CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

Deep Learning Essentials

9. Task-specific Architectures

Semantic segmentation, Object detection, generative models, ...

Lukáš Neumann

Adapted from B3B33UROB slides of Karel Zimmerman

Semantic segmentation

road

sidewalk

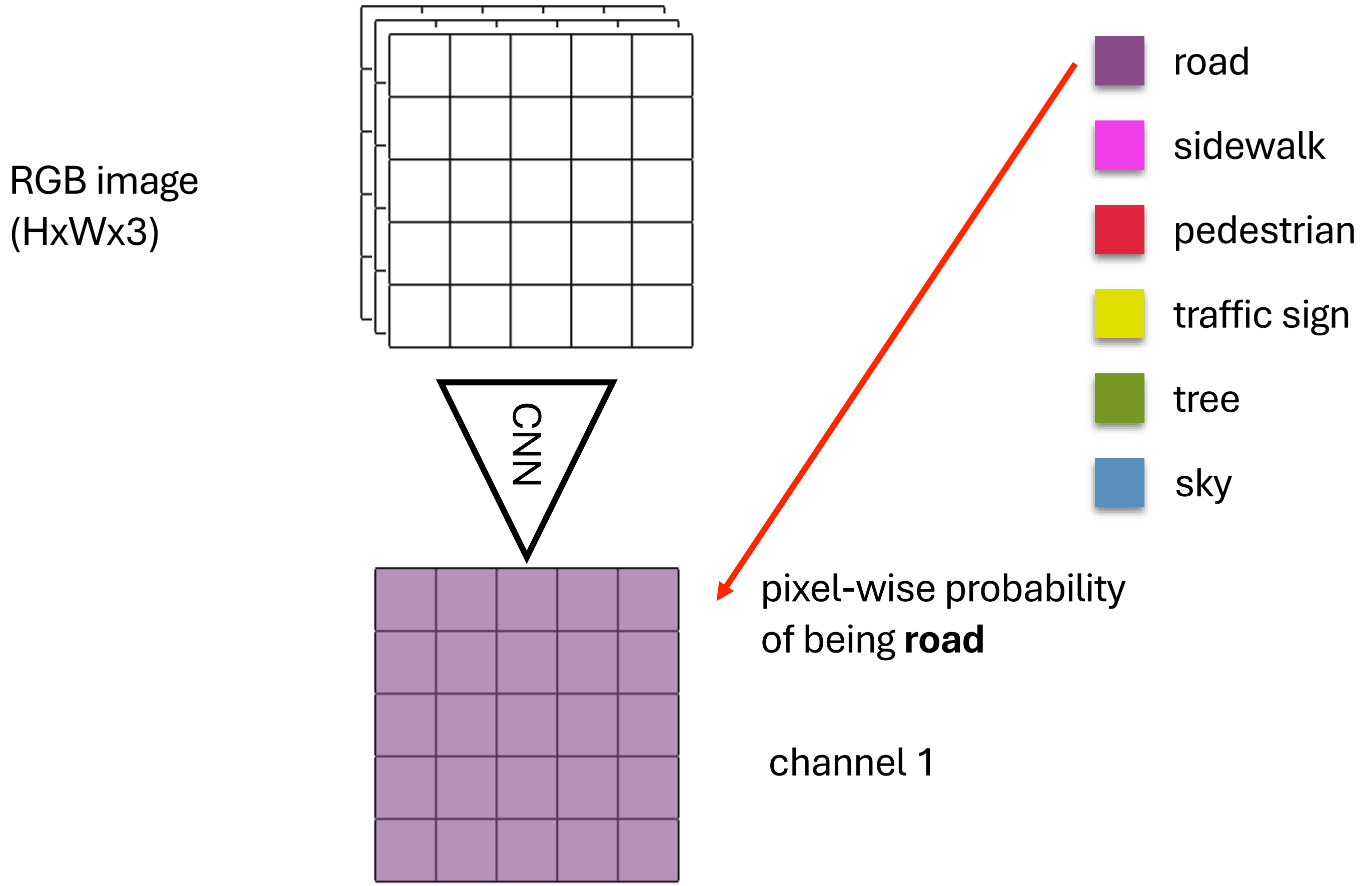
pedestrian

traffic sign

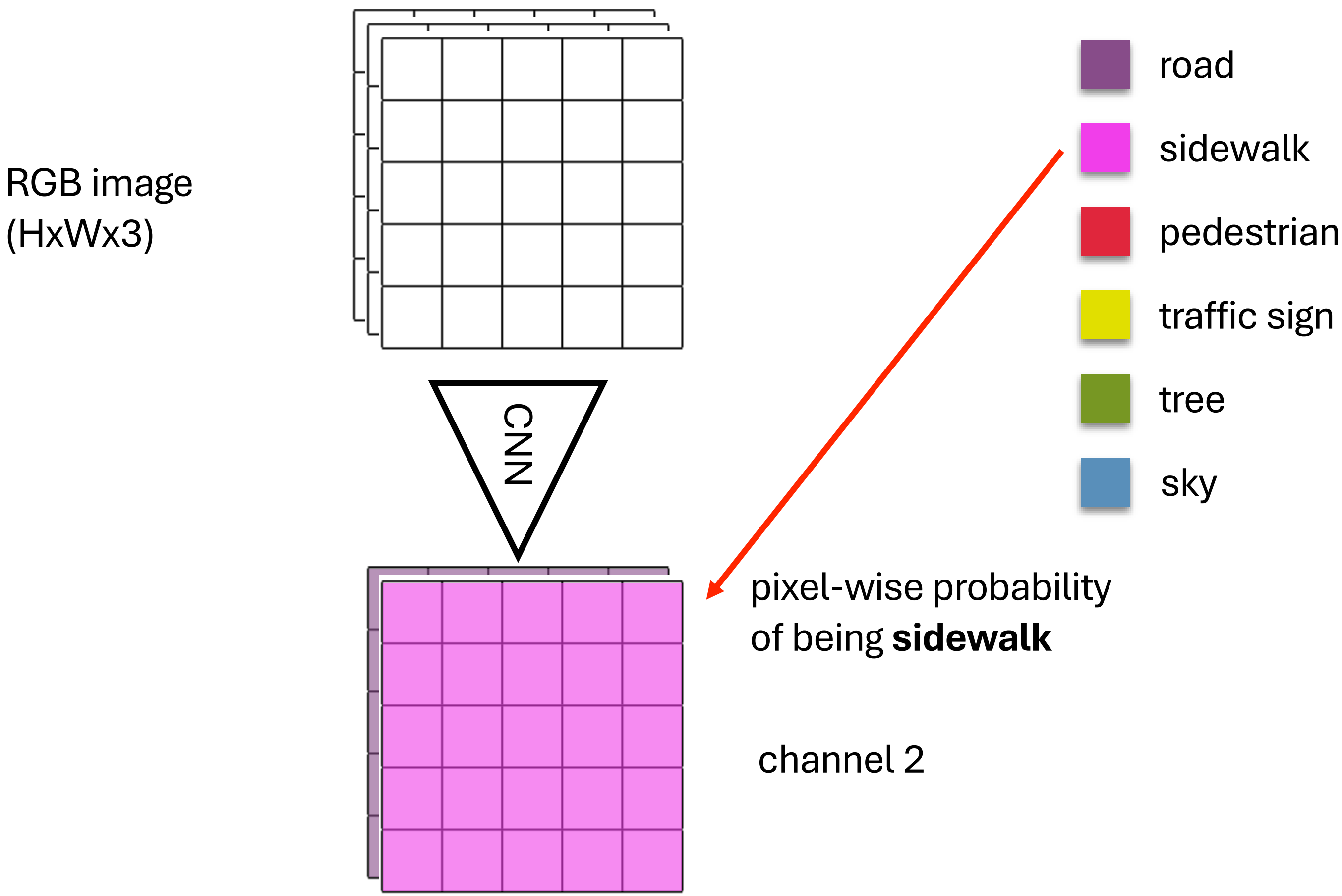
tree

sky

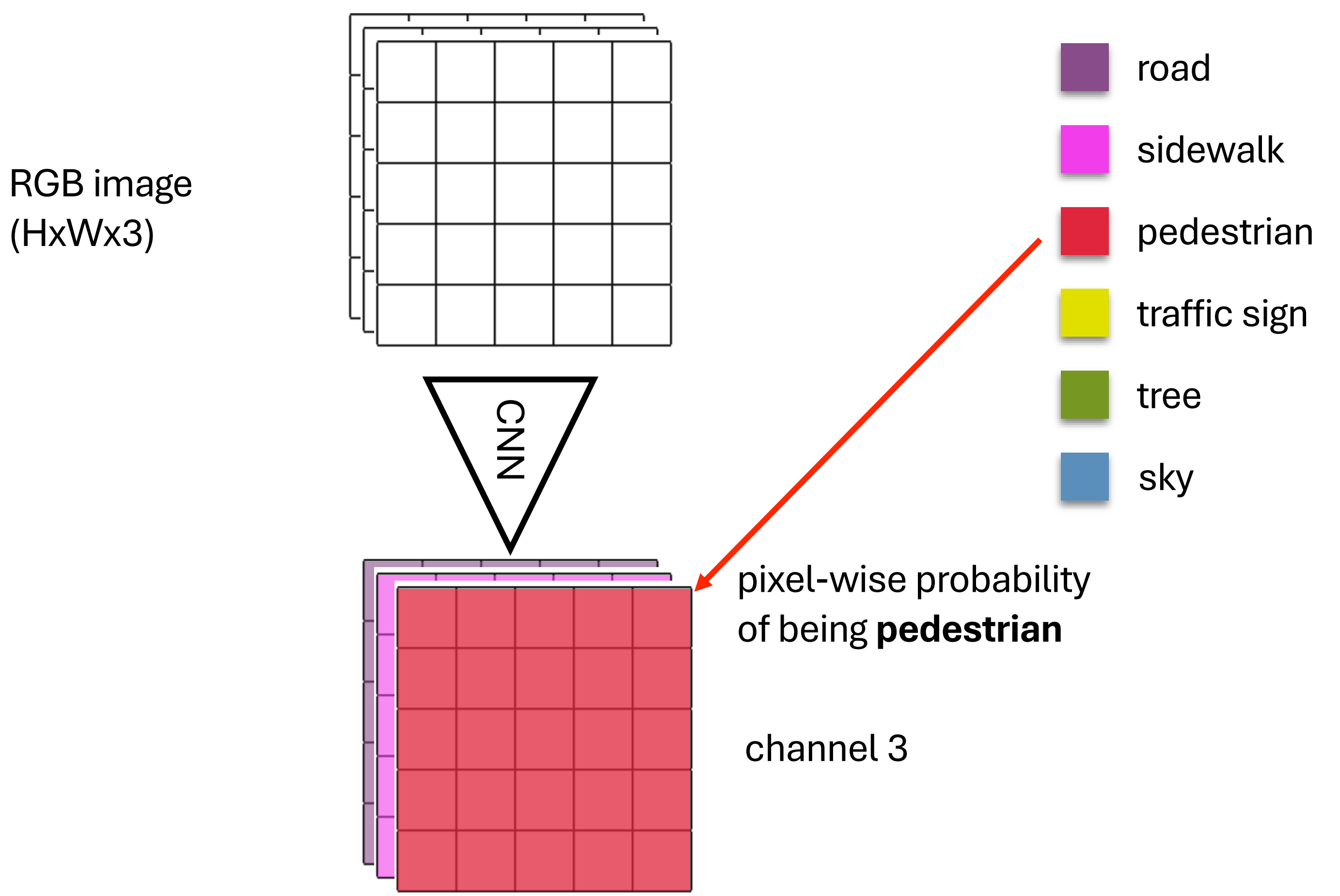
Semantic segmentation



Semantic segmentation

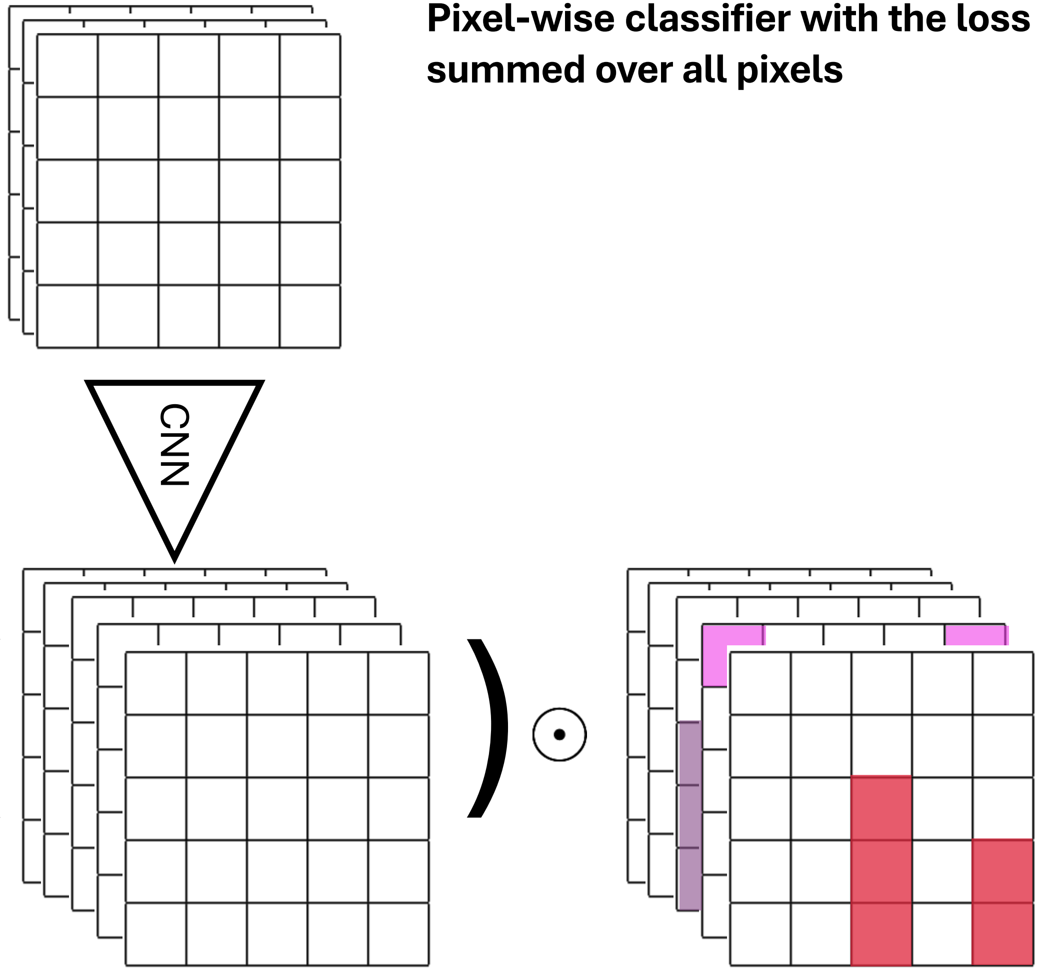


Semantic segmentation



Semantic segmentation

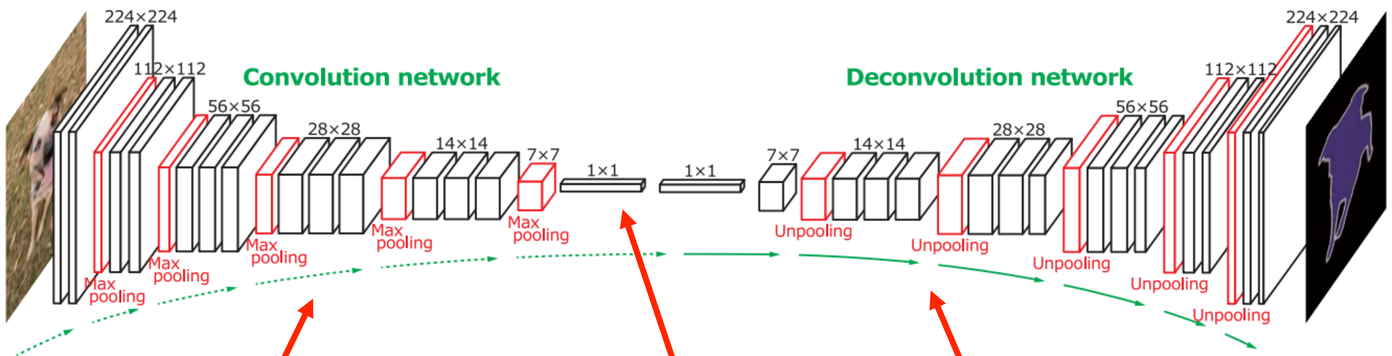
Pixel-wise classifier with the loss summed over all pixels



$$\sum_{\substack{\text{pixels}\\ \text{channels}}} -\log\Big(\quad \Big) \odot$$

Semantic segmentation

- UNet architecture



Convolution layers:
- decrease spatial resolution
- increase number of channels

Deconvolution layers
- Increase spatial resolution
- Decrease number of channels to K classes

Spatial context encoded in channels

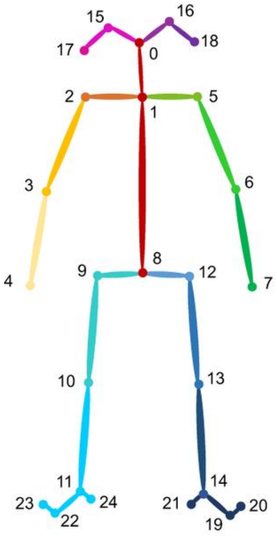
Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention, 2015.

