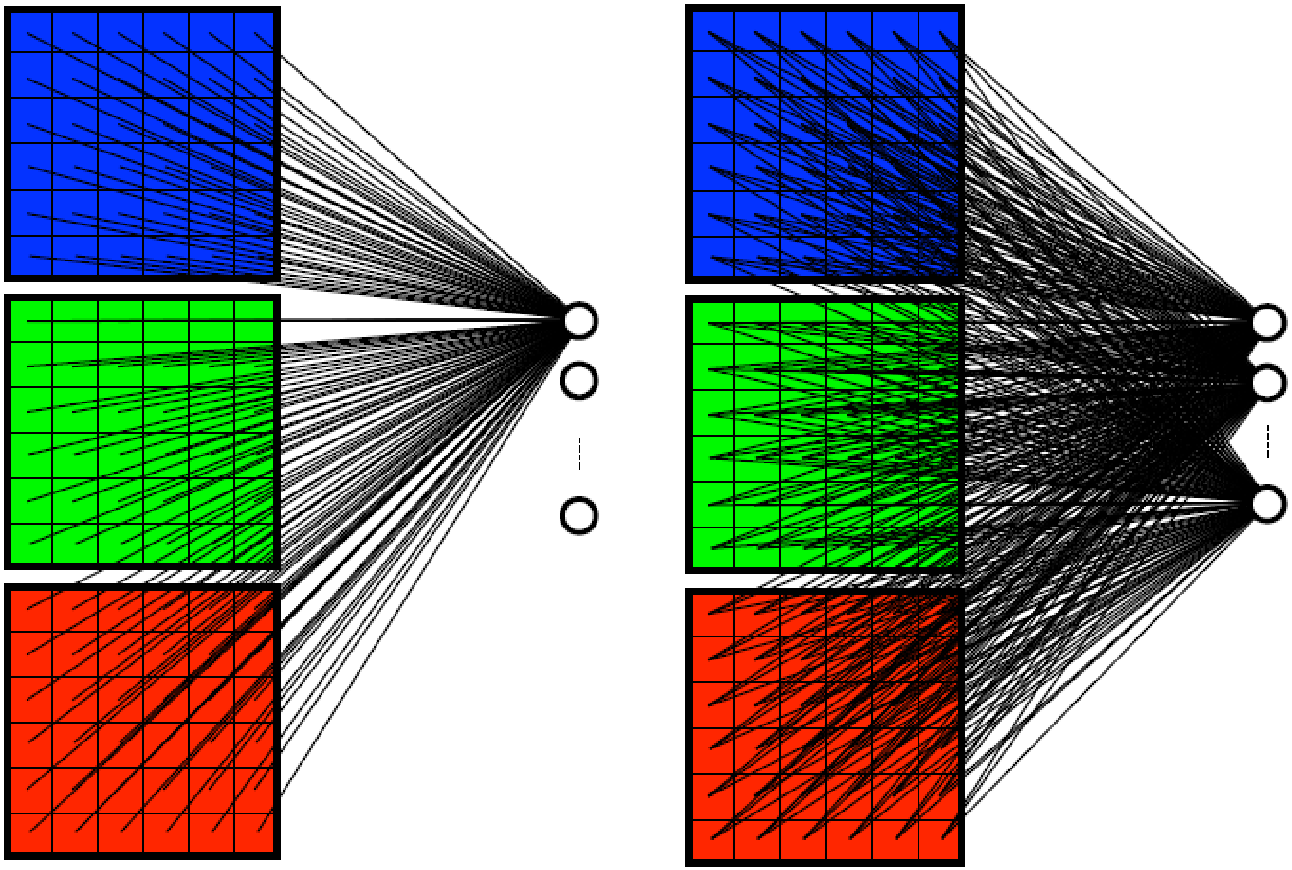
(A Very Quick) Intro to Neural Networks & CNN

- Neural Network & Multilayer Perceptron
- Convolutional Neural Networks



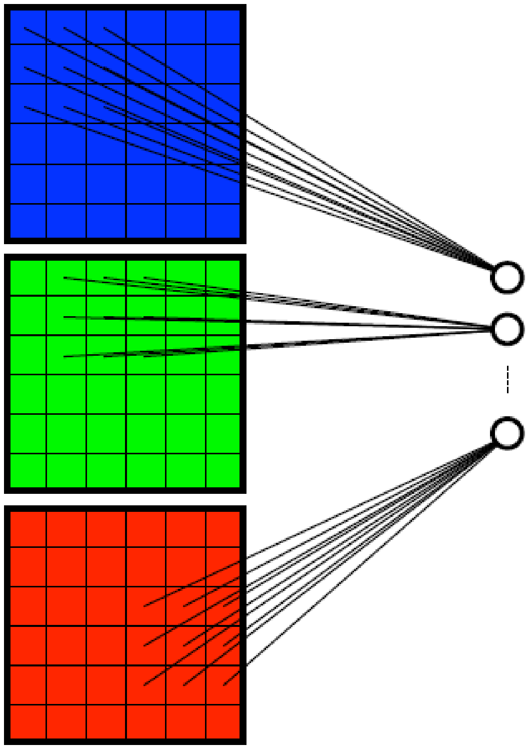
Convolutional Neural Networks

- How many weights for MLPs for images?



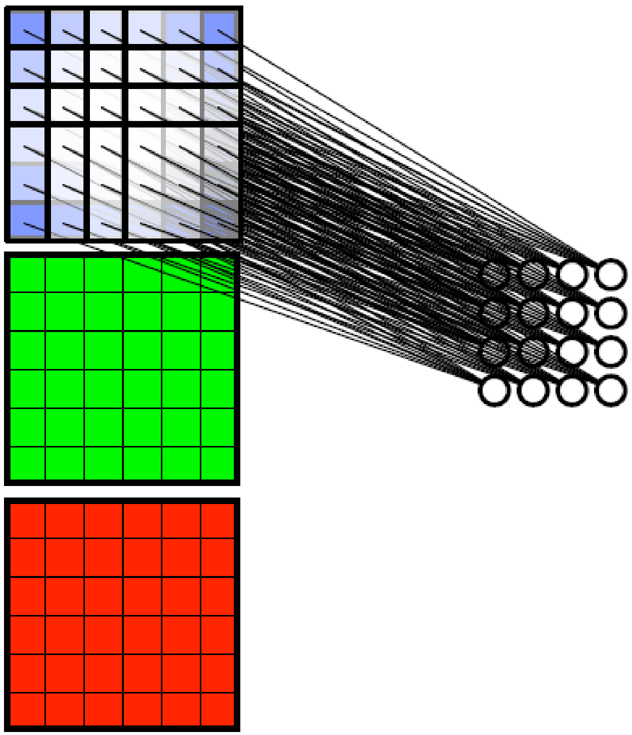
Convolutional Neural Networks

- Property I of CNN: Local Connectivity
  - Each neuron takes info only from a neighborhood of pixels.

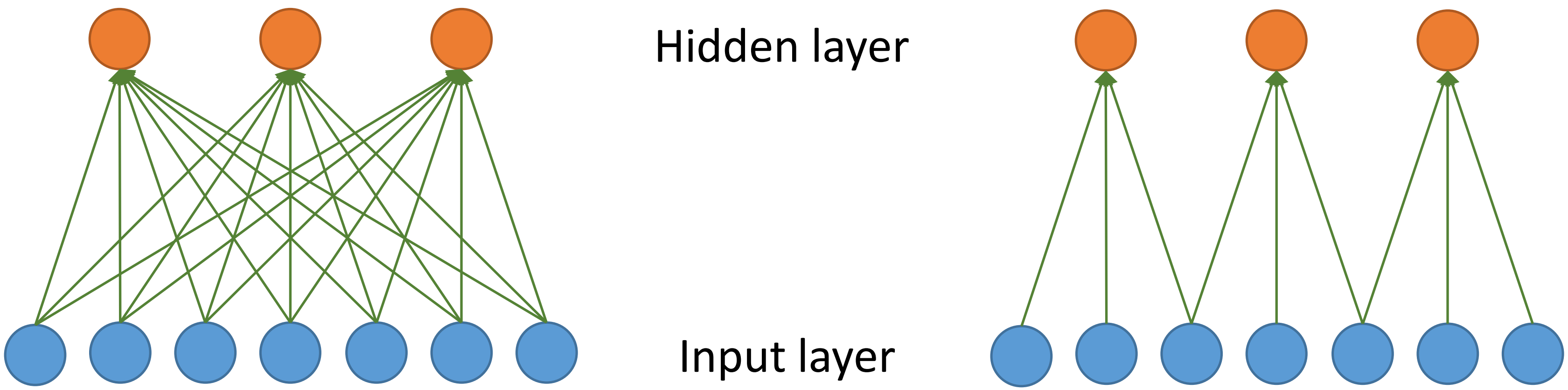


Convolutional Neural Networks

- Property II of CNN: Weight Sharing
  - Neurons connecting all neighborhoods have identical weights.



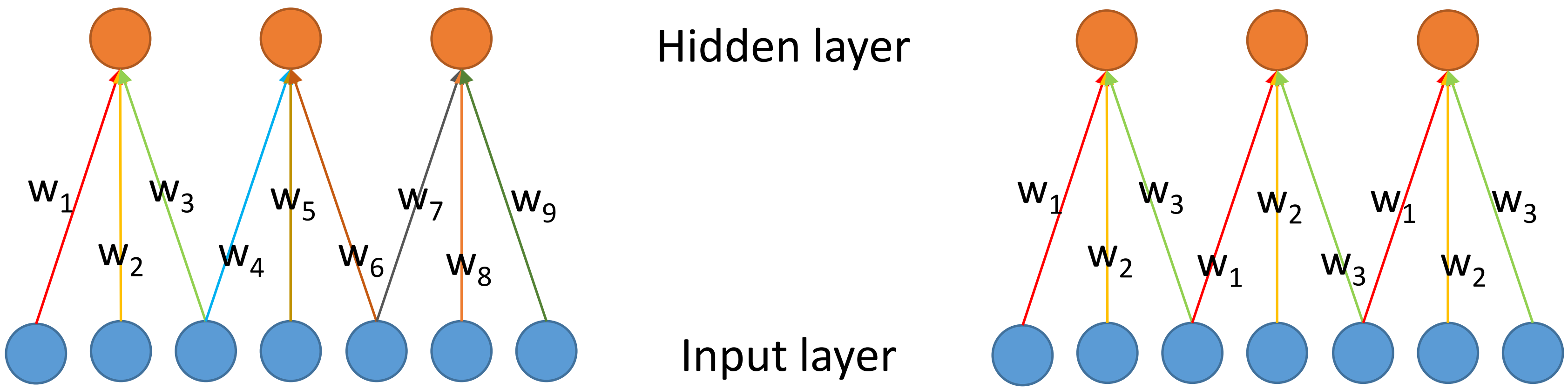
CNN: Local Connectivity



Global connectivity **Local** connectivity

- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Global connectivity:
  - Local connectivity:

CNN: Weight Sharing

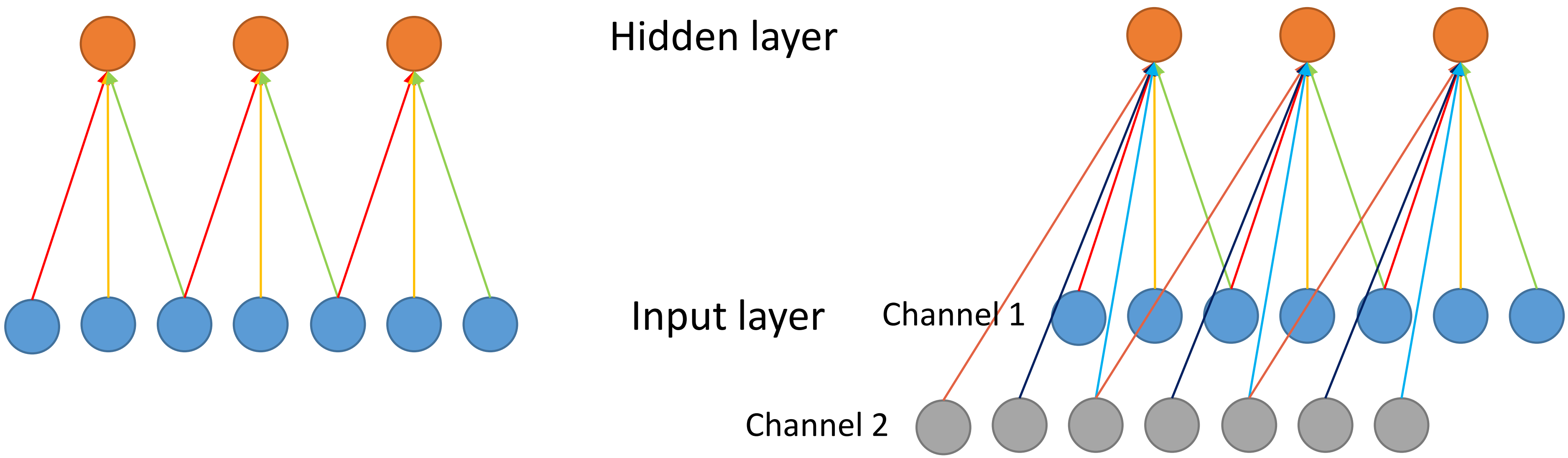


Without weight sharing

With weight sharing

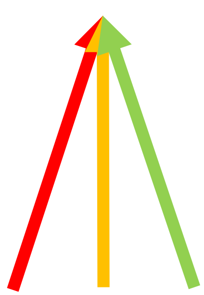
- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Without weight sharing:
  - With weight sharing :

CNN with Multiple Input Channels

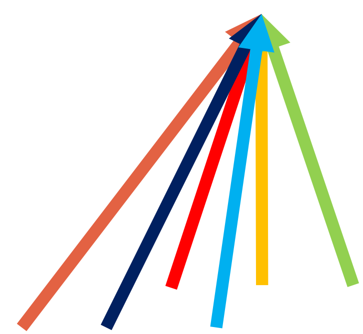


Single input channel

Multiple input channels

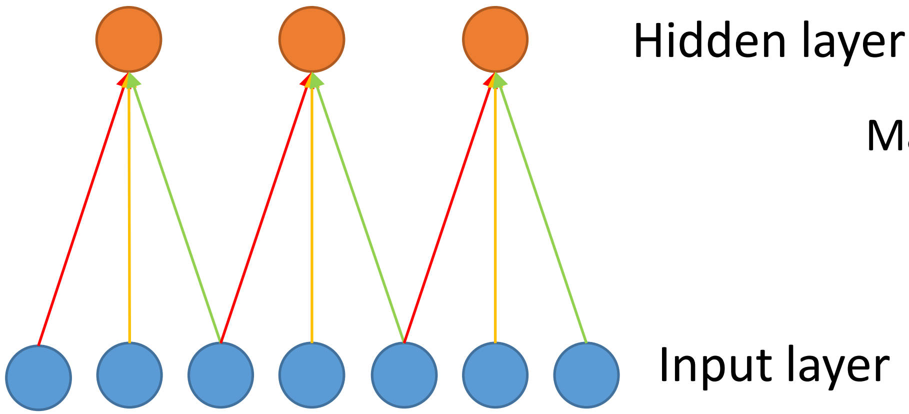


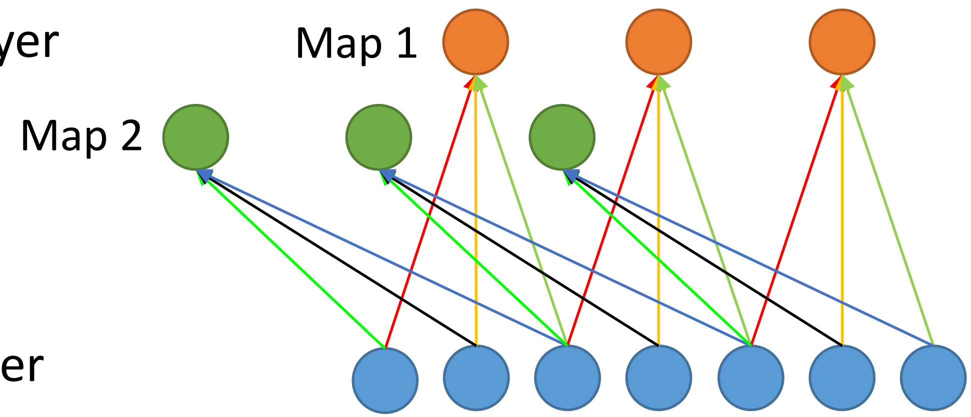
Filter weights



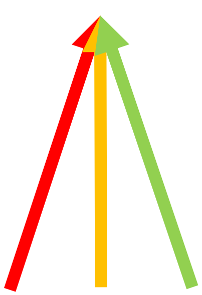
Filter weights

CNN with Multiple Output Maps



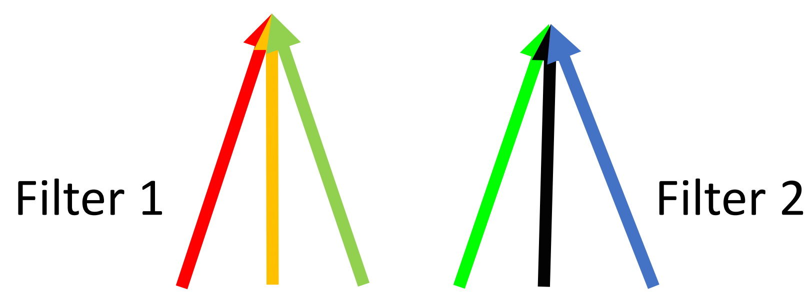


Single output map



Filter weights

Multiple output maps



Filter weights

Putting them together

- Local connectivity
- Weight sharing
- Handling multiple input channels
- Handling multiple output maps

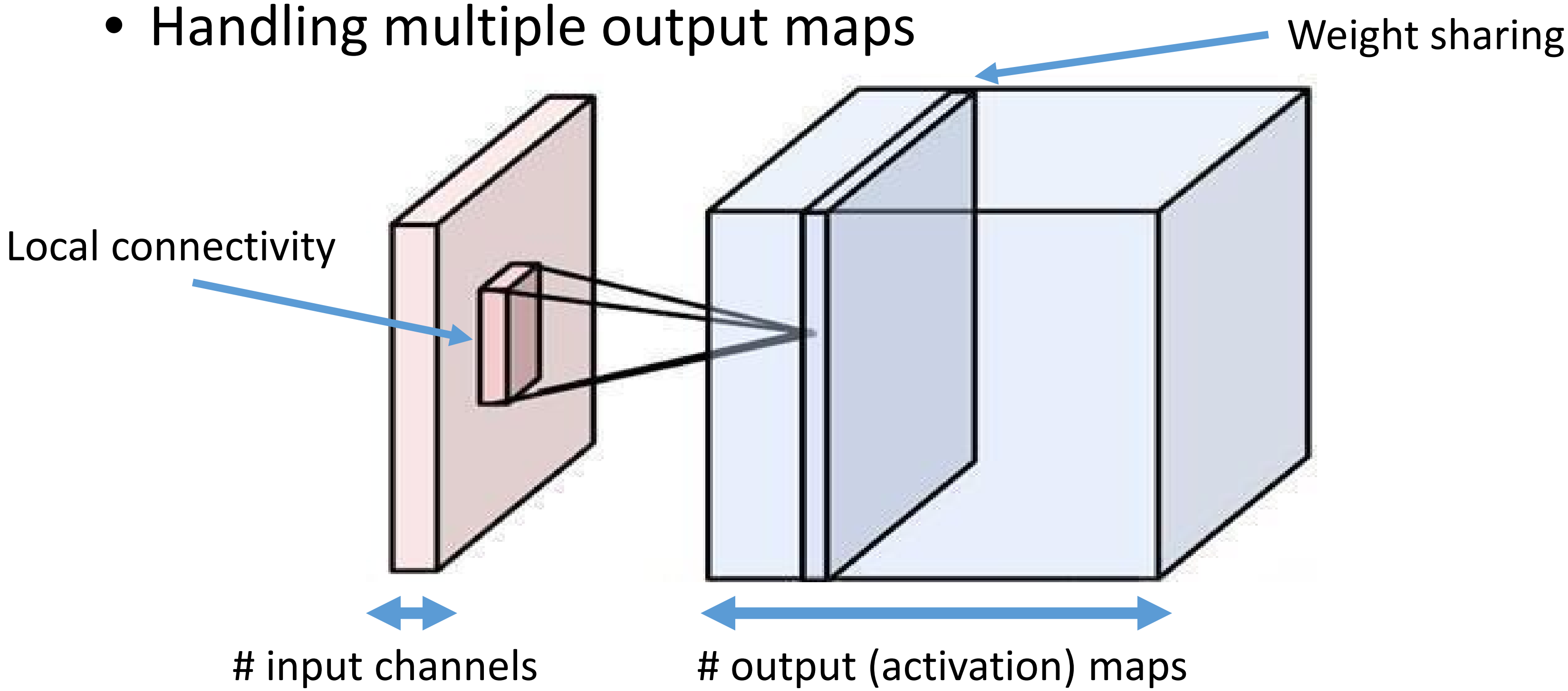
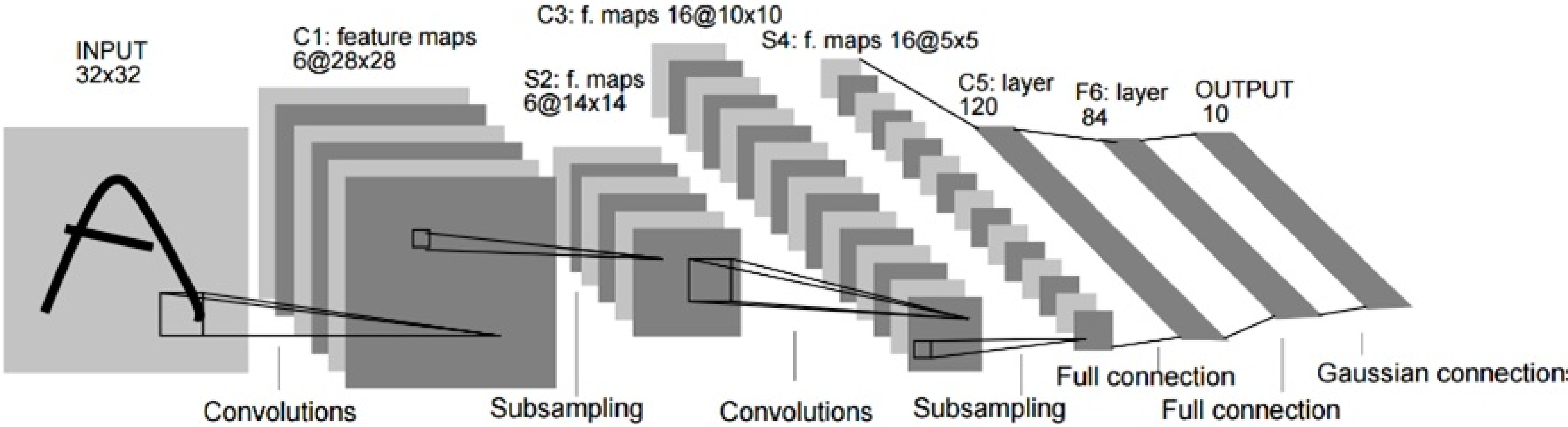
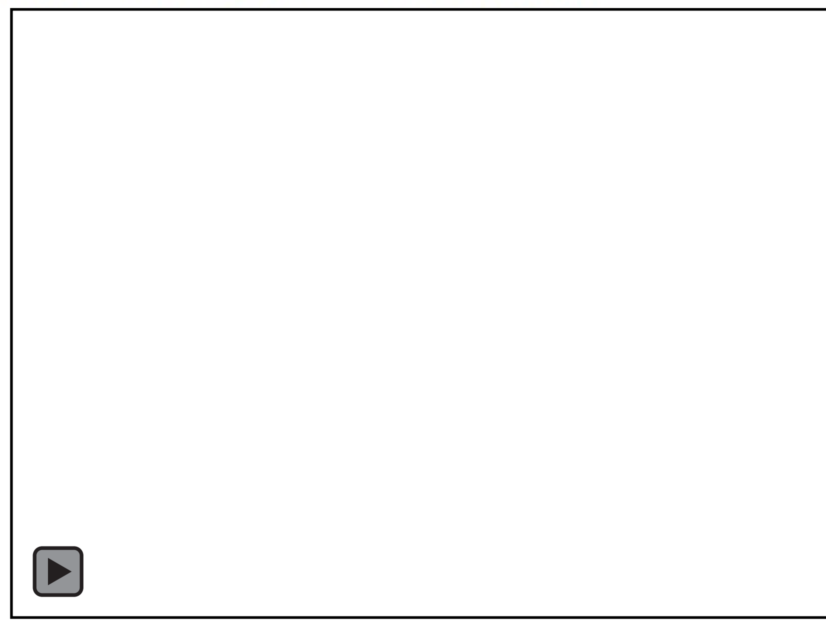


Image credit: A. Karpathy

LeNet [LeCun et al. 1998]

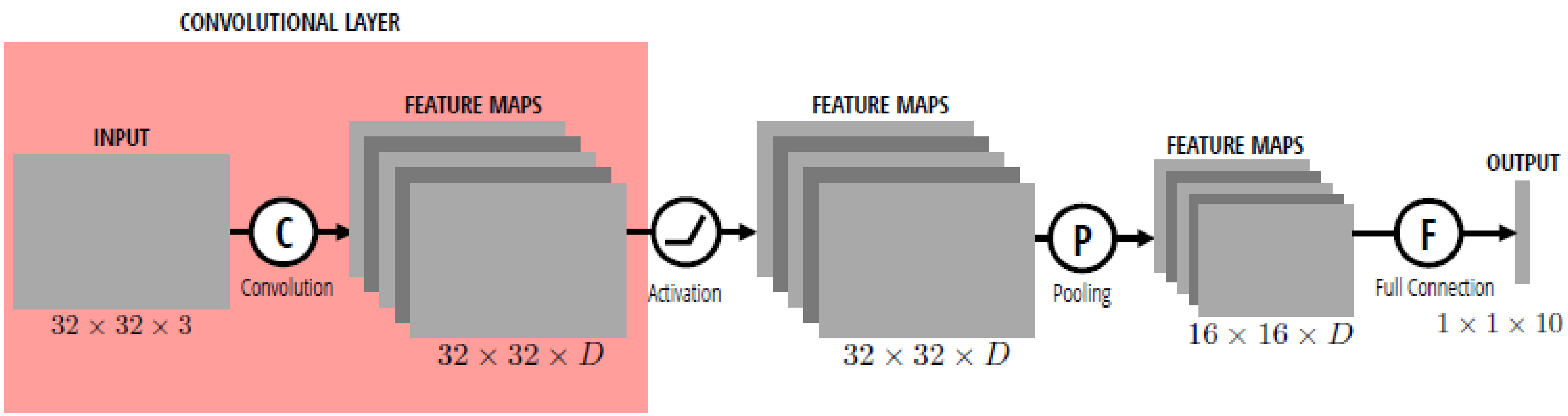




LeNet-1 from 1993

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

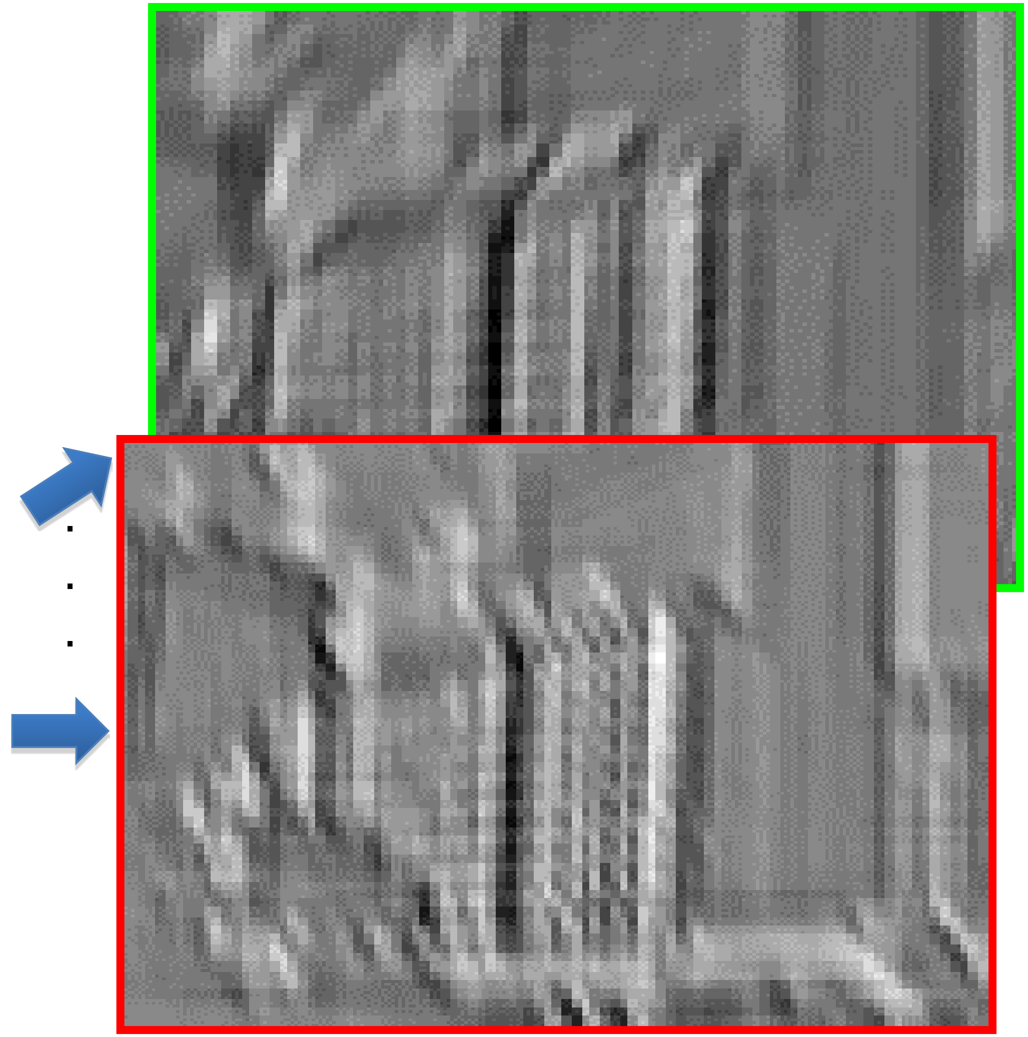
Convolution Layer in CNN



What is a Convolution?

- Weighted moving sum

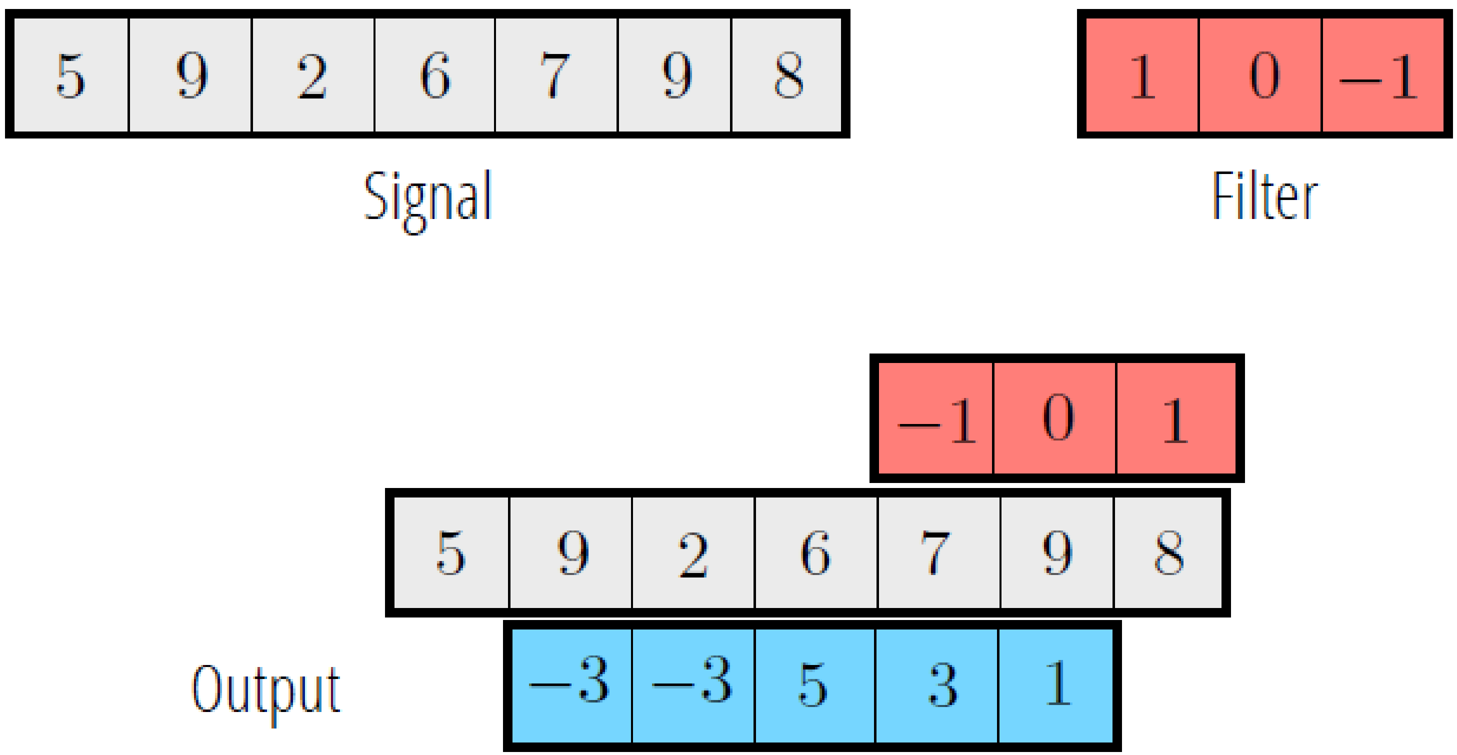
Input



Feature Activation Map

slide credit: S. Lazebnik

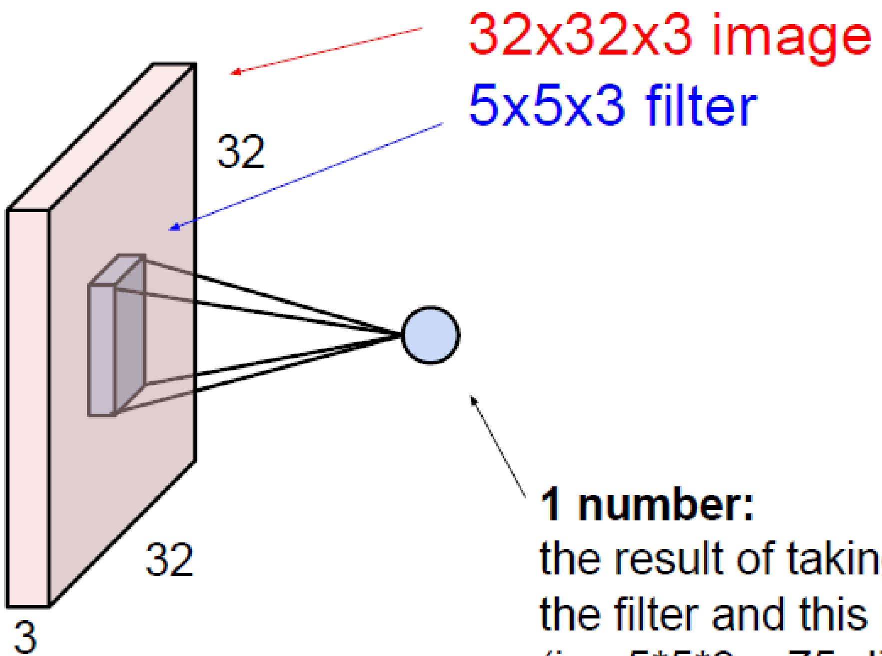
What is a Convolution?

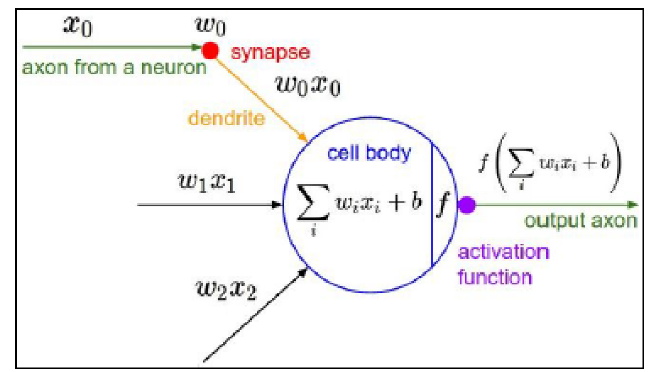


Convolution is a local linear operator

Putting them together (cont’d)

- The brain/neuron view of CONV layer



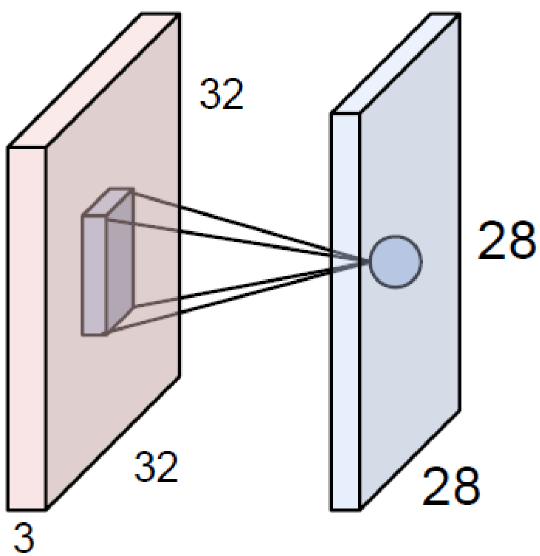


It’s just a neuron with local connectivity...

1 number:
the result of taking a dot product between the filter and this part of the image
(i.e. 5*5*3 = 75-dimensional dot product)

Putting them together (cont'd)

- The brain/neuron view of CONV layer



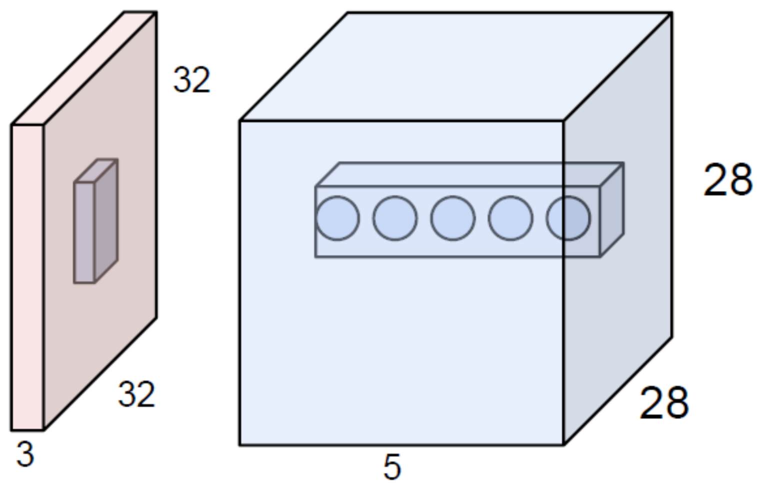
An activation map is a 28x28 sheet of neuron outputs:

1. Each is connected to a small region in the input
2. All of them share parameters

"5x5 filter" -> "5x5 receptive field for each neuron"

Putting them together (cont'd)

- The brain/neuron view of CONV layer

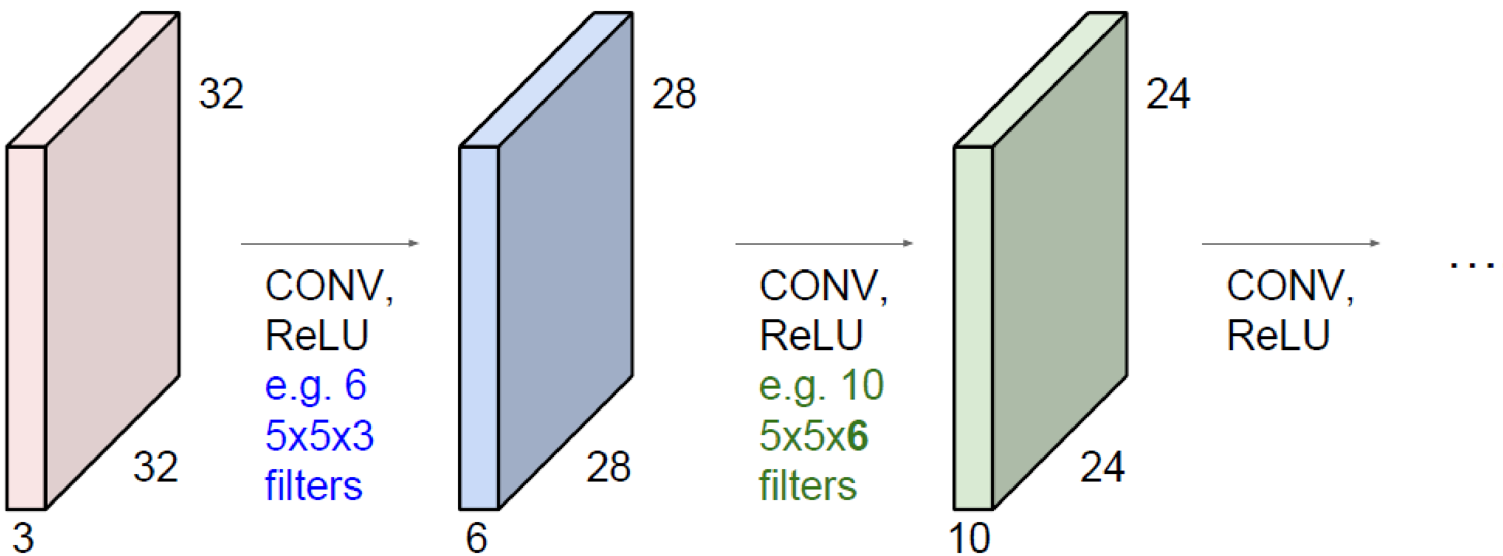


E.g. with 5 filters,
CONV layer consists of
neurons arranged in a 3D grid
(28x28x5)

There will be 5 different
neurons all looking at the same
region in the input volume

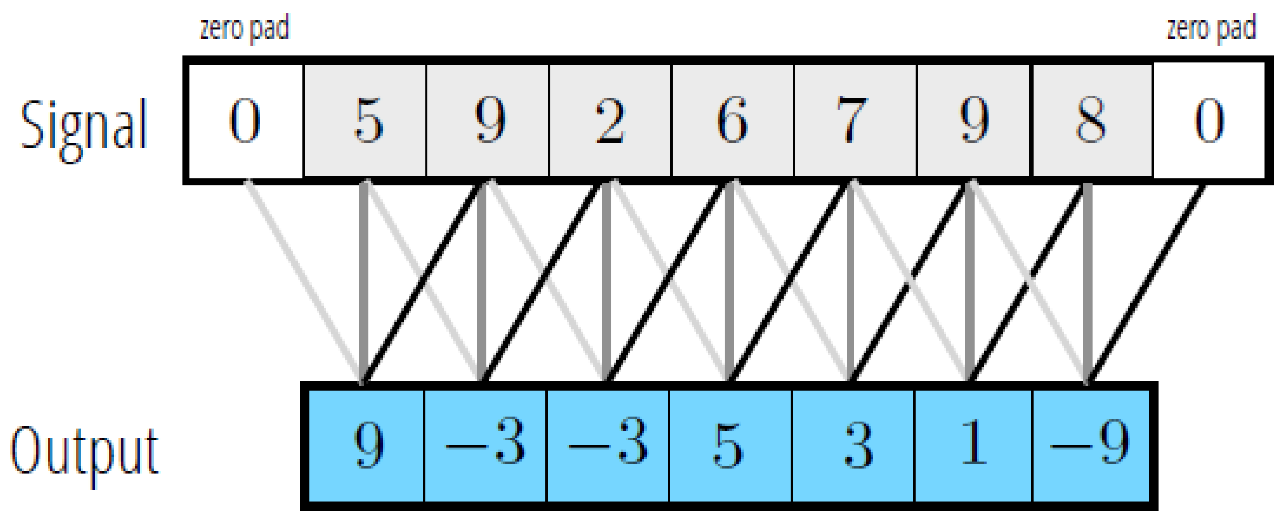
Putting them together (cont'd)

- Image input with 32 x 32 pixels convolved repeatedly with 5 x 5 x 3 filters shrinks volumes spatially (32 -> 28 -> 24 -> …).



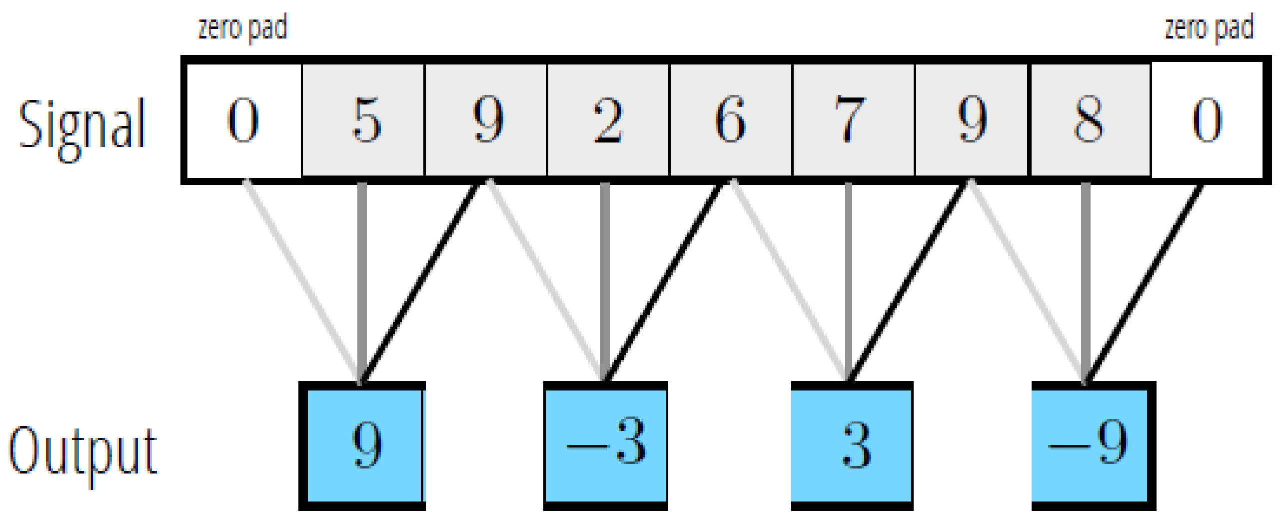
What is a Convolution?

- Zero Padding
  - Output is the same size as input (doesn't shrink as the network gets deeper).



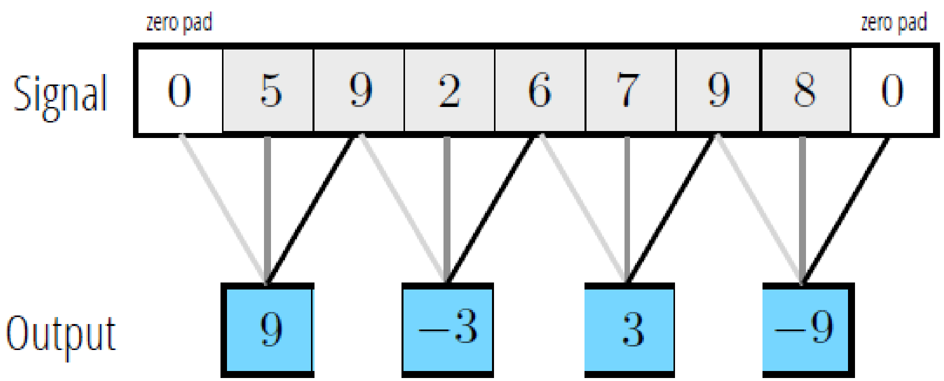
What is a Convolution?

- Stride
  - Step size across signals



What is a Convolution?

- Stride
  - Step size across signals



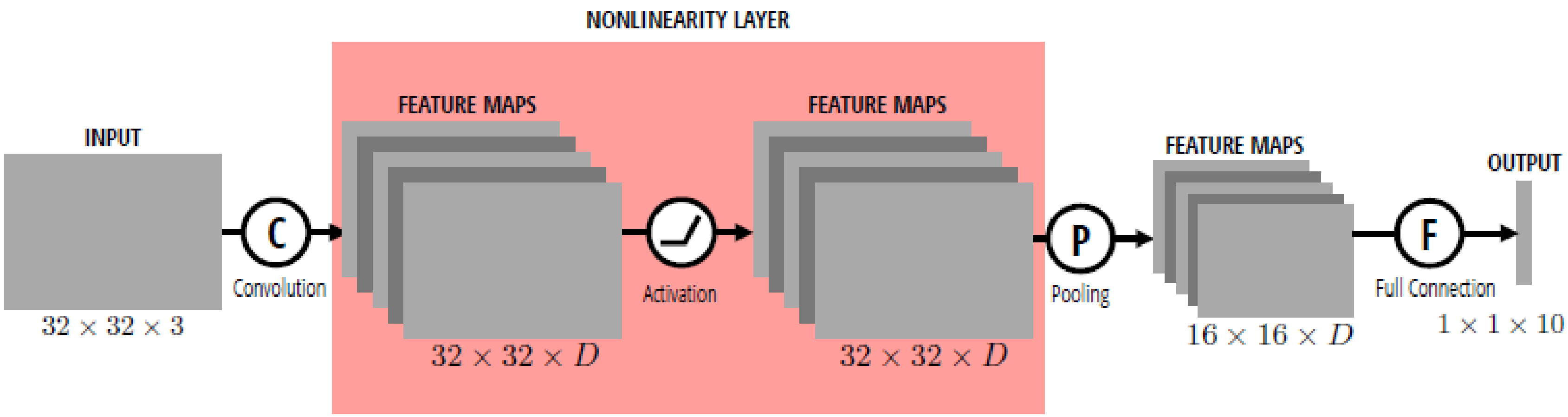
Output Size $\dfrac{N - c}{s} + 1$

Input Size

Filter Size

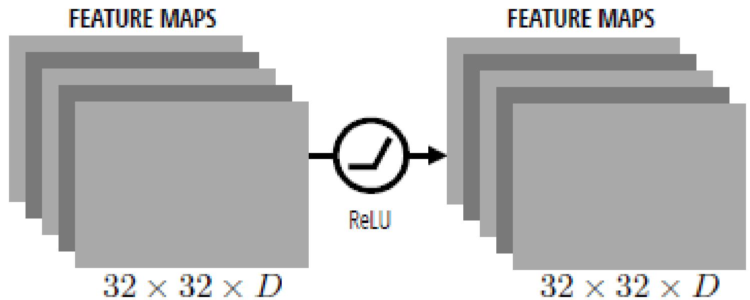
Stride: step size across the signal

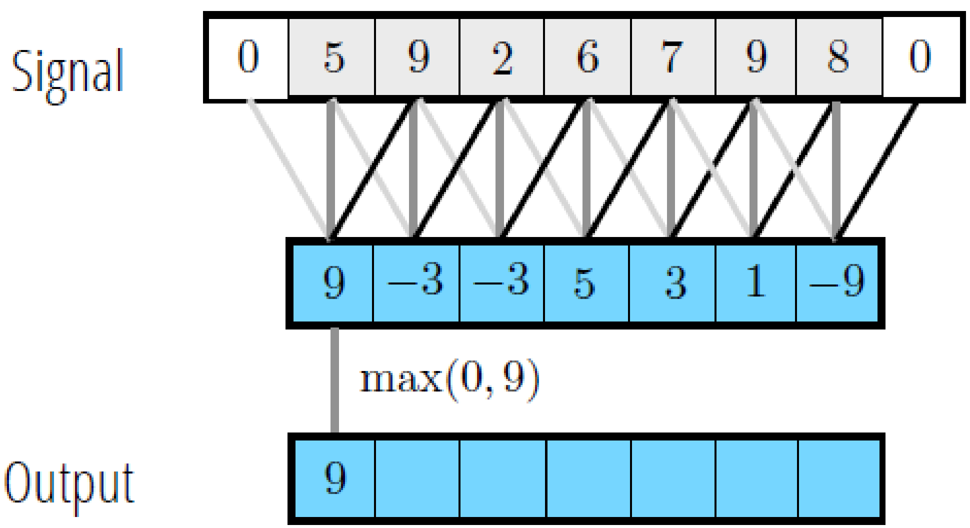
Nonlinearity Layer in CNN



Nonlinearity Layer

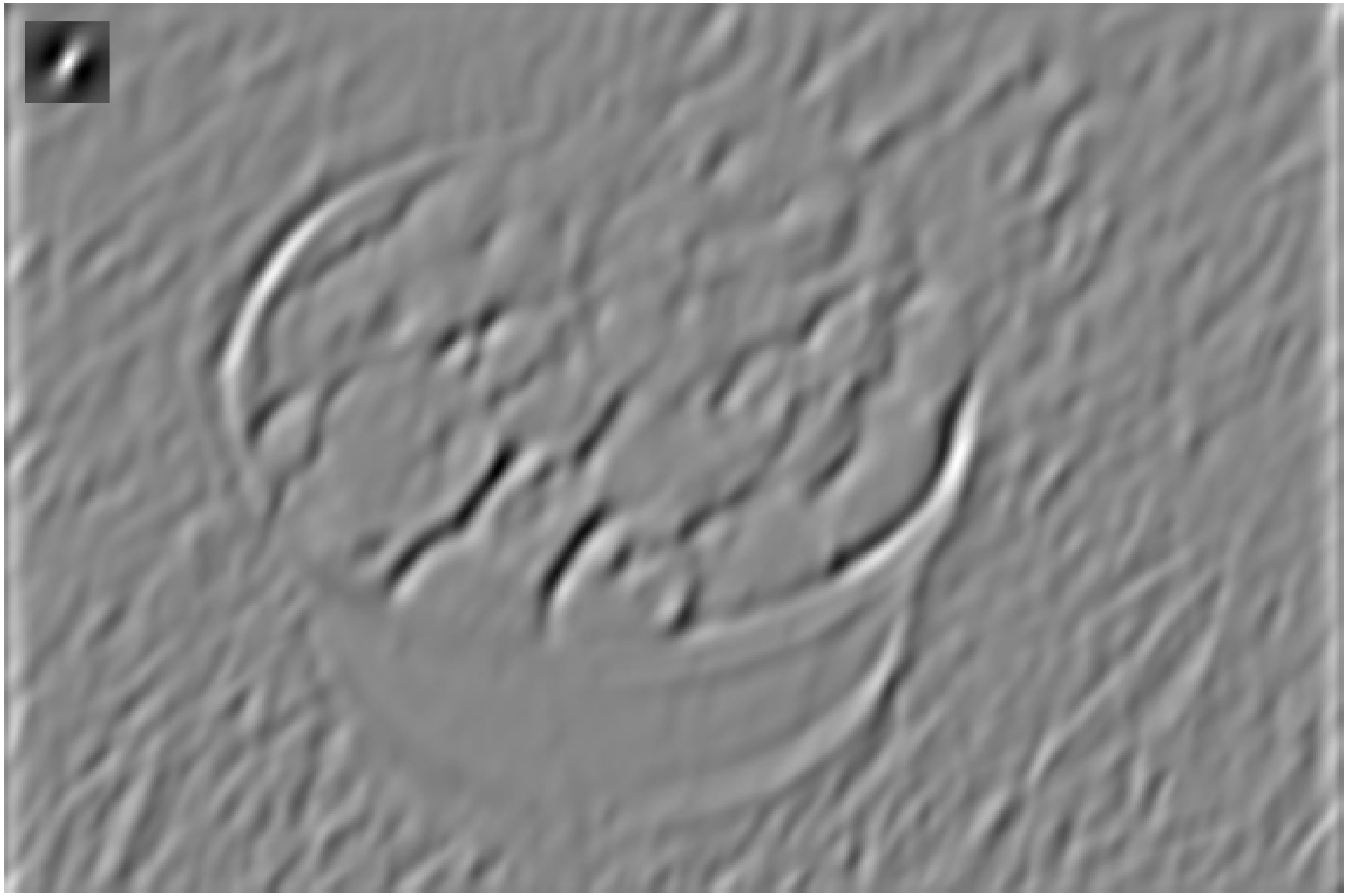
- E.g., ReLU (Rectified Linear Unit)
  - Pixel by pixel computation of max(0, x)





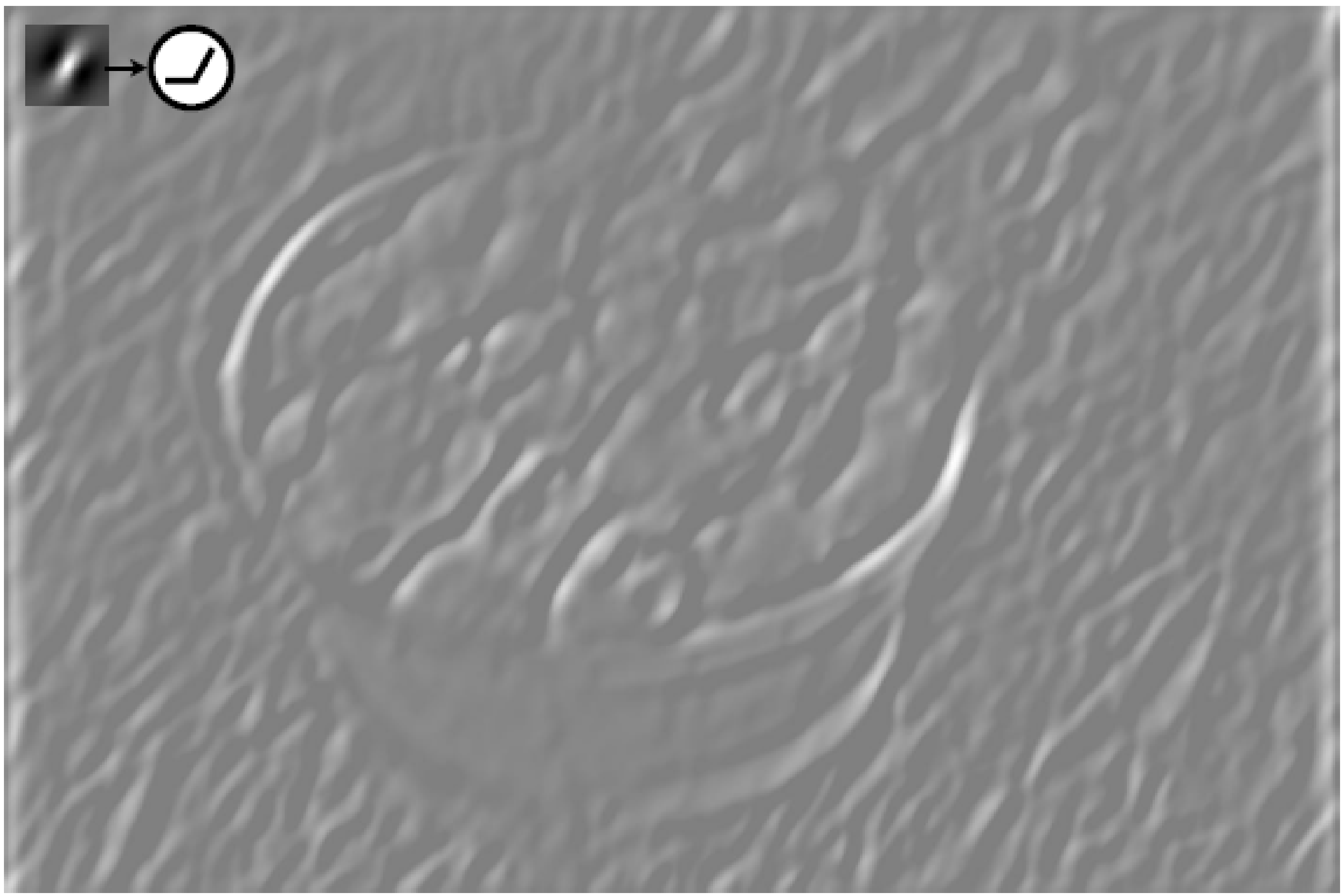
Nonlinearity Layer

- E.g., ReLU (Rectified Linear Unit)
  - Pixel by pixel computation of max(0, x)

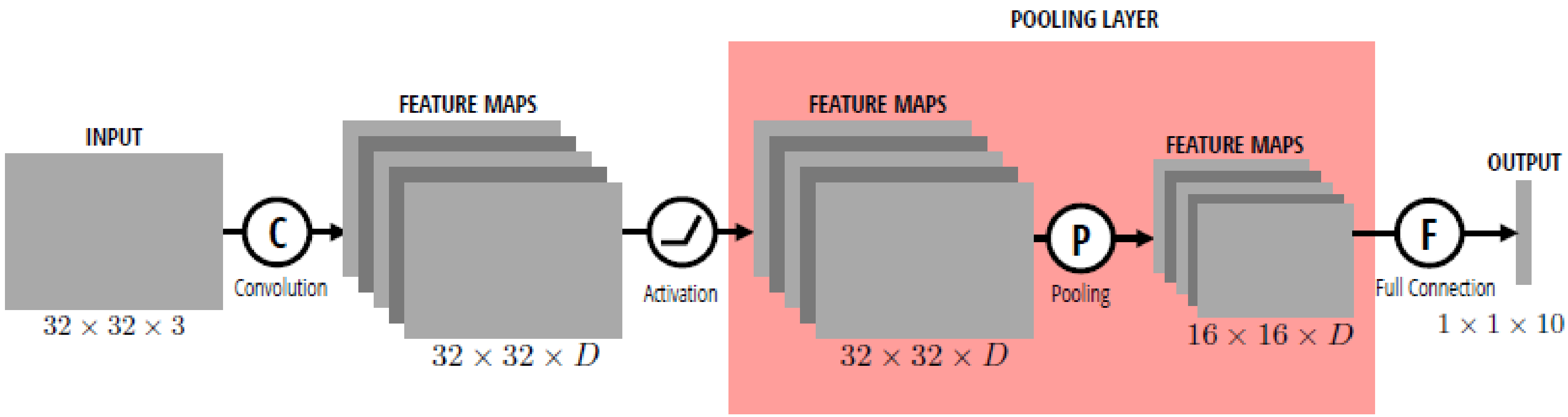


Nonlinearity Layer

- E.g., ReLU (Rectified Linear Unit)
  - Pixel by pixel computation of max(0, x)

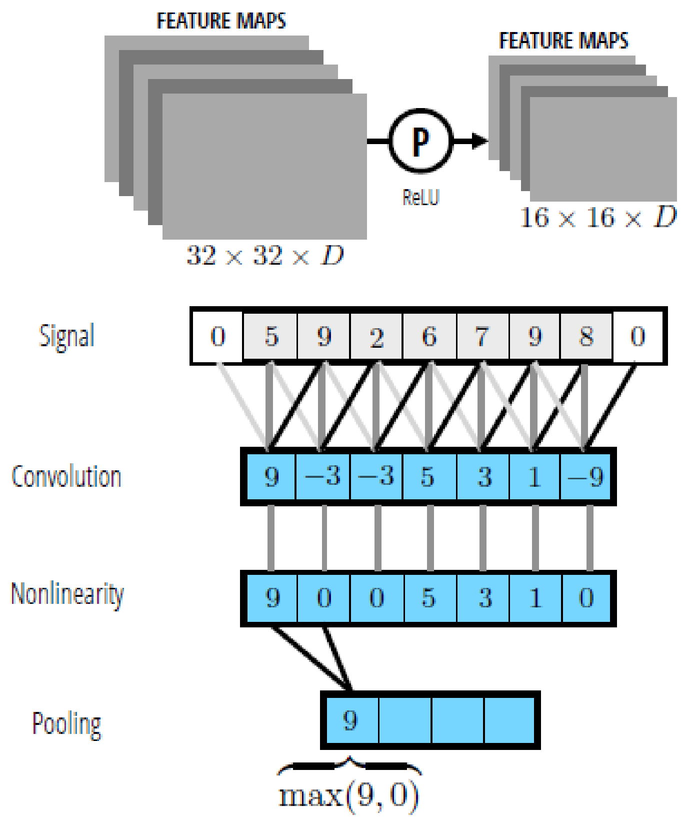


Pooling Layer in CNN



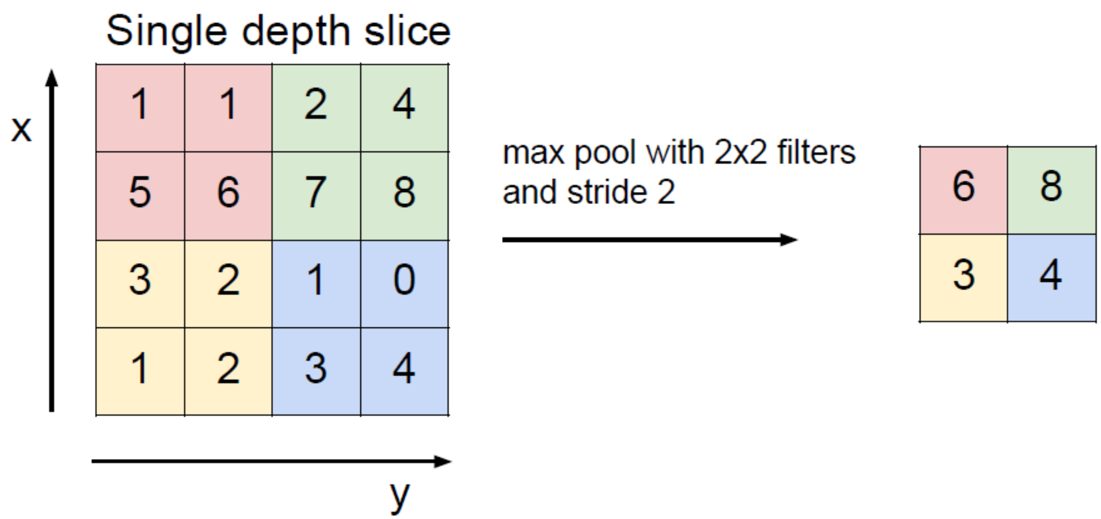
Pooling Layer

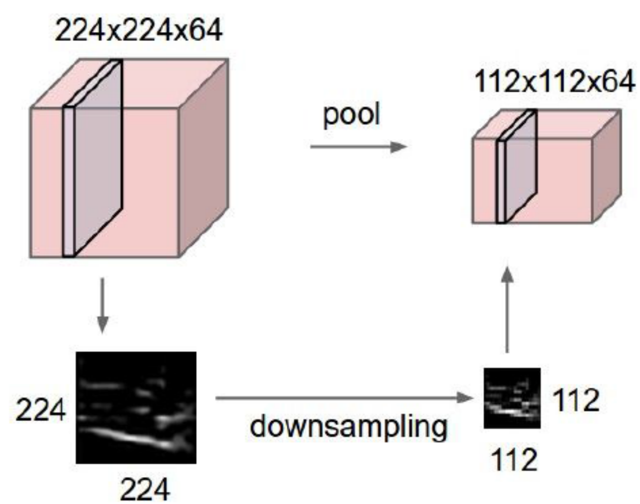
- Makes the representations smaller and more manageable
- Operates over each activation map independently
- E.g., Max Pooling