Pose estimation

- Estimate position of human body joints in 2D (3D)

input

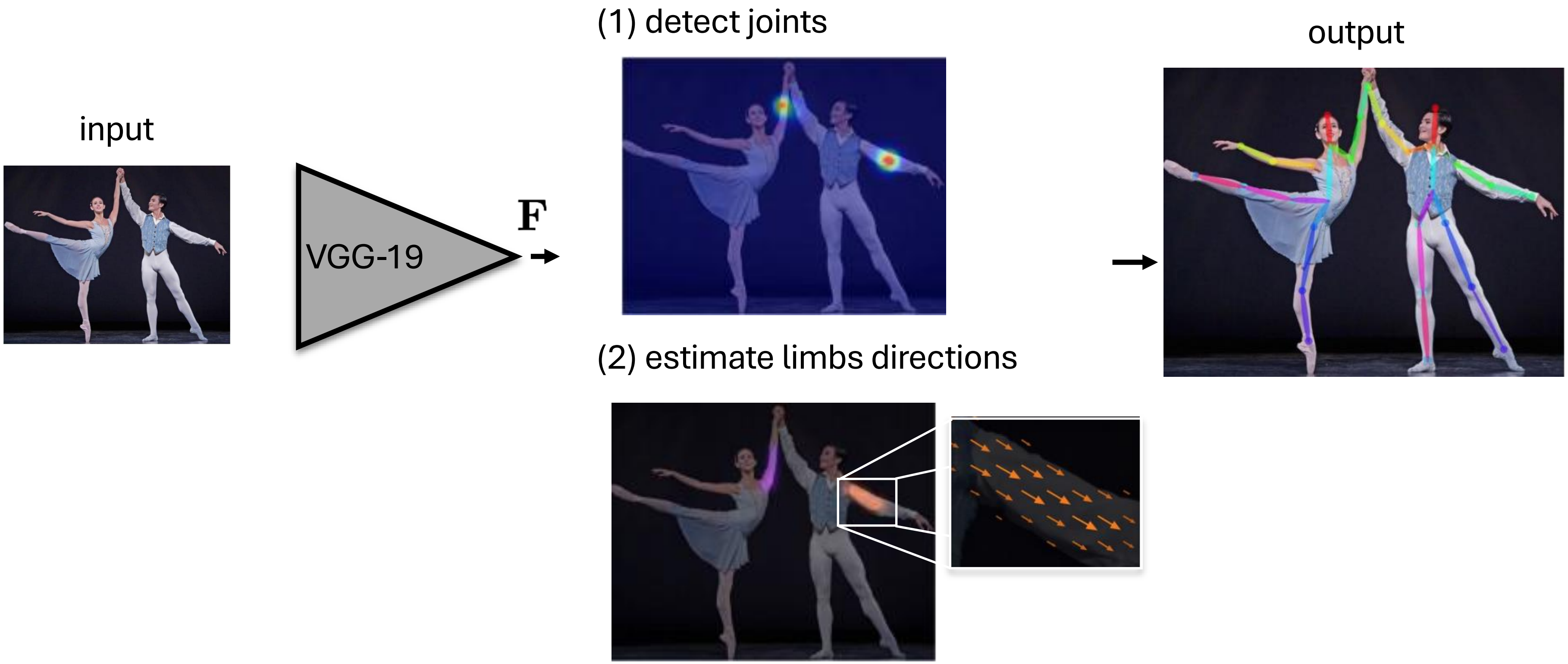
output



0 - Nose
1 - Neck
2 - Right shoulder
3 - Right elbow
4 - Right wrist
5 - Left shoulder
6 - Left elbow
7 - Left wrist
8 - Mid hip
9 - Right hip
10 - Right knee
11 - Right ankle
12 - Left hip
13 - Left knee
14 - Left ankle
15 – Right eye
16 – Left eye
17 – Right ear
18 – Left ear
19 - Left big-toe
20 - Left small-toe
21 - Left heel
22 - Right big-toe
23 - Right small-toe
24 - Right heel

Pose estimation

- OpenPose

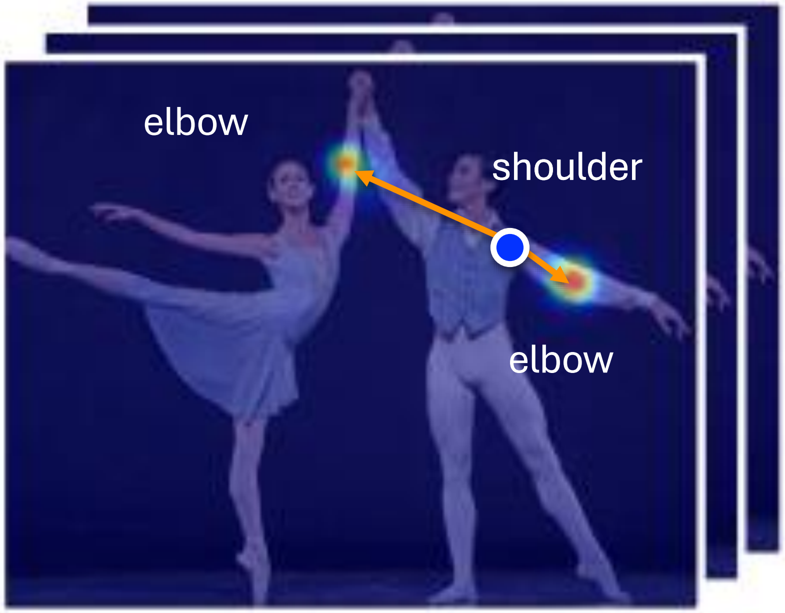


Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." CVPR 2017

Pose estimation

- There are inherent ambiguities how joint positions can be connected together

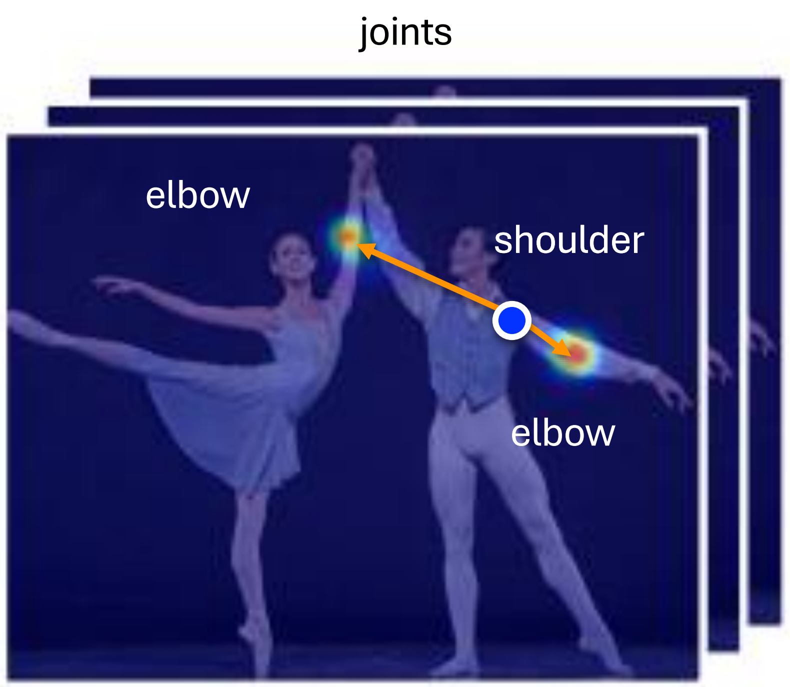
joints



PAFs

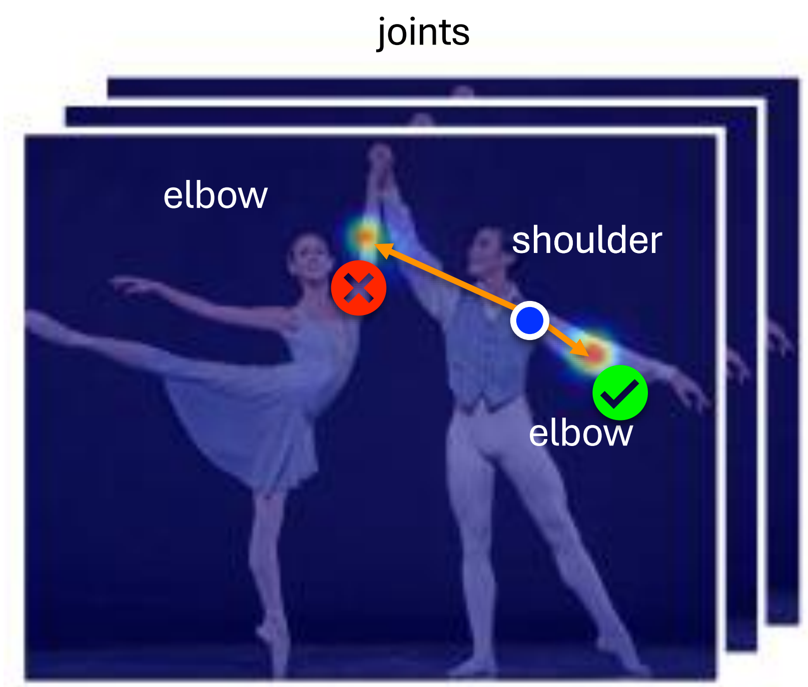
Pose estimation

- There are inherent ambiguities how joint positions can be connected together



Pose estimation

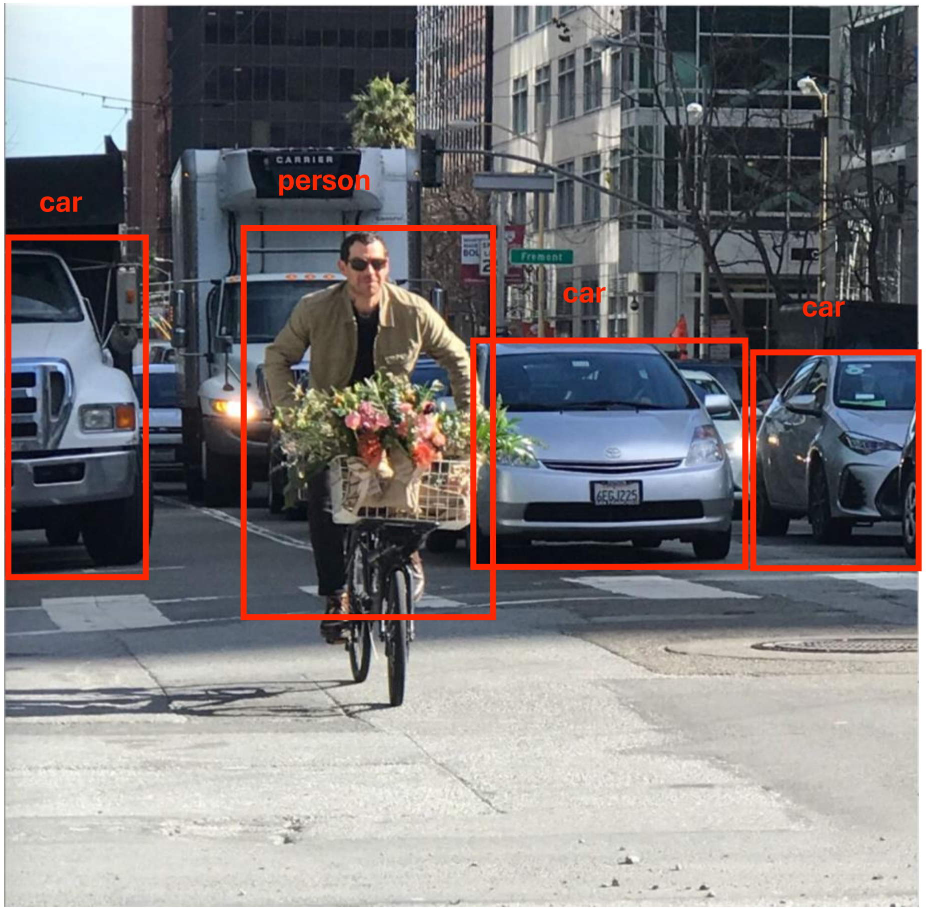
- Point Affinity Fields (PAFs) allow to correctly connect joints