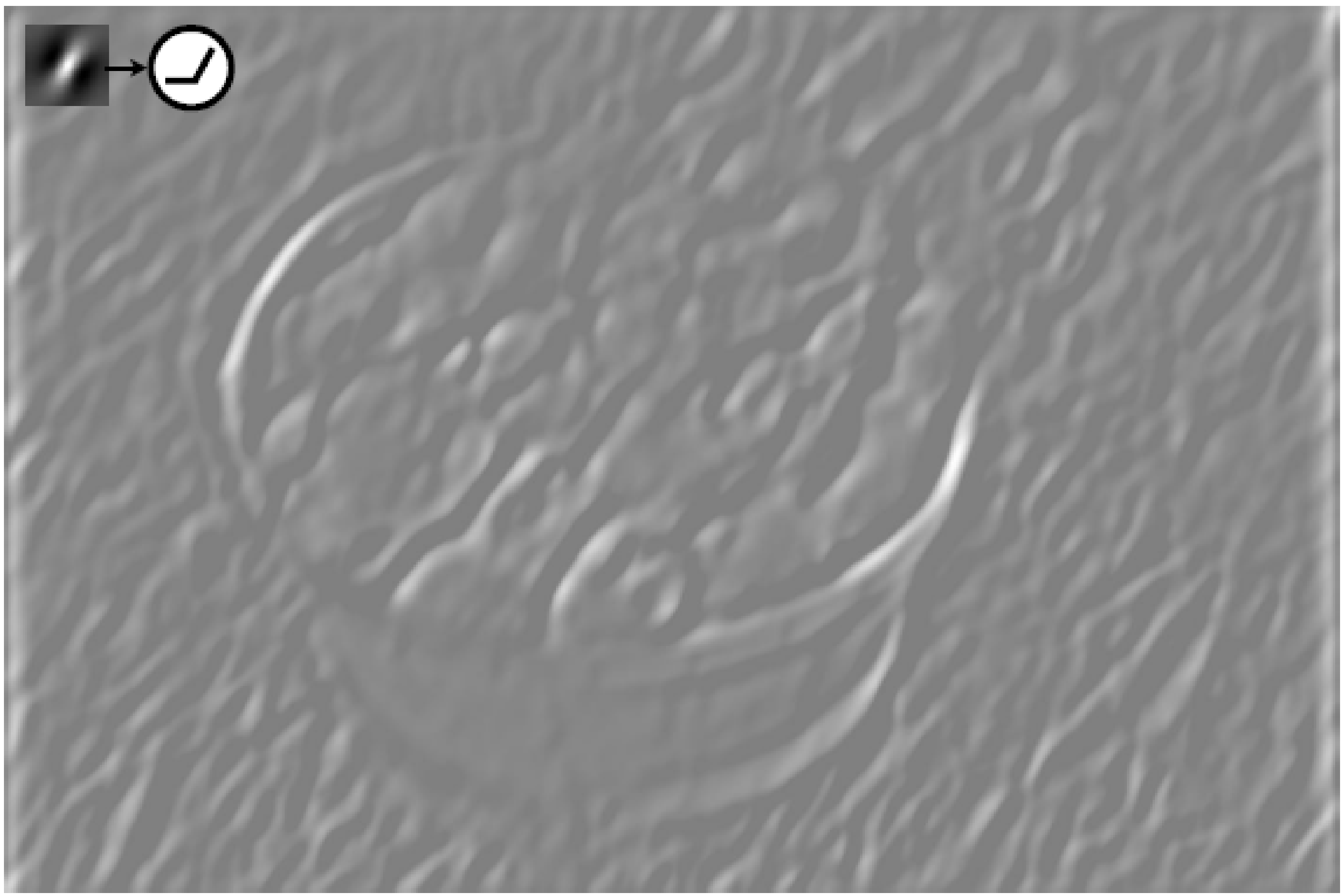
Pooling Layer

- Reduces the spatial size and provides spatial invariance

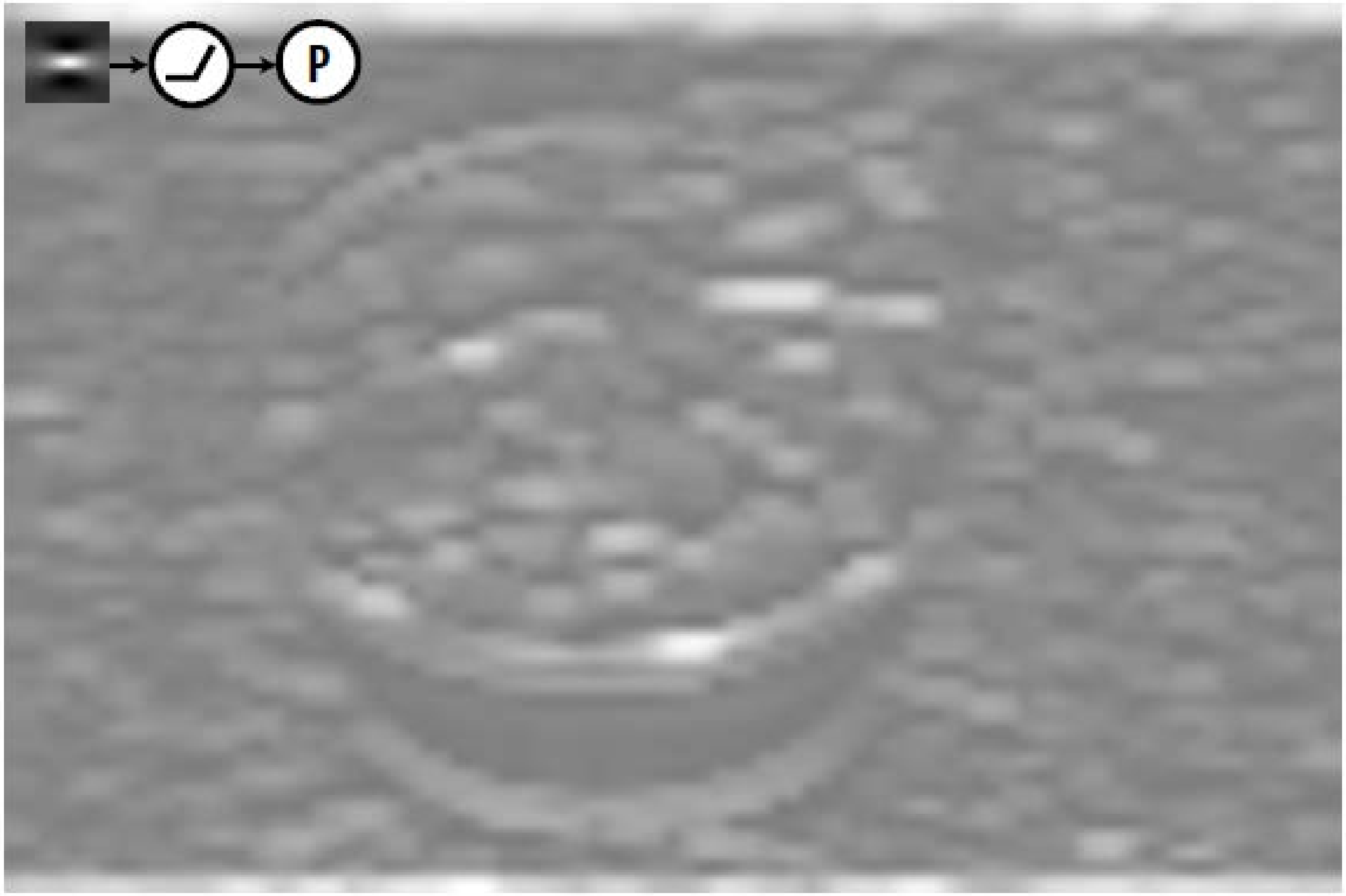




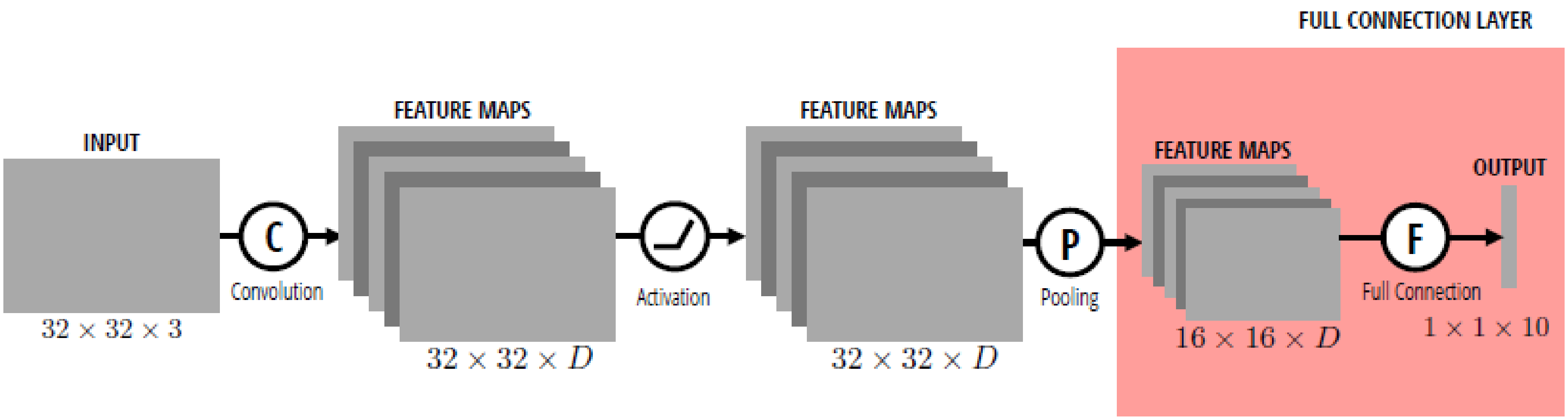
- Example
  - Nonlinearity by ReLU



- Example
  - Max pooling

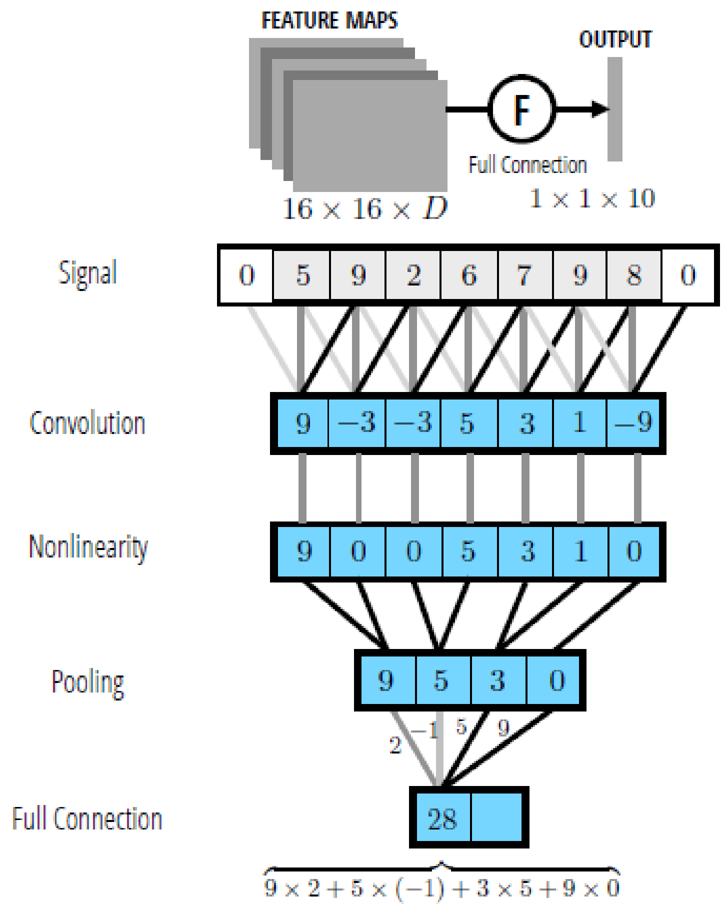


Fully Connected (FC) Layer in CNN



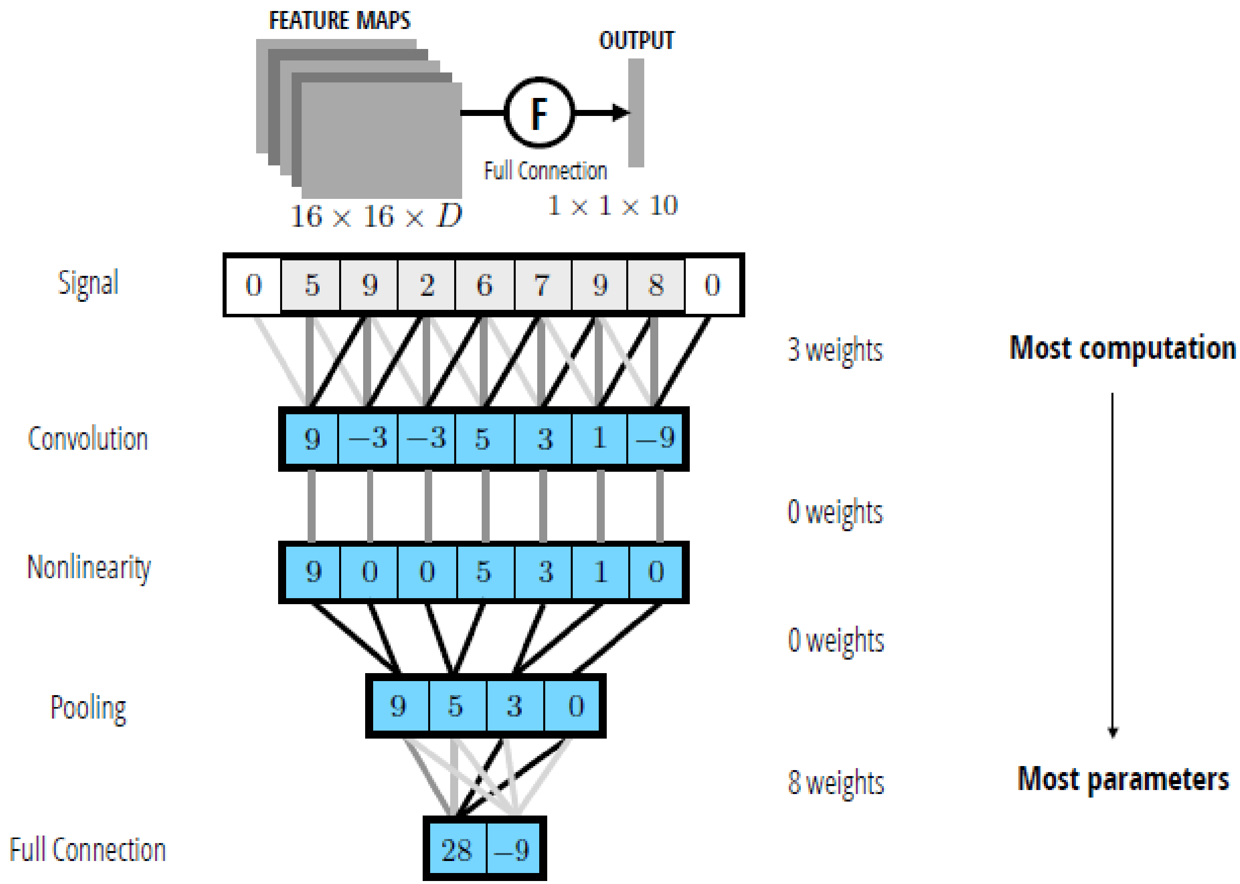
FC Layer

- Contains neurons that connect to the entire input volume, as in ordinary neural networks

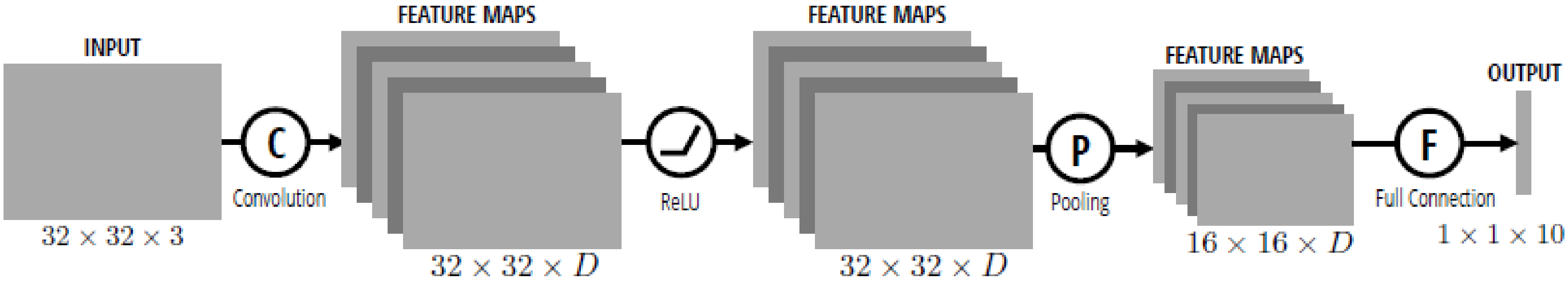


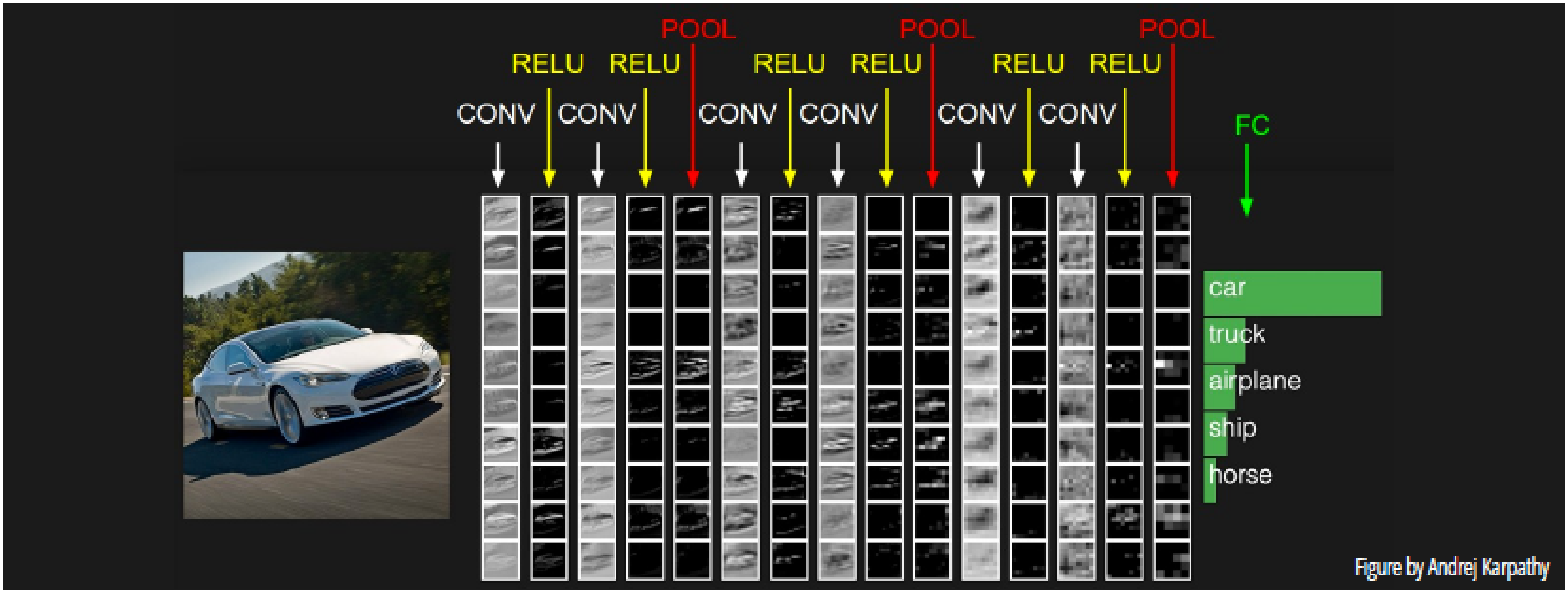
FC Layer

- Contains neurons that connect to the entire input volume, as in ordinary neural networks



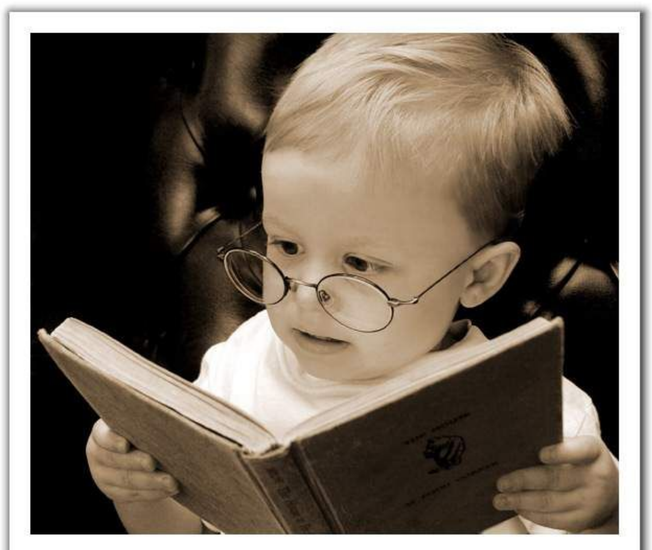
CNN


INPUT
FEATURE MAPS
FEATURE MAPS
FEATURE MAPS
OUTPUT
Convolution
ReLU
Pooling
Full Connection
32 × 32 × 3
32 × 32 × D
32 × 32 × D
16 × 16 × D
1 × 1 × 10


POOL
POOL
POOL
RELU RELU RELU RELU RELU RELU
CONV CONV CONV CONV CONV CONV
FC
car
truck
airplane
ship
horse
Figure by Andrej Karpathy

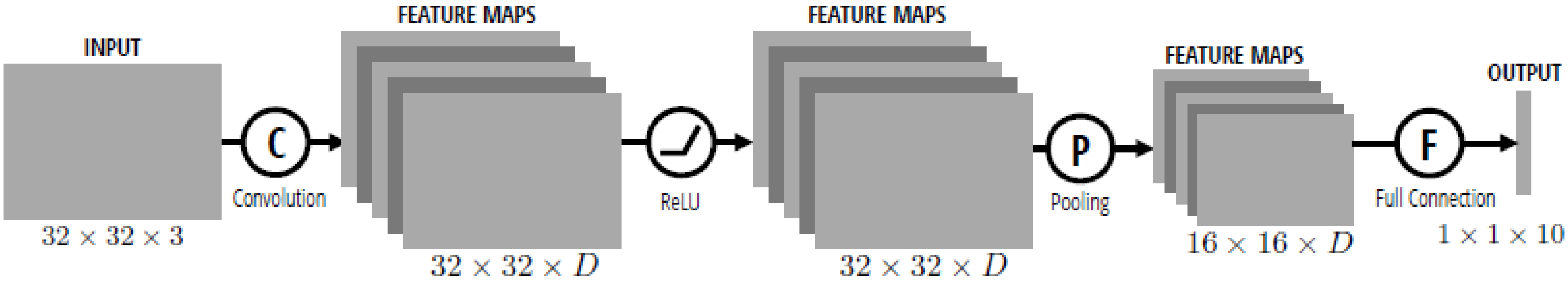
What Will Be Covered in Today's Lecture?

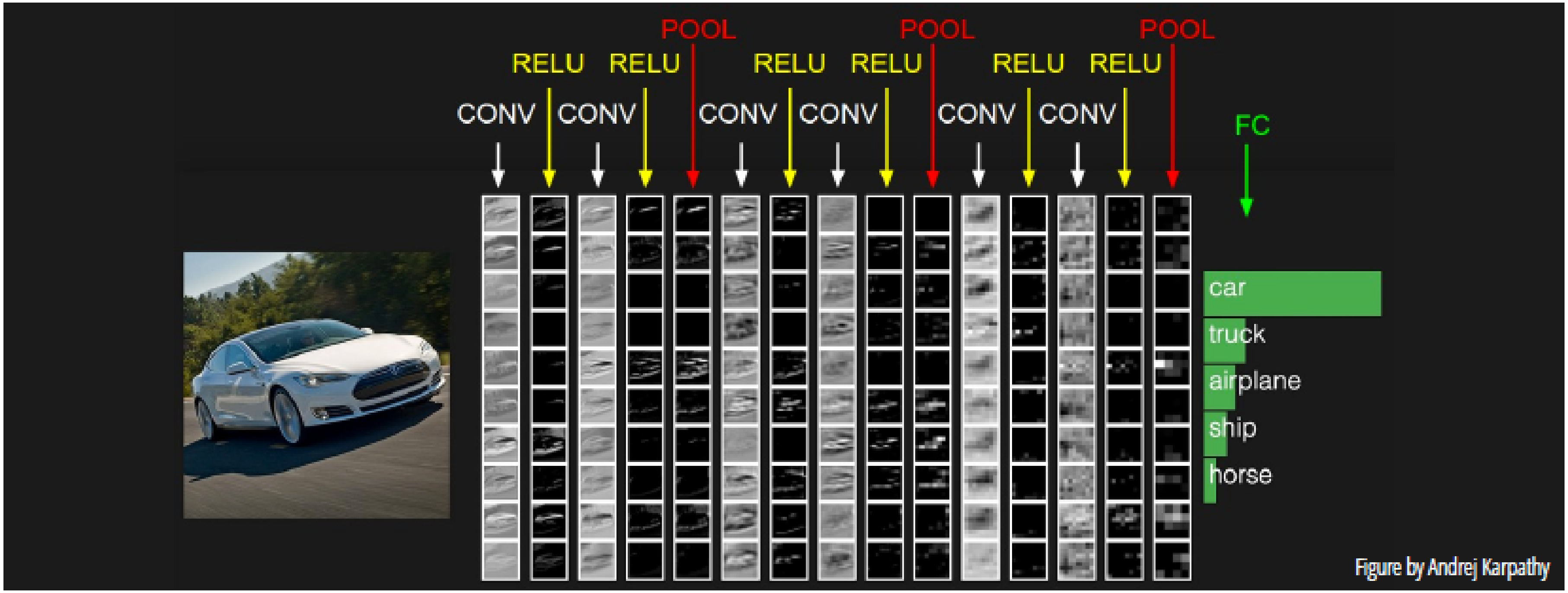
- Brief Review to CV/ML Backgrounds
- Recent Advances in Deep Learning for Computer Vision
- Transfer Learning and Its Applications to Image Analysis and Synthesis
- Beyond Transfer Learning: Representation Disentanglement*



*: if time permits

Revisit of CNN





Transfer Learning: What, When, and Why?

- What is Transfer Learning?
  - “Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.” – Wikipedia

- What is the common assumption in Machine Learning?
  - Training data (typically annotated) would be available.
  - Training and test data are drawn from the same feature space and with the same distribution.

(Traditional) Machine Learning vs. Transfer Learning

- Machine Learning
  - Collecting/annotating data is typically expensive.

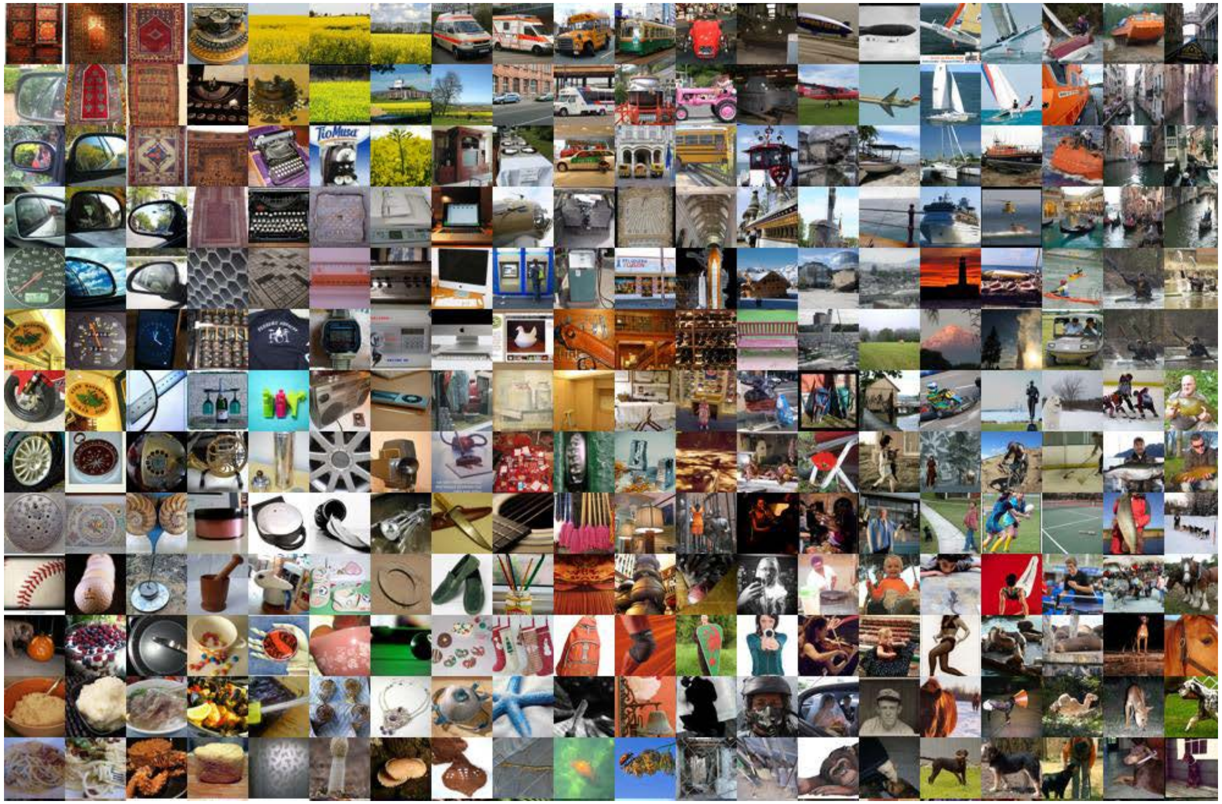
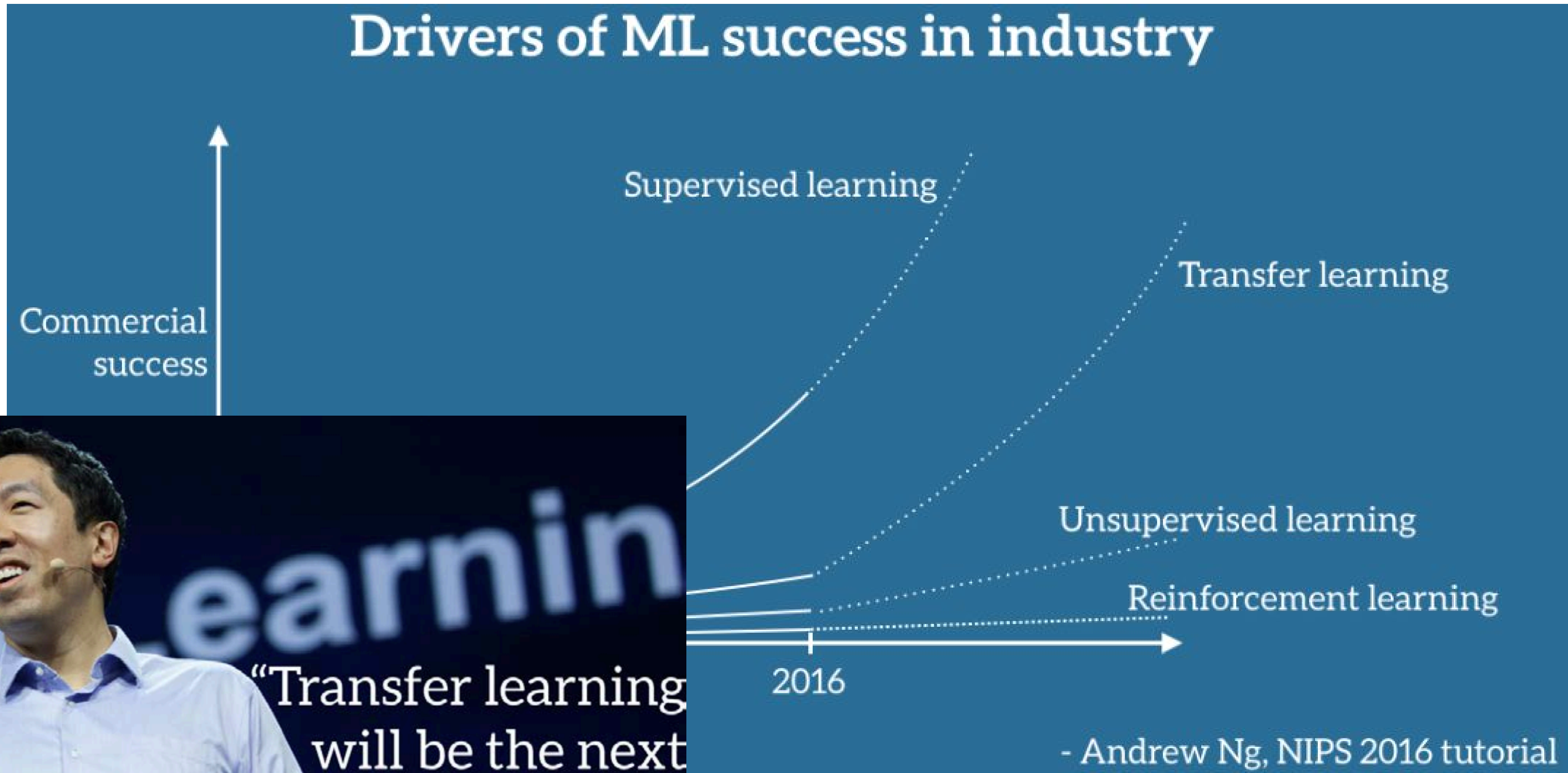
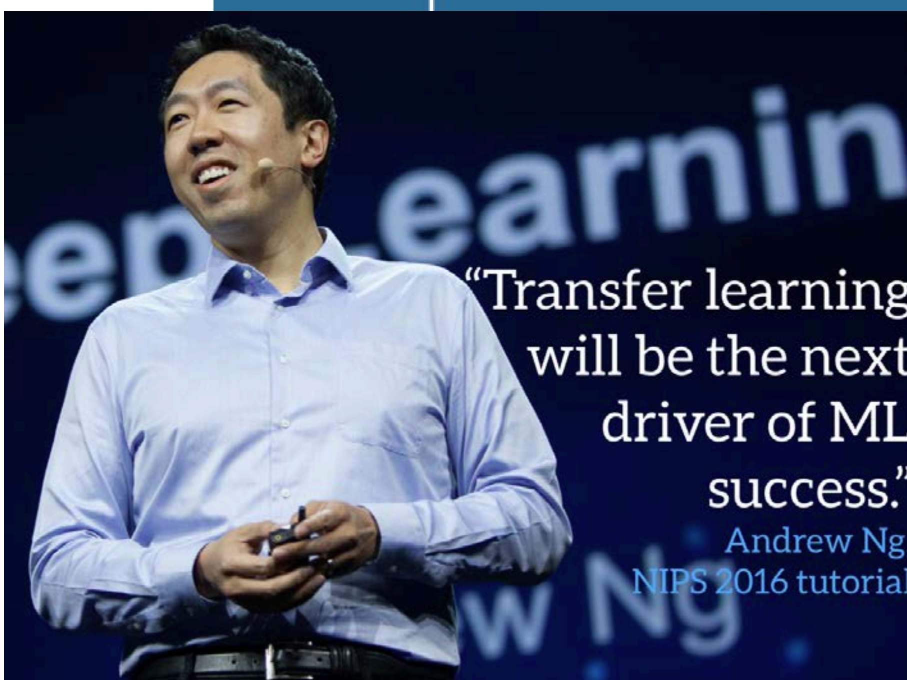


*Image Courtesy: Andrej Karpathy

Why You Should Know Transfer Learning?



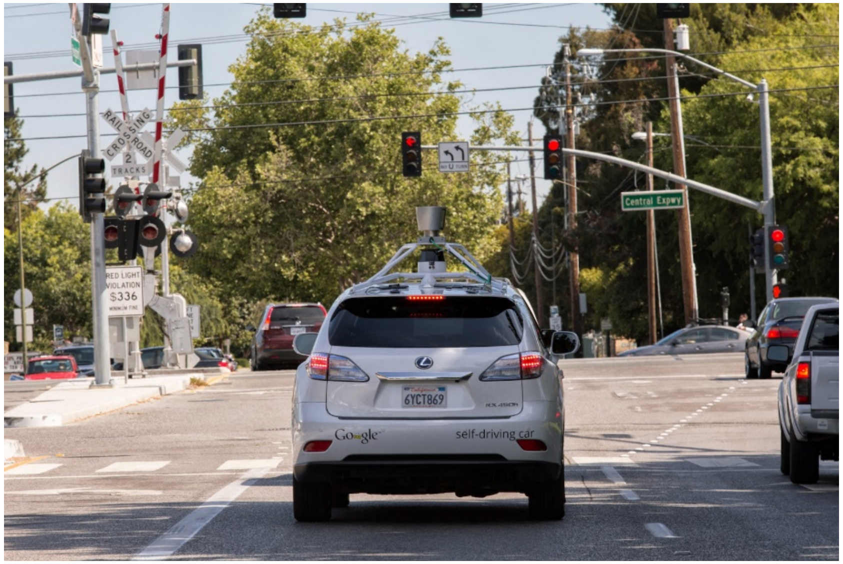


Transfer Learning: What, When, and Why? (cont'd)

- Examples #2

vs.

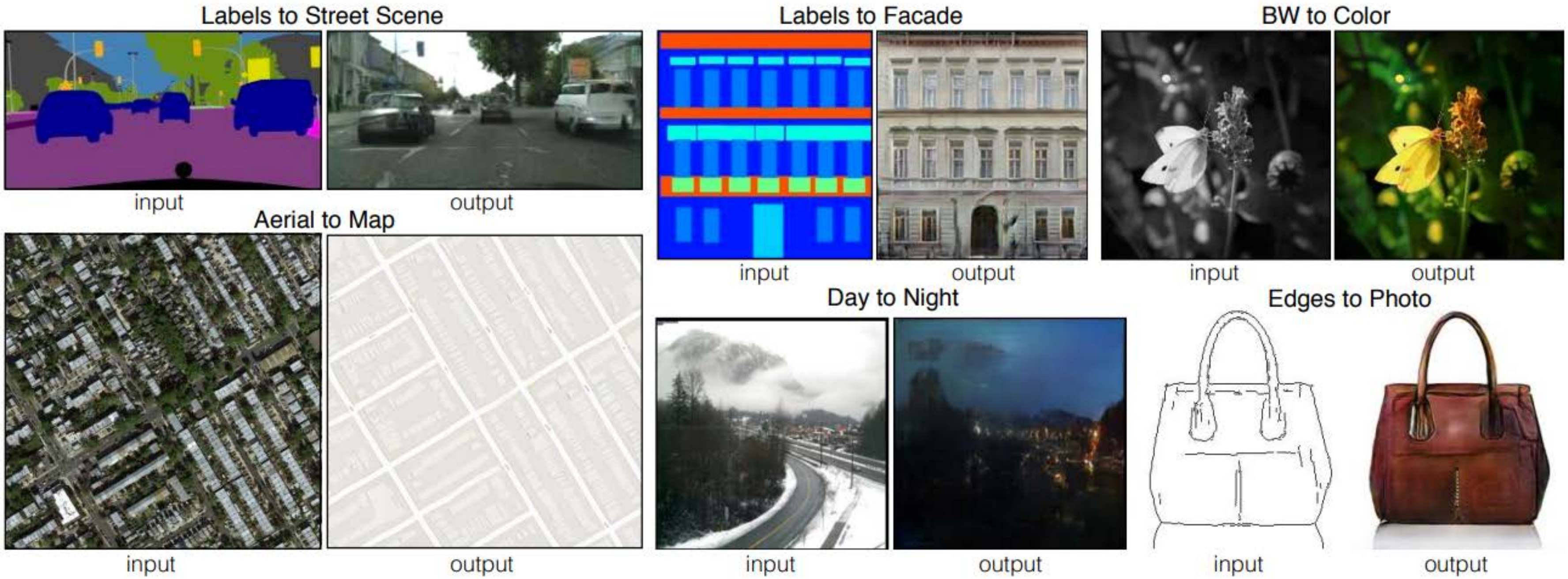
Why You Should Know Transfer Learning?



https://techcrunch.com/2017/02/08/udacity-open-sources-its-self-driving-car-simulator-for-anyone-to-use/
https://googleblog.blogspot.tw/2014/04/the-latest-chapter-for-self-driving-car.html

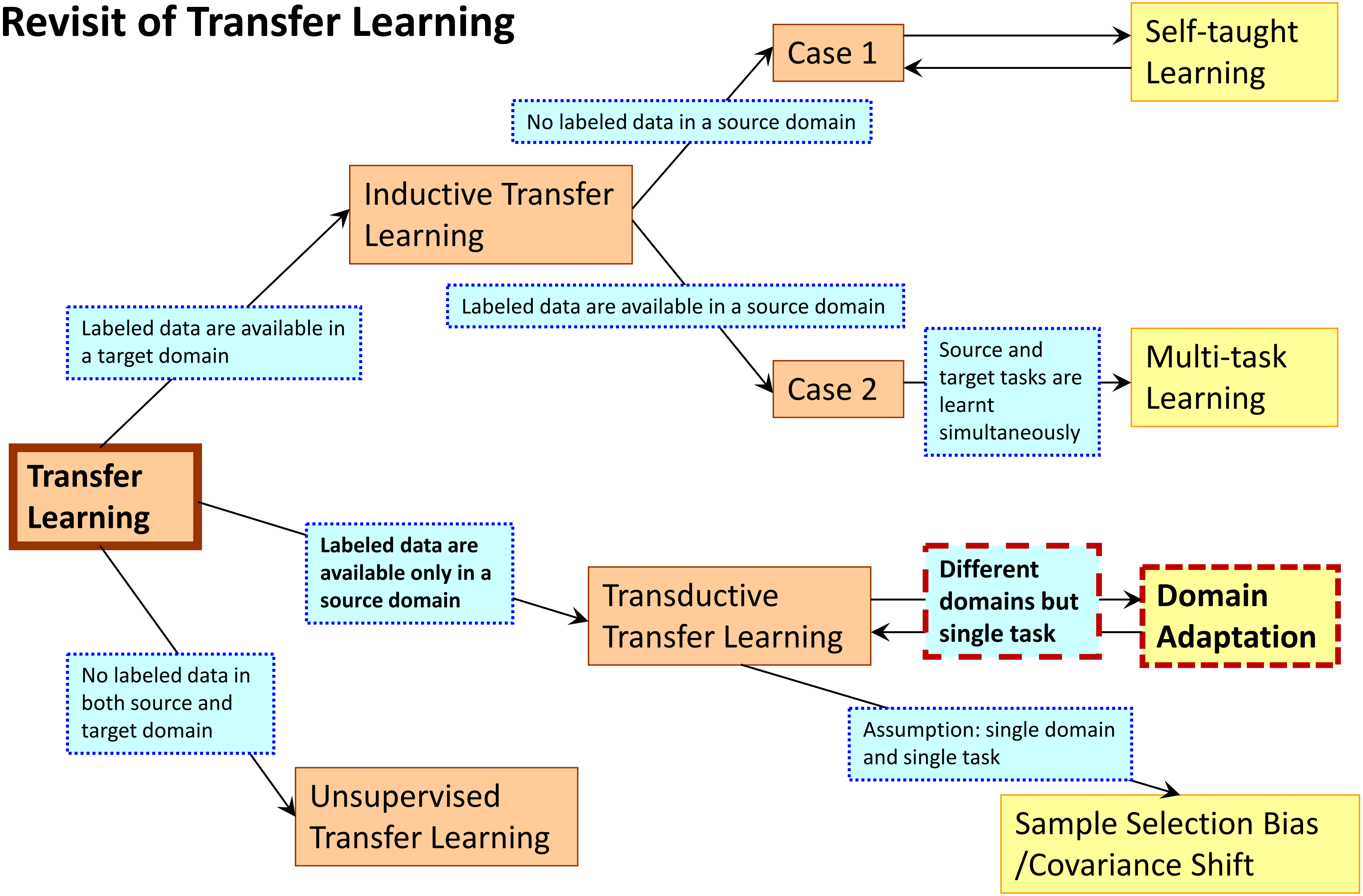
- Beyond standard classification, we might need to address image translation/manipulation/style transfer tasks.

- More image translation/manipulation/style transfer tasks



Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

Revisit of Transfer Learning



S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE TKDE, 2010.

Domain Adaptation

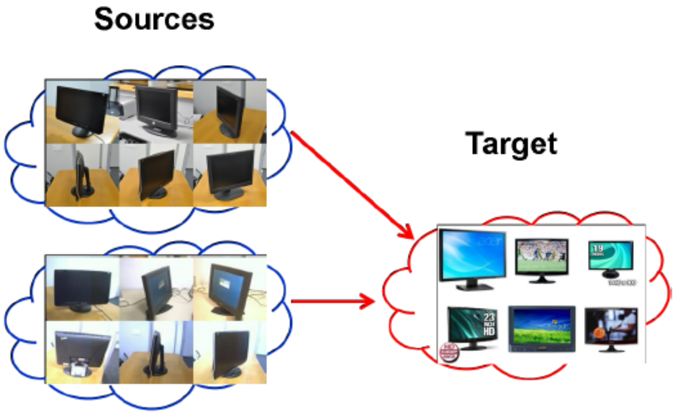


Image: Courtesy to S.J. Pan

- What's DA?
  - Leveraging info from one or more source domains, so that the same learning task in the target domain can be addressed.
  - Typically all the source-domain data are labeled.
- Settings
  - Semi-supervised DA: few target-domain data are with labels.
  - Unsupervised DA: no label info available in the target-domain. (shall we address supervised DA?)
  - Imbalanced DA: fewer classes of interest in the target domain
  - Homogeneous vs. heterogeneous DA

Deep Feature is Sufficiently Promising.

- DeCAF
  - Leveraging an auxiliary large dataset to train CNN.
  - The resulting features exhibit sufficient representation ability.
  - Supporting results on Office+Caltech datasets, etc.

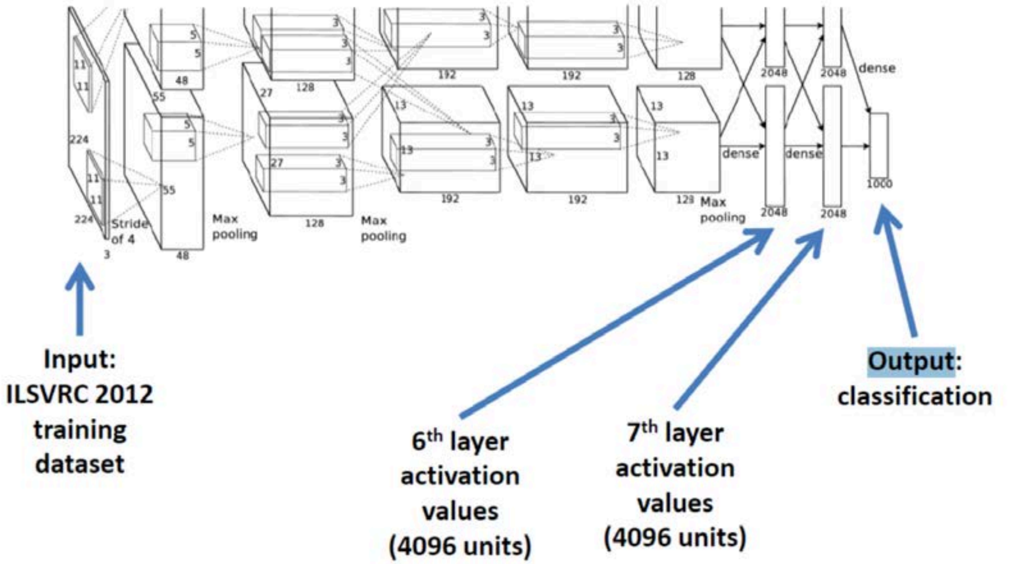
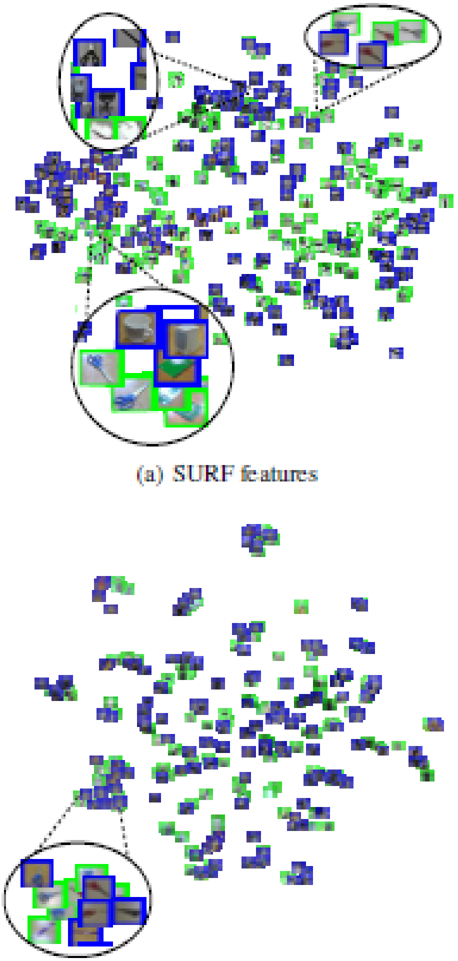


Image: Courtesy to A. Krizhevsky.

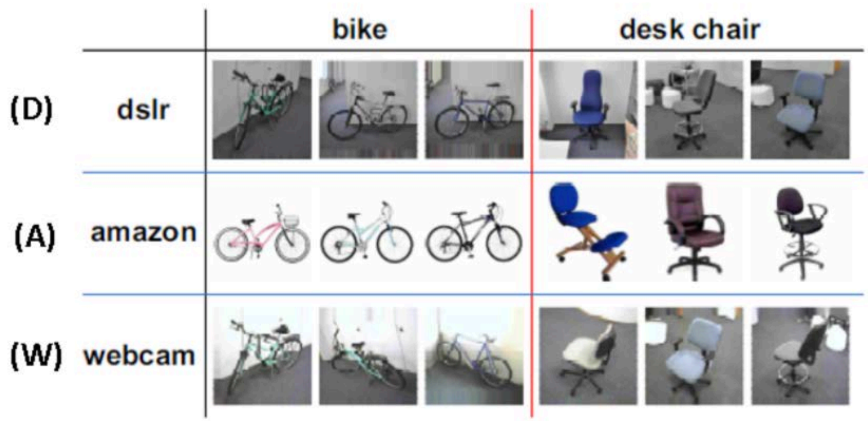


(a) SURF features

(b) $DeCAF_6$

Deep Feature is Sufficiently Promising.

- DeCAF
  - Leveraging an auxiliary large dataset to train CNN.
  - The resulting features exhibit sufficient representation ability.
  - Supporting results on Office+Caltech, etc. object image datasets



Feature	SURF											$Decaf_6$				
data	Raw	SA	SDA	GFK	TCA	JDA	TJM	SCA	JGSA primal	JGSA linear	JGSA RBF	JDA	OTGL	JGSA primal	JGSA linear	JGSA RBF
A→D	35.67	33.76	33.76	40.13	33.76	39.49	45.22	39.49	**47.13**	45.86	45.22	81.53	85.00	**88.54**	85.35	84.71
A→W	31.19	33.22	30.85	36.95	36.27	37.97	42.03	34.92	45.76	**49.49**	45.08	80.68	83.05	81.02	**84.75**	80.00
D→A	28.29	**39.87**	38.73	28.71	31.00	33.09	32.78	31.63	38.00	36.01	38.73	91.96	**92.31**	91.96	92.28	91.96
D→W	83.73	76.95	76.95	80.34	86.10	89.49	85.42	84.41	91.86	91.86	**93.22**	99.32	96.29	**99.66**	98.64	98.64
W→A	31.63	39.25	39.25	27.56	28.91	32.78	29.96	29.96	39.87	**41.02**	40.81	90.71	90.62	90.71	**91.44**	91.34
W→D	84.71	75.16	75.80	85.35	89.17	89.17	89.17	87.26	**90.45**	**90.45**	88.54	**100**	96.25	**100**	**100**	**100**

Donahue et al., DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, ICML 2014

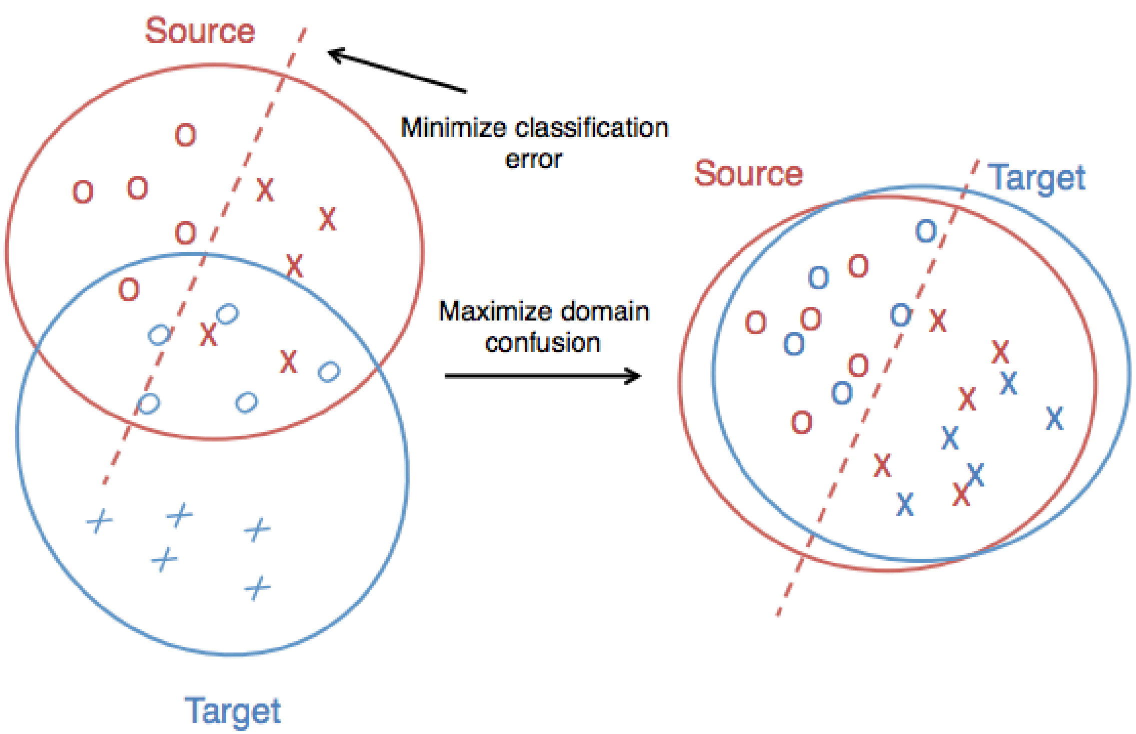
Recent Deep Learning Methods for TL

- Deep Domain Confusion (DDC)
- Domain-Adversarial Training of Neural Networks (DANN)
- Adversarial Discriminative Domain Adaptation (ADDA)
- Domain Separation Network (DSN)
- Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks (PixelDA)
- No More Discrimination: Cross City Adaptation of Road Scene Segmenters

	Shared weights	Adaptation loss	Generative model
DDC	✓	MMD	✗
DANN	✓	Adversarial	✗
ADDA	✗	Adversarial	✗
DSN	Partially shared	MMD/Adversarial	✗
PixelDA	✗	Adversarial	✓

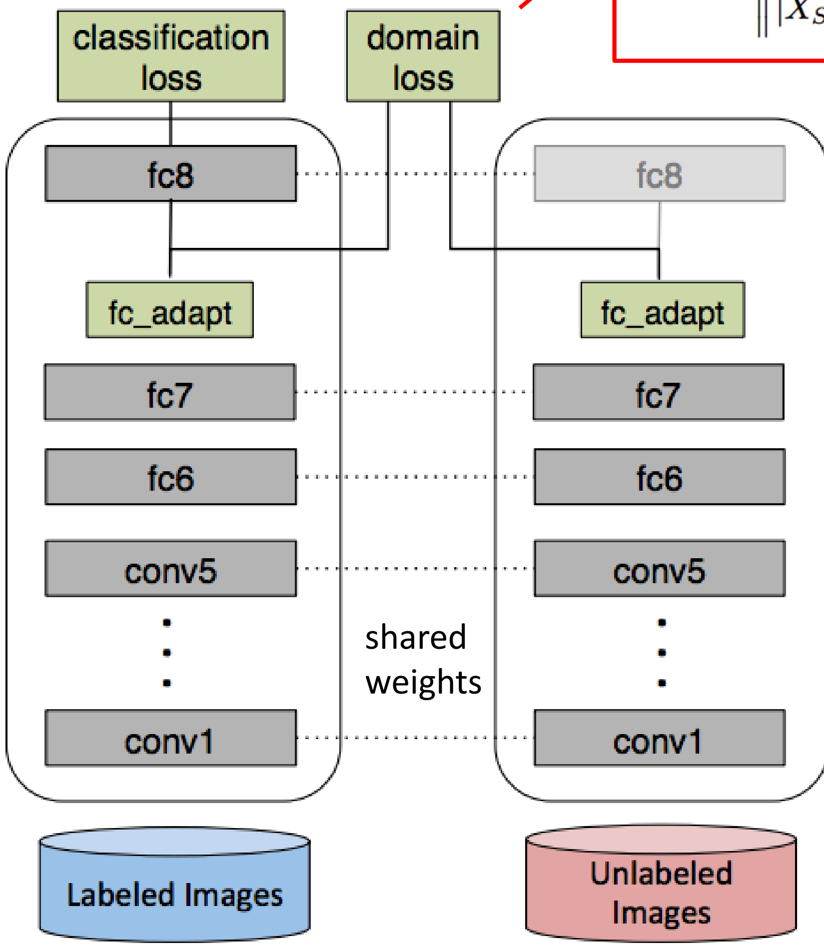
Deep Domain Confusion (DDC)

- Deep Domain Confusion: Maximizing for Domain Invariance
  - Tzeng et al., arXiv: 1412.3474, 2014



Deep Domain Confusion (DDC)

$$\mathrm{MMD}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|$$

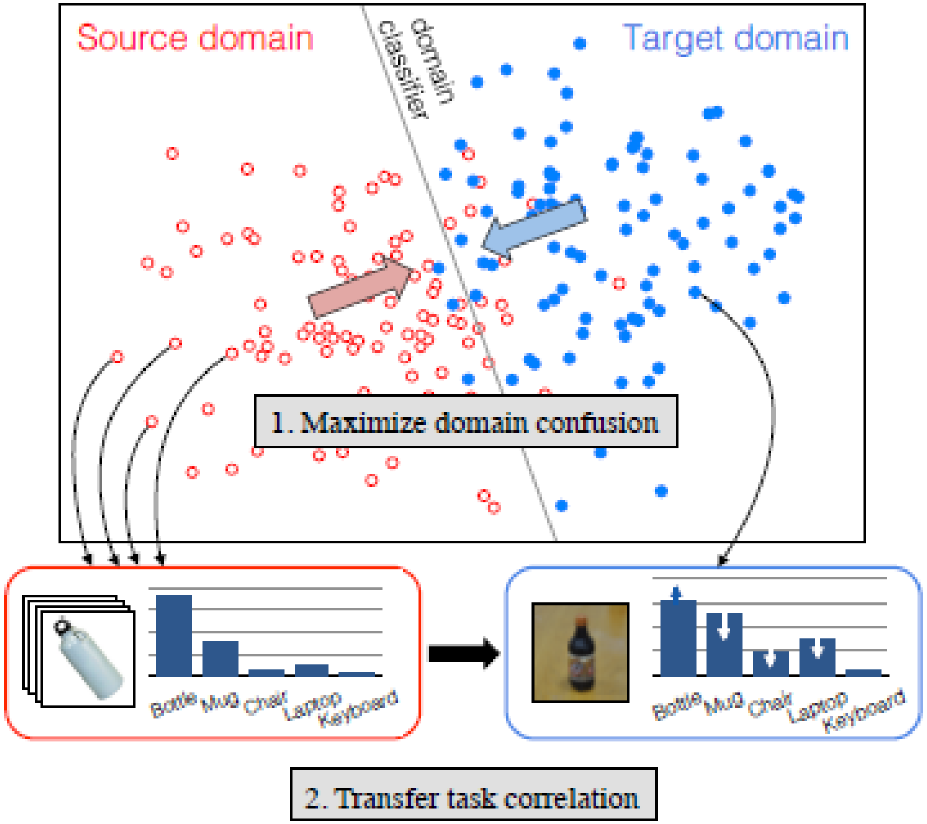


✓ Minimize classification loss:

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \, \mathrm{MMD}^2(X_S, X_T)$$

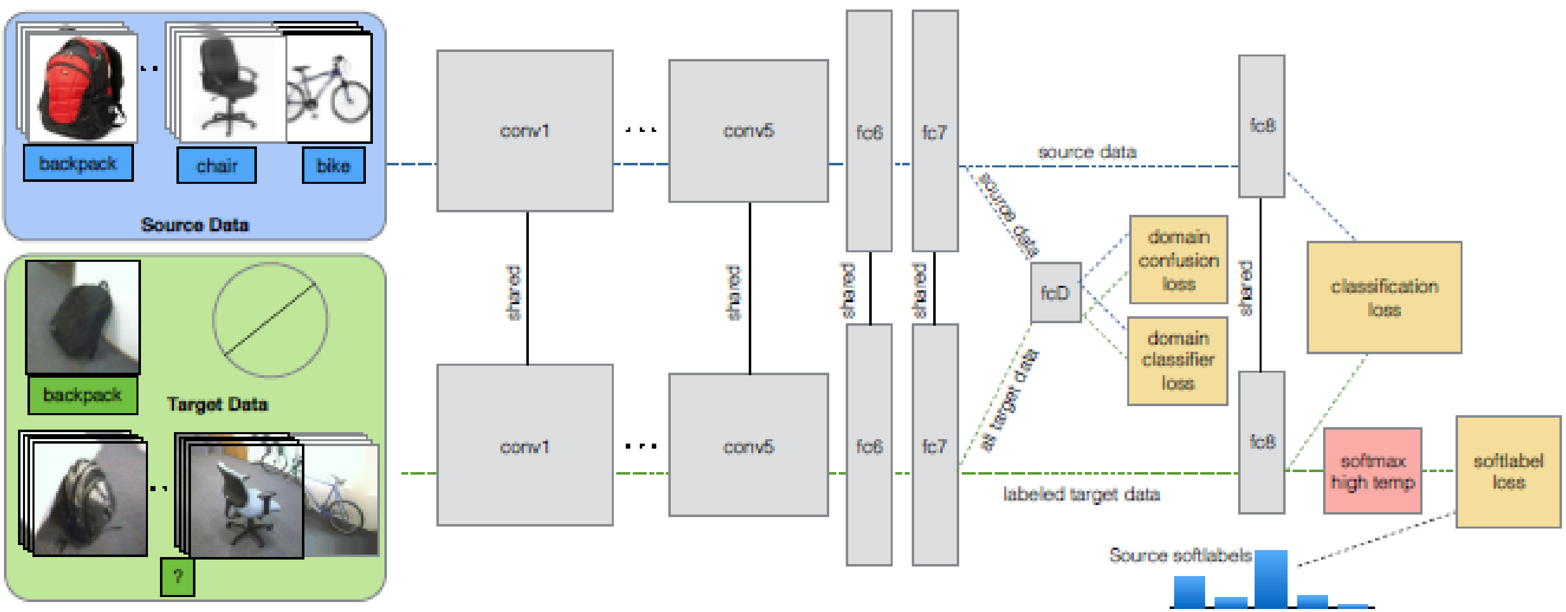
Deep Domain Confusion (DDC)

- Simultaneous Deep Transfer Across Domains and Tasks
  - Tzeng et al., ICCV, 2015



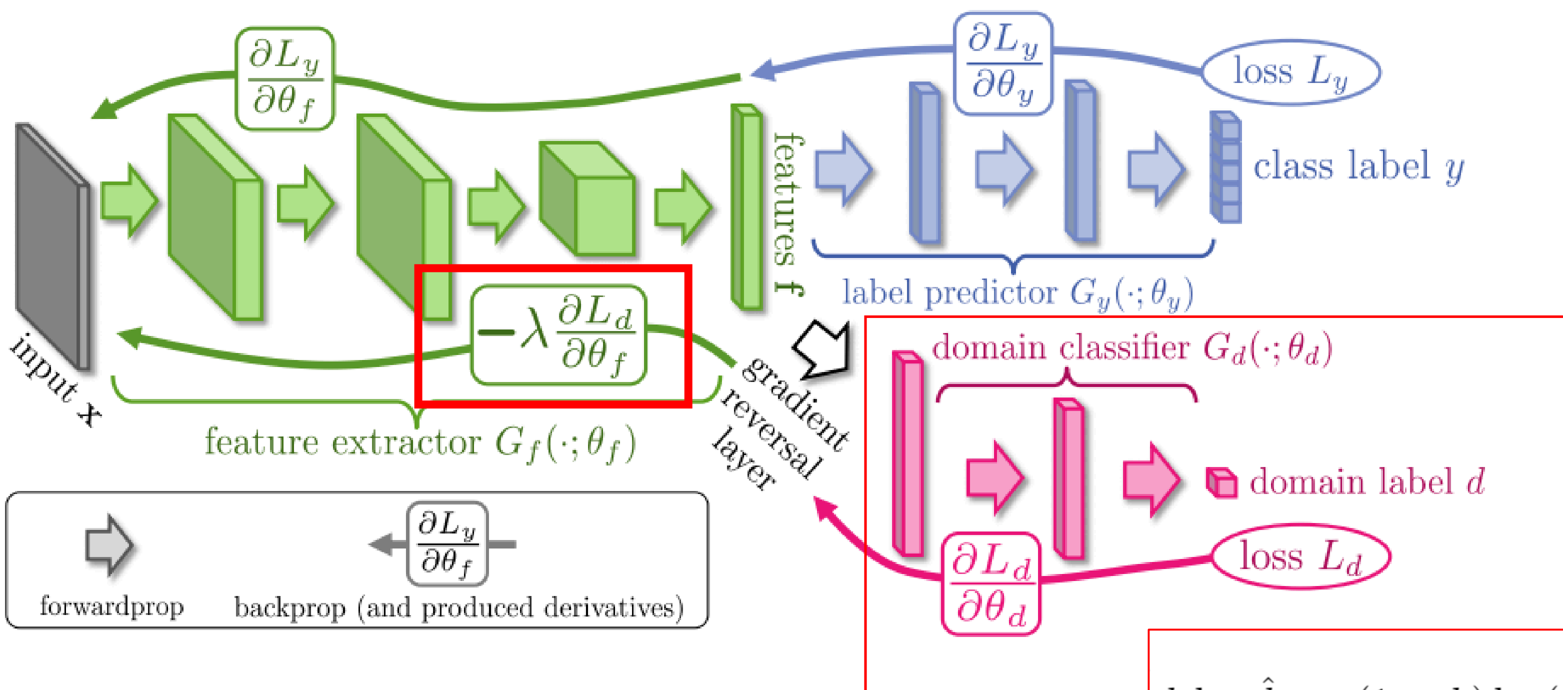
Deep Domain Confusion (DDC)

- Simultaneous Deep Transfer Across Domains and Tasks
  - Tzeng et al., ICCV, 2015
  - Soft label loss is additionally introduced.



Domain Confusion by Domain-Adversarial Training

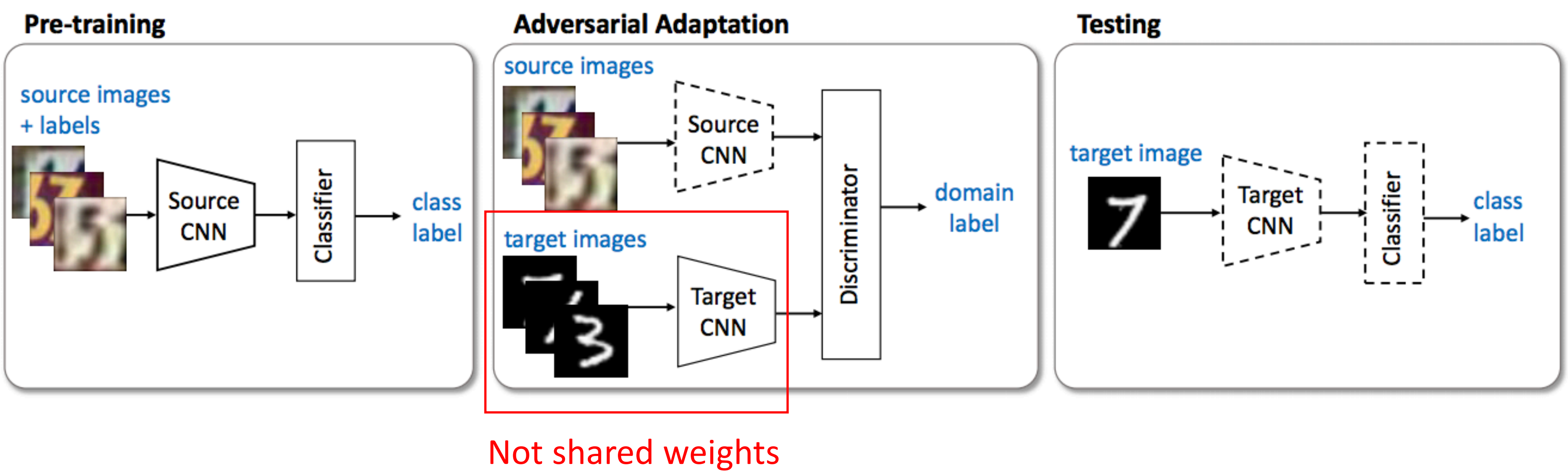
- Domain-Adversarial Training of Neural Networks (DANN)
  - Y. Ganin et al., ICML 2015
  - Maximize domain confusion = maximize domain classification loss
  - Minimize source-domain data classification loss



$$d_i \log \hat{d}_i \; + \; (1 - d_i) \log(1 - \hat{d}_i)$$

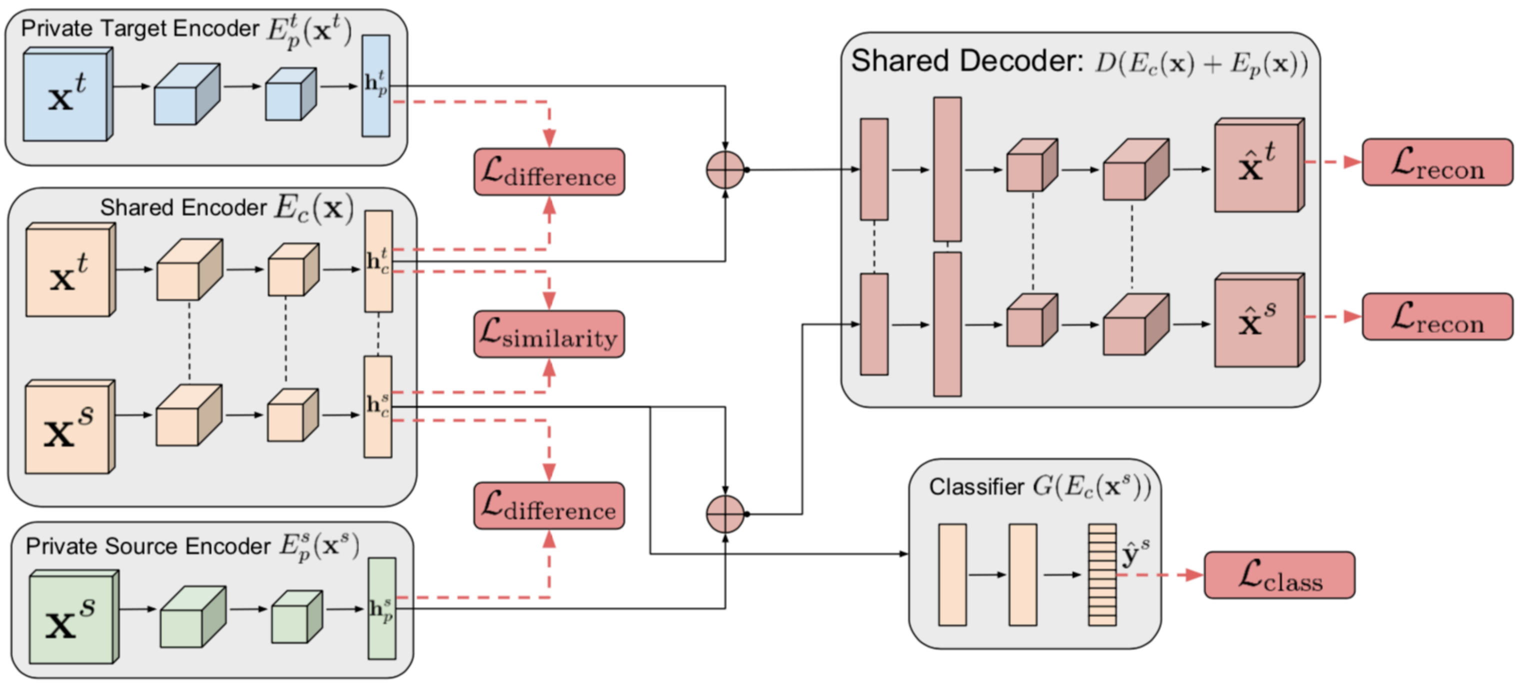
Domain Confusion by Domain-Adversarial Training

- Adversarial Discriminative Domain Adaptation
  - Tzeng et al., CVPR 2017
  - Maximize domain confusion = maximize domain classification loss
  - Minimize source-domain data classification loss
  - Compared to DANN, a distinct decoder for the target domain is considered.



Beyond Domain Confusion

- Domain Separation Network
  - Bousmalis et al., NIPS 2016
  - Separate encoders for domain-invariant and domain-specific features



Beyond Domain Confusion

- Domain Separation Network, NIPS 2016
  - Example results

Source-domain image X_s

Reconstruct private + shared features $D(E_c(x_s)+E_p(x_s))$

Reconstruct private feature $D(E_p(x_s))$

(a) MNIST (source) (b) MNIST-M (target)

Reconstruct shared feature only $D(E_c(x_s))$

Target-domain image X_T

Beyond Domain Confusion

- Domain Separation Network, NIPS 2016
  - Example results

Reconstruct private feature $D(E_p(x_T))$

Source-domain image X_s

Reconstruct private + shared features $D(E_c(x_s)+E_p(x_s))$

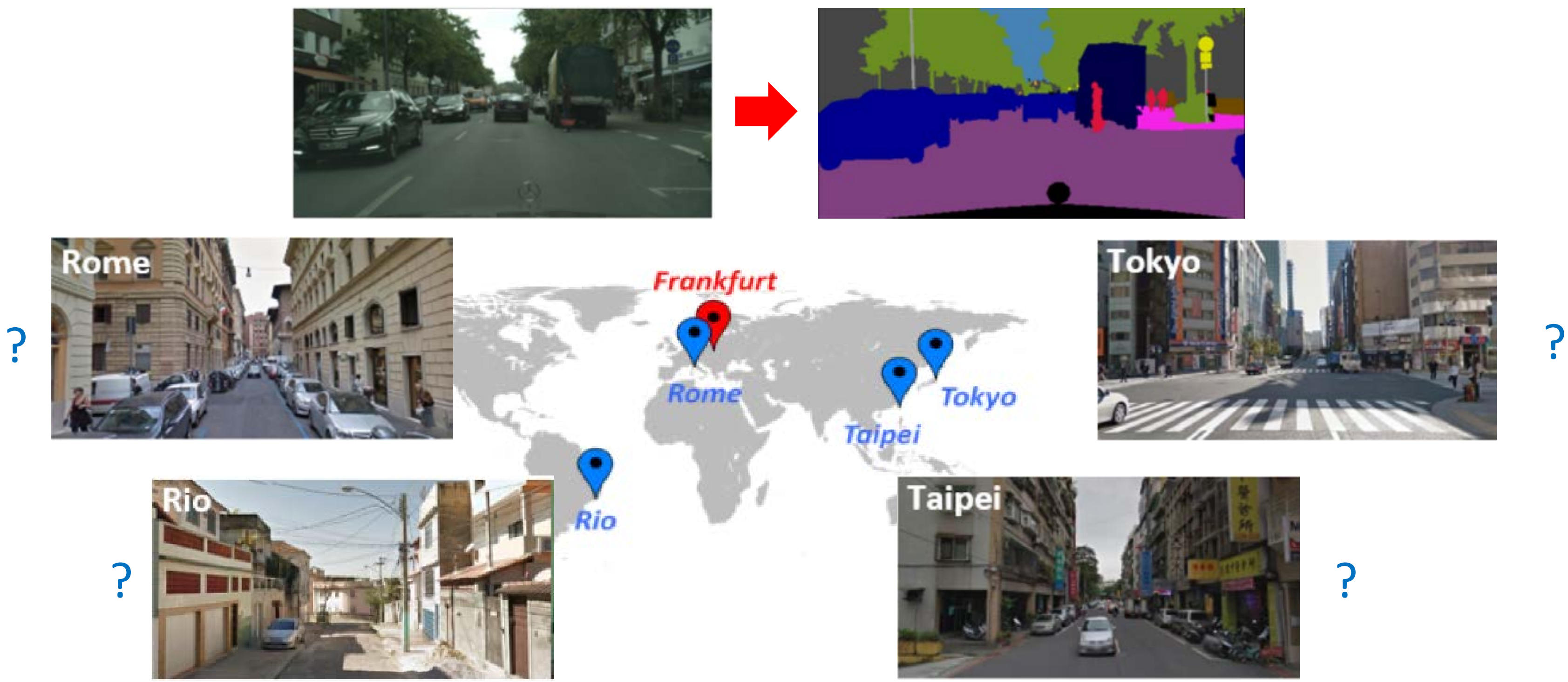
(a) MNIST (source) (b) MNIST-M (target)

Target-domain image X_T

Reconstruct shared feature only $D(E_c(x_T))$

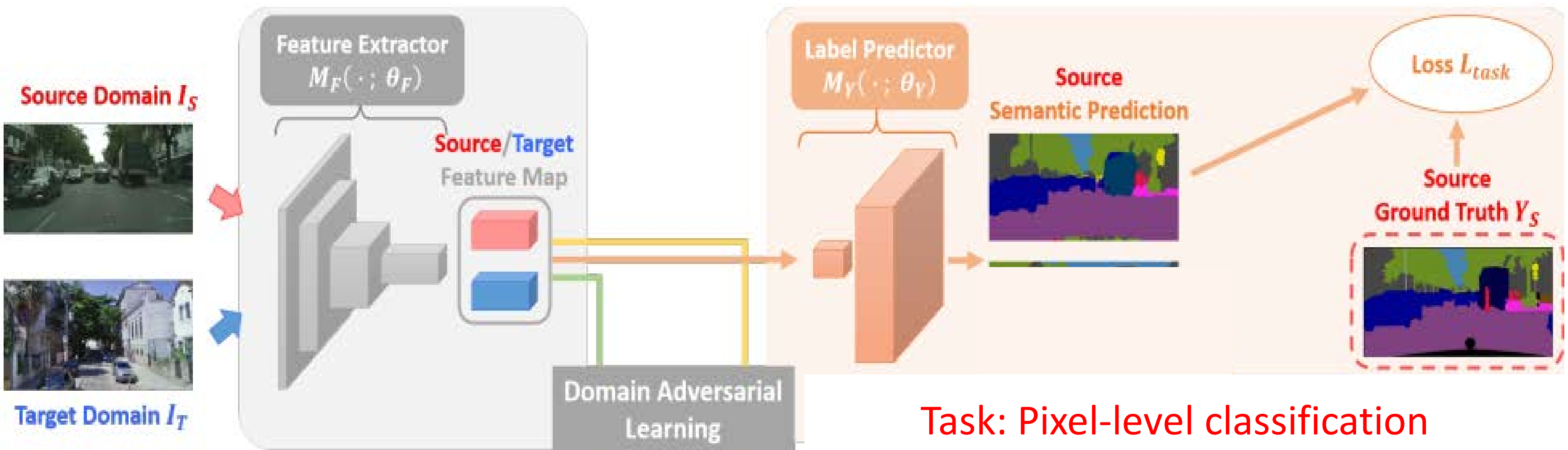
Semantic Segmentation Across Cities

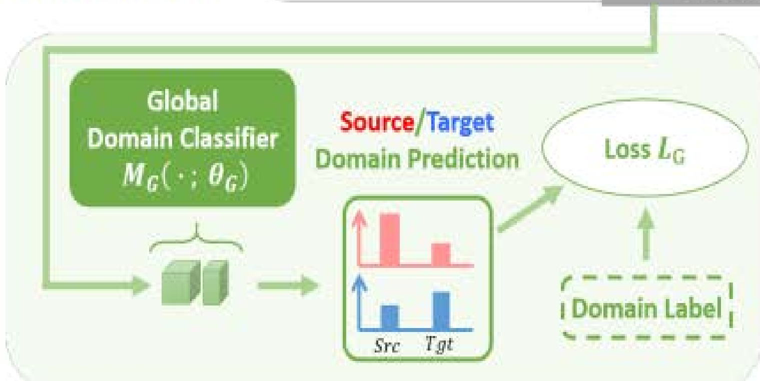
- No More Discrimination: Cross City Adaptation of Road Scene Segmenters
  - Chen et al., ICCV 2017
  - Weakly supervised DA for semantic segmentation



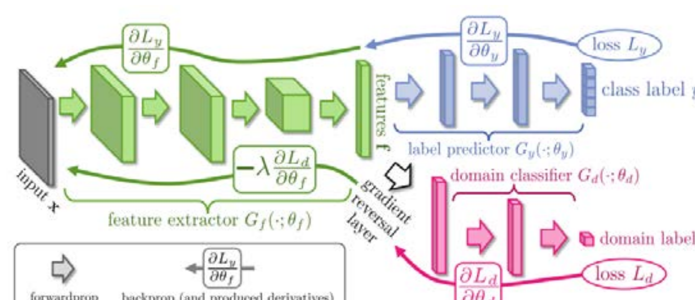
Semantic Segmentation Across Cities

- No More Discrimination: Cross City Adaptation of Road Scene Segmenters
  - Chen et al., ICCV 2017
  - Weakly supervised DA for semantic segmentation