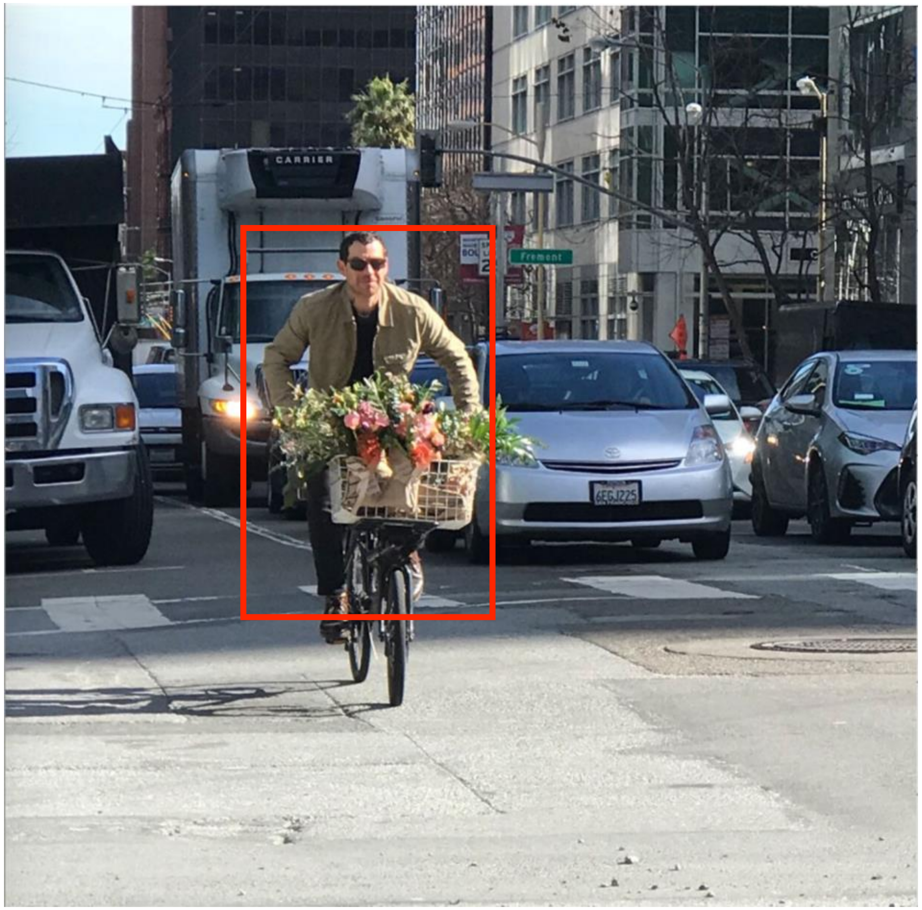
Pose estimation

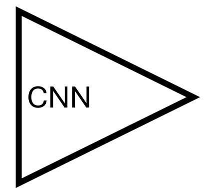
Object detection

Object detection



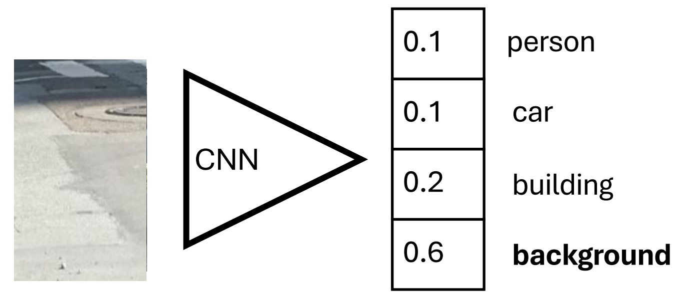
Object detection



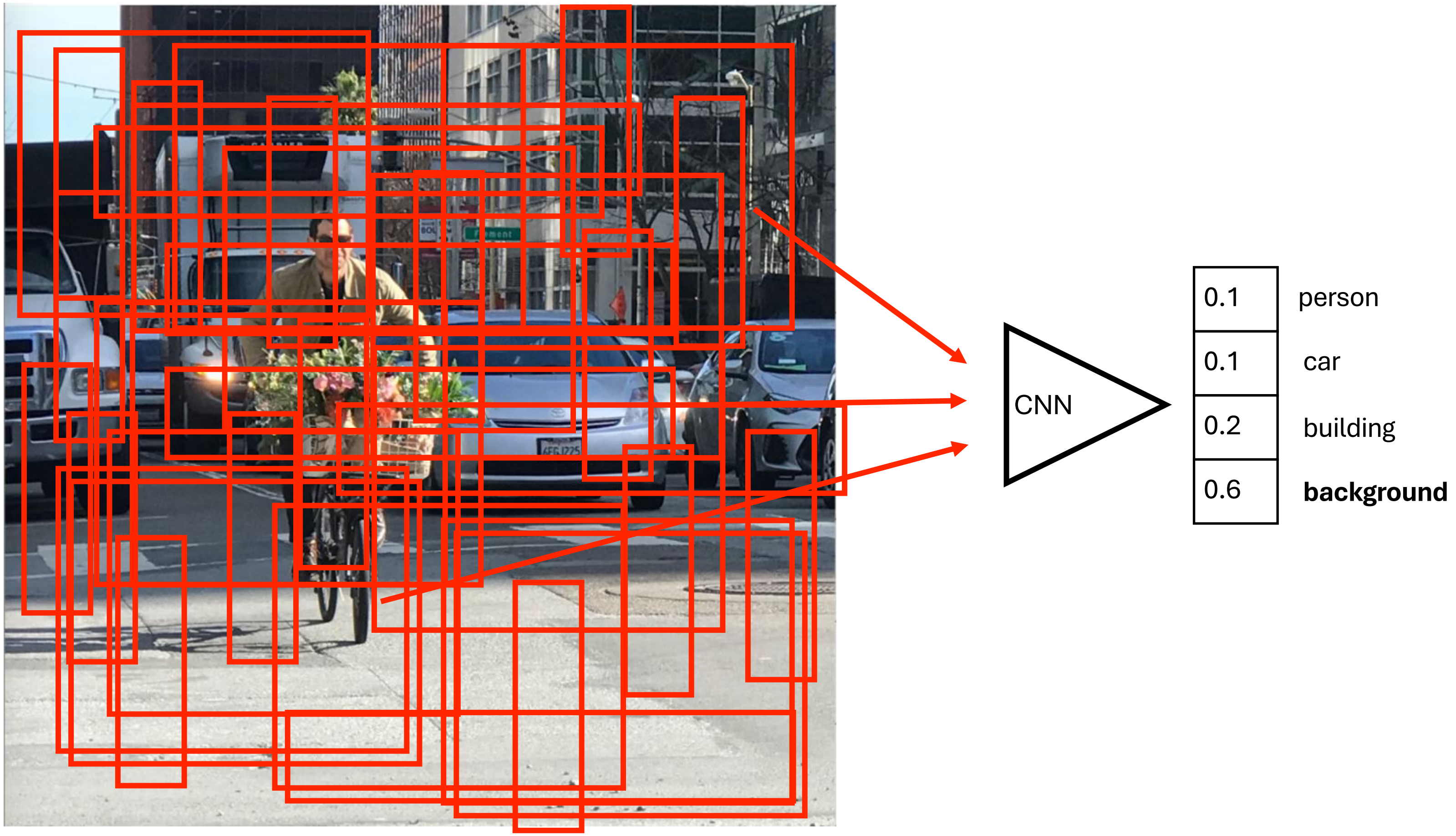


0.6	**person**
0.1	car
0.2	building
0.1	background

Object detection



Object detection

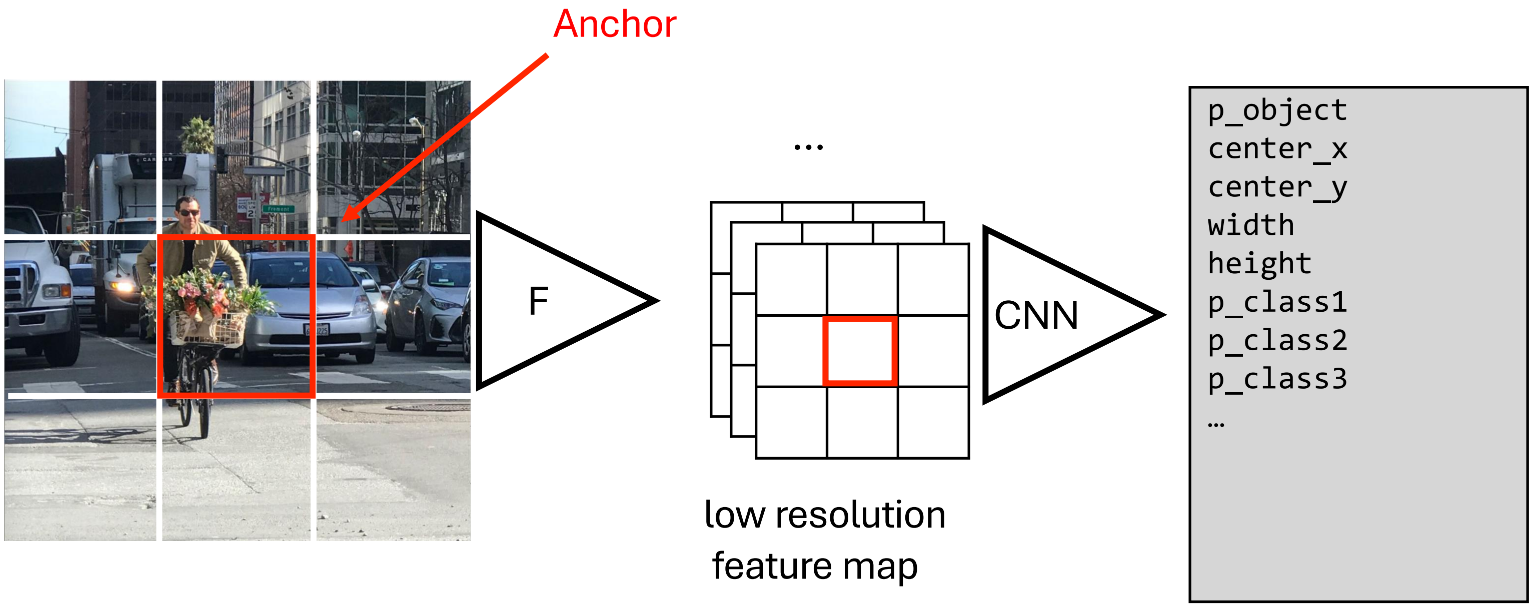


Classify all rectangles?

H x W x AspectRatio x Scales x 0.001 sec = **months**

Object detection

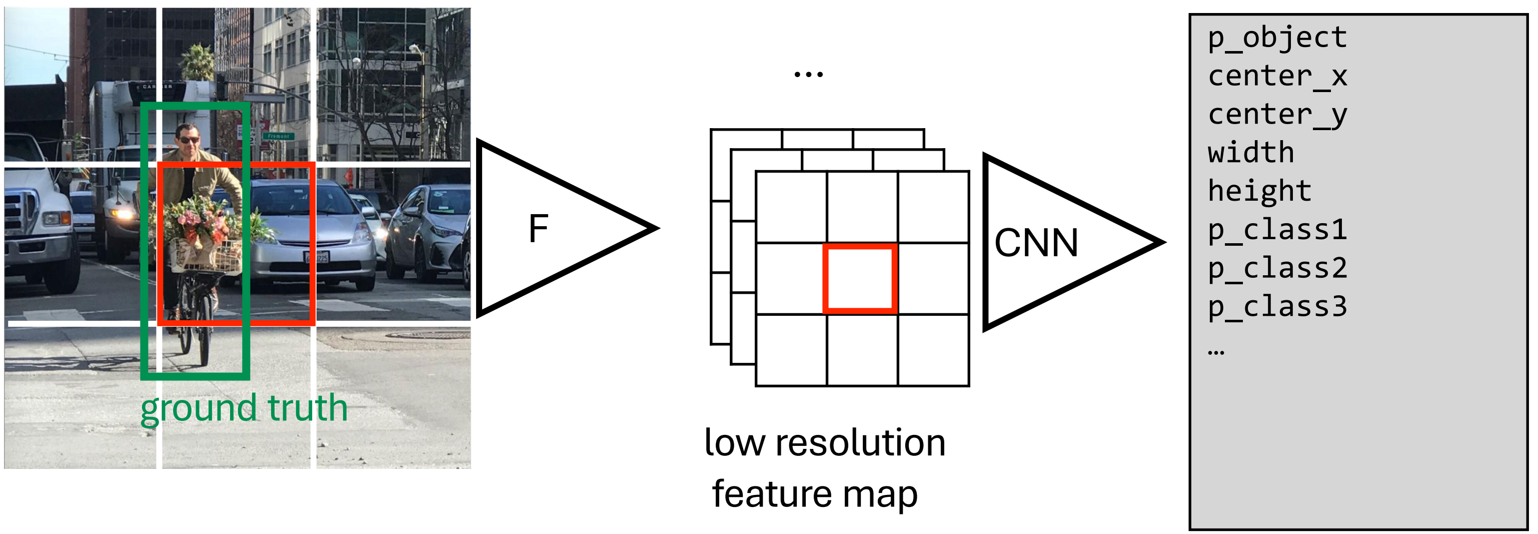
- YOLO (You Only Look Once)
- Divide image into MxM sub images (corresponding to its receptive fields)
- Predict relative position, objectness, class for each patch



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

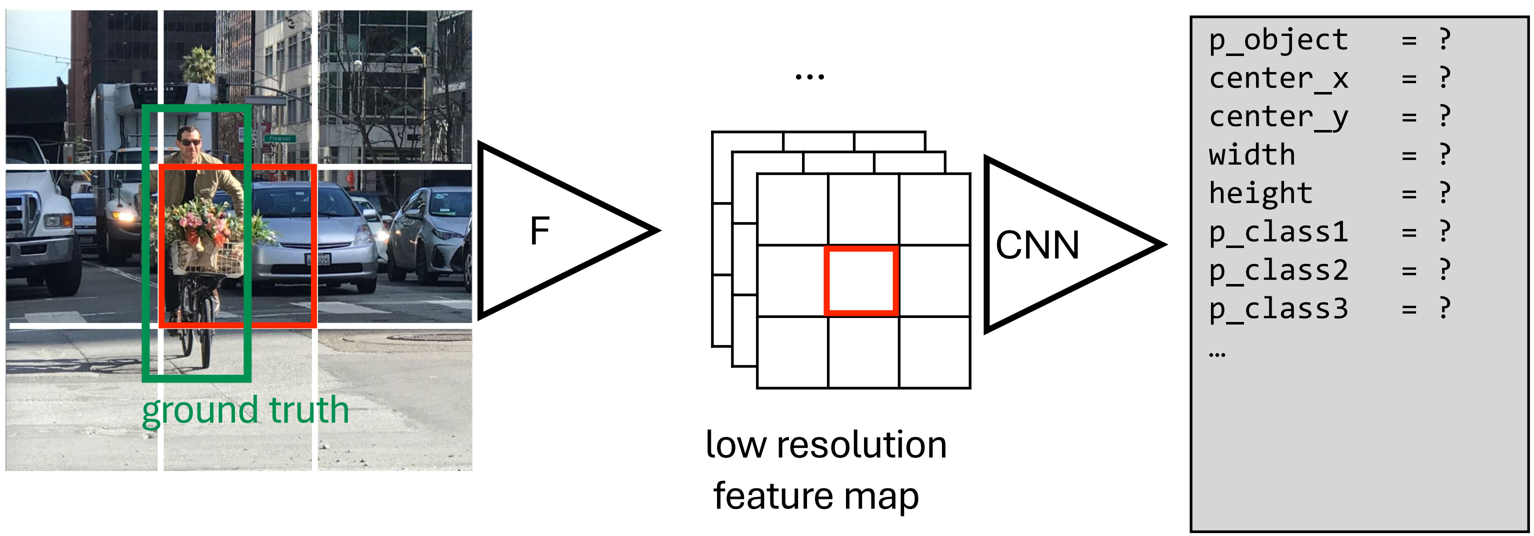
- YOLO (You Only Look Once)
- Divide image into MxM sub images (corresponding to its receptive fields)
- Predict relative position, objectness, class for each patch



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

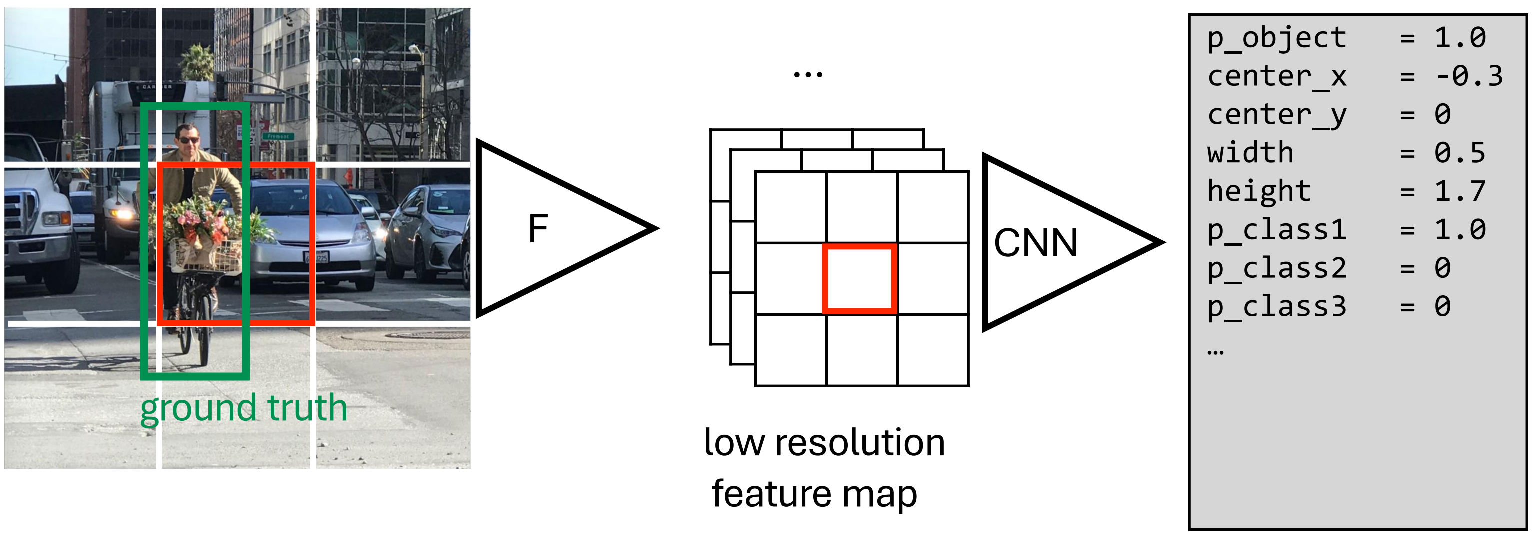
- YOLO (You Only Look Once)
- Divide image into MxM sub images (corresponding to its receptive fields)
- Predict relative position, objectness, class for each patch