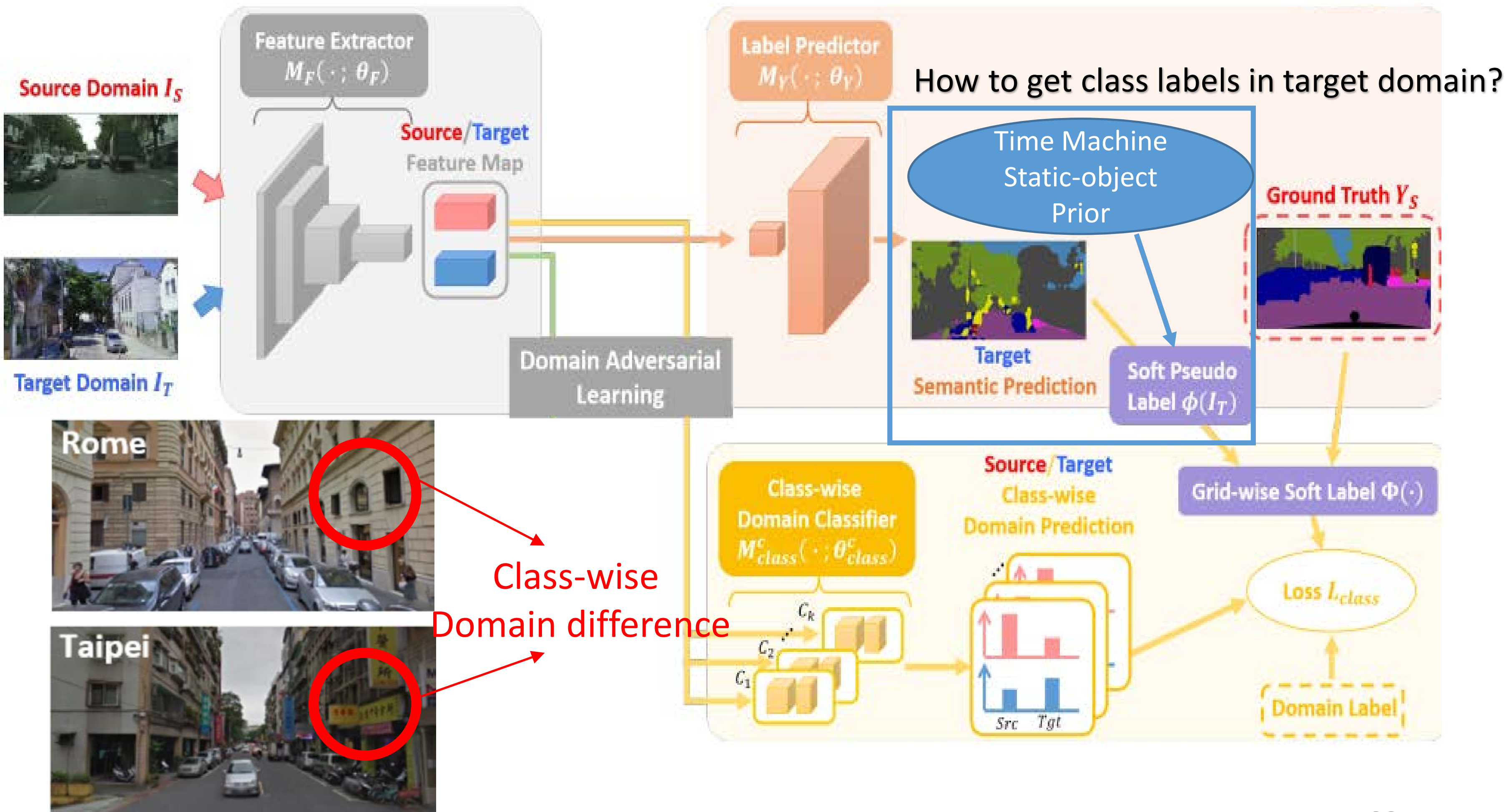


Task: Pixel-level classification



Semantic Segmentation Across Cities

- No More Discrimination: Cross City Adaptation of Road Scene Segmenters

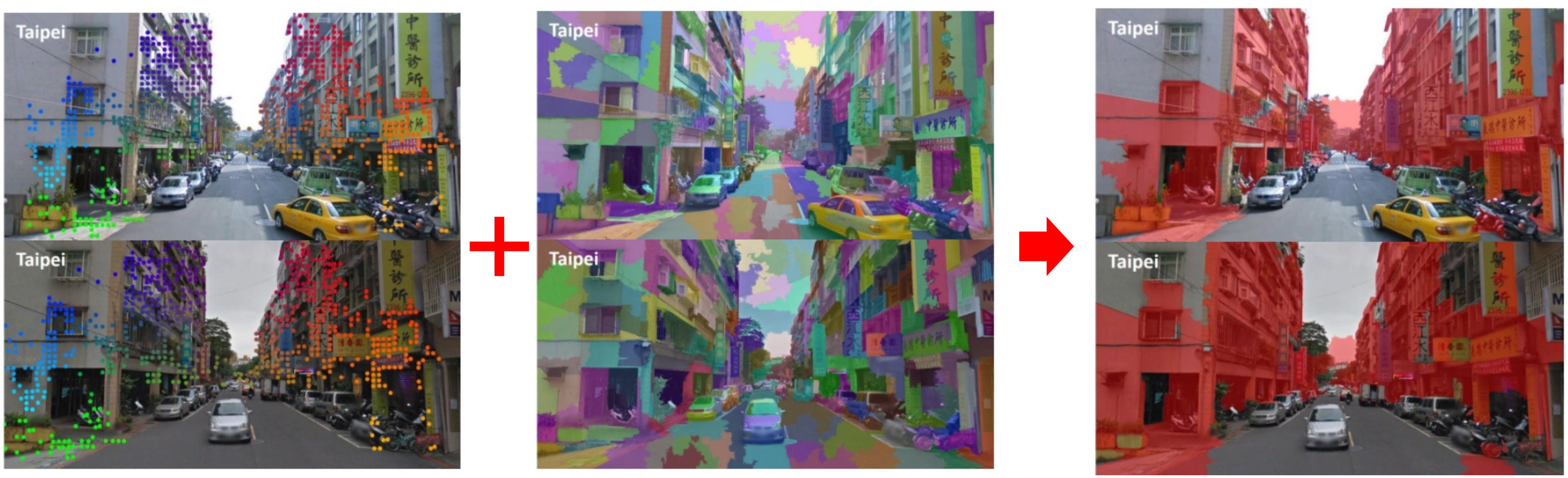


Semantic Segmentation Across Cities

- No More Discrimination: Cross City Adaptation of Road Scene Segmenters
  - Chen et al., ICCV 2017
  - Weakly supervised DA for semantic segmentation
  - Static-object prior from Google Map Time Machine features

2015

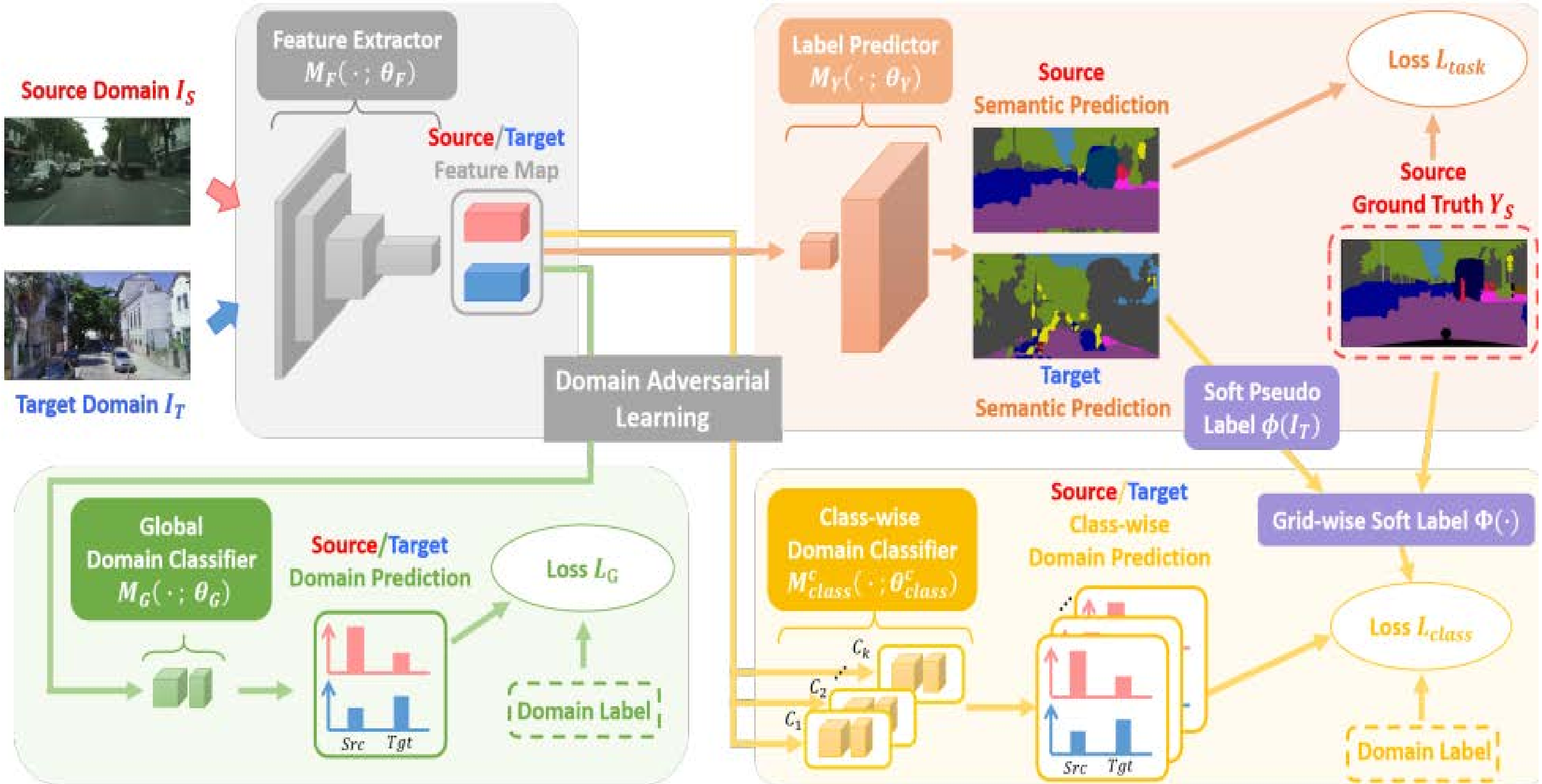
2016



Dense Match | Superpixel | Static classes

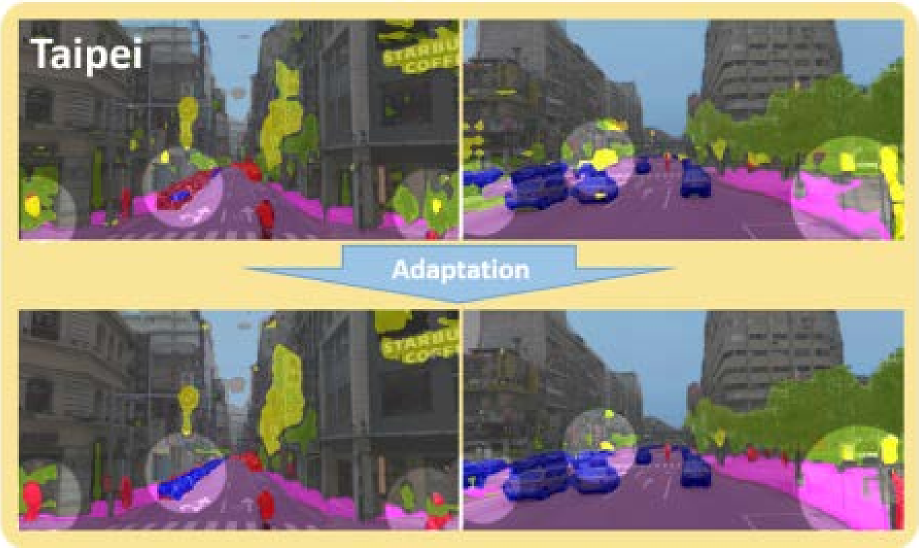
Semantic Segmentation Across Cities

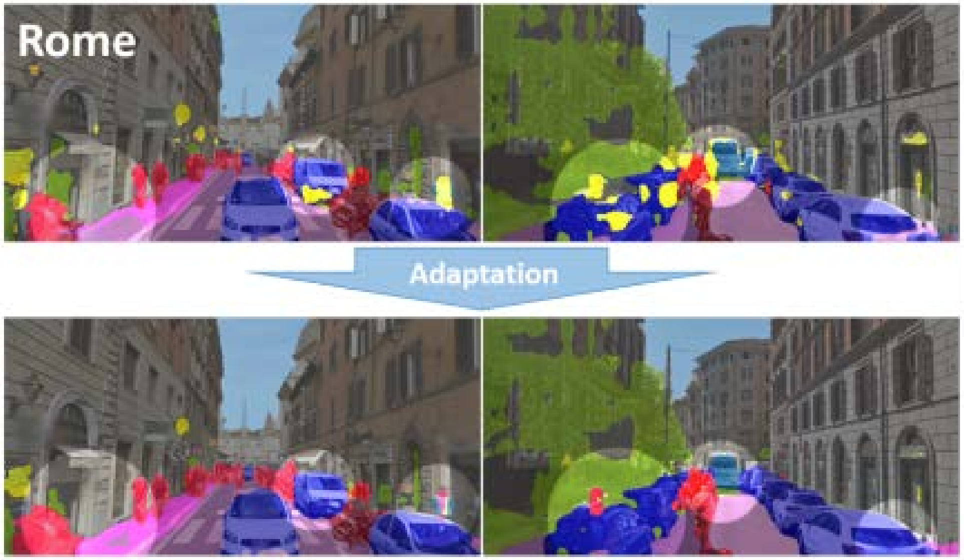
- No More Discrimination: Cross City Adaptation of Road Scene Segmenters



Semantic Segmentation Across Cities

- No More Discrimination: Cross City Adaptation of Road Scene Segmenters
  - Chen et al., ICCV 2017
  - Weakly supervised DA for semantic segmentation
  - Static-object prior from Google Map Time Machine features
  - Qualitative example results

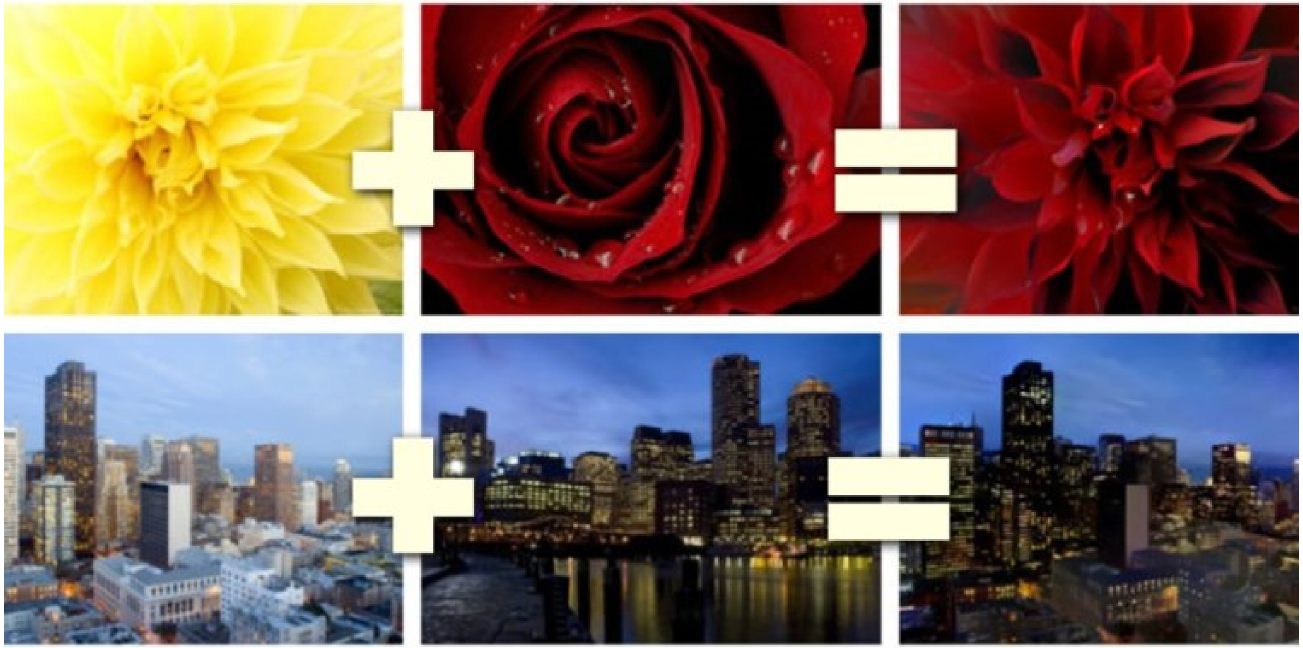




Transfer Learning for Manipulating Data?

- TL not only addresses cross-domain classification tasks.
- Let's see how we can synthesize and manipulate data across domains.
- As a computer vision guy, let's focus on visual data in this lecture…

Source Domain



Target Domain

What to Cover?

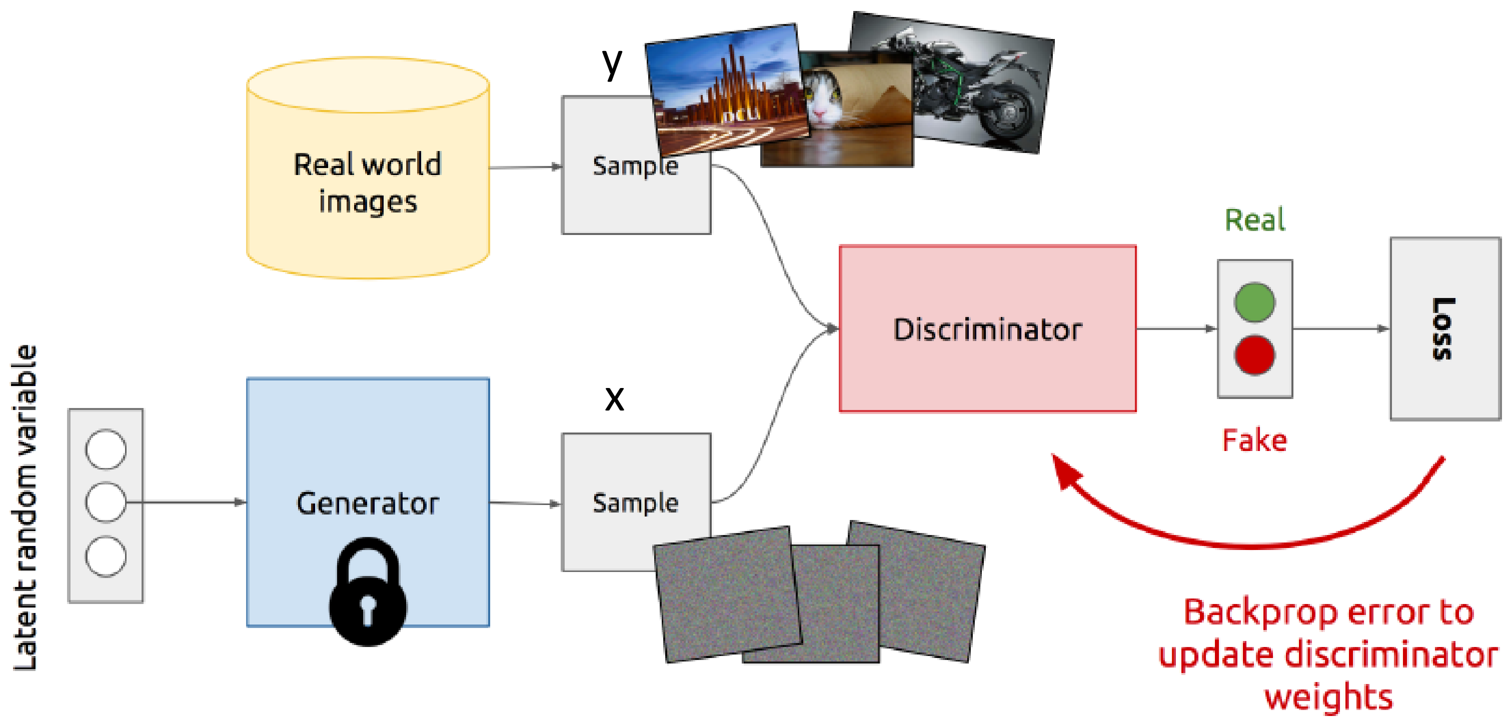
- Cross-Domain Image Translation
  - Pix2pix (CVPR’17)
  - CycleGAN (ICCV’17), DualGAN (ICCV’17), DiscoGAN (ICML’17)
  - UNIT (NIPS’17)
  - DTN (ICLR’17)
  - Beyond image translation

What to Cover in Transfer Learning?

- Cross-Domain Image Translation
  - Pix2pix (CVPR’17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS’17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR’17) : Learning cross-domain image representation (with unpaired training data)
  - Beyond image translation

A Super Brief Review for *Generative Adversarial Networks (GAN)*

- Architecture of GAN
  - Loss: $\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D\ (G(x)))] + \mathbb{E}[\log D(y)]$



Goodfellow et al., Generative Adversarial Nets, NIPS, 2014

Pix2pix

- Image-to-image translation with conditional adversarial networks (CVPR'17)
  - Can be viewed as image style transfer

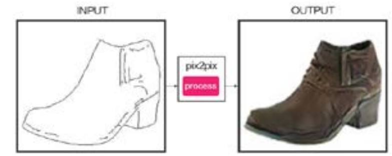
Sketch

Photo

Isola et al. " Image-to-image translation with conditional adversarial networks." *CVPR* 2017.

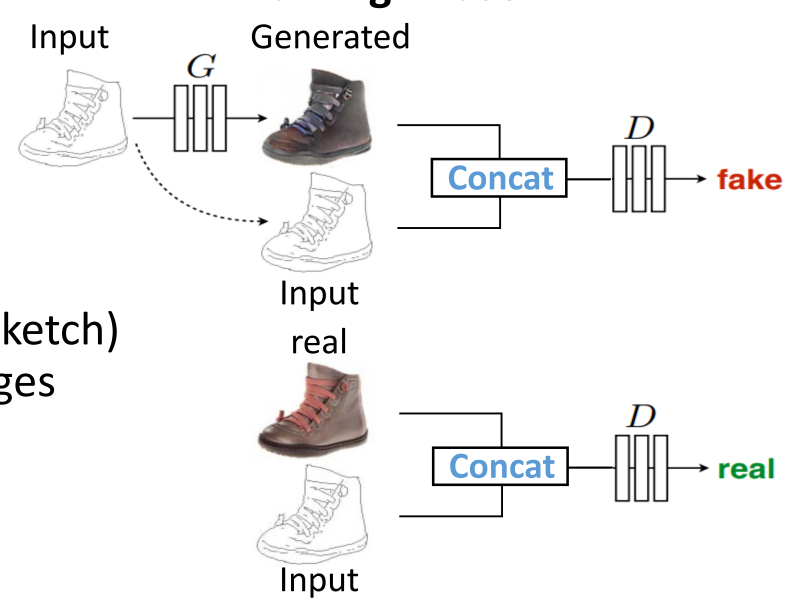
Pix2pix

Testing Phase



- **Goal / Problem Setting**
  - Image translation across two distinct domains (e.g., sketch v.s. photo)
  - Pairwise training data
- Method: Conditional GAN
  - Example: Sketch to Photo
    - Generator
      Input: Sketch
      Output: Photo
    - Discriminator
      Input: Concatenation of Input(Sketch) & Synthesized/Real(Photo) images
      Output: Real or Fake

Training Phase

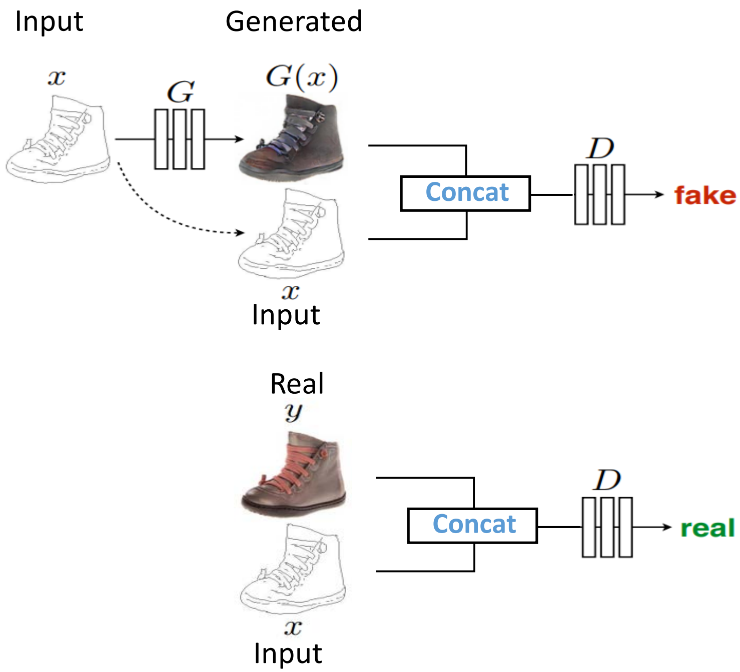


Isola et al. " Image-to-image translation with conditional adversarial networks." *CVPR* 2017.

Pix2pix

- ***Learning the model***

Training Phase



Overall objective function

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

Conditional GAN loss

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_x[\underbrace{\log(1 - D(\overbrace{x, G(x)}^{\text{Concatenate}}))]}_{\text{Fake (Generated)}} + \mathbb{E}_{x,y}[\underbrace{\log D(\overbrace{x, y}^{\text{Concatenate}})]}_{\text{Real}}$$

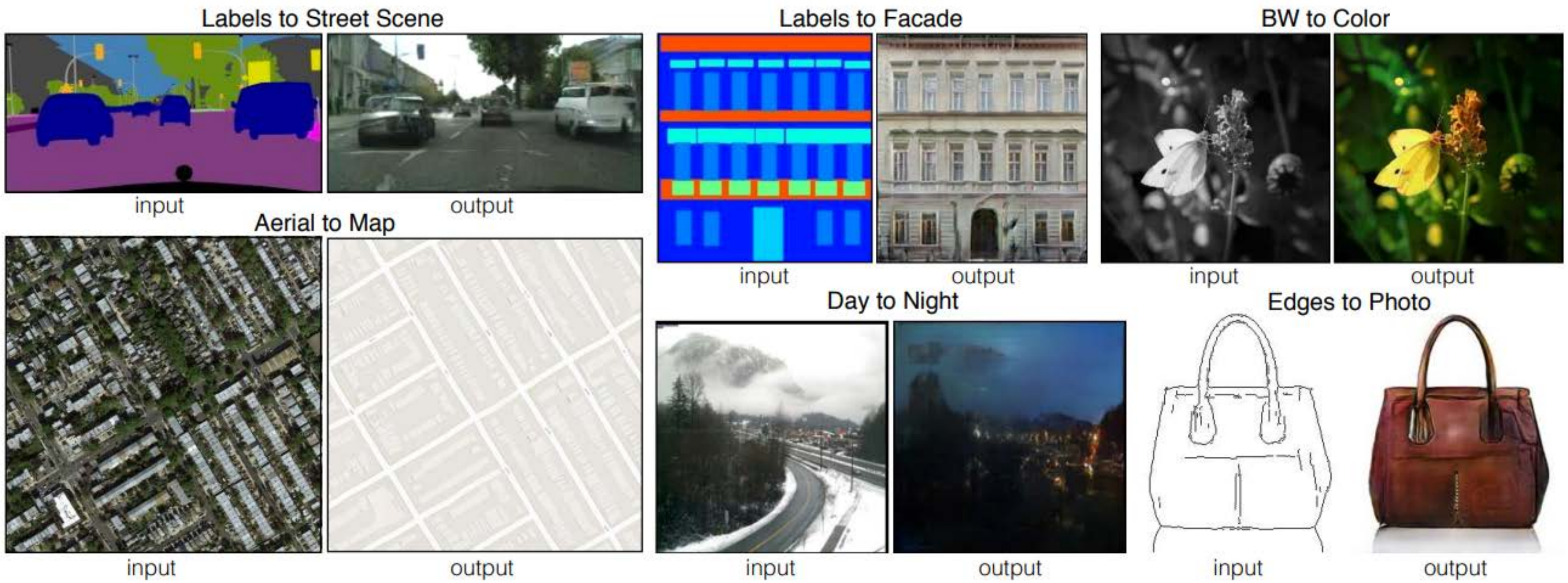
Reconstruction Loss

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[\|y - G(x)\|_1]$$

Isola et al. " Image-to-image translation with conditional adversarial networks." *CVPR* 2017.

Pix2pix

- ***Experiment results***



Demo page: https://affinelayer.com/pixsrv/

Isola et al. " Image-to-image translation with conditional adversarial networks." *CVPR* 2017.

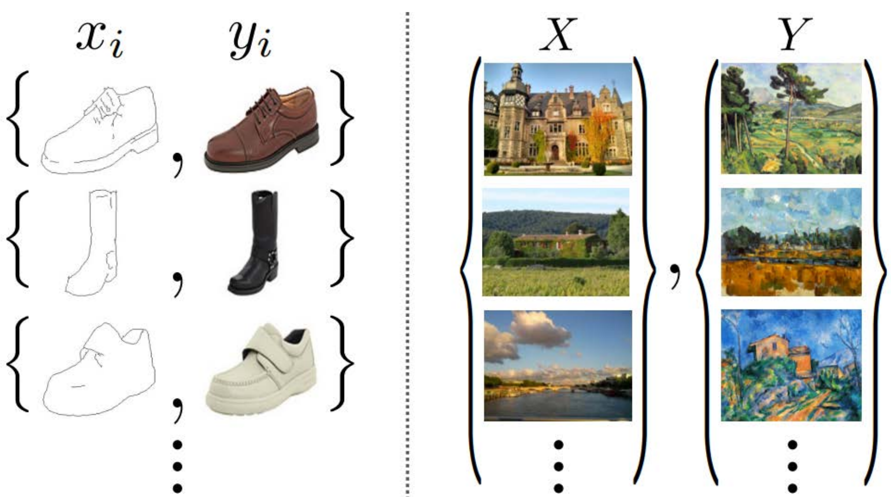
What to Cover?

- Cross-Domain Image Translation
  - Pix2pix (CVPR'17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS'17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR'17) : Learning cross-domain image representation (with unpaired training data)
  - Beyond image translation
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR'18) : Image translation via representation disentanglement
  - CDRD (CVPR'18) : Cross-domain representation disentanglement and translation

CycleGAN/DiscoGAN/DualGAN

- CycleGAN (CVPR'17)
  - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks -to-image translation with conditional adversarial networks

Paired **Unpaired**



1-to-1 Correspondence

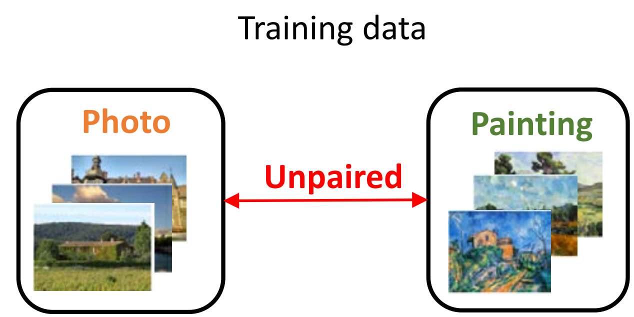
No Correspondence

- Easier to collect training data
- More practical

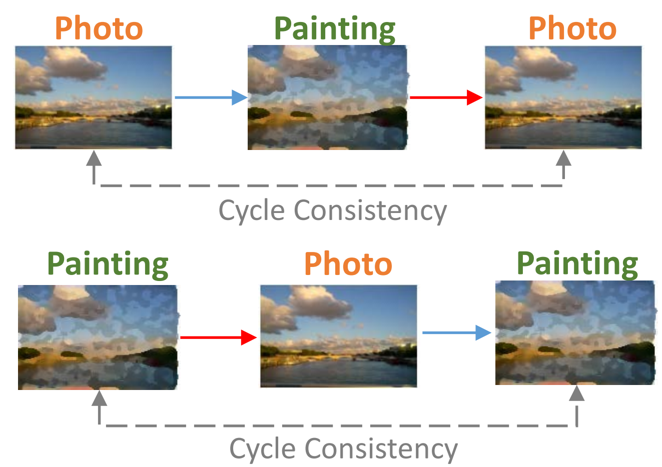
Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN

- **Goal / Problem Setting**
  - Image translation across two distinct domains
  - Unpaired training data



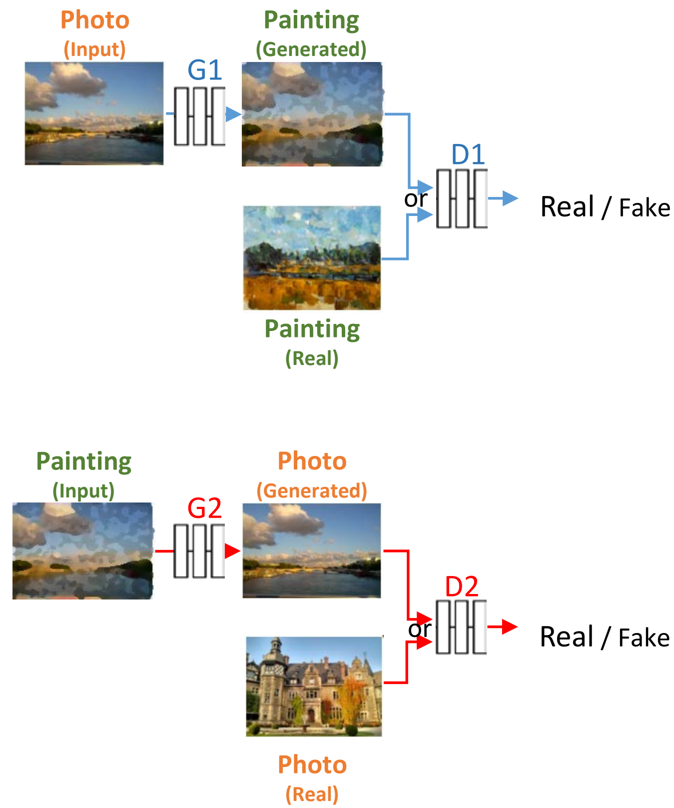
- **Idea**
  - Autoencoding-like image translation
  - Cycle consistency between two domains



Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN

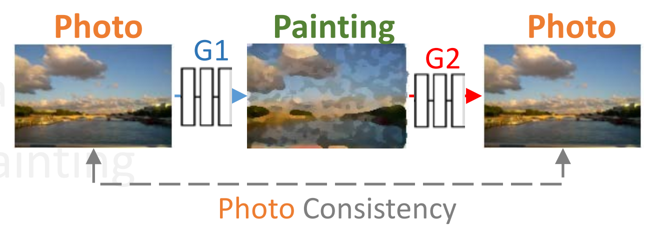
- Method (Example: Photo & Painting)
  - Based on 2 GANs
    - First GAN (G1, D1): Photo to Painting
    - Second GAN (G2, D2): Painting to Photo
  - Cycle Consistency
    - Photo consistency
    - Painting consistency



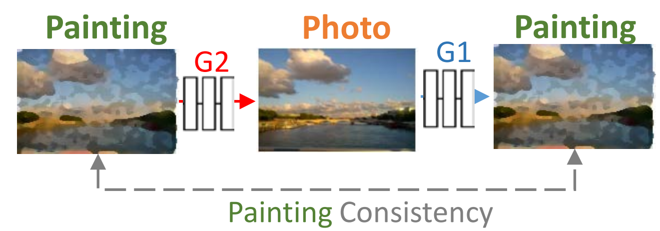
Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN

- Method (Example: Photo vs. Painting)
  - Based on 2 GANs
    - First GAN (G1, D1): Photo to Pa
    - Second GAN (G2, D2): Photo to Painting



  - Cycle Consistency
    - Photo consistency
    - Painting consistency



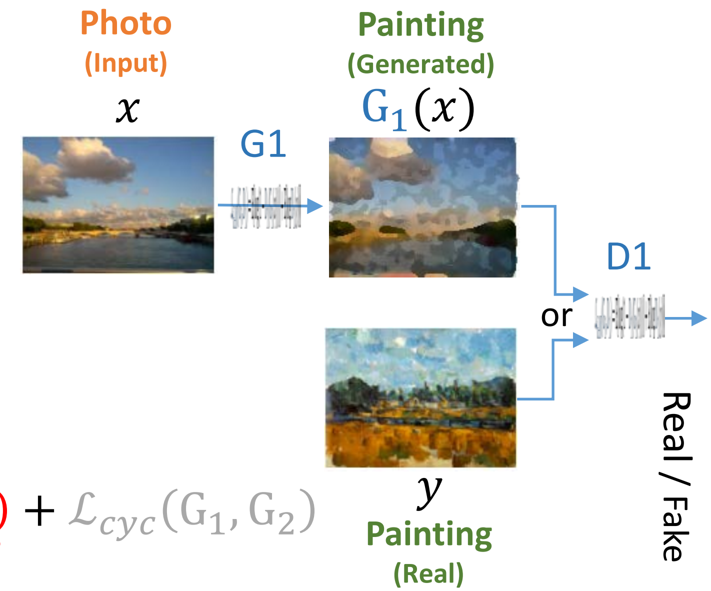
Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN

- **Learning**

Overall objective function

$$G_1^*, G_2^* = \arg\min_{G_1,G_2}\max_{D_1,D_2} \underbrace{\mathcal{L}_{GAN}(G_1, D_1)}_{\text{First GAN}} + \underbrace{\mathcal{L}_{GAN}(G_2, D_2)}_{\text{Second GAN}} + \mathcal{L}_{cyc}(G_1, G_2)$$

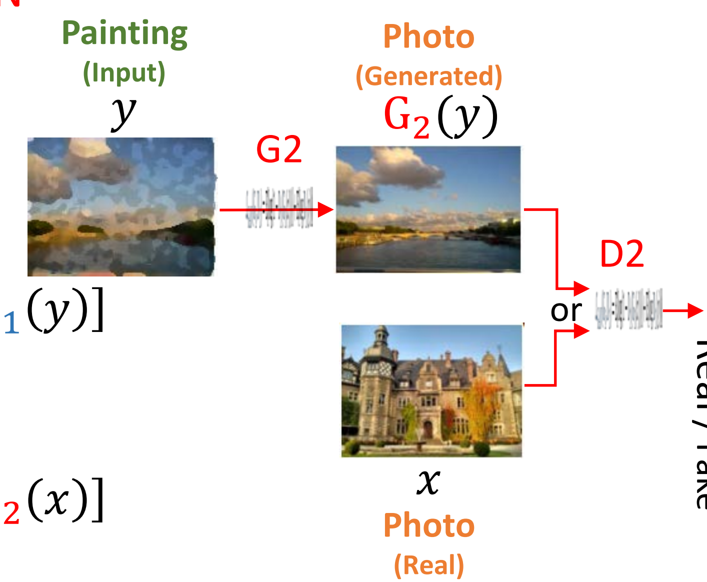


- **Adversarial Loss**
  - First GAN (G1, D1):

$$\mathcal{L}_{GAN}(G_1, D_1) = \mathbb{E}[\log(1 - D_1(G_1(x)))] + \mathbb{E}[\log D_1(y)]$$

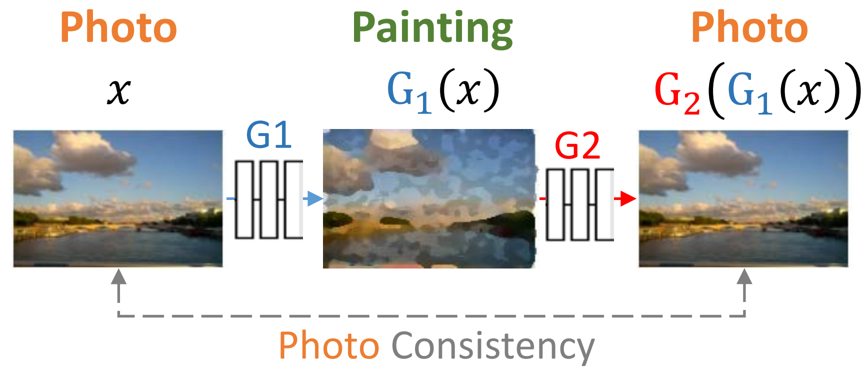
  - Second GAN (G2, D2):

$$\mathcal{L}_{GAN}(G_2, D_2) = \mathbb{E}[\log(1 - D_2(G_2(y)))] + \mathbb{E}[\log D_2(x)]$$



Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN



- **Learning**

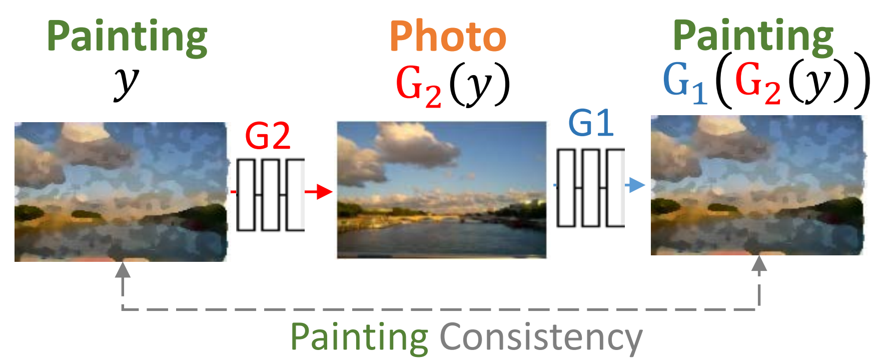
Overall objective function

$$G_1^*, G_2^* = \arg \min_{G_1, G_2} \max_{D_1, D_2} \mathcal{L}_{GAN}(G_1, D_1) + \mathcal{L}_{GAN}(G_2, D_2) + \mathcal{L}_{cyc}(G_1, G_2)$$

Cycle Consistency

- **Consistency Loss**
  - Photo and Painting consistency

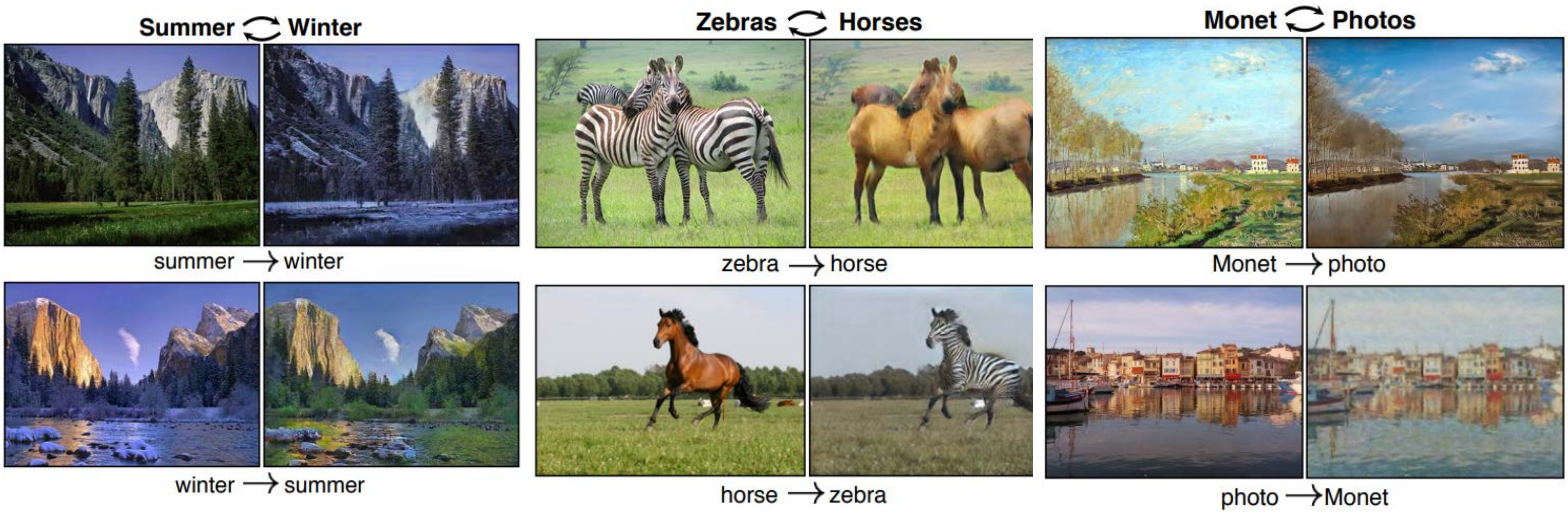
$$\mathcal{L}_{cyc}(G_1, G_2) = \mathbb{E}\left[\left\|G_2(G_1(x)) - x\right\|_1\right] + \left[\left\|G_1(G_2(y)) - y\right\|_1\right]$$



Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

CycleGAN

- Example results

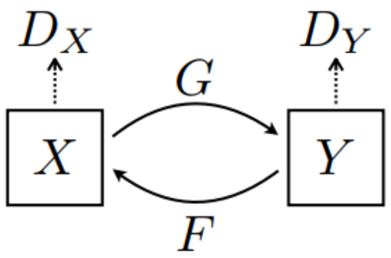


Project Page: https://junyanz.github.io/CycleGAN/

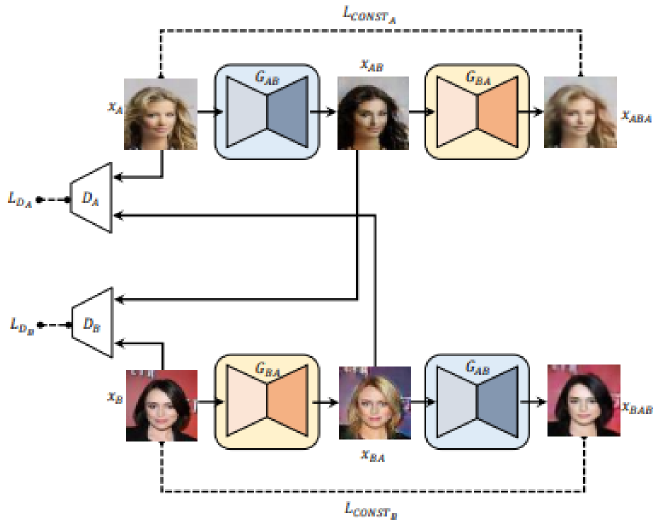
Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.

Image Translation Using Unpaired Training Data

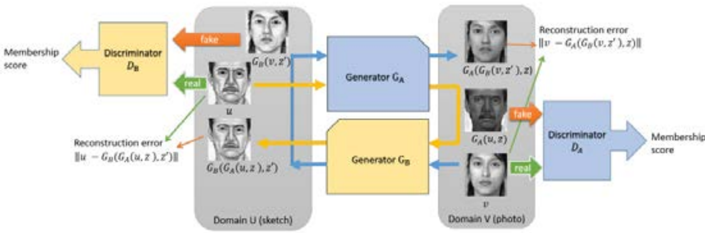
- CycleGAN, DiscoGAN, and DualGAN



CycleGAN
ICCV'17



DiscoGAN
ICML'17



DualGAN
ICCV'17

Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." *CVPR* 2017.
Kim et al. "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks.", *ICML 2017*
Yi, Zili, et al. "Dualgan: Unsupervised dual learning for image-to-image translation." *ICCV 2017*

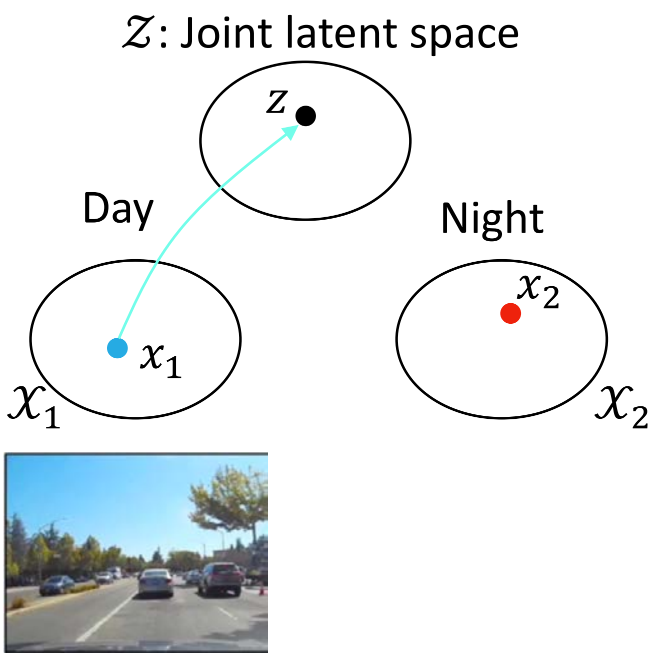
What to Cover in Transfer Learning?

- Cross-Domain Image Translation
  - Pix2pix (CVPR'17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS'17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR'17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR'18) : Image translation via representation disentanglement
  - CDRD (CVPR'18) : Cross-domain representation disentanglement and translation

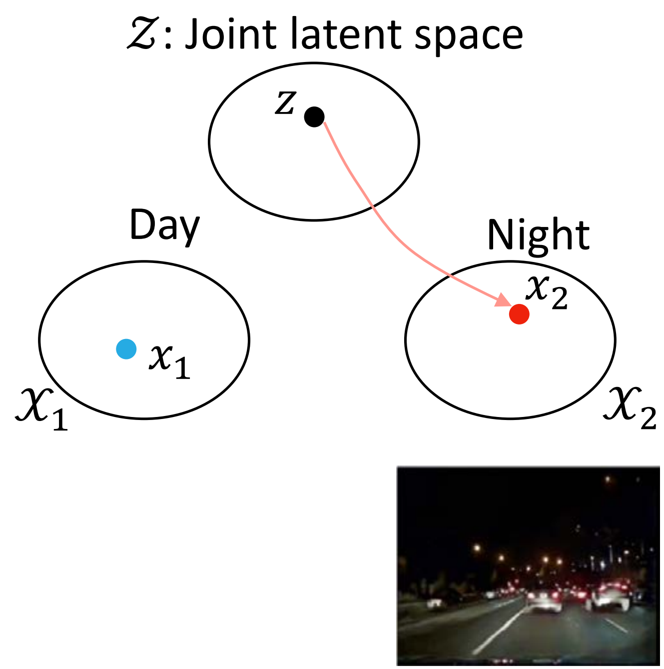
UNIT

- Unsupervised Image-to-Image Translation Networks (NIPS'17)
  - Image translation via learning cross-domain joint representation

Stage1: Encode to the joint space



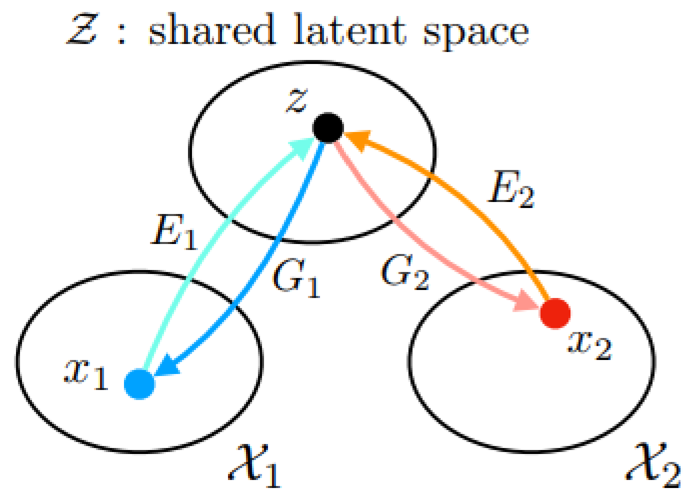
Stage2: Generate cross-domain images

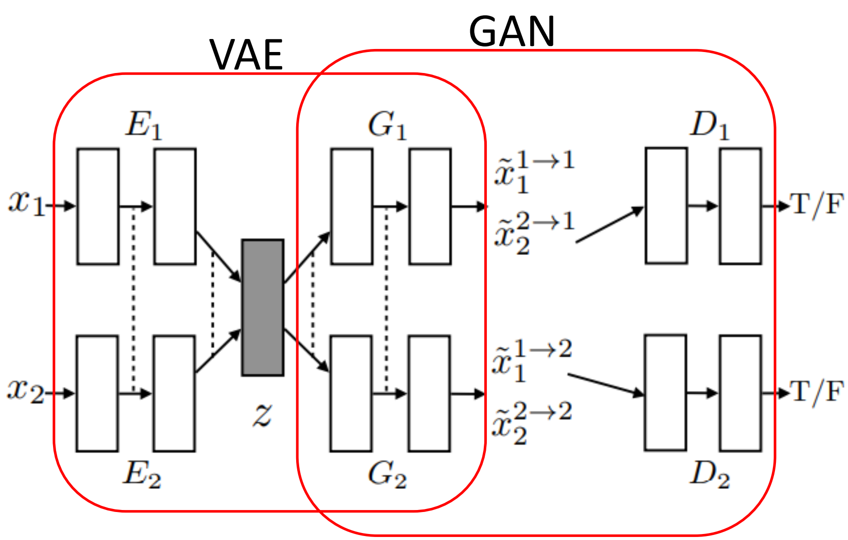


Liu et al., "Unsupervised image-to-image translation networks.", NIPS 2017

UNIT

- Goal/Problem Setting
  - Image translation
    across two distinct domains
  - Unpaired training image data
- Idea
  - Based on two parallel VAE-GAN models

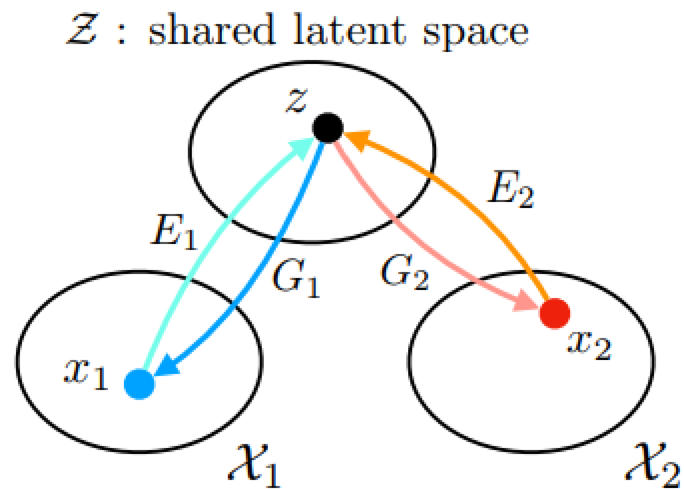


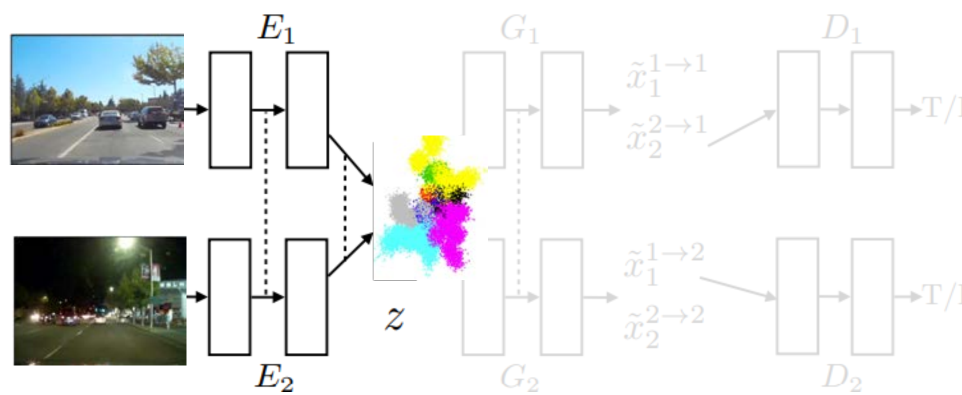


Liu et al., "Unsupervised image-to-image translation networks.“, NIPS 2017

UNIT

- Goal/Problem Setting
  - Image translation across two distinct domains
  - Unpaired training image data
- Idea
  - Based on two parallel VAE-GAN models
  - Learning of joint representation across image domains

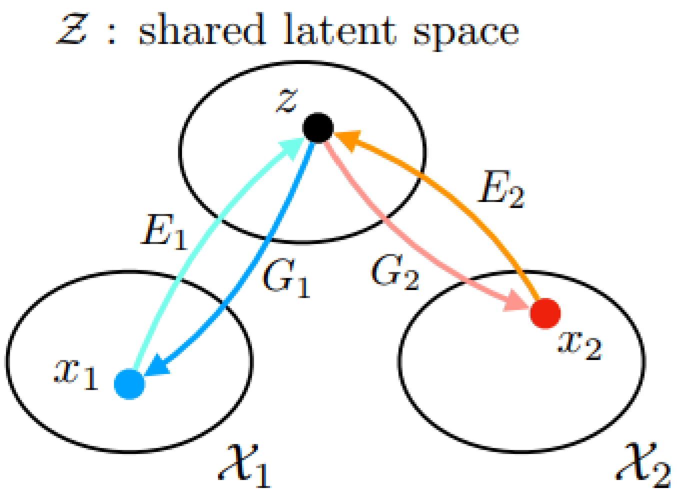


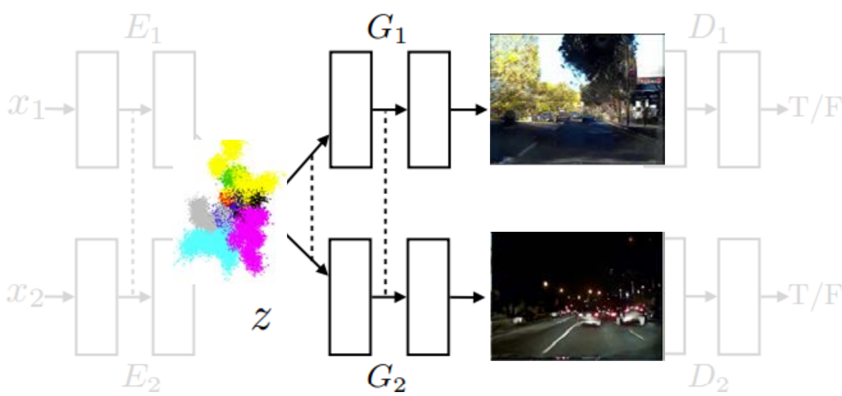


Liu et al., "Unsupervised image-to-image translation networks.", NIPS 2017

UNIT

- Goal/Problem Setting
  - Image translation
    across two distinct domains
  - Unpaired training image data
- Idea
  - Based on two parallel VAE-GAN models
  - Learning of joint representation
    across image domains
  - Generate cross-domain images
    from joint representation





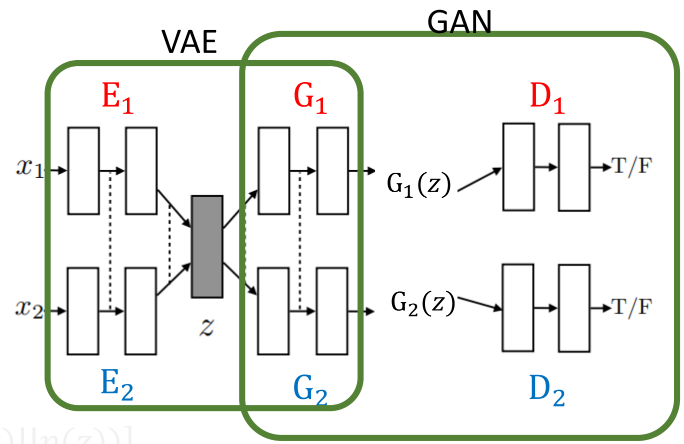
Liu et al., "Unsupervised image-to-image translation networks.", NIPS 2017

UNIT

- ***Learning***

Overall objective function

$$G^* = \arg\min_{G}\max_{D} \underbrace{\mathcal{L}_{VAE}(E_1, G_1, E_2, G_2)}_{\textbf{Variation Autoencoder}} + \underbrace{\mathcal{L}_{GAN}(G_1, D_1, G_2, D_2)}_{\textbf{Adversarial}}$$



Variation Autoencoder Loss

$$\mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) = \mathbb{E}[\|G_1(E_1(x_1)) - x_1\|_2] + \mathbb{E}[\mathcal{KL}(q_1(z)\|p(z))]$$
$$\mathbb{E}[\|G_2(E_2(x_2)) - x_2\|_2] + \mathbb{E}[\mathcal{KL}(q_2(z)\|p(z))]$$

Adversarial Loss

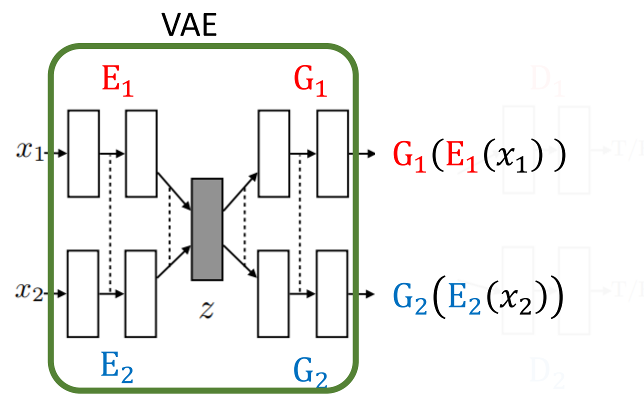
$$\mathcal{L}_{GAN}(G_1, D_1, G_2, D_2) = \mathbb{E}[\log(1 - D_1(G_1(z))] + \mathbb{E}[\log D_1(y_1)]$$
$$\mathbb{E}[\log(1 - D_2(G_2(z))] + \mathbb{E}[\log D_2(y_2)]$$

UNIT

- ***Learning***

Overall objective function

$$G = \arg\min_{G}\max_{D} \mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) + \mathcal{L}_{GAN}(G_1, D_1, G_2, D_2)$$



Variation Autoencoder Loss

$$\mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) = \mathbb{E}\left[\left\|G_1\left(E_1(x_1)\right) - x_1\right\|_2\right] + \mathbb{E}[\mathcal{KL}(q_1(z)||p(z))]$$
$$\mathbb{E}\left[\left\|G_2\left(E_2(x_2)\right) - x_2\right\|_2\right] + \mathbb{E}[\mathcal{KL}(q_2(z)||p(z))]$$

Reconstruction

Adversarial Loss

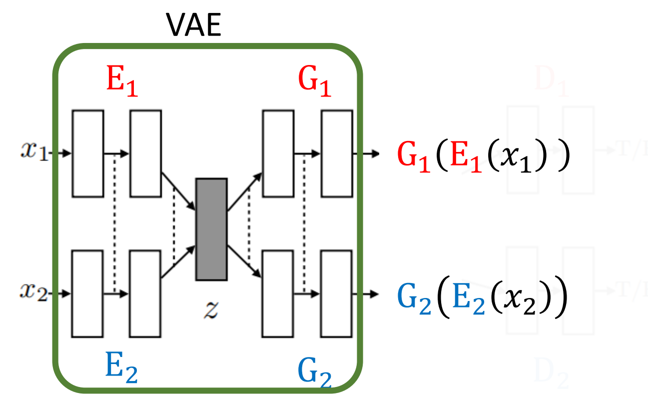
$$\mathcal{L}_{GAN}(G_1, D_1, G_2, D_2) = \mathbb{E}[\log(1 - D_1(G_1(z))] + \mathbb{E}[\log D_1(y_1)]$$
$$\mathbb{E}[\log(1 - D_2(G_2(z))] + \mathbb{E}[\log D_2(y_2)]$$

UNIT

- ***Learning***

Overall objective function

$$G = \arg\min_{G}\max_{D} \mathcal{L}_{VAE}(\mathrm{E}_1, \mathrm{G}_1, \mathrm{E}_2, \mathrm{G}_2) + \mathcal{L}_{GAN}(\mathrm{G}_1, \mathrm{D}_1, \mathrm{G}_2, \mathrm{D}_2)$$



Variation Autoencoder Loss

$$\mathcal{L}_{VAE}(\mathrm{E}_1, \mathrm{G}_1, \mathrm{E}_2, \mathrm{G}_2) = \mathbb{E}\left[\left\|\mathrm{G}_1\left(\mathrm{E}_1(x_1)\right) - x_1\right\|_2\right] + \mathbb{E}[\mathcal{KL}(q_1(z)||p(z))]$$
$$\mathbb{E}\left[\left\|\mathrm{G}_2\left(\mathrm{E}_2(x_2)\right) - x_2\right\|_2\right] + \mathbb{E}[\mathcal{KL}(q_2(z)||p(z))]$$

Prior Loss

Adversarial Loss

$$\mathcal{L}_{GAN}(\mathrm{G}_1, \mathrm{D}_1, \mathrm{G}_2, \mathrm{D}_2) = \mathbb{E}[\log(1 - \mathrm{D}_1(\mathrm{G}_1(z))] + \mathbb{E}[\log \mathrm{D}_1(y_1)]$$
$$\mathbb{E}[\log(1 - \mathrm{D}_2(\mathrm{G}_2(z))] + \mathbb{E}[\log \mathrm{D}_2(y_2)]$$

UNIT

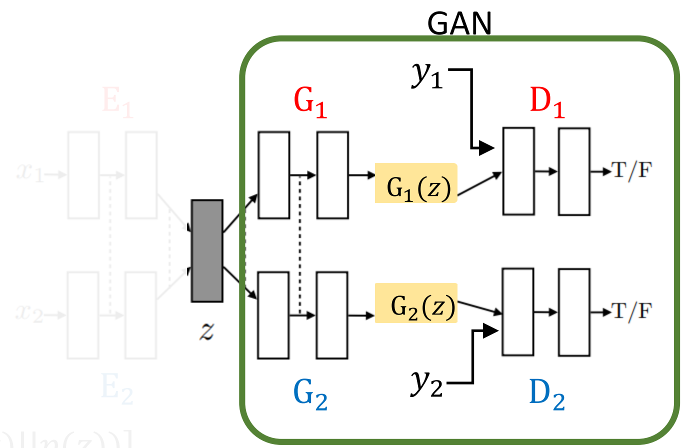
- ***Learning***

Overall objective function

$$G = \arg\min_{G}\max_{D} \mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) + \mathcal{L}_{GAN}(G_1, D_1, G_2, D_2)$$

Variation Autoencoder Loss

$$\mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) = \mathbb{E}[||G_1(E_1(x_1)) - x_1||_2] + \mathbb{E}[\mathcal{KL}(q_1(z)||p(z))]$$
$$\mathbb{E}[||G_2(E_2(x_2)) - x_2||_2] + \mathbb{E}[\mathcal{KL}(q_2(z)||p(z))]$$



Adversarial Loss

$$\mathcal{L}_{GAN}(G_1, D_1, G_2, D_2) = \mathbb{E}[\log(1 - D_1(G_1(z))] + \mathbb{E}[\log D_1(y_1)]$$
$$\mathbb{E}[\log(1 - D_2(G_2(z))] + \mathbb{E}[\log D_2(y_2)]$$

Generated

UNIT

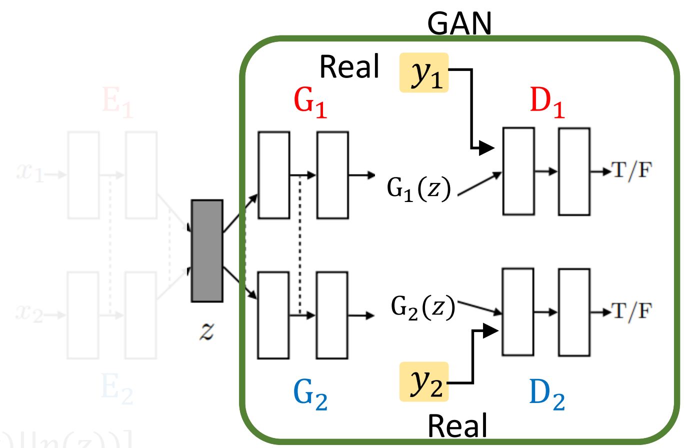
- ***Learning***

Overall objective function

$$G = \arg\min_{G}\max_{D} \mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) + \mathcal{L}_{GAN}(G_1, D_1, G_2, D_2$$

Variation Autoencoder Loss

$$\mathcal{L}_{VAE}(E_1, G_1, E_2, G_2) = \mathbb{E}[||G_1(E_1(x_1)) - x_1||_2] + \mathbb{E}[\mathcal{KL}(q_1(z)||p(z))]$$
$$\mathbb{E}[||G_2(E_2(x_2)) - x_2||_2] + \mathbb{E}[\mathcal{KL}(q_2(z)||p(z))]$$



Adversarial Loss

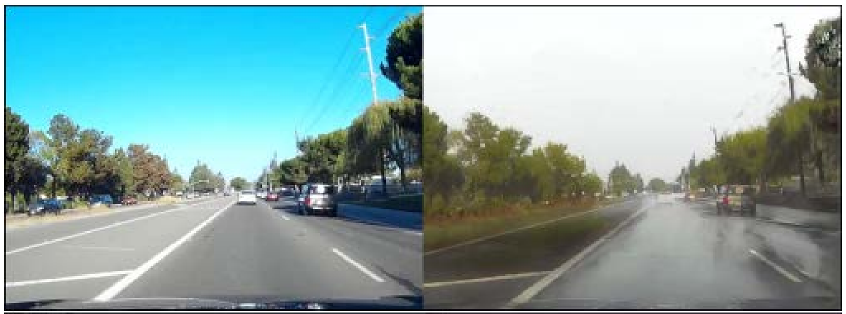
$$\mathcal{L}_{GAN}(G_1, D_1, G_2, D_2) = \mathbb{E}[\log(1 - D_1(G_1(z))] + \mathbb{E}[\log D_1(y_1)]$$
$$\mathbb{E}[\log(1 - D_2(G_2(z))] + \mathbb{E}[\log D_2(y_2)]$$

Real

UNIT

- Example results

Sunny → Rainy



Real Street-view → Synthetic Street-view

Rainy → Sunny

Synthetic Street-view → Real Street-view

Github Page: https://github.com/mingyuliutw/UNIT

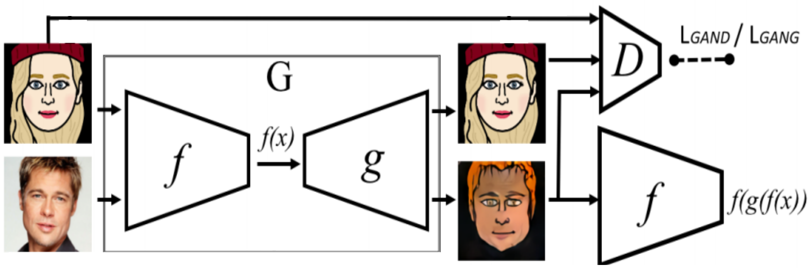
Liu et al., "Unsupervised image-to-image translation networks.“, NIPS 2017

What to Cover?

- Cross-Domain Image Translation
  - Pix2pix (CVPR'17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS'17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR'17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR'18) : Image translation via representation disentanglement
  - CDRD (CVPR'18) : Cross-domain representation disentanglement and translation
- Final Remarks

Domain Transfer Networks

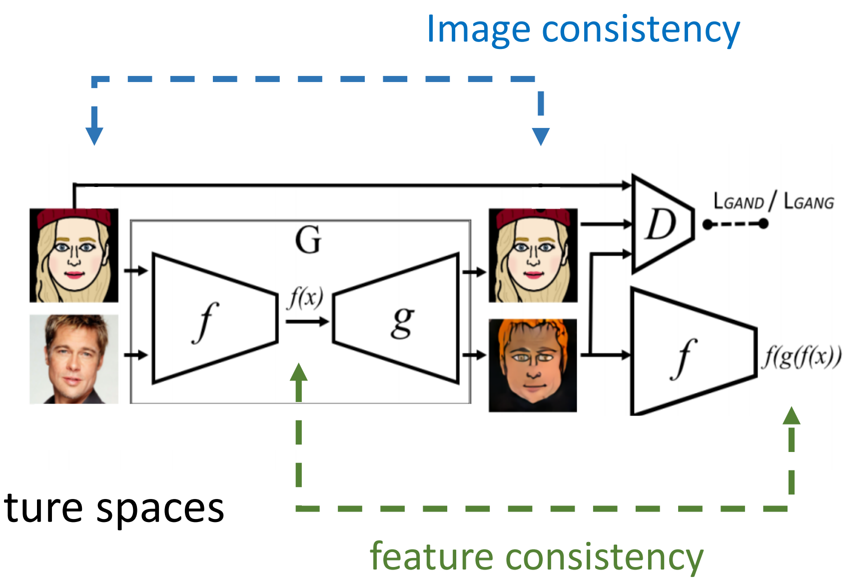
- Unsupervised Cross-Domain Image Generation (ICLR'17)
- Goal/Problem Setting
  - Image translation across two domains
  - One-way only translation
  - Unpaired training data
- Idea
  - Apply unified model to learn joint representation across domains.