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

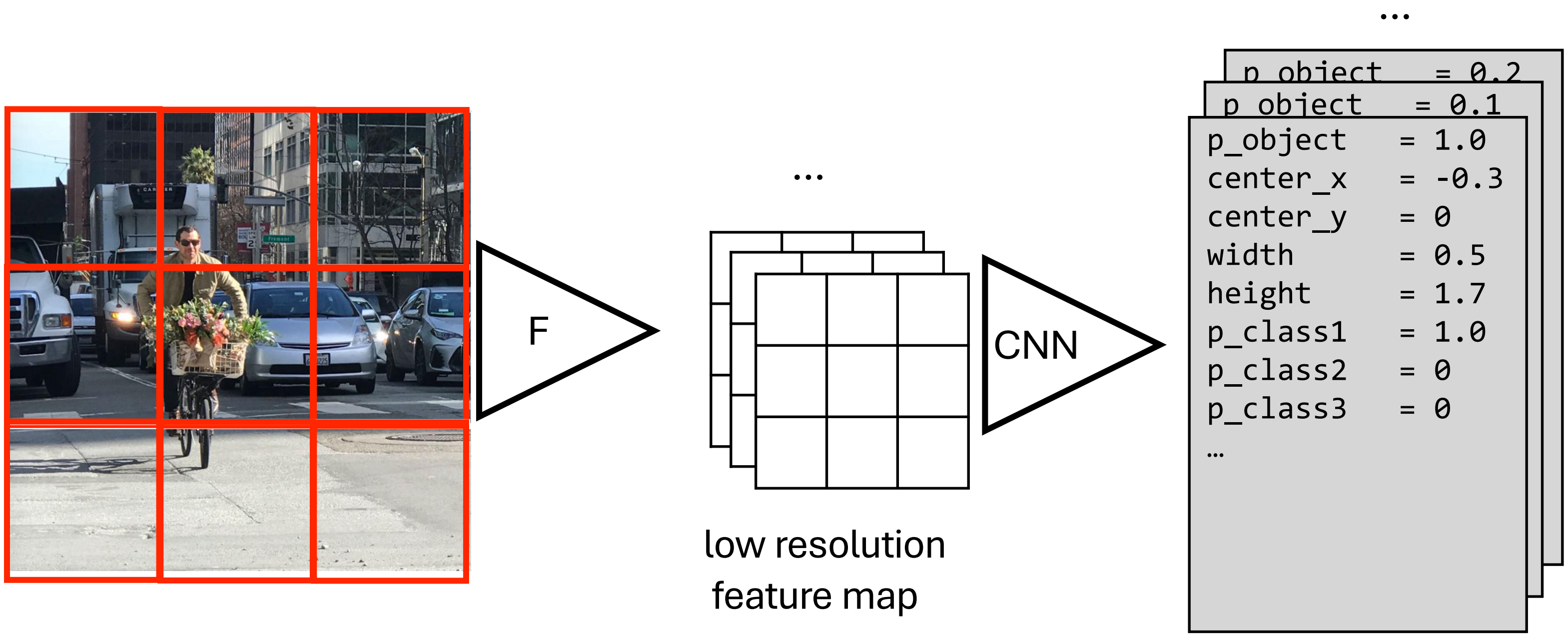
- YOLO (You Only Look Once)
- Divide image into MxM sub images (corresponding to its receptive fields)
- Predict relative position, objectness, class for each patch



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

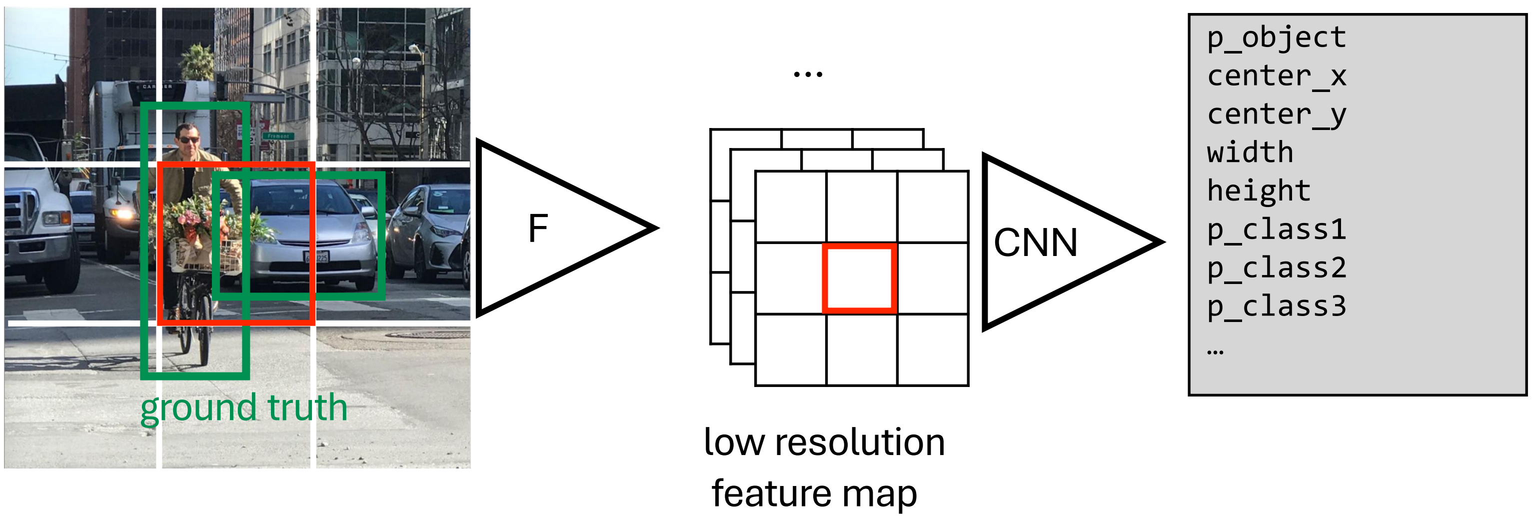
- Each image patch has its own set of outputs



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

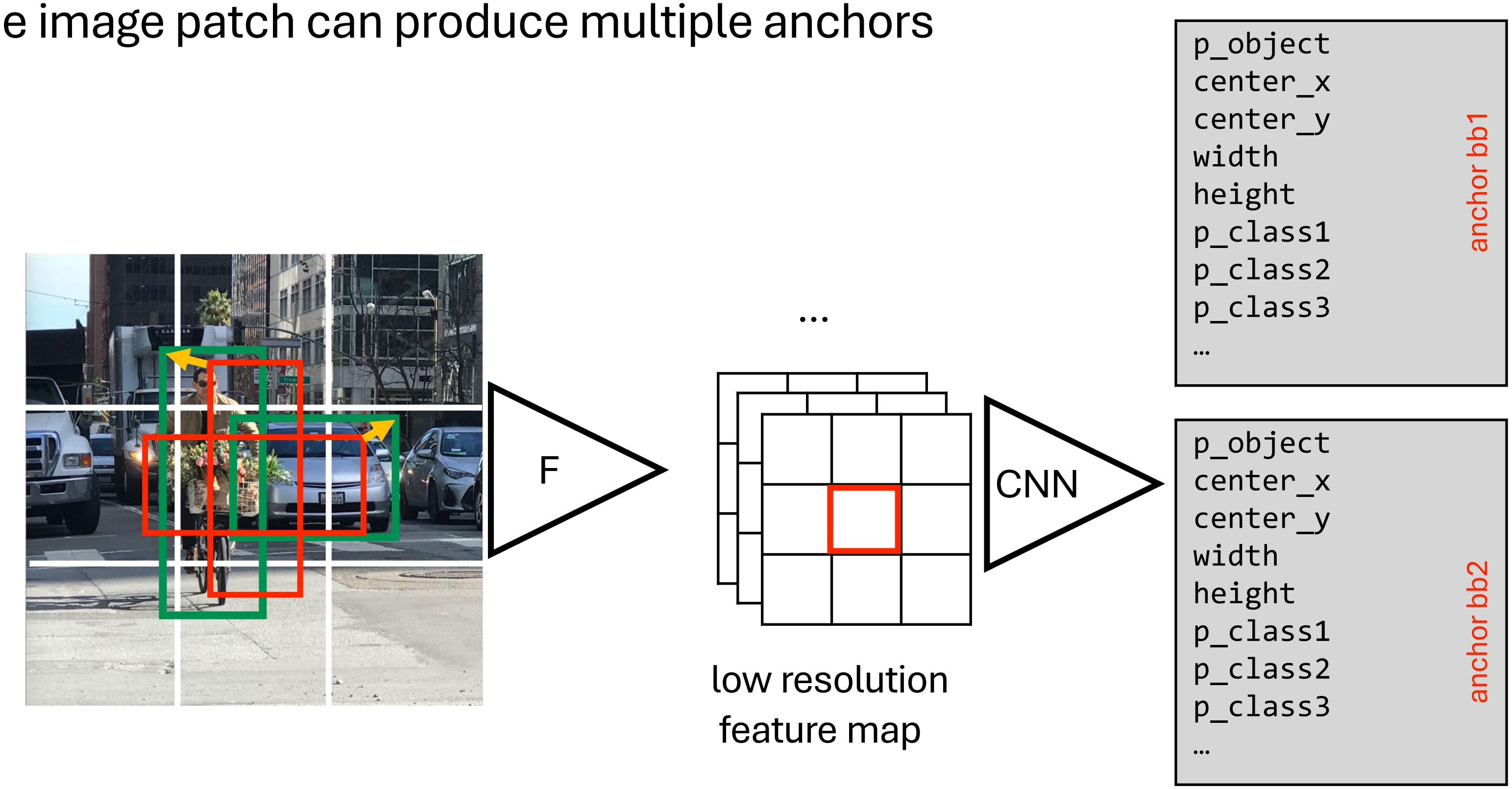
- Do you see a problem?
- Multiple objects in a single patch!



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

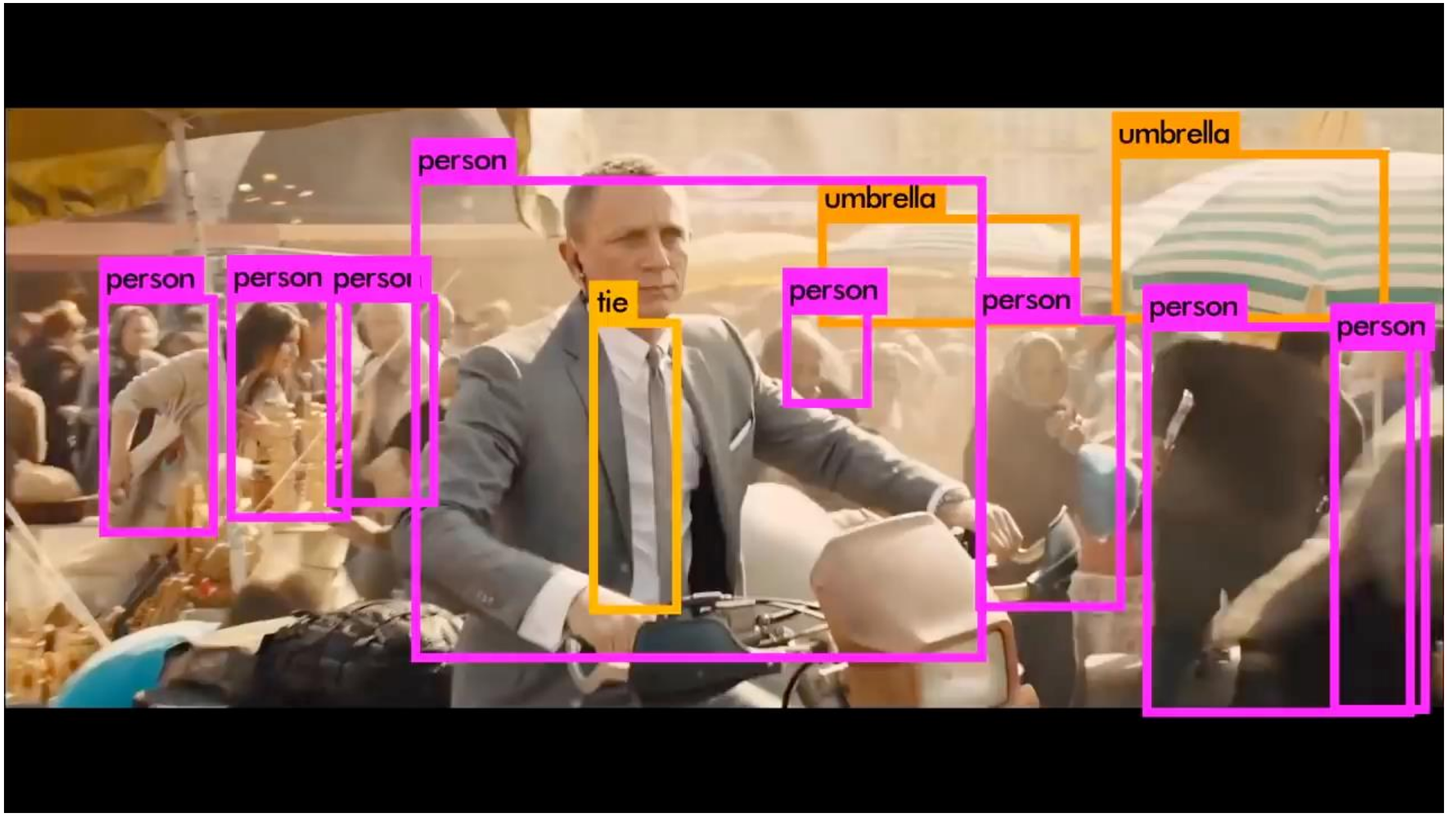
- Multiple anchors with different aspect ratios
- One image patch can produce multiple anchors



Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR 2016

Object detection

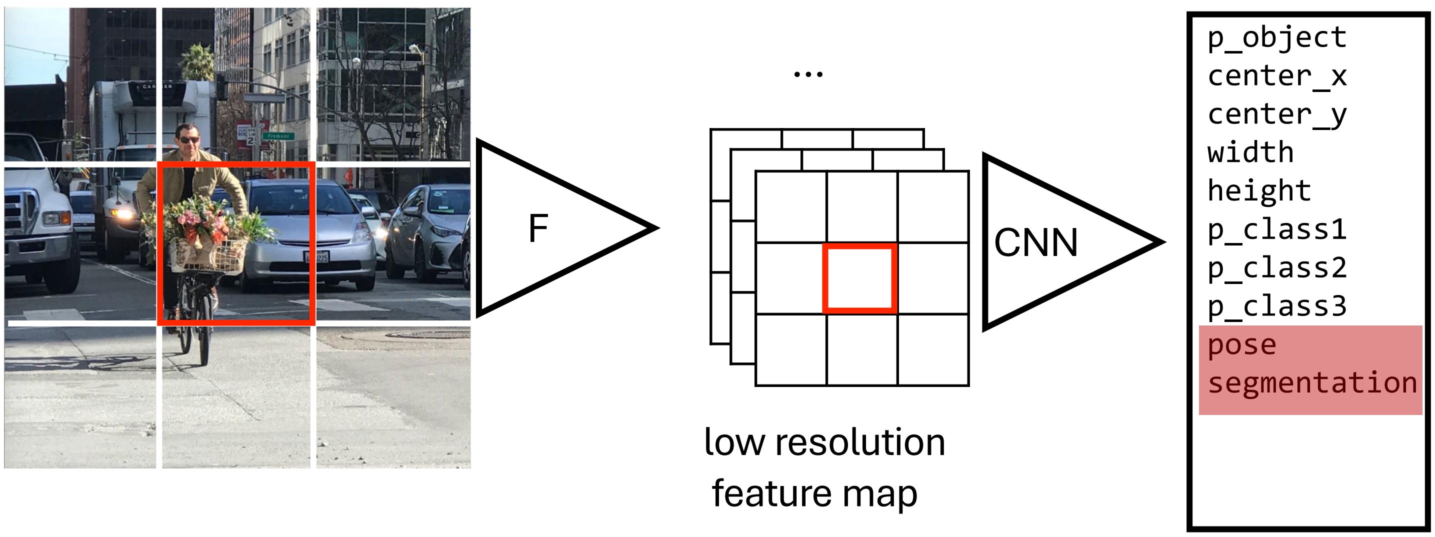
- YOLOv2



Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger.", CVPR2017

Instance segmentation

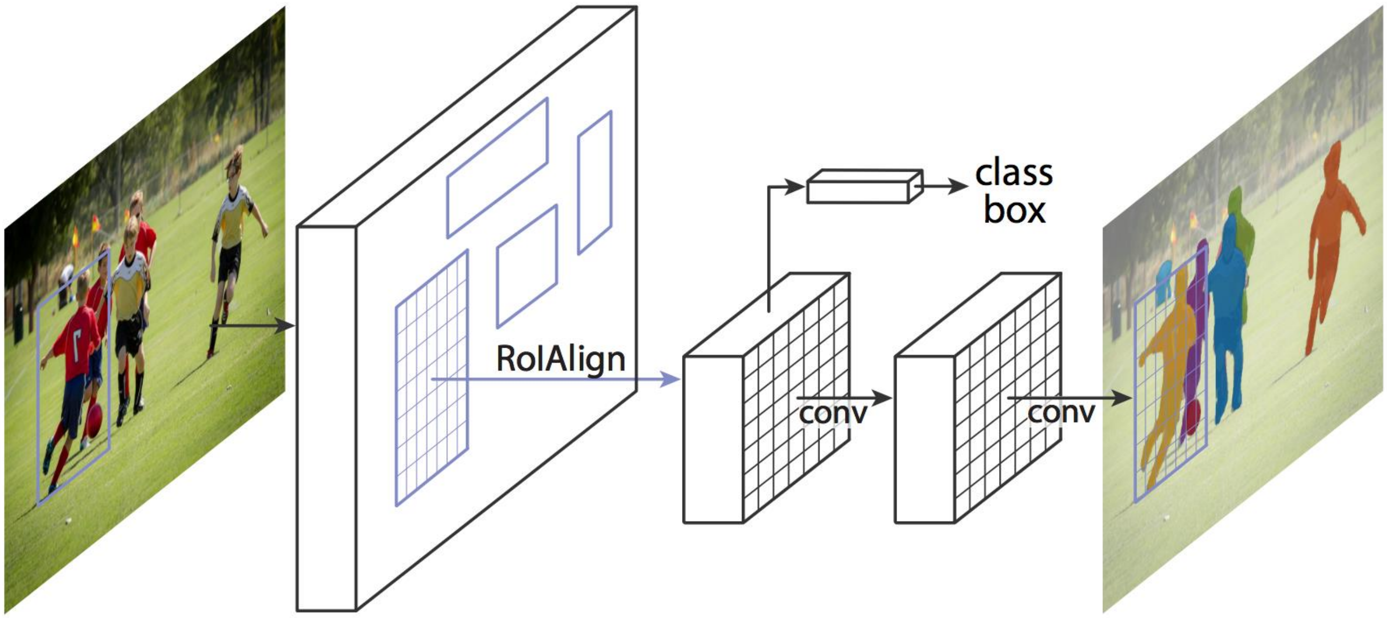
- Mask R-CNN
- Added pose and segmentation head for each detected object



He, Kaiming, et al. "Mask R-CNN" ICCV 2017

Instance segmentation

- Mask R-CNN
- Added pose and segmentation head for each detected object



He, Kaiming, et al. "Mask R-CNN" ICCV 2017

Instance segmentation

- Mask R-CNN
- Added pose and segmentation head for each detected object

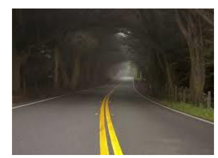
He, Kaiming, et al. "Mask R-CNN" ICCV 2017

Object detector evaluation

- Two classes – positive (object is present) & negative (object is not present)

- Example - Car detector
  - Car is present (positive class)
  - Car is not present = background only (negative class)






Object detector evaluation

Positive class

Negative class

Classifier output:






Object detector evaluation

Positive class

Negative class

Classifier output:






true positive (TP) … classifier correctly found the object (e.g. car) = the **positive** class

Object detector evaluation

Positive class

Negative class

Classifier output:






true positive (TP) … classifier correctly found the object (e.g. car) = the **positive** class

true negative (TN) … classifier correctly indicated the object is missing = the **negative** class

Object detector evaluation

Positive class

Negative class

Classifier output:






true positive (TP) ... classifier correctly found the object (e.g. car) = the **positive** class

true negative (TN) ... classifier correctly indicated the object is missing = the **negative** class

false negative (FN) ... classifier **falsely** indicates **negative** class in presence of the object
→ **missed danger**

Object detector evaluation

Positive class

Negative class

Classifier output:

true positive (TP) … classifier correctly found the object (e.g. car) = the **positive** class

true negative (TN) … classifier correctly indicated the object is missing = the **negative** class

false negative (FN) … classifier **falsely** indicates **negative** class in presence of the object
→ **missed danger**

false positive (FP) … classifier **falsely** indicates **positive** class where the object is missing
→ **false alarm**

Object detector evaluation

Positive class

Negative class

Classifier output:






true positive (TP) =1

true negative (TN) =2

false negative (FN) =1

false positive (FP) =2

“1/3 of of samples classified as CARS are actually CARS”

$$\text{Precision (P)} = \frac{TP}{TP + FP} = \frac{1}{1+2} = 1/3$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = \frac{1}{1+1} = 1/2$$

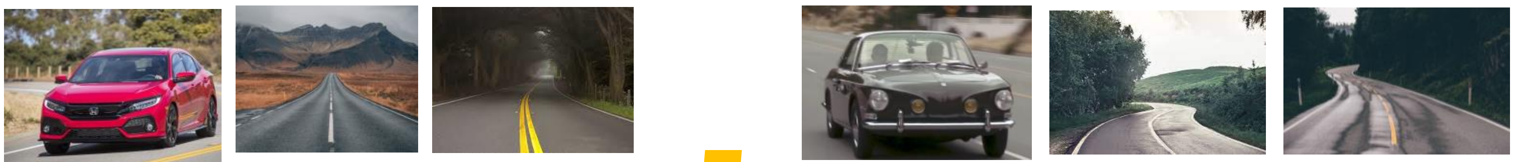
“1/2 of all CARS are discovered.”

What is the best possible outcome? Oracle: Precision = Recall = 1

Object detector evaluation

Positive class

Negative class



Classifier score:

0.9 0.5 0.1 -0.1 -0.4 -0.6

Threshold = 0

true positive (TP) =1

true negative (TN) =2

false negative (FN) =1

false positive (FP) =2

$$\text{Precision (P)} = \frac{TP}{TP + FP} = \frac{1}{1 + 2} = 1/3$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = \frac{1}{1 + 1} = 1/2$$

Object detector evaluation

Positive class




Negative class


Classifier score:	0.9	0.5	0.1	-0.1	-0.4	-0.6

Threshold = -0.2

true positive (TP) =2

true negative (TN) =2

false negative (FN) =0

false positive (FP) =2

$$\text{Precision (P)} = \frac{TP}{TP + FP} = \frac{2}{2+2} = 1/2$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = \frac{2}{2+0} = 1$$

Object detector evaluation

Positive class Negative class

Classifier score: 0.9 0.5 0.1 -0.1 -0.4 -0.6

Threshold = -0.5

true positive (TP) =2

true negative (TN) =1

false negative (FN) =0

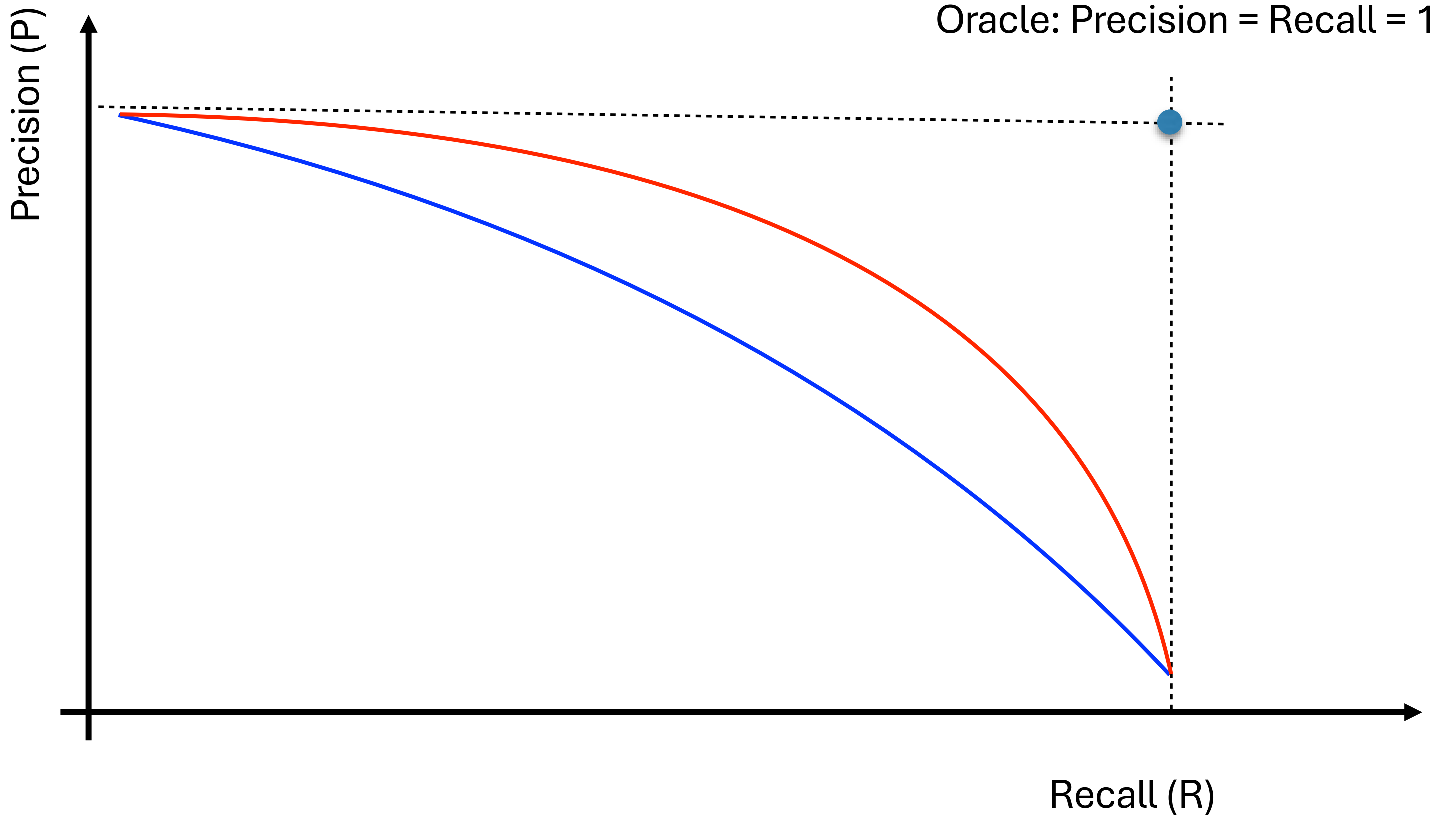
false positive (FP) =3

$$\text{Precision (P)} = \frac{TP}{TP + FP} = \frac{2}{2+3} = 2/5$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = \frac{2}{2+0} = 1$$