Taigman et al., "Unsupervised cross-domain image generation.", ICLR 2016

Domain Transfer Networks

- Unsupervised Cross-Domain Image Generation (ICLR'17)
- Goal/Problem Setting
  - Image translation across two domains
  - One-way only translation
  - Unpaired training data
- Idea
  - Apply unified model to learn joint representation across domains.
  - Consistency observed in image and feature spaces

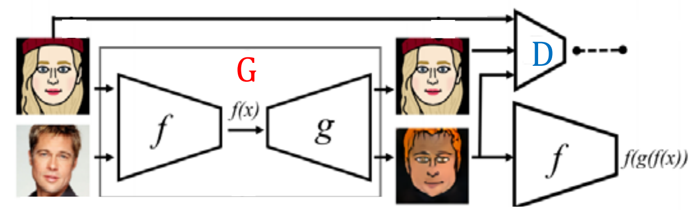


Taigman et al., "Unsupervised cross-domain image generation.", ICLR 2016

Domain Transfer Networks

- ***Learning***
  - **Unified model** to translate across domains

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{img}(G) + \mathcal{L}_{feat}(G) + \mathcal{L}_{GAN}(G, D)$$



  - Consistency of feature and image space

$$\mathcal{L}_{img}(G) = \mathbb{E}\left[\|g(f(y)) - y\|_2\right]$$

$$\mathcal{L}_{feat}(G) = \mathbb{E}\left[\|f(g(f(x))) - f(x)\|_2\right]$$

  - Adversarial loss

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x))] + \mathbb{E}[\log(1 - D(G(y))] + \mathbb{E}[\log D(y)]$$

Domain Transfer Networks

- ***Learning***
  - **Unified model** to translate across domains

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{img}(G) + \mathcal{L}_{feat}(G) + \mathcal{L}_{GAN}(G, D)$$

  - **Consistency of image and feature space**

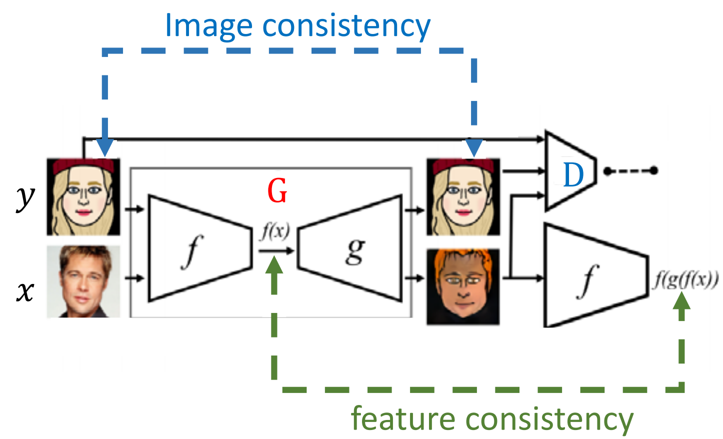
$$\mathcal{L}_{img}(G) = \mathbb{E}\left[\left\|g\big(f(y)\big) - y\right\|_2\right]$$

$$\mathcal{L}_{feat}(G) = \mathbb{E}\left[\left\|f\big(g\big(f(x)\big)\big) - f(x)\right\|_2\right]$$

$$G = \{f, g\}$$

  - Adversarial loss

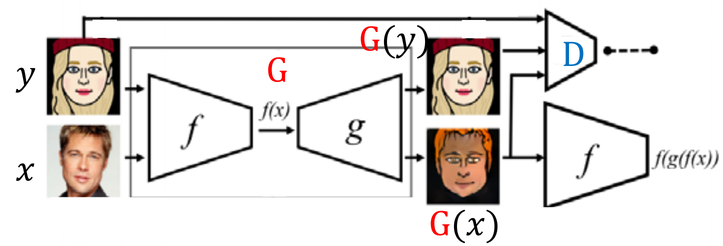
$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x))] + \mathbb{E}[\log(1 - D(G(y))] + \mathbb{E}[\log D(y)]$$



Domain Transfer Networks

- ***Learning***
  - **Unified model** to translate across domains

    $$G^{*} = \arg\min_{G}\max_{D} \mathcal{L}_{img}(G) + \mathcal{L}_{feat}(G) + \mathcal{L}_{GAN}(G, D)$$



  - Consistency of feature and image space

    $$\mathcal{L}_{img}(G) = \mathbb{E}\left[\left\|g(f(y)) - y\right\|_2\right]$$

    $$\mathcal{L}_{feat}(G) = \mathbb{E}\left[\left\|f(g(f(x))) - f(x)\right\|_2\right]$$

  - **Adversarial loss**

    $$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x))] + \mathbb{E}[\log(1 - D(G(y))] + \mathbb{E}[\log D(y)]$$

Domain Transfer Networks

- ***Learning***
  - **Unified model** to translate across domains

    $$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{img}(G) + \mathcal{L}_{feat}(G) + \mathcal{L}_{GAN}(G, D)$$

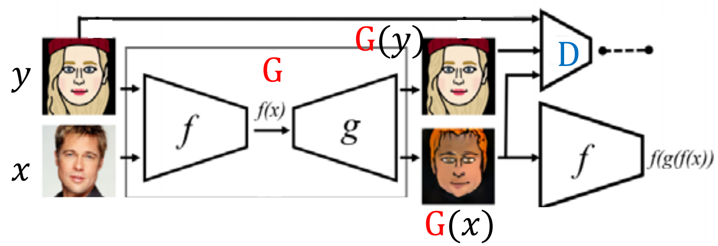
  - Consistency of feature and image space

    $$\mathcal{L}_{img}(G) = \mathbb{E}\left[\left\|g\big(f(y)\big) - y\right\|_2\right]$$

    $$\mathcal{L}_{feat}(G) = \mathbb{E}\left[\left\|f\big(g\big(f(x)\big)\big) - f(x)\right\|_2\right]$$

  - Adversarial loss

    $$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x))] + \mathbb{E}[\log(1 - D(G(y))] + \mathbb{E}[\log D(y)]$$

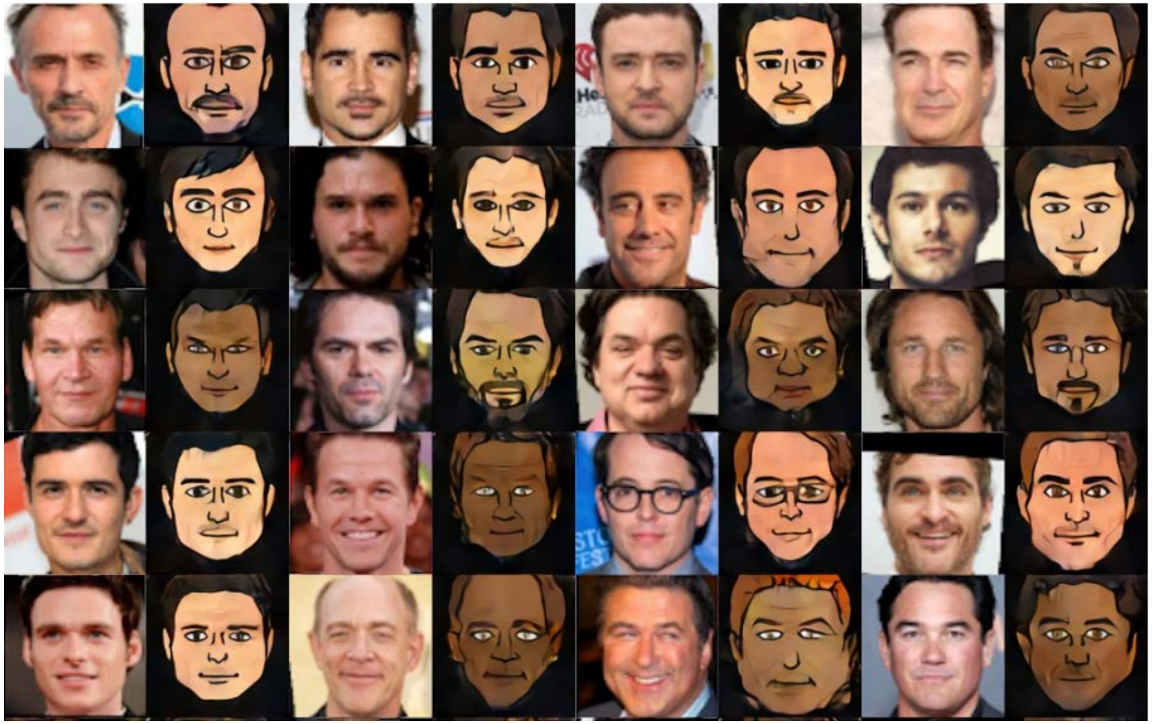


DTN

- Example results

SVHN 2 MNIST

Photo 2 Emoji



Taigman et al., "Unsupervised cross-domain image generation.“, ICLR 2016

Beyond Transfer Learning

- Cross-Domain Image Translation
  - Pix2pix (CVPR’17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS’17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR’17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR’18) : Image translation via representation disentanglement
  - UFDN (NIPS’18): A Unified Feature Disentangler for Multi-Domain Image Translation and Maniuplation

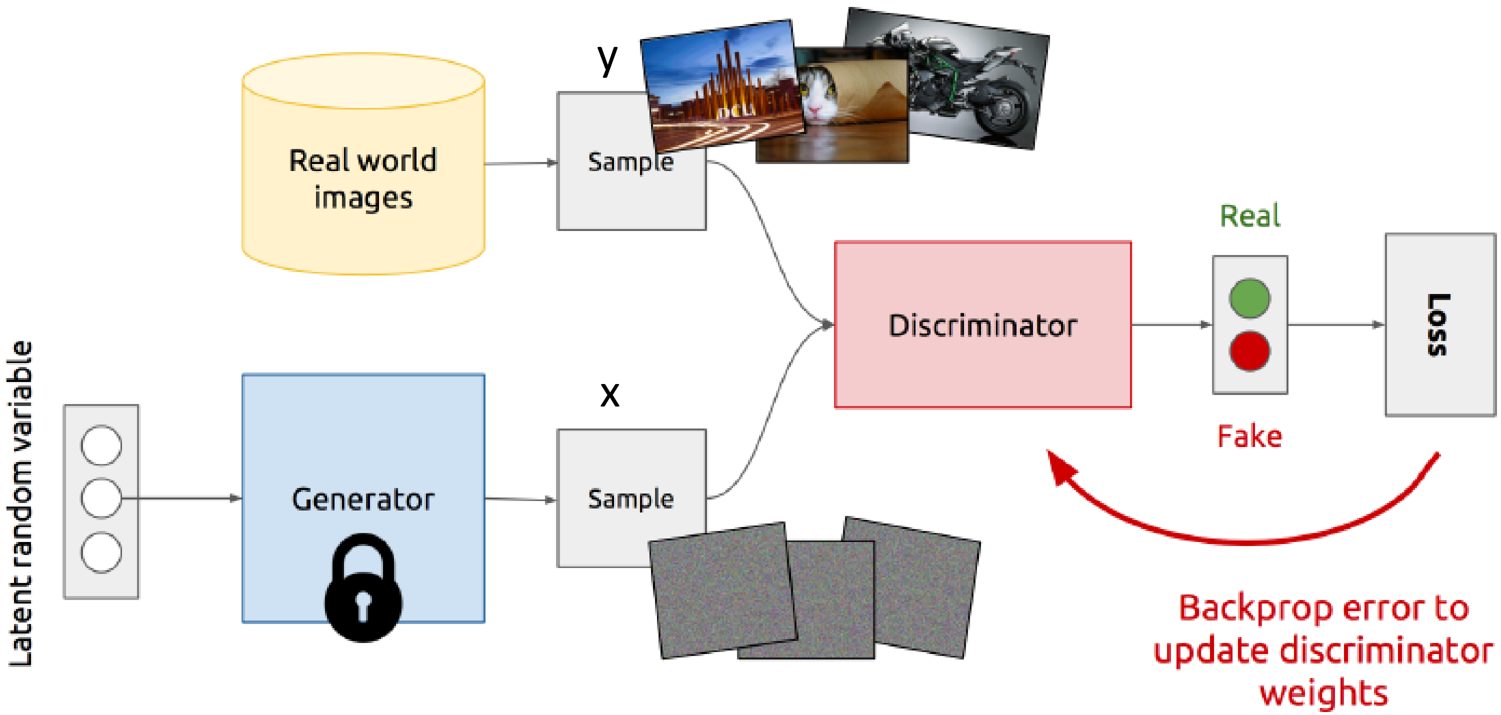
Beyond Image Style Transfer:
Learning Interpretable Deep Representations

- Faceapp – Putting a smile on your face!
  - Deep learning for representation disentanglement
  - Interpretable deep feature representation

Input
Mr. Takeshi Kaneshiro

Recall: Generative Adversarial Networks (GAN)

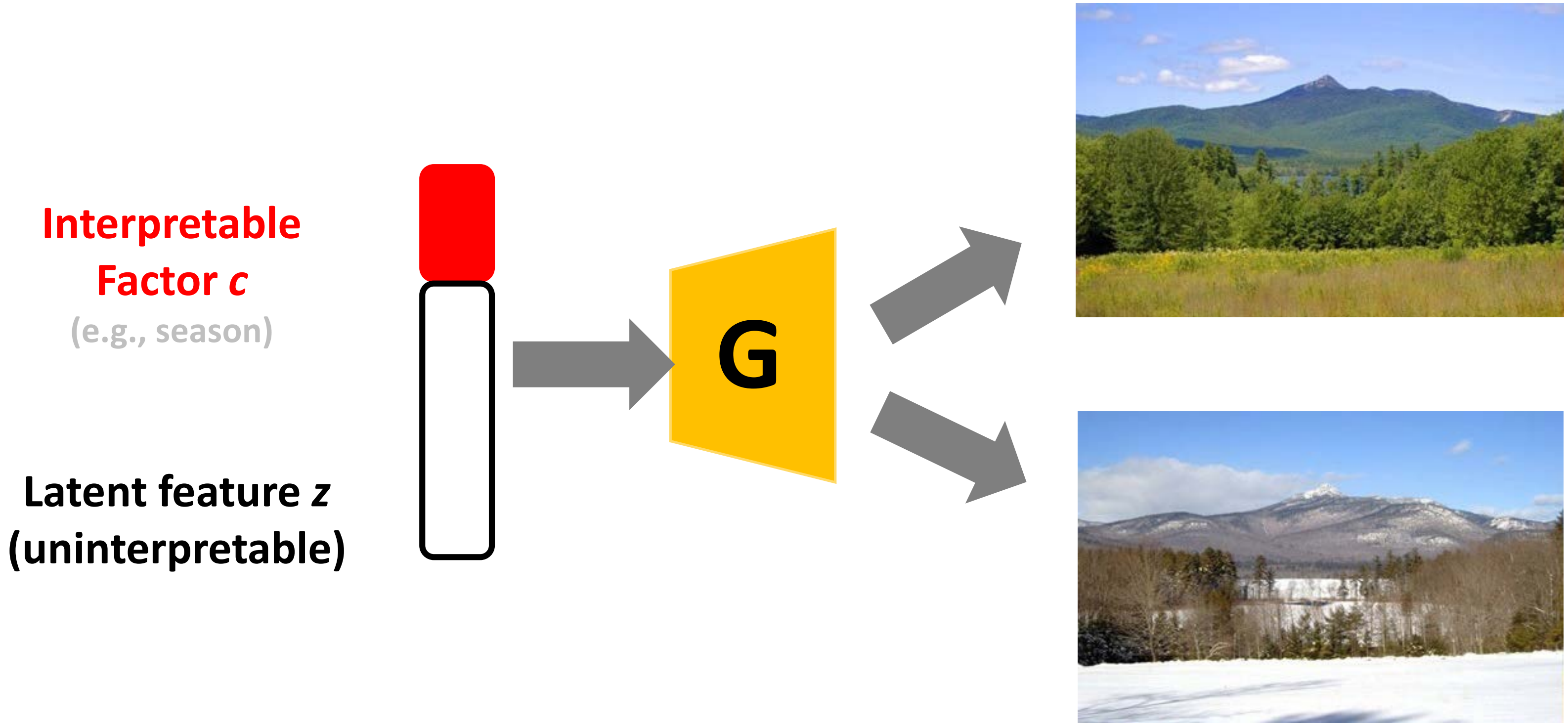
- Architecture of GAN
  - Loss $\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D\ (G(x)))] + \mathbb{E}[\log D(y)]$



Goodfellow et al., Generative Adversarial Nets, NIPS, 2014

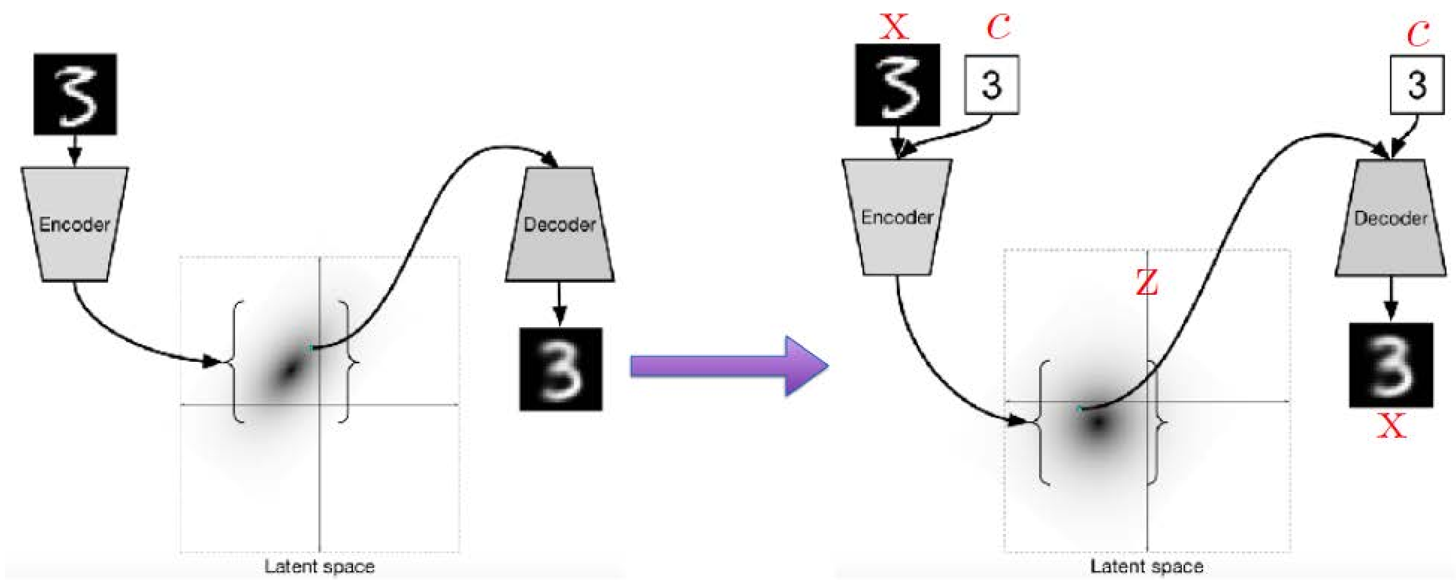
Representation Disentanglement

- Goal
  - Interpretable deep feature representation
  - Disentangle attribute of interest ***c*** from the derived latent representation ***z***
  - Possible solutions: VAE, GAN, or mix of them…



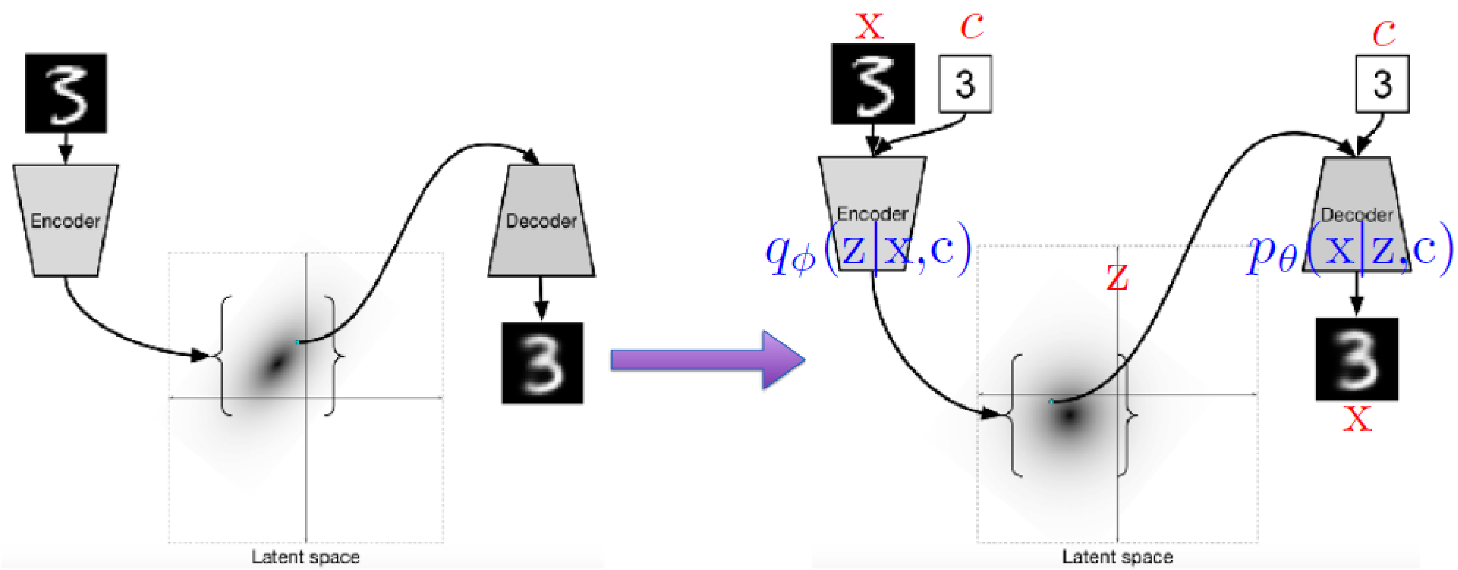
Representation Disentanglement

- Goal
  - Interpretable deep feature representation
  - Disentangle attribute of interest ***c*** from the derived latent representation ***z***
    - Supervised setting: from VAE to conditional VAE



Representation Disentanglement

- Conditional VAE
  - Given training data ***x*** and attribute of interest c, we model the conditional distribution $p_\theta(x|c)$.



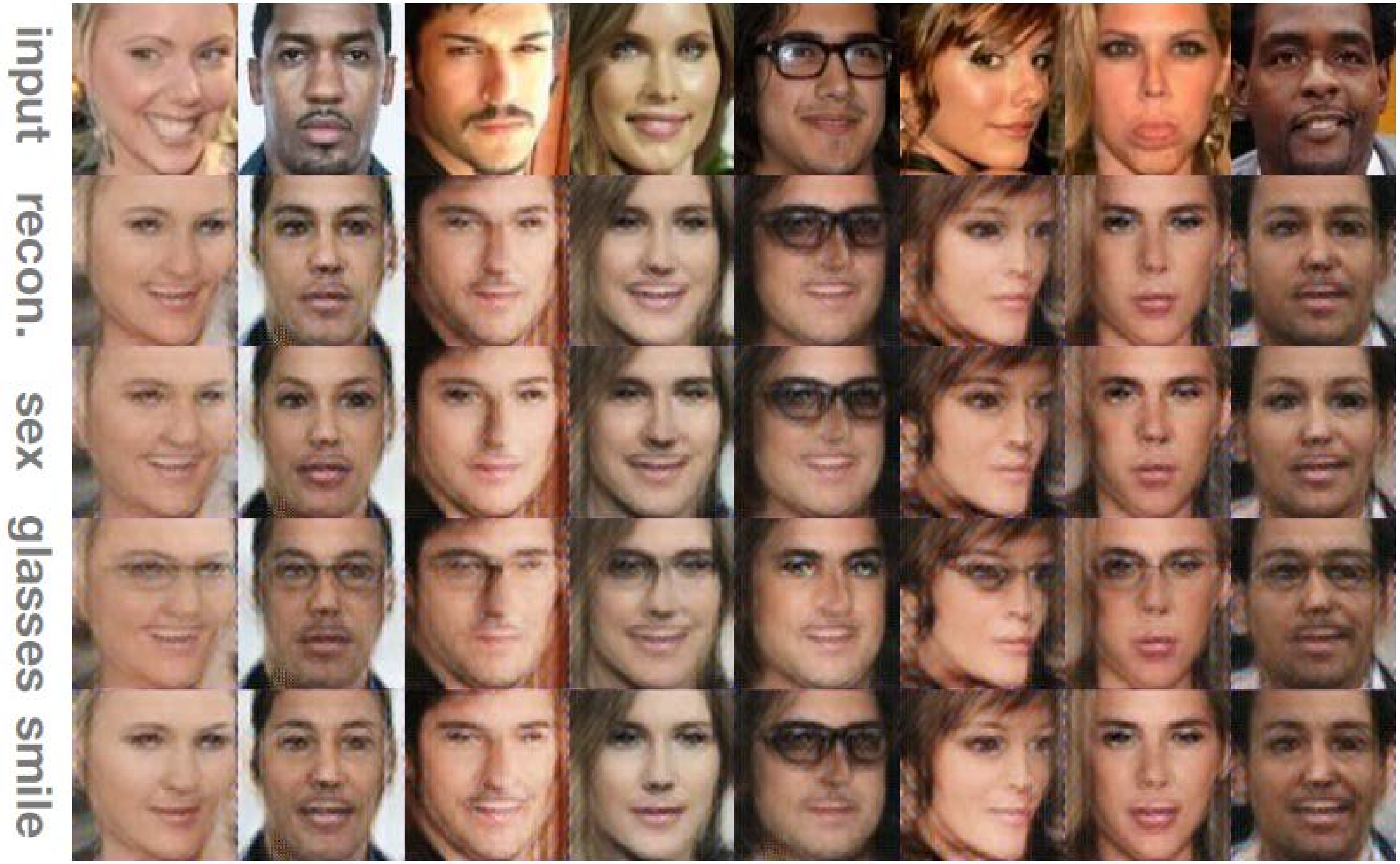
impose prior **reconstruction**

$$= \underline{-KL(q_\phi(z|x,c)||p_\theta(z|\cancel{c}))} + \underline{\mathbb{E}_{q_\phi(z|x,c)} \log p_\theta(x|c,z)}$$

https://zhuanlan.zhihu.com/p/25518643

Representation Disentanglement

- Conditional VAE
  - Example Results



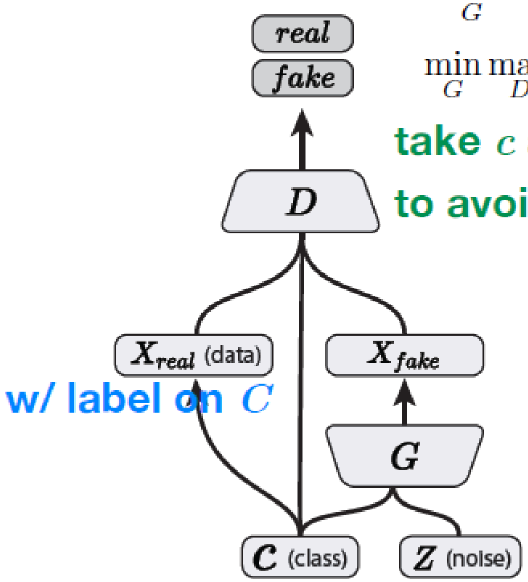
Representation Disentanglement

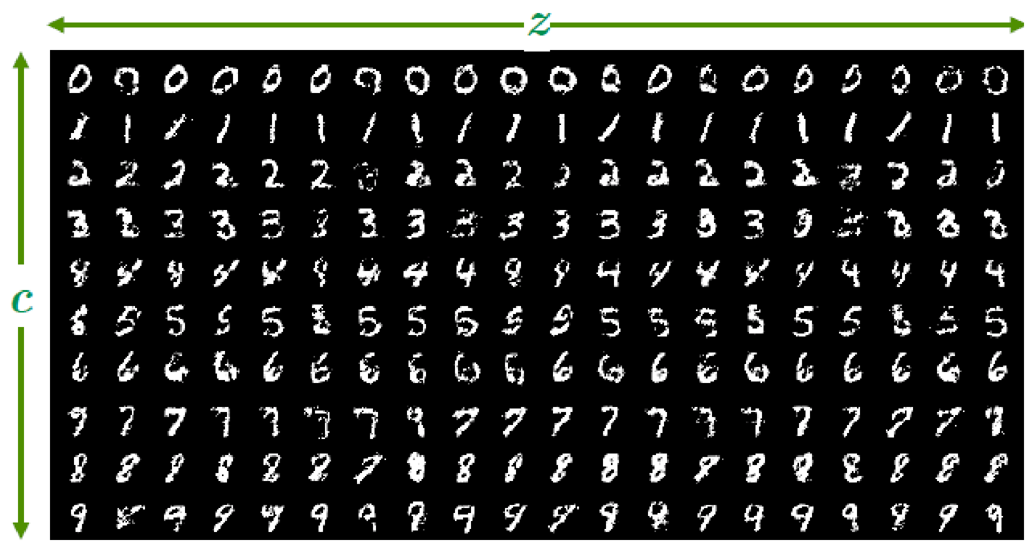
- Conditional GAN
  - Interpretable latent factor ***c***
  - Latent representation ***z***

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

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

take c as also input for D
to avoid ignoring c in G

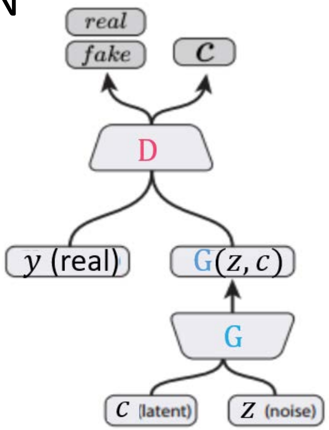




https://arxiv.org/abs/1411.1784

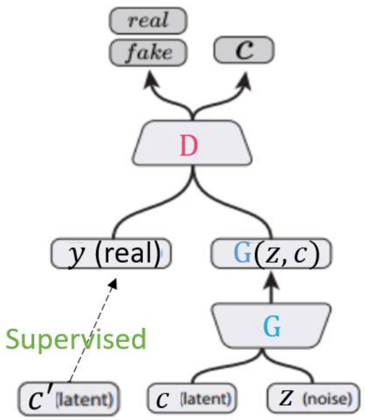
Representation Disentanglement

- Goal
  - Interpretable deep feature representation
  - Disentangle attribute of interest ***c*** from the derived latent representation ***z***
    - Unsupervised: InfoGAN
    - Supervised: AC-GAN



InfoGAN

Chen et al.
NIPS ’16



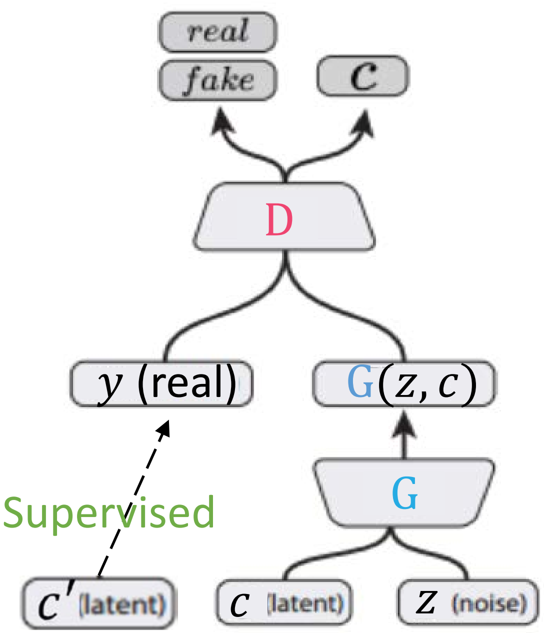
ACGAN

Odena et al.
ICML ’17

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.
Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML’17

AC-GAN

- Supervised Disentanglement



- **Learning**
  - **Overall objective function**

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D)$$

  - **Adversarial Loss**

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(z, c)))] + \mathbb{E}[\log D(y)]$$

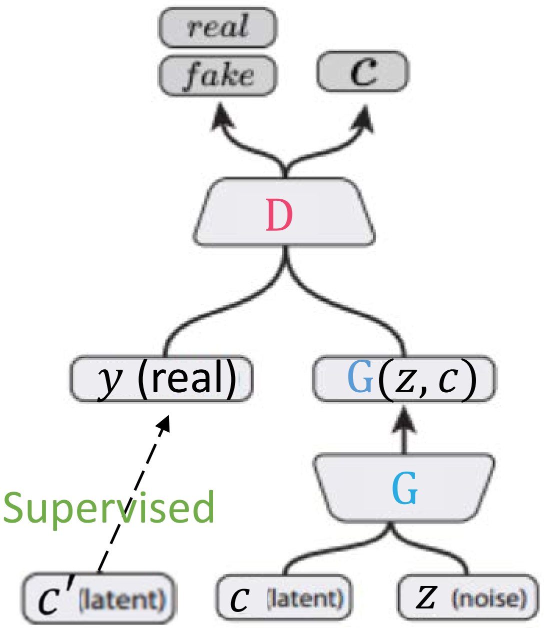
  - **Disentanglement loss**

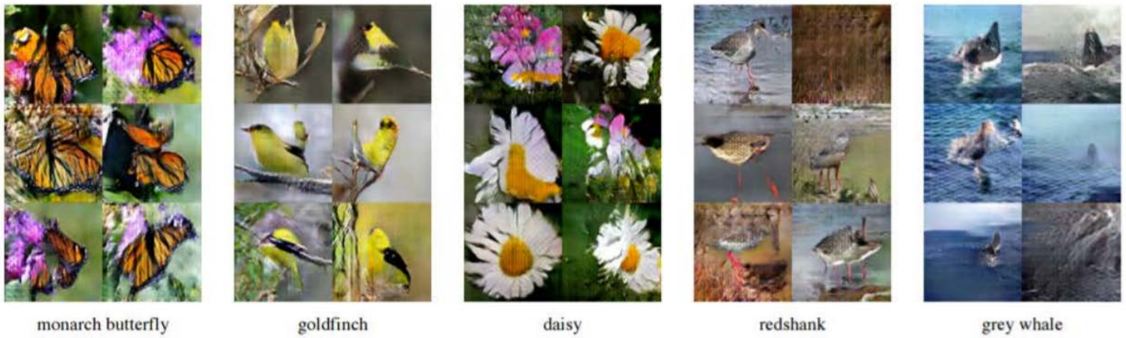
$$\mathcal{L}_{cls}(G, D) = \underbrace{\mathbb{E}[-\log D_{\text{cls}}(c'|y)]}_{\substack{\text{Real data} \\ \text{w.r.t. its domain label}}} + \underbrace{\mathbb{E}[-\log D_{\text{cls}}(c|G(x, c))]}_{\substack{\text{Generated data} \\ \text{w.r.t. assigned label}}}$$

Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML'17

AC-GAN

- Supervised Disentanglement



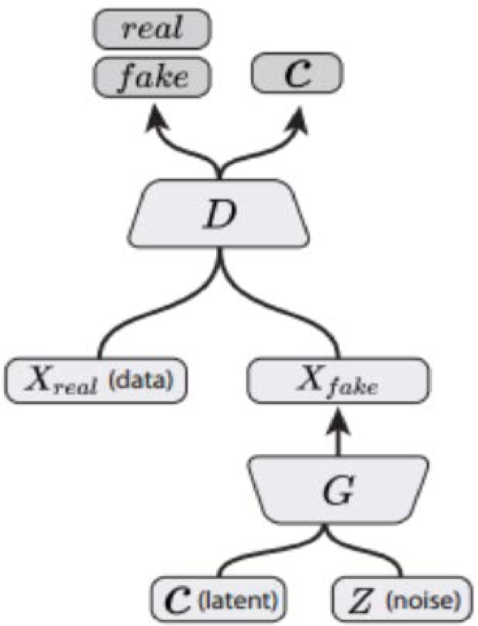


Different c values

Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML'17

InfoGAN

- Unsupervised Disentanglement



- Learning
  - **Overall objective function**

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D)$$

  - **Adversarial Loss**

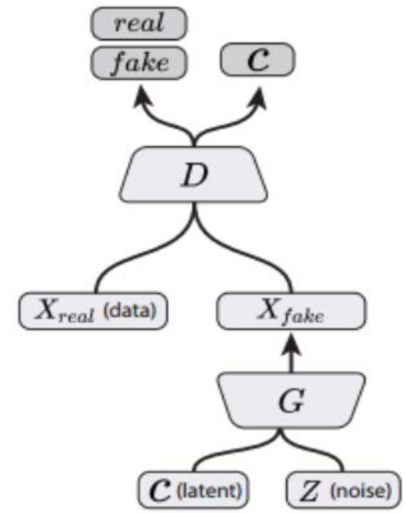
$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(z, c)))] + \mathbb{E}[\log D(y)]$$

  - **Disentanglement loss**

$$\mathcal{L}_{cls}(G, D) = \underbrace{\mathbb{E}[-\log D_{cls}(c|G(x, c))]}_{\text{Generated data w.r.t. assigned label}}$$

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.

InfoGAN

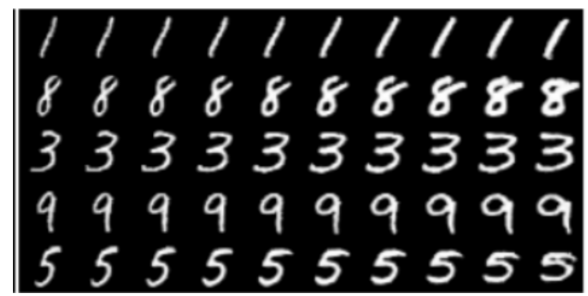


- Unsupervised Disentanglement
  - No guarantee in disentangling particular semantics

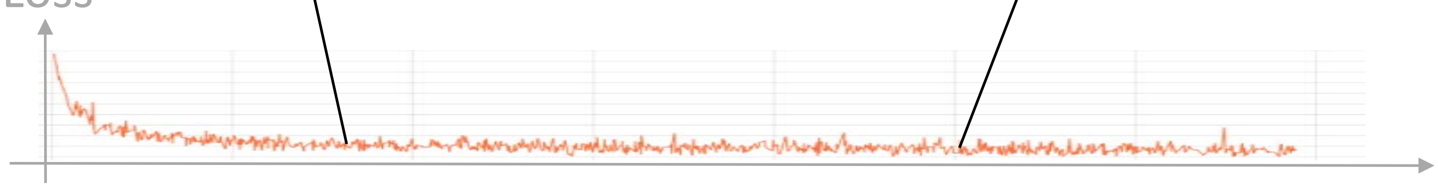
Different c

Rotation Angle

Different c



Width



Training process

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.