Object detector evaluation

- PR-curve



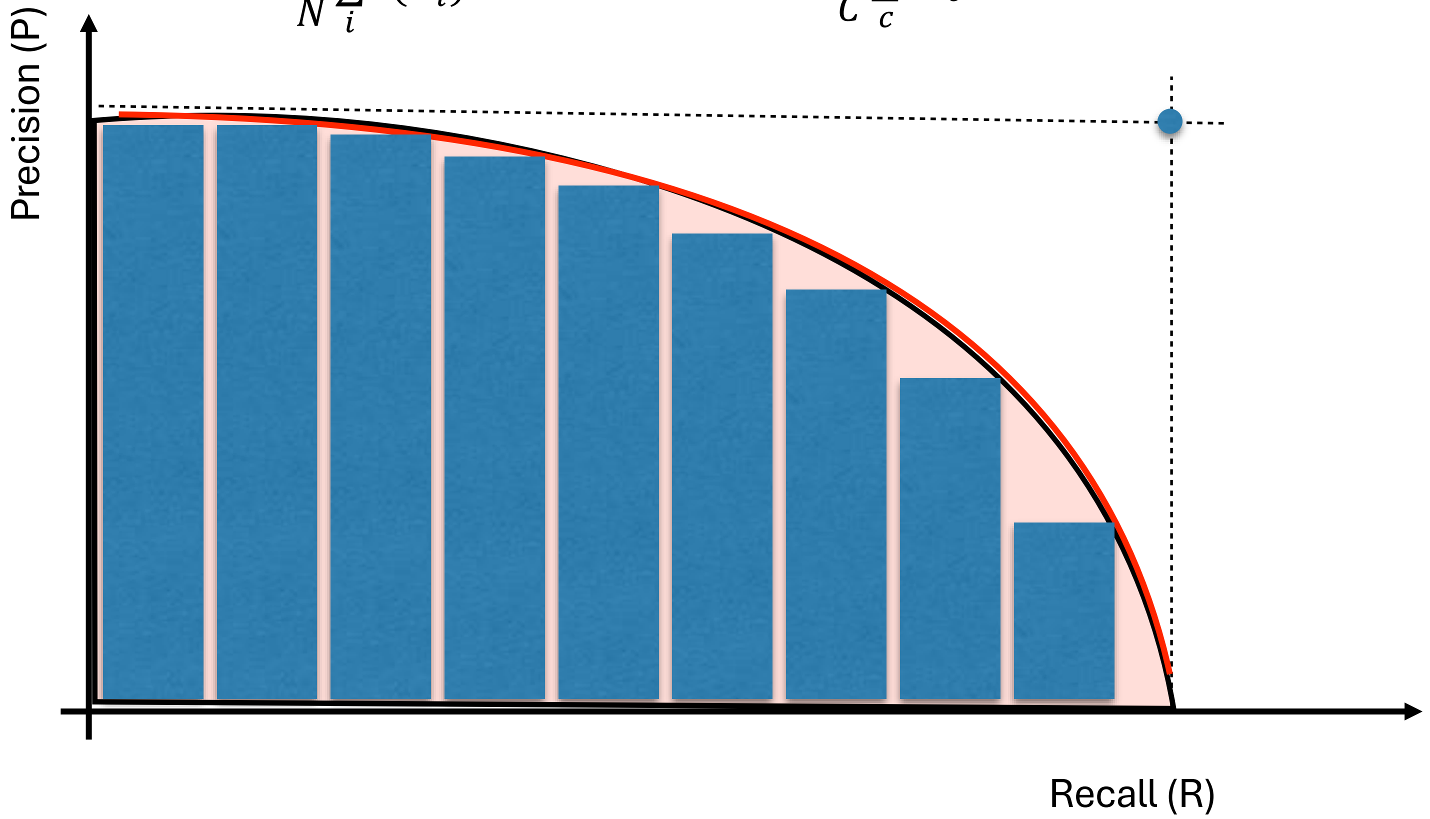
Object detector evaluation

Average Precission

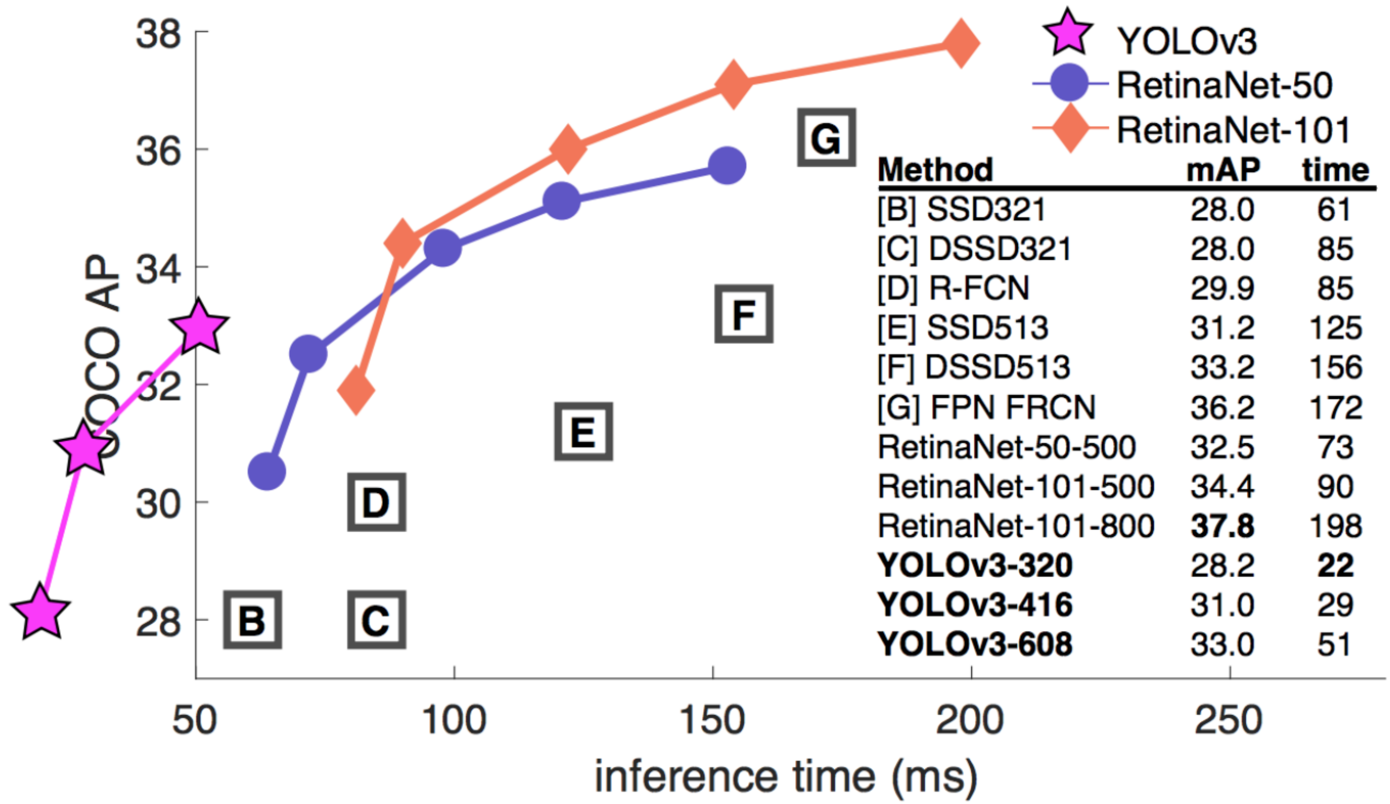
$$\mathrm{AP} = \frac{1}{N}\sum_i \mathrm{P}(\mathrm{R}_i)$$

mean Average Precission

$$\mathrm{mAP} = \frac{1}{C}\sum_c \mathrm{AP}_c$$



Object detector evaluation



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

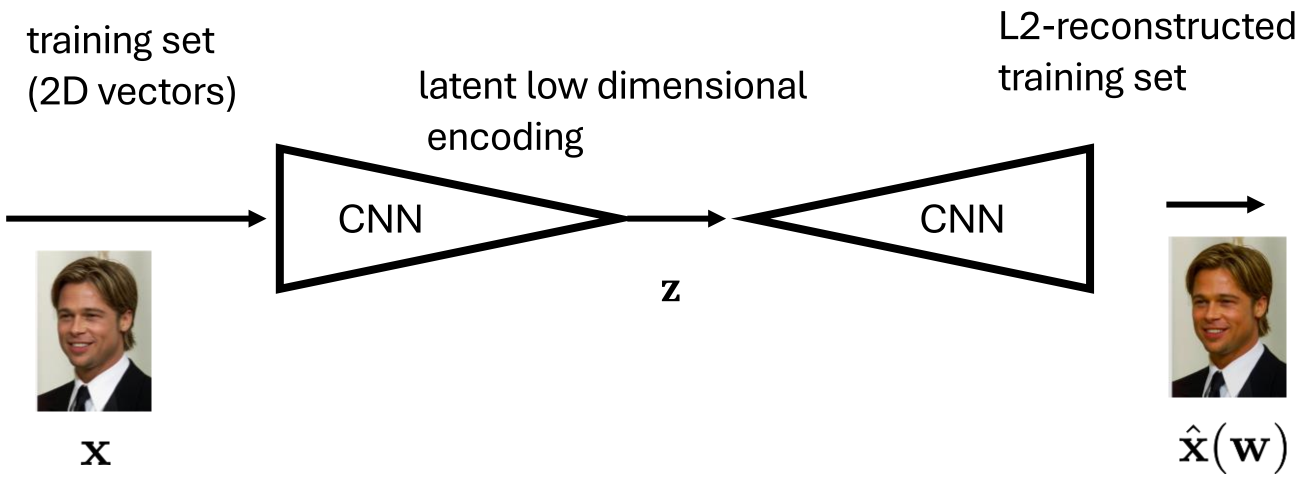
Generative models

- Variational Auto-Encoders (VAEs)
- Generative Adversarial Networks (GANs)
- Diffusion models

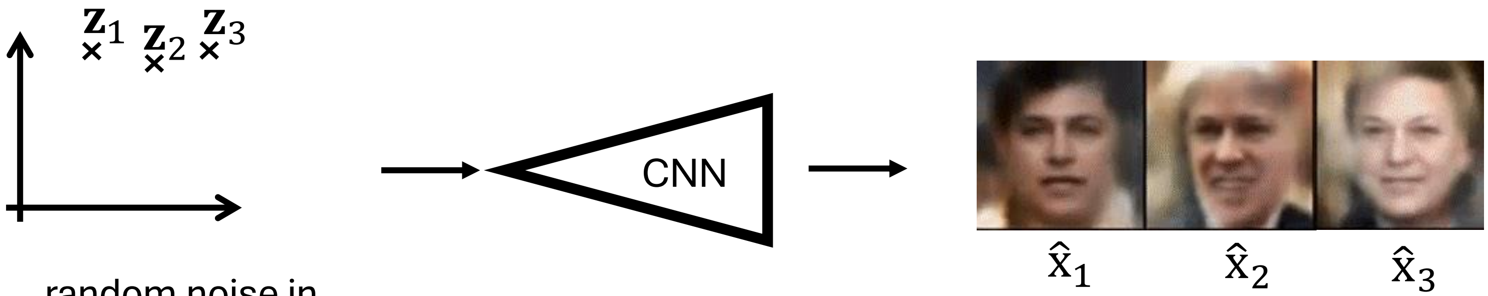
Variational Auto-Encoders (VAEs)

- Latent space $\mathbf{z}$ is pushed towards a Gaussian distribution
- Learning the self-reconstruction with L2 reconstruction loss

$$\arg\min_{\mathbf{w}} \|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{w})\|_2^2$$

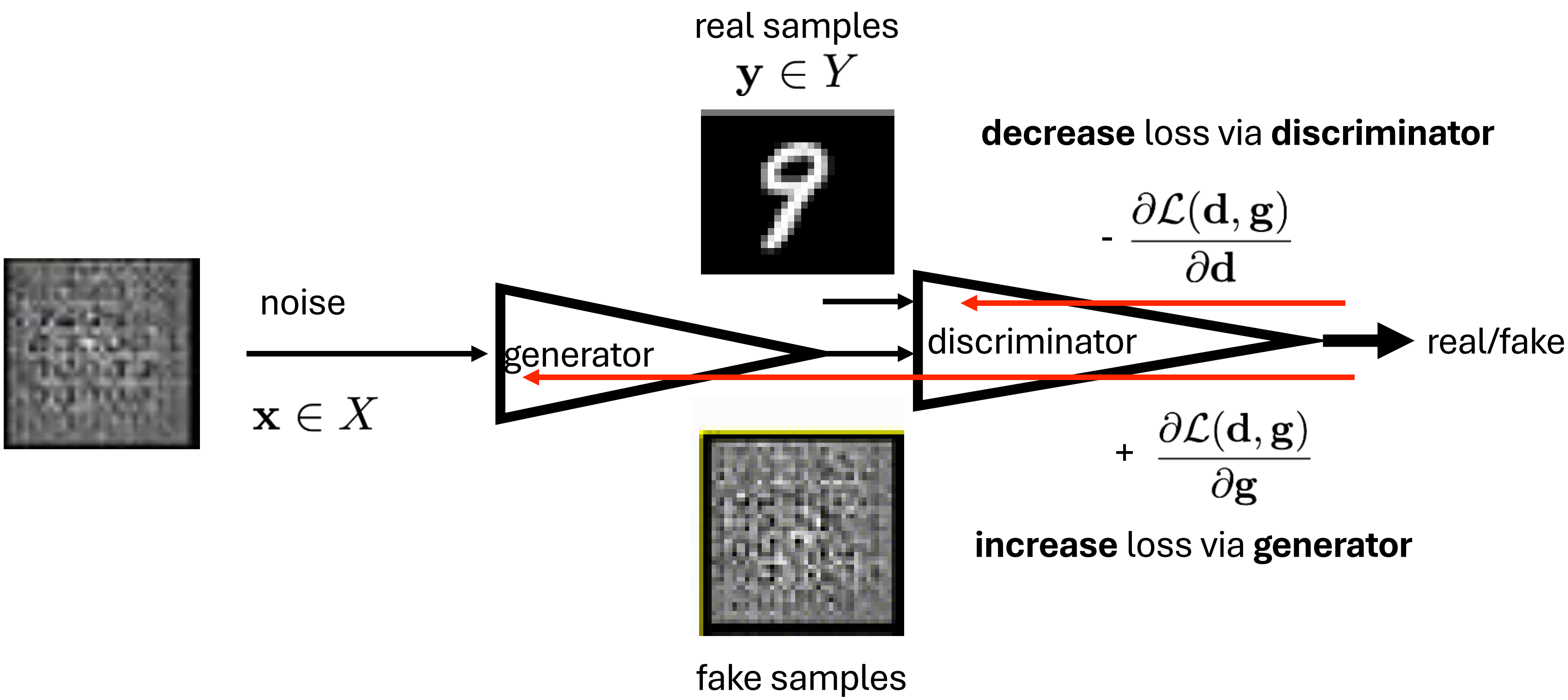


Variational Auto-Encoders (VAEs)



random noise in
low dimensional
encoding

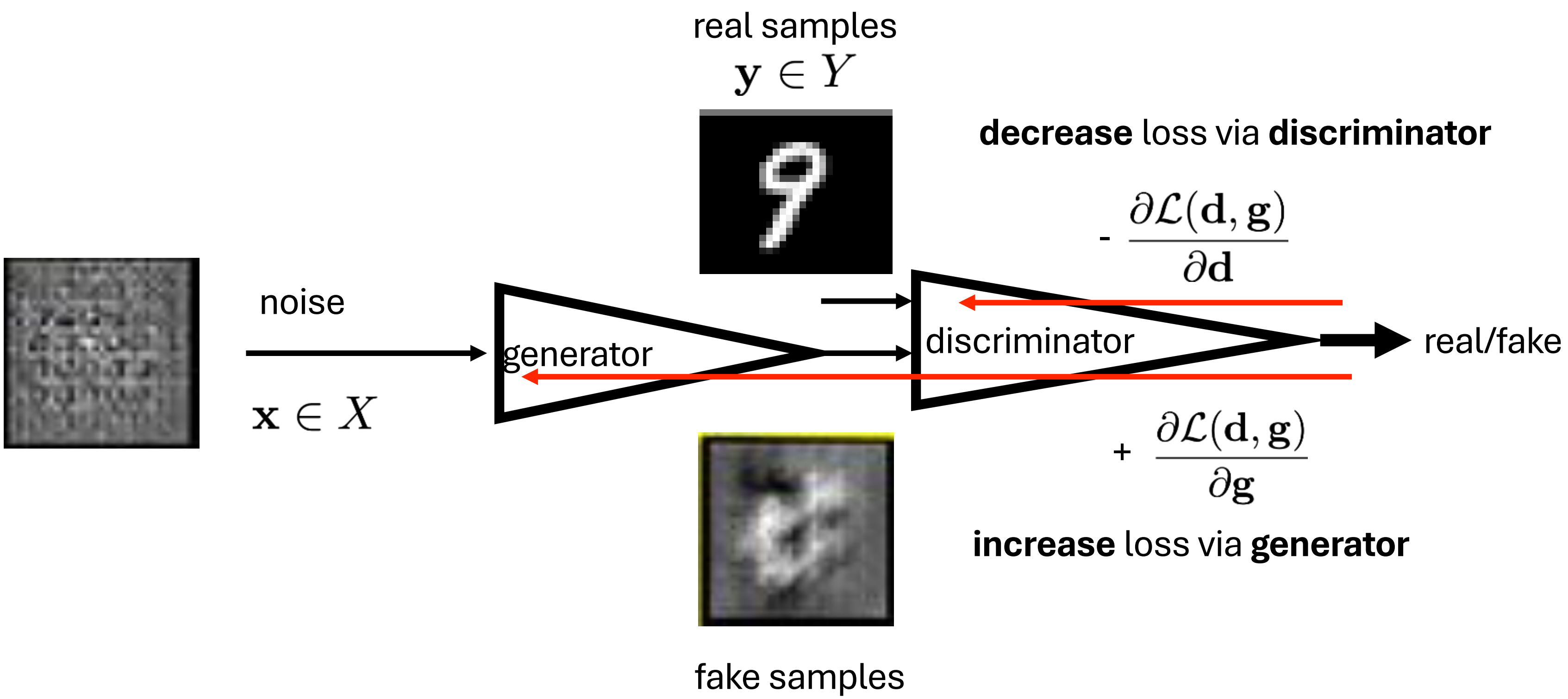
Generative Adversarial Networks (GANs)



Classification loss: $$\mathcal{L}(\mathbf{d}, \mathbf{g}) = \sum_{\mathbf{x} \in X} - \log(\mathbf{d}(\mathbf{g}(\mathbf{x}))) + \sum_{\mathbf{y} \in Y} - \log(1 - \mathbf{d}(\mathbf{y}))$$

Goodfellow, Ian J., et al. "Generative adversarial nets." NIPS 2014

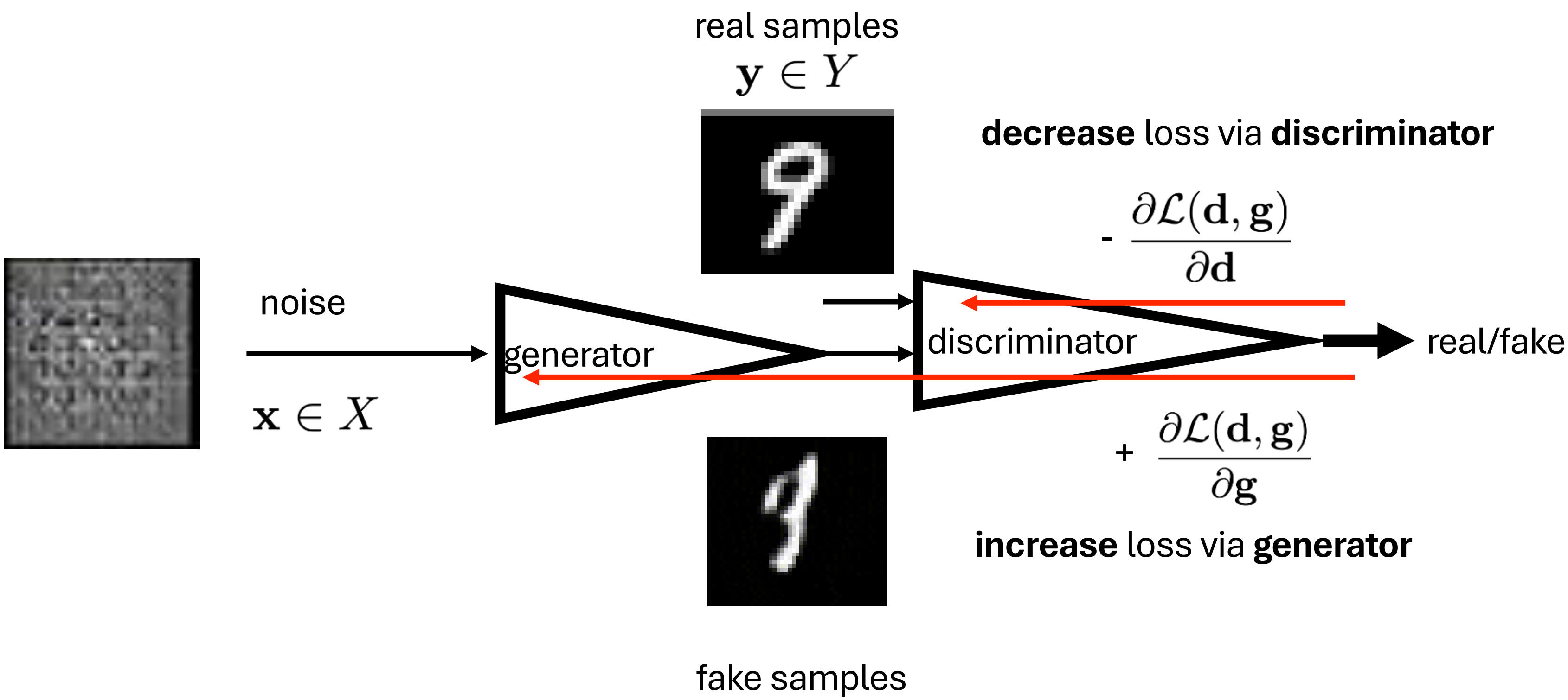
Generative Adversarial Networks (GANs)



Classification loss: $$\mathcal{L}(\mathbf{d},\mathbf{g}) = \sum_{\mathbf{x}\in X} -\log(\mathbf{d}(\mathbf{g}(\mathbf{x}))) + \sum_{\mathbf{y}\in Y} -\log(1-\mathbf{d}(\mathbf{y}))$$

Goodfellow, Ian J., et al. "Generative adversarial nets." NIPS 2014

Generative Adversarial Networks (GANs)

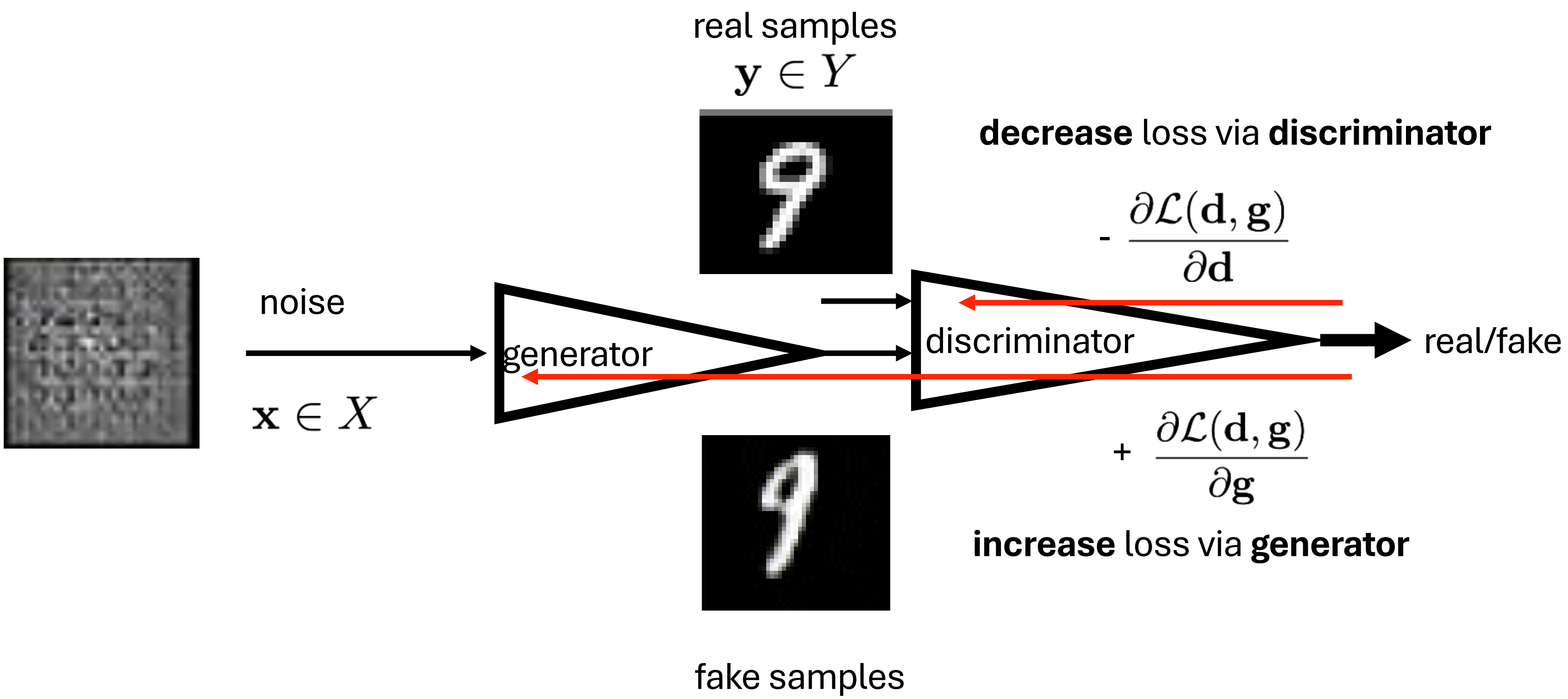


Classification loss:

$$\mathcal{L}(\mathbf{d}, \mathbf{g}) = \sum_{\mathbf{x} \in X} -\log(\mathbf{d}(\mathbf{g}(\mathbf{x}))) + \sum_{\mathbf{y} \in Y} -\log(1 - \mathbf{d}(\mathbf{y}))$$

Goodfellow, Ian J., et al. "Generative adversarial nets." NIPS 2014

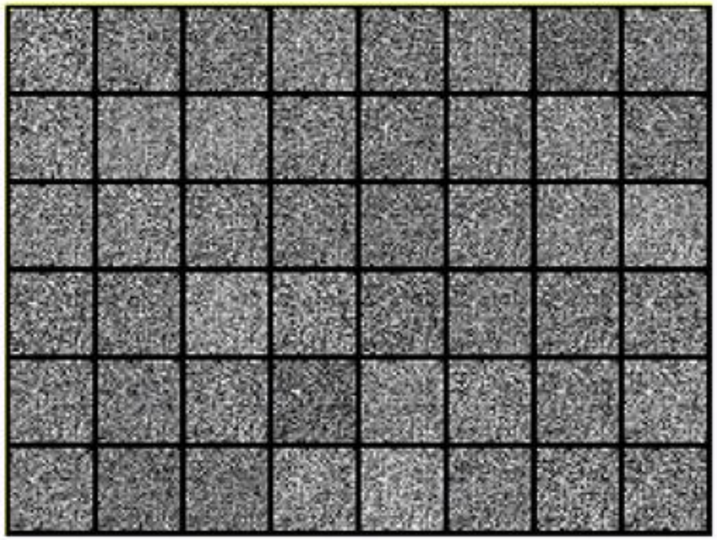
Generative Adversarial Nets

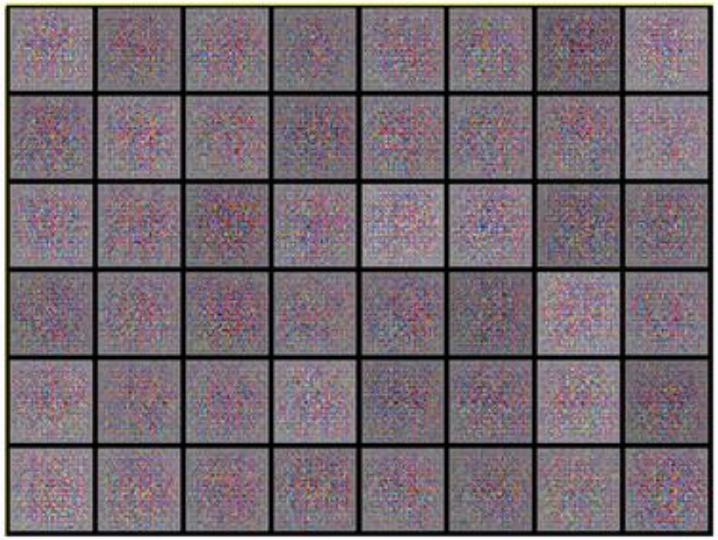


Classification loss: $$\mathcal{L}(\mathbf{d}, \mathbf{g}) = \sum_{\mathbf{x} \in X} -\log(\mathbf{d}(\mathbf{g}(\mathbf{x}))) + \sum_{\mathbf{y} \in Y} -\log(1 - \mathbf{d}(\mathbf{y}))$$

Goodfellow, Ian J., et al. "Generative adversarial nets." NIPS 2014

Generative Adversarial Networks (GANs)





Generative Adversarial Networks (GANs)

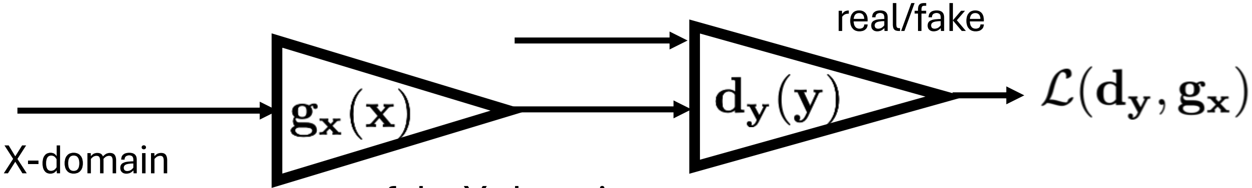
Gao, Hongchang, Jian Pei, and Heng Huang. "ProGAN: Network embedding via proximity generative adversarial network." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

Generative Adversarial Networks (GANs)

- CycleGAN

real Monet’s paining

real Y-domain samples $\mathbf{y} \in Y$



real X-domain samples $\mathbf{x} \in X$

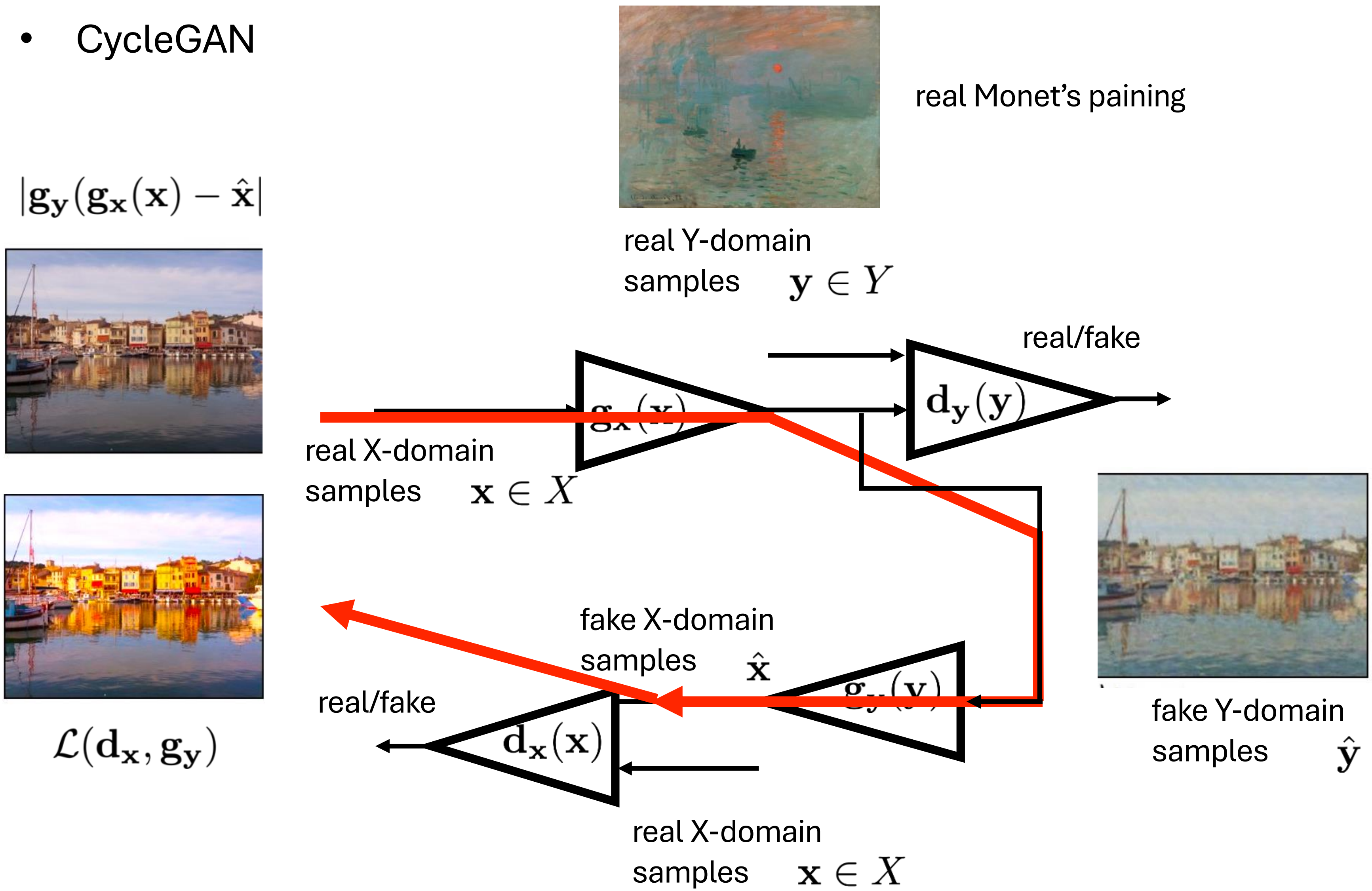
fake Y-domain samples $\hat{\mathbf{y}}$

fake Monet’s paining

Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.“ ICCV 2017

Generative Adversarial Networks (GANs)

- CycleGAN



Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV 2017

Generative Adversarial Networks (GANs)

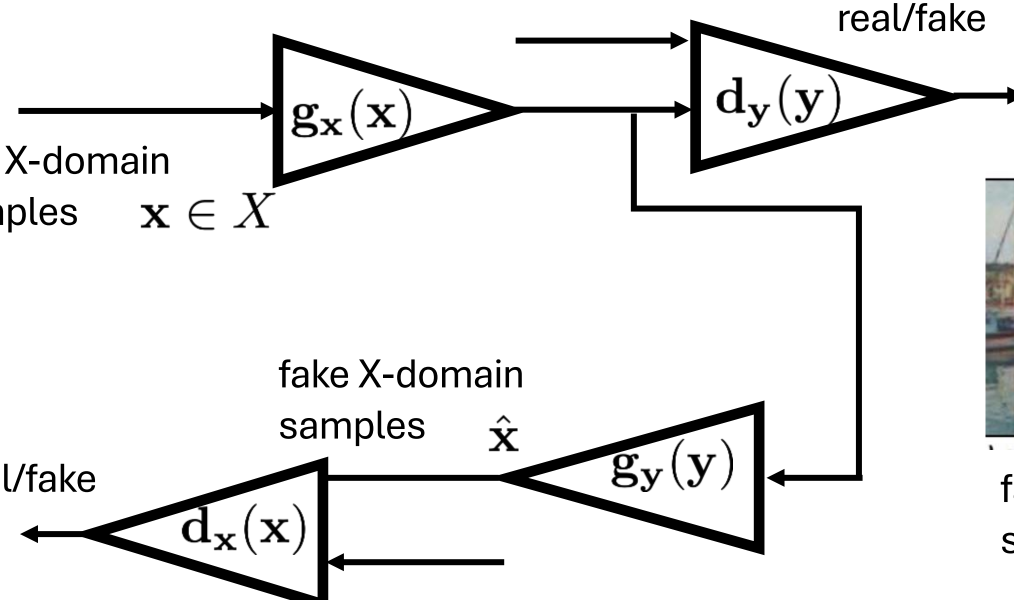
- CycleGAN

$$\mathcal{L}_{GAN}(\mathbf{d_x}, \mathbf{d_y}, \mathbf{g_x}, \mathbf{g_y}) = \mathcal{L}(\mathbf{d_x}, \mathbf{g_y}) + \mathcal{L}(\mathbf{d_y}, \mathbf{g_x}) + |\mathbf{g_y}(\mathbf{g_x}(\mathbf{x}) - \hat{\mathbf{x}}|$$

real Y-domain samples $\mathbf{y} \in Y$

real/fake

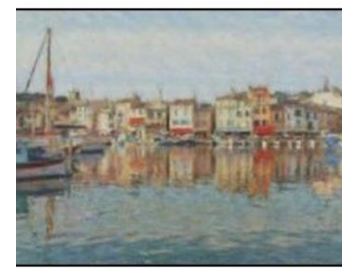
real X-domain samples $\mathbf{x} \in X$



fake X-domain samples $\hat{\mathbf{x}}$

real/fake

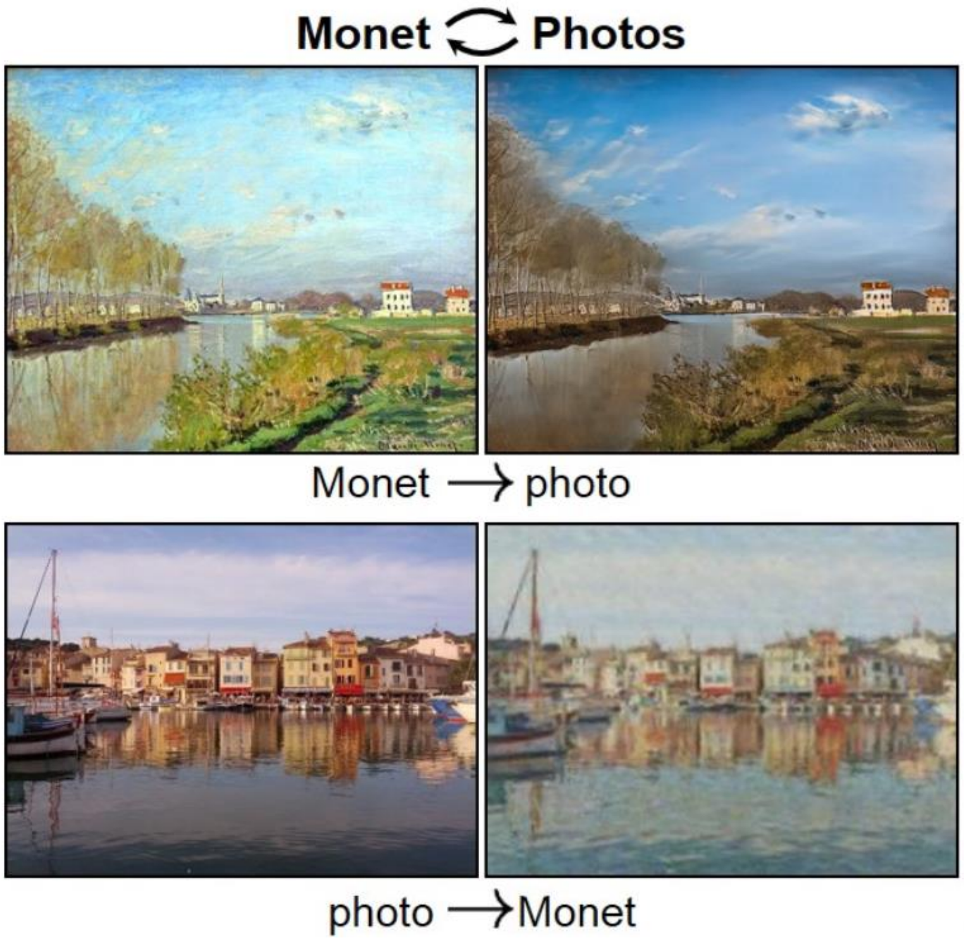
real X-domain samples $\mathbf{x} \in X$



fake Y-domain samples $\hat{\mathbf{y}}$

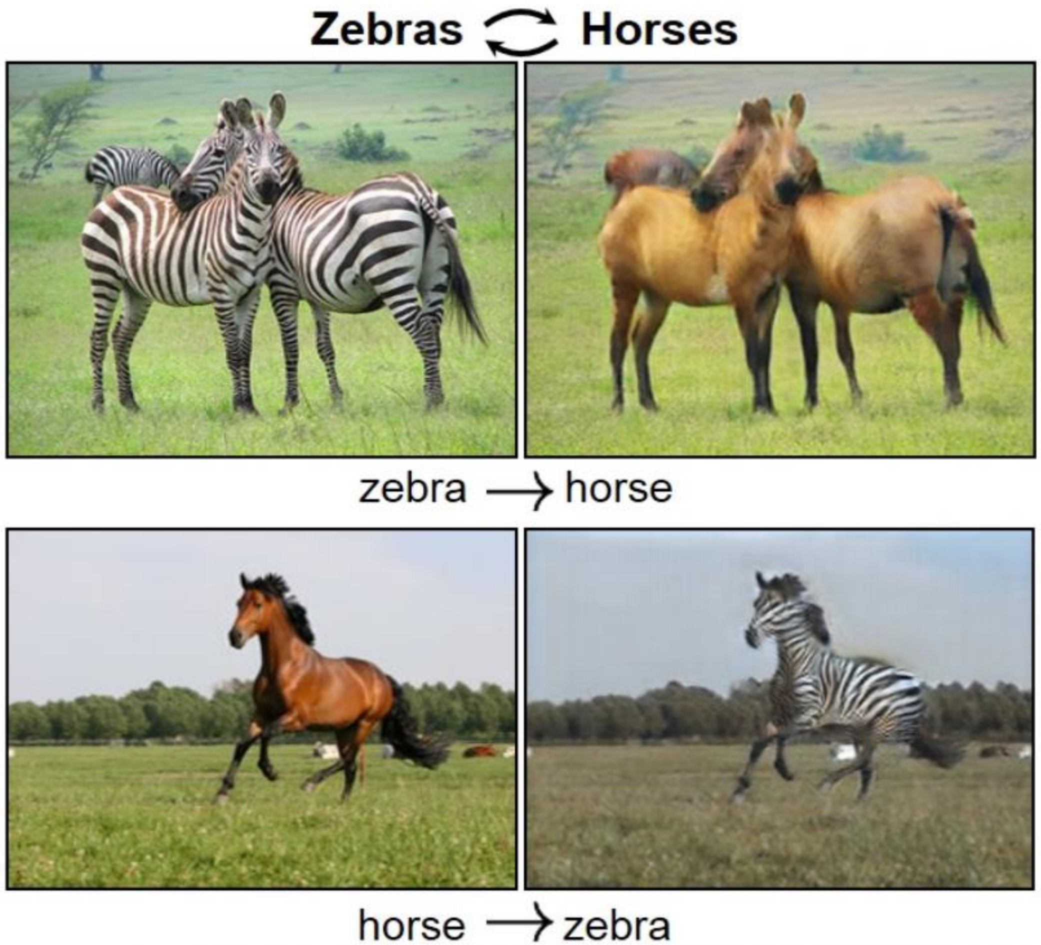
Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV 2017

Generative Adversarial Networks (GANs)



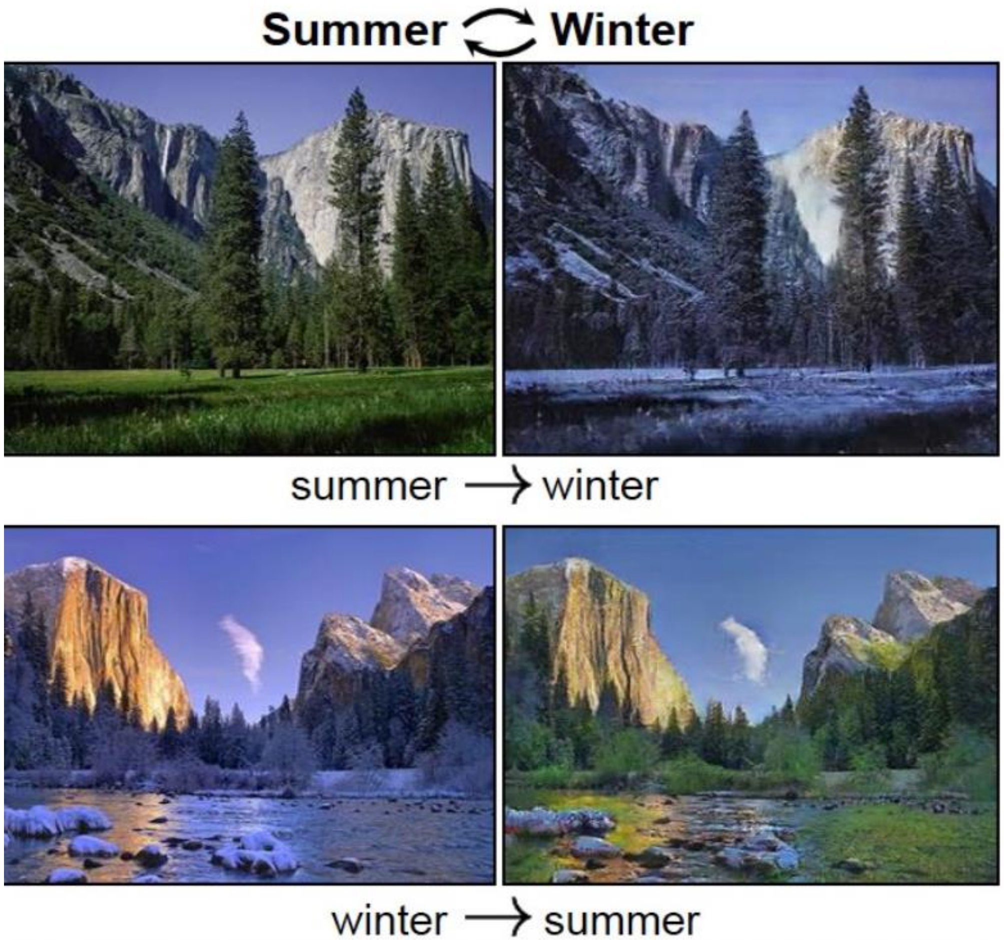
Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV 2017

Generative Adversarial Networks (GANs)



Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV 2017

Generative Adversarial Networks (GANs)



Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV 2017

Diffusion models

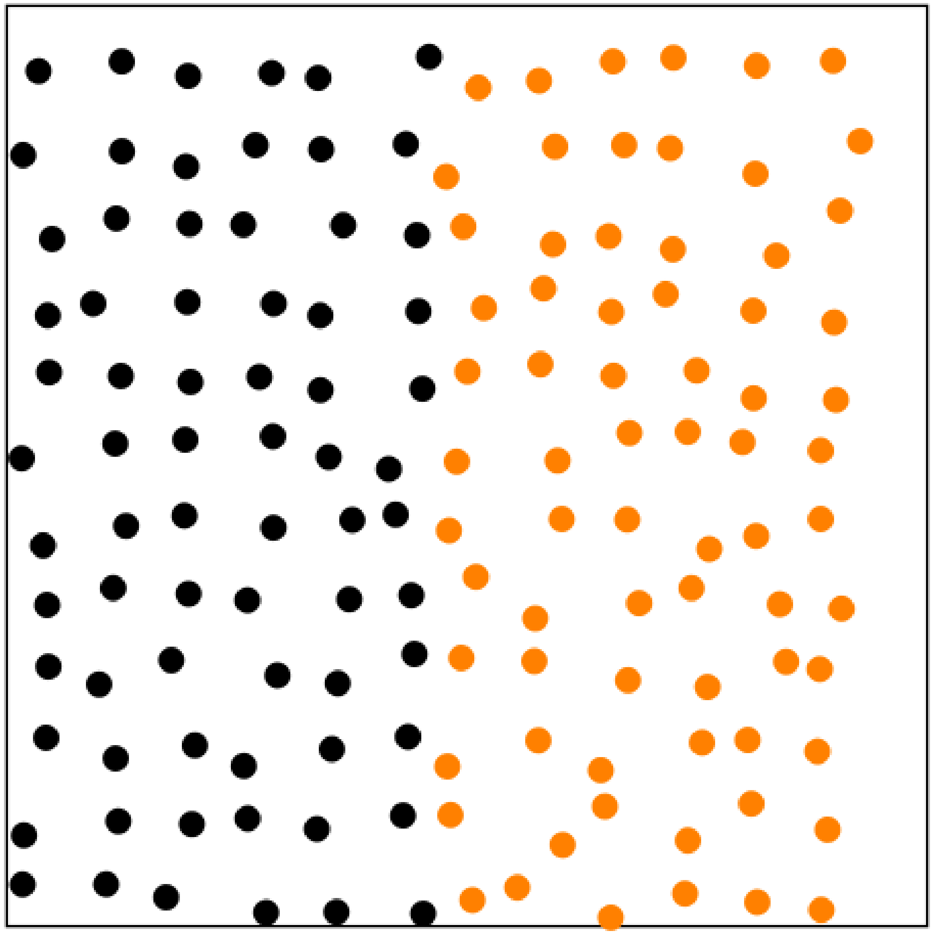
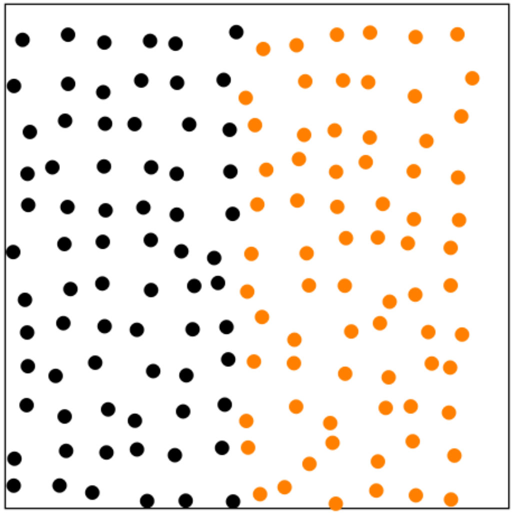
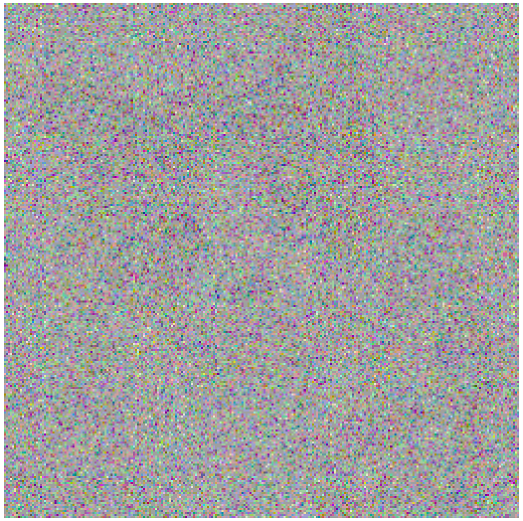

Particle diffusion


Image diffusion

Is