Beyond Transfer Learning

- Cross-Domain Image Translation
  - Pix2pix (CVPR’17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS’17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR’17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR’18) : Joint image translation and representation disentanglement

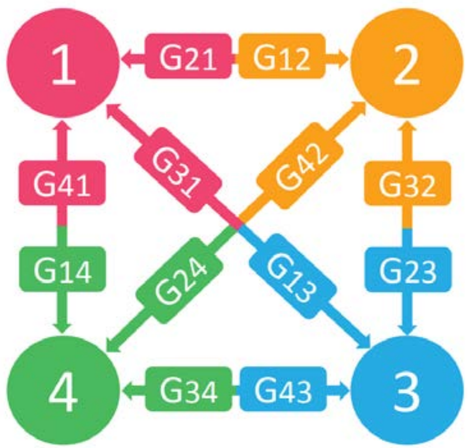
Beyond Transfer Learning

- Cross-Domain Image Translation
  - Pix2pix (CVPR'17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS'17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR'17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR'18) : Image translation via representation disentanglement
  - UFDN (NIPS'18): A Unified Feature Disentangler for Multi-Domain Image Translation and Maniuplation

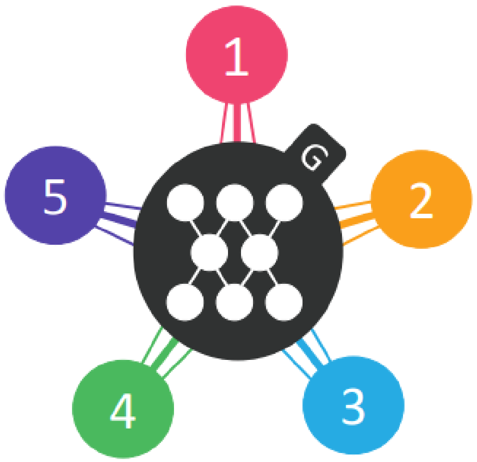
StarGAN

- Goal
  - Unified GAN for multi-domain image-to-image translation

Traditional Cross-Domain Models



Unified Multi-Domain Model (StarGAN)

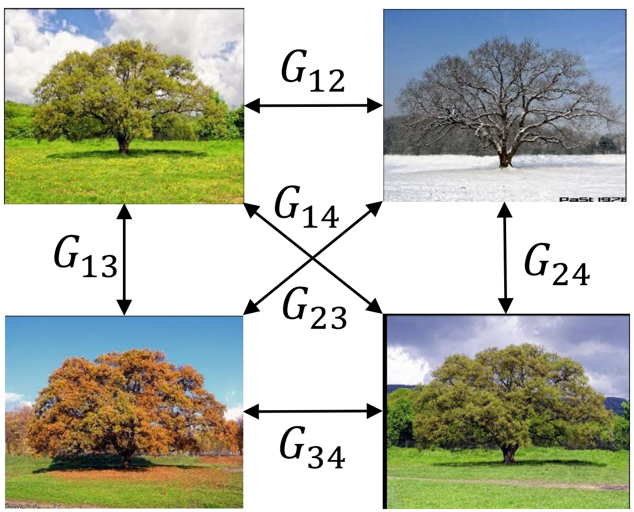


Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." CVPR 2018

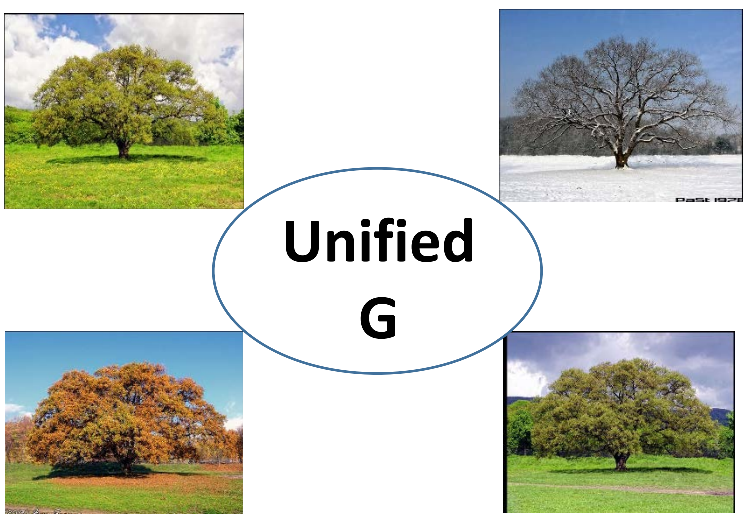
StarGAN

- Goal
  - Unified GAN for multi-domain image-to-image translation

Traditional Cross-Domain Models



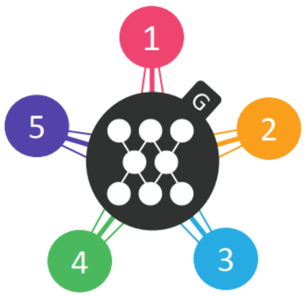
Unified Multi-Domain Model (StarGAN)



Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." CVPR 2018

StarGAN

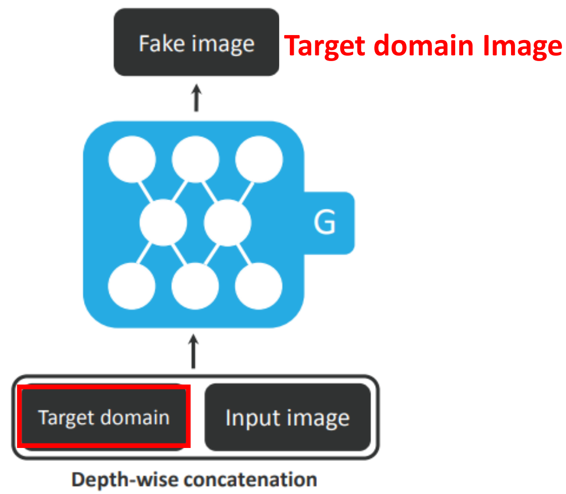
- ***Goal / Problem Setting***
  - **Single** image translation model across **multiple** domains
  - Unpaired training data



StarGAN

- ***Goal / Problem Setting***
  - **Single** Image translation model across **multiple** domains
  - Unpaired training data

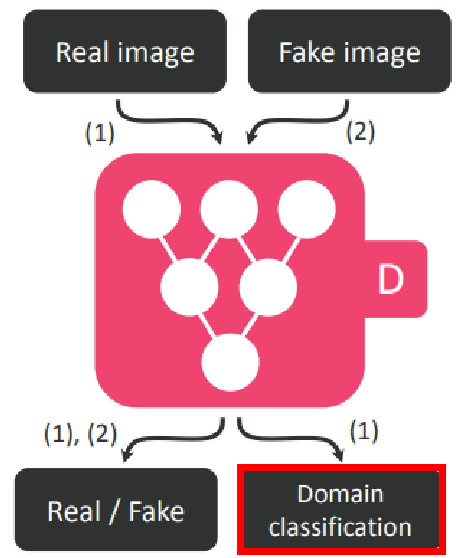
- ***Idea***
  - Concatenate image and target domain label as input of generator
  - Auxiliary domain classifier on Discriminator



StarGAN

- ***Goal / Problem Setting***
  - **Single** Image translation model across **multiple** domains
  - Unpaired training data

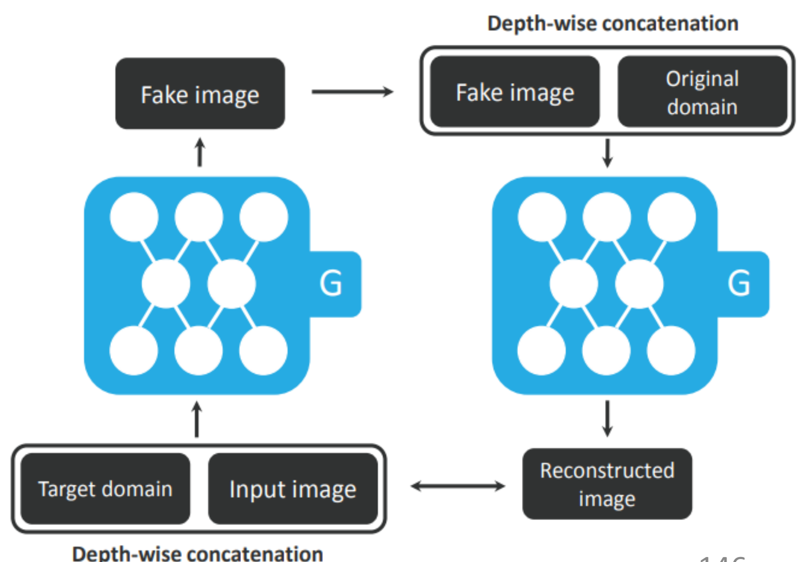
- ***Idea***
  - Concatenate image and target domain label as input of Generator
  - Auxiliary <u>domain classifier</u> as discriminator too



StarGAN

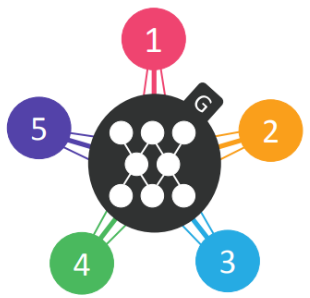
- ***Goal / Problem Setting***
  - **Single** Image translation model across **multiple** domains
  - Unpaired training data

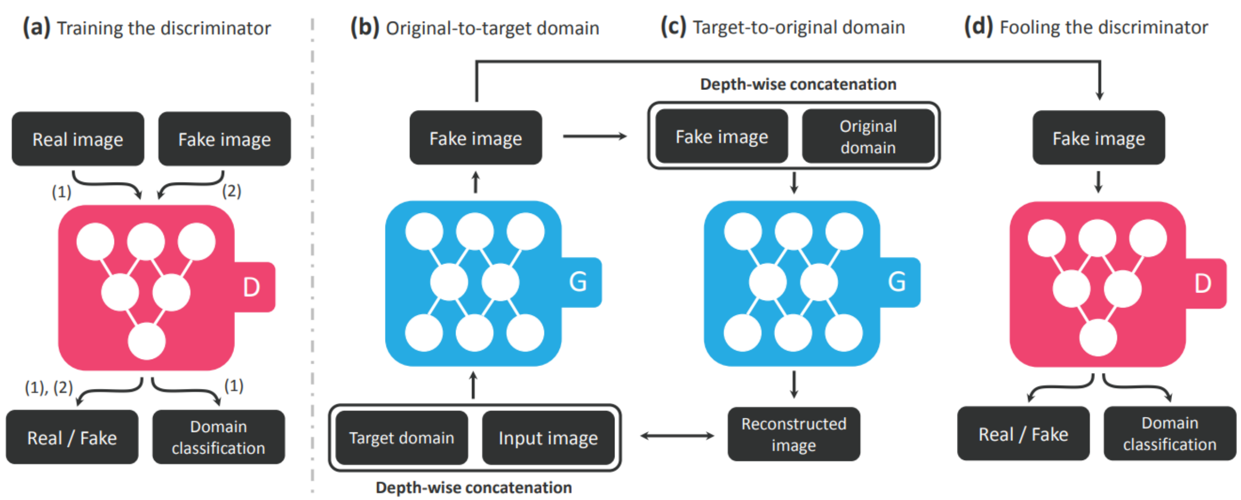
- ***Idea***
  - Concatenate image and target domain label as in
    Generator
  - Auxiliary domain classifier on Discriminator
  - Cycle consistency across domains



StarGAN

- ***Goal / Problem Setting***
  - **Single** Image translation model across **multiple** domains
  - Unpaired training data
- ***Idea***
  - Auxiliary domain classifier as discriminator
  - Concatenate image and target domain label as input
  - Cycle consistency across domains



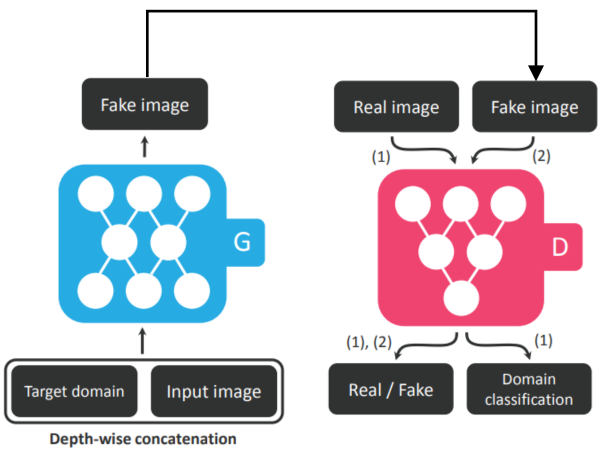


StarGAN

- ***Learning***

Overall objective function

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$



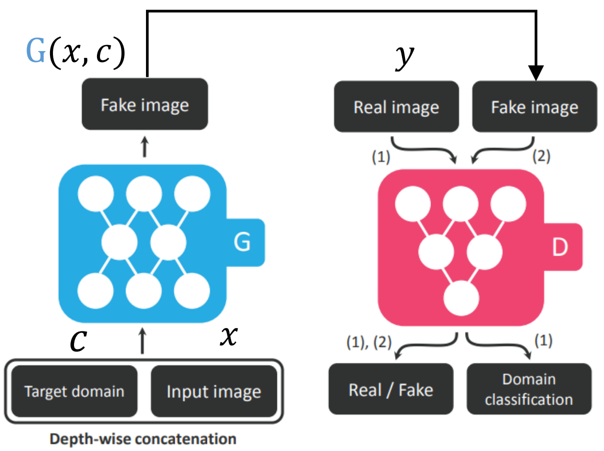
StarGAN

- ***Learning***

Overall objective function

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$

Adversarial Loss



- **Adversarial Loss**

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x, c)))] + \mathbb{E}[\log D(y)]$$

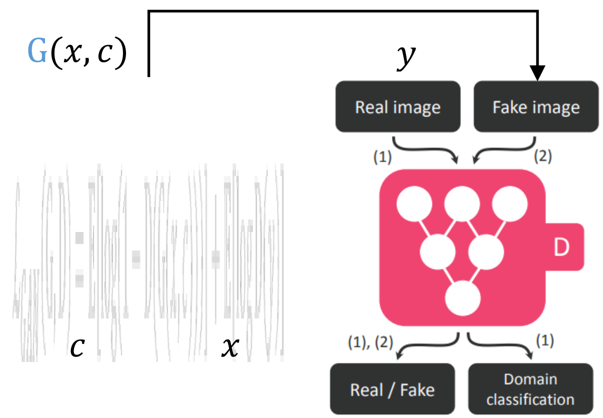
StarGAN

- ***Learning***

Overall objective function

$$\mathrm{G}^* = \arg\min_{\mathrm{G}} \max_{\mathrm{D}} \mathcal{L}_{GAN}(\mathrm{G},\mathrm{D}) + \mathcal{L}_{cls}(\mathrm{G},\mathrm{D}) + \mathcal{L}_{cyc}(\mathrm{G})$$

Domain Classification Loss



- **Adversarial Loss**

$$\mathcal{L}_{GAN}(\mathrm{G},\mathrm{D}) = \mathbb{E}[\log(1-\mathrm{D}(\mathrm{G}(x,c)))] + \mathbb{E}[\log \mathrm{D}(y)]$$

- **Domain Classification Loss (Disentanglement)**

$$\mathcal{L}_{cls}(\mathrm{G},\mathrm{D}) = \mathbb{E}[-\log \mathrm{D_{cls}}(c'|y)] + \mathbb{E}[-\log \mathrm{D_{cls}}(c|\mathrm{G}(x,c))]$$

StarGAN

- ***Learning***

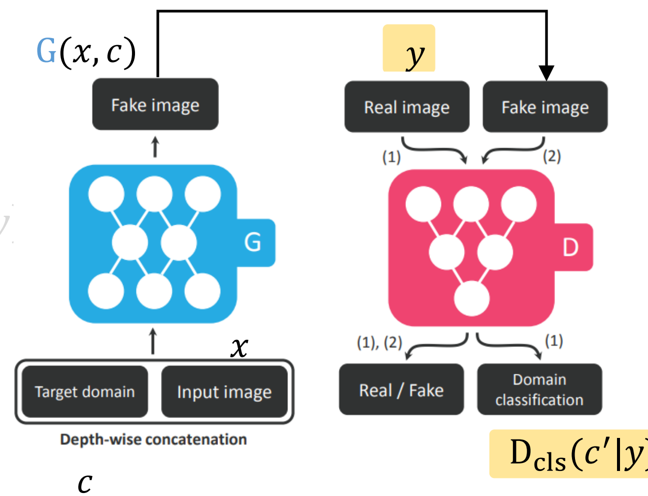
Overall objective function

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$

Domain Classification Loss

- Adversarial Loss

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x, c)))] + \mathbb{E}[\log D(y$$



- **Domain Classification Loss (Disentanglement)**

$$\mathcal{L}_{cls}(G, D) = \underbrace{\mathbb{E}[-\log D_{cls}(c'|y)]}_{\text{Real data w.r.t. its domain label}} + \mathbb{E}[-\log D_{cls}(c|G(x, c))]$$

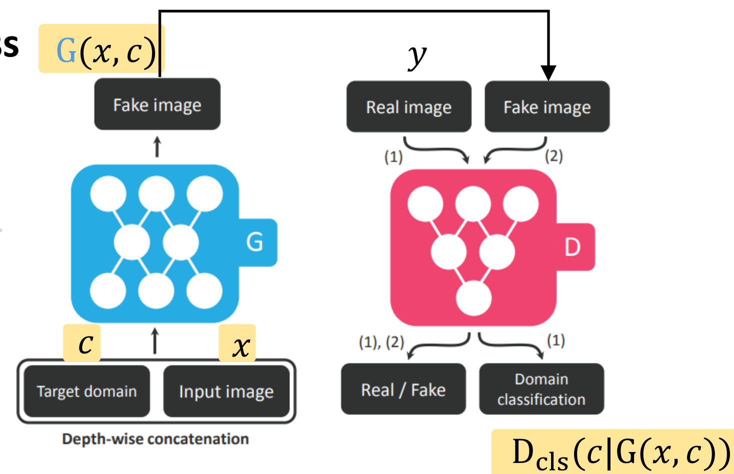
StarGAN

- ***Learning***

Overall objective function

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$

Domain Classification Loss



- Adversarial Loss

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x, c)))] +$$

- **Domain Classification Loss** **(Disentanglement)**

$$\mathcal{L}_{cls}(G, D) = \mathbb{E}[-\log D_{cls}(c'|y)] + \mathbb{E}[-\log \underbrace{D_{cls}(c|G(x, c))}_{\text{Generated data w.r.t. assigned label}}]$$

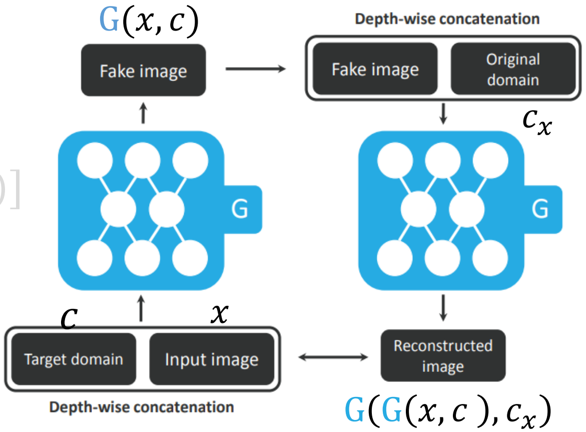
StarGAN

- ***Learning***

Overall objective function

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$

Consistency Loss



- **Adversarial Loss**

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x, c)))] + \mathbb{E}[\log D(y)]$$

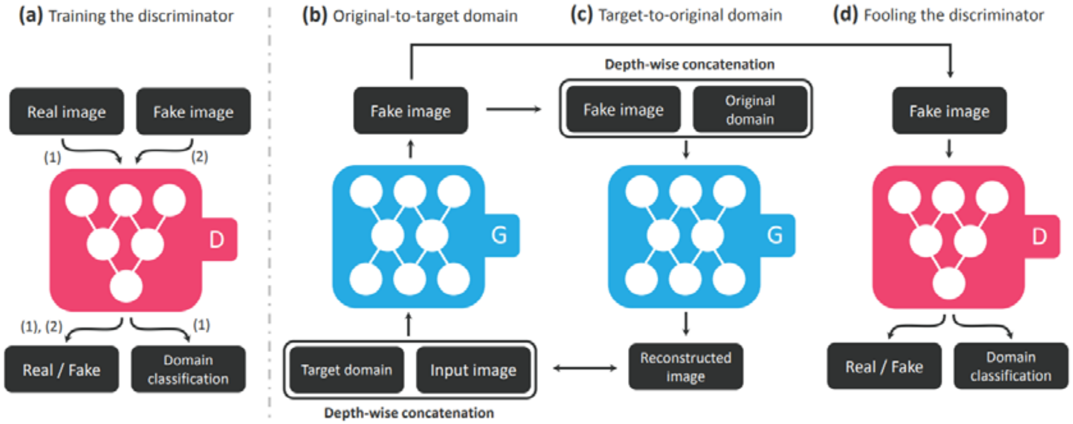
- **Domain Classification Loss (Disentanglement)**

$$\mathcal{L}_{cls}(G, D) = \mathbb{E}[-\log D_{cls}(c'|y)] + \mathbb{E}[-\log D_{cls}(c|G(x, c))]$$

- **Cycle Consistency Loss**

$$\mathcal{L}_{cyc}(G) = \mathbb{E}[\|G(G(x, c), c_x) - x\|_1]$$

StarGAN



- ***Learning***

Overall objective function

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D) + \mathcal{L}_{cyc}(G)$$

- **Adversarial Loss**

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(x, c)))] + \mathbb{E}[\log D(y)]$$

- **Domain Classification Loss**

$$\mathcal{L}_{cls}(G, D) = \mathbb{E}[-\log D_{cls}(c'|y)] + \mathbb{E}[-\log D_{cls}(c|G(x, c))]$$

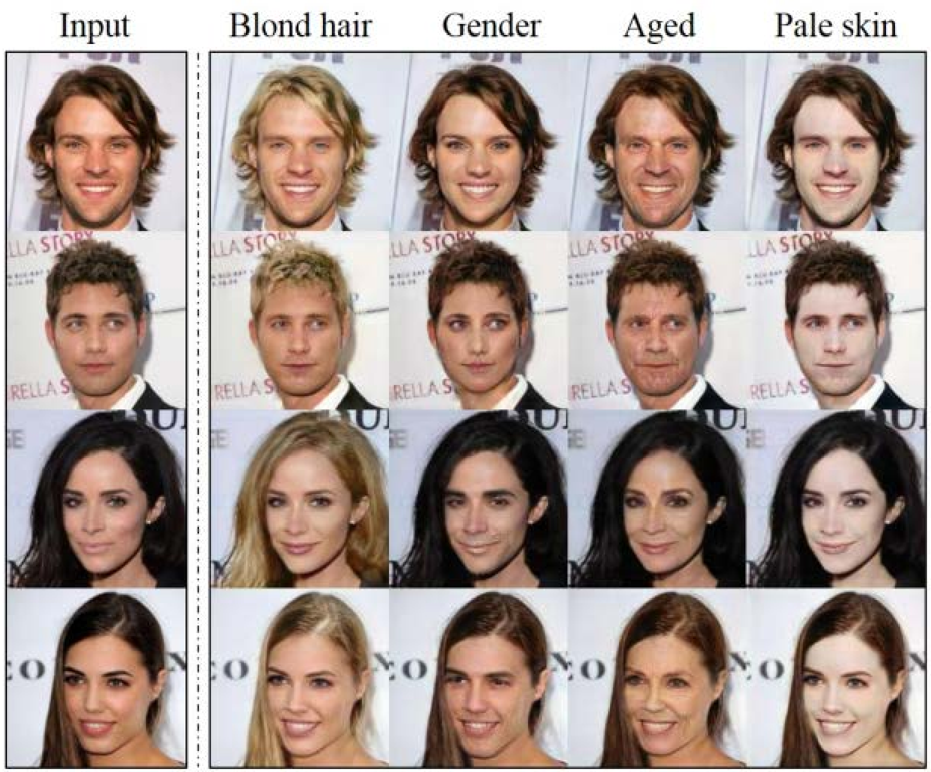
- **Cycle Consistency Loss**

$$\mathcal{L}_{cyc}(G) = \mathbb{E}[\|G(G(x, c\ ), c_x) - x\|_1]$$

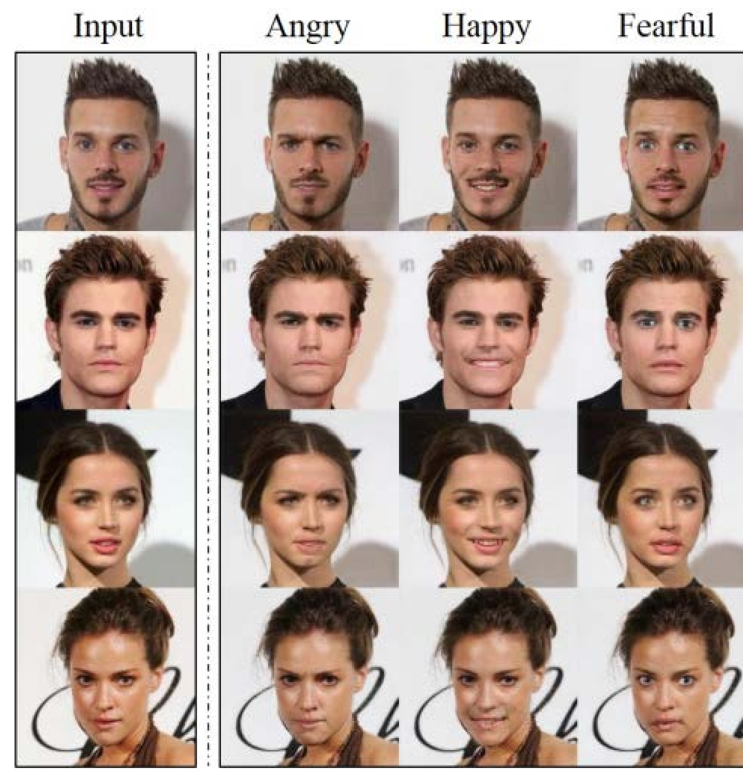
StarGAN

- Example results
  - StarGAN can somehow be viewed as a representation disentanglement model, instead of an image translation one.

Multiple Domains



Multiple Domains



Github Page: https://github.com/yunjey/StarGAN

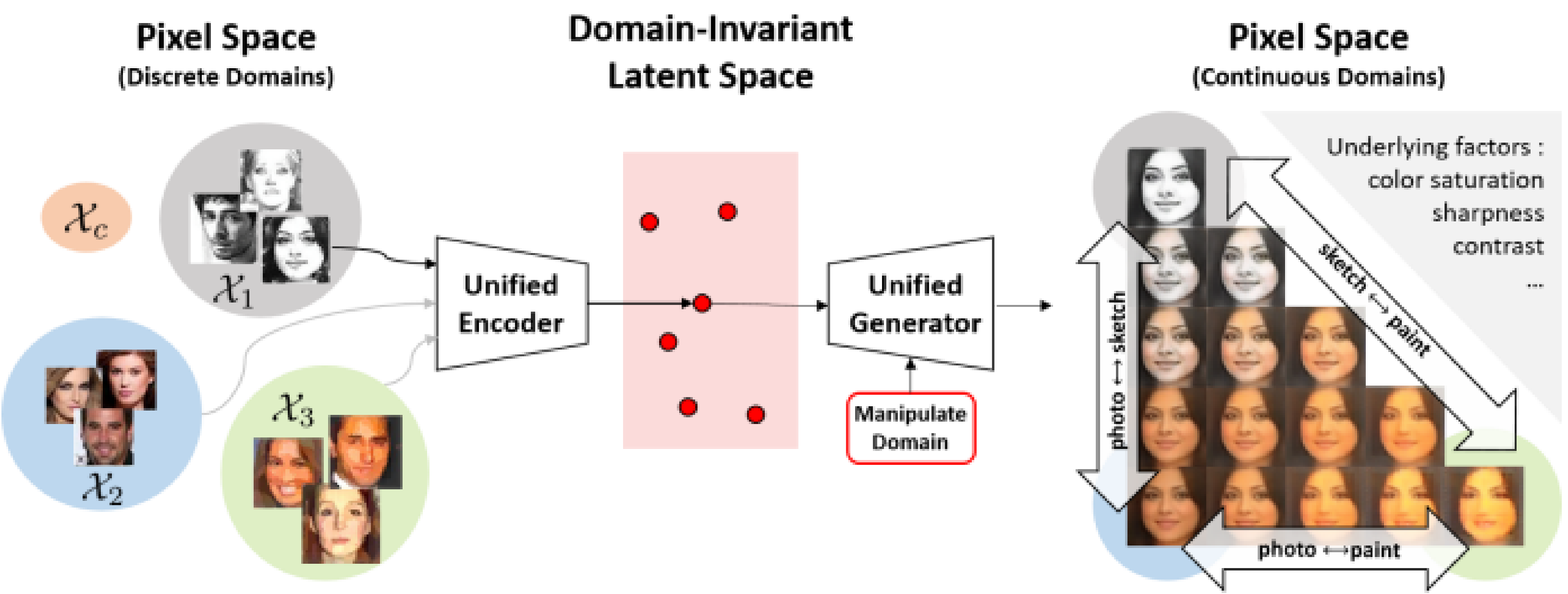
Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." CVPR 2018

Beyond Transfer Learning

- Cross-Domain Image Translation
  - Pix2pix (CVPR’17): Pairwise cross-domain training data
  - CycleGAN/DualGAN/DiscoGAN: Unpaired cross-domain training data
  - UNIT (NIPS’17): Learning cross-domain image representation (with unpaired training data)
  - DTN (ICLR’17) : Learning cross-domain image representation (with unpaired training data)
- Representation Disentanglement
  - InfoGAN & AC-GAN: Representation disentanglement in a single domain
  - StarGAN (CVPR’18) : Image translation via representation disentanglement
  - UFDN (NIPS’18): A Unified Feature Disentangler for Multi-Domain Image Translation and Maniuplation

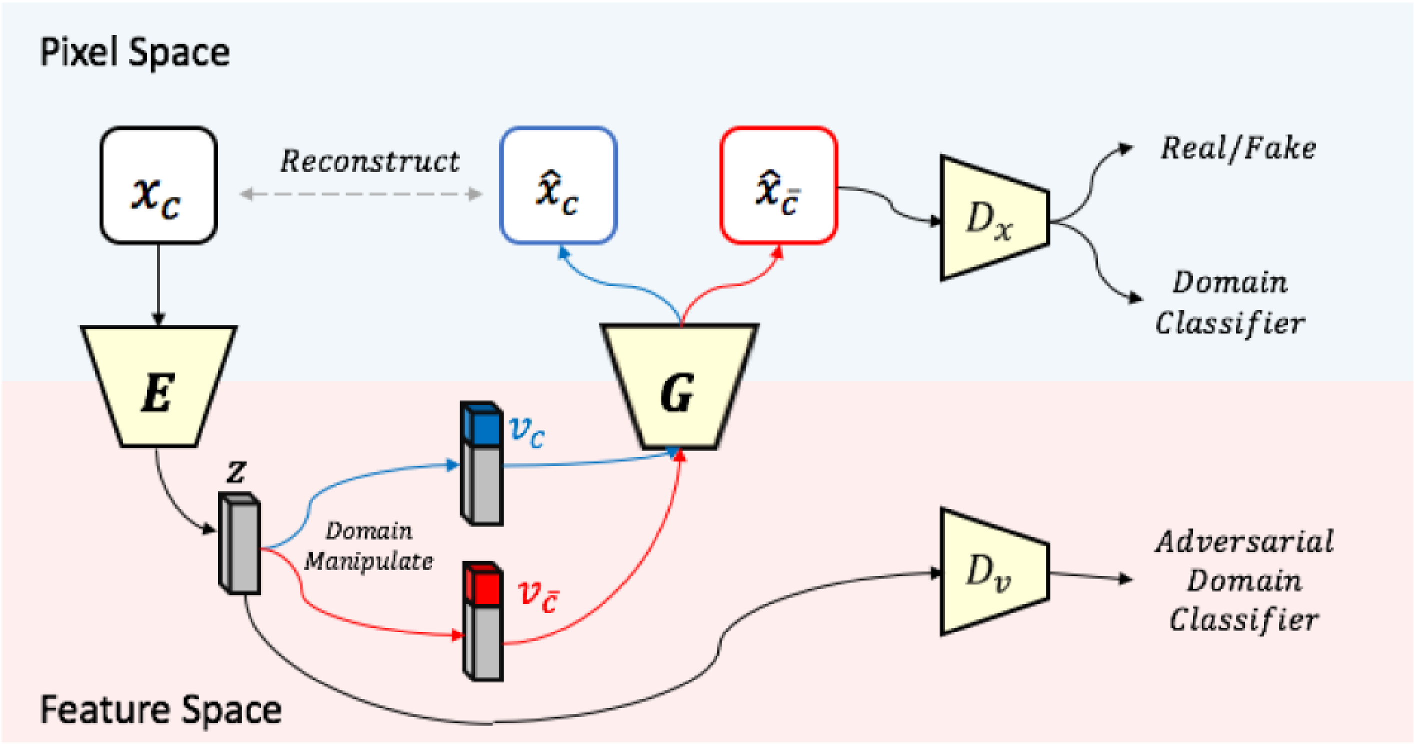
A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation

- Learning interpretable representations



A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation

- Learning interpretable representations



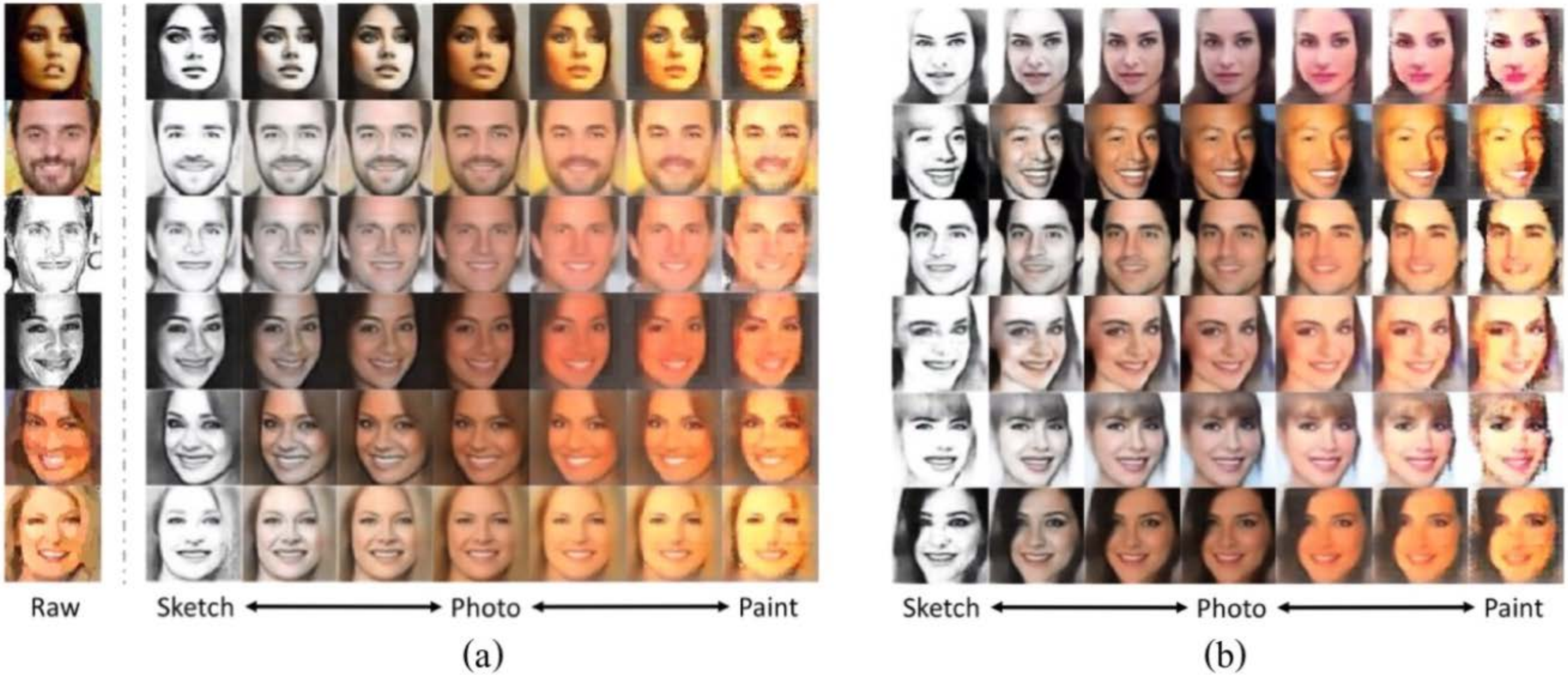
A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation

- Learning interpretable representations

	Unpaired data	Bidirectional translation	Unified structure	Multiple domains	Joint representation	Feature disentanglement
Pix2Pix [10]	-	-	-	-	-	-
CycleGAN [30]	✓	✓	-	-	-	-
StarGAN [4]	✓	✓	✓	✓	-	-
DTN [24]	✓	-	-	-	✓	-
UNIT [16]	✓	✓	-	-	✓	-
E-CDRD [18]	✓	✓	-	✓	✓	✓
UFDN (Ours)	✓	✓	✓	✓	✓	✓

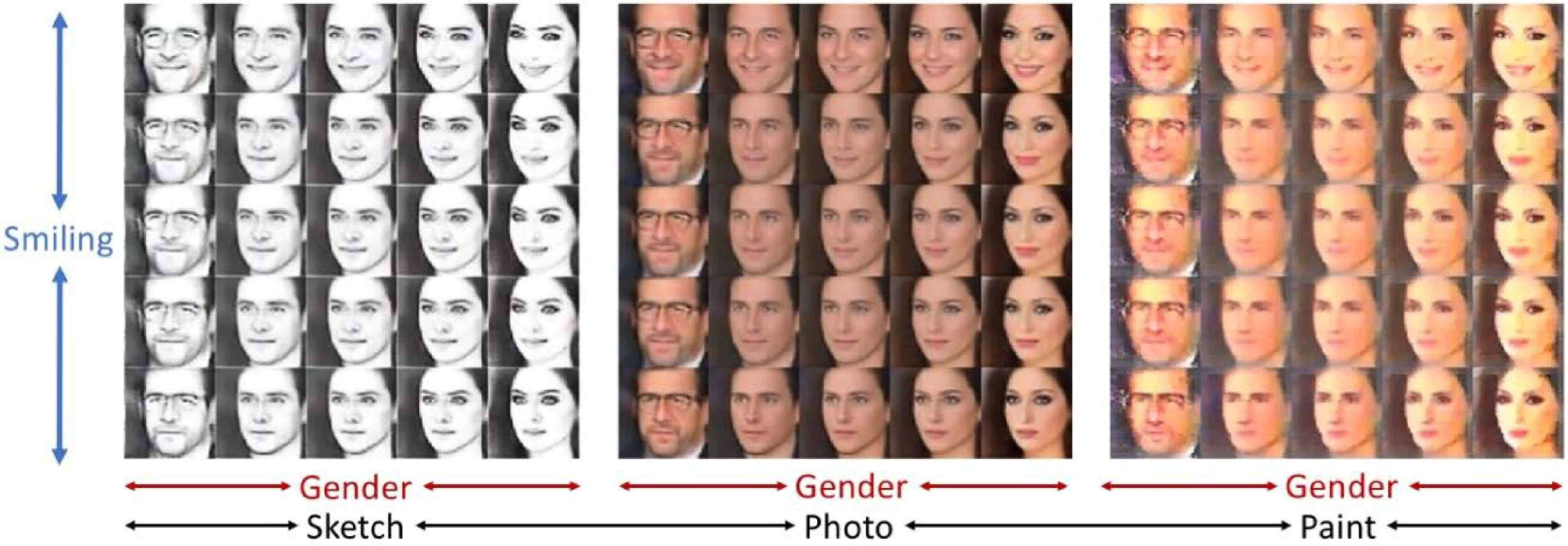
Example Results

- Face image translation



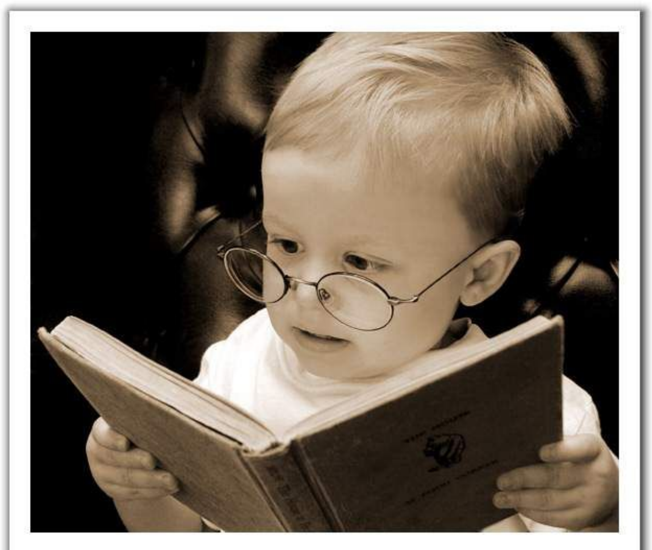
Example Results

- Multi-attribute image translation



What Have We Covered in Today's Lecture?

- Brief Review to CV/ML Backgrounds
- Recent Advances in Deep Learning for Computer Vision
- Transfer Learning and Representation Disentanglement



Resources

- http://deeplearning.net/
  - Hub to many other deep learning resources
- https://github.com/ChristosChristofidis/awesome-deep-learning
  - A resource collection deep learning
- https://github.com/kjw0612/awesome-deep-vision
  - A resource collection deep learning for computer vision
- http://cs231n.stanford.edu/syllabus.html
  - Nice course on CNN for visual recognition
- http://deeplearning.ai
  - Lots of online course videos by Andrew Ng
- http://vllab.ee.ntu.edu.tw/dlcv.html
  - DLCV course at NTU

Vision & Learning Lab at NTU

http://vllab.ee.ntu.edu.tw/

Thank You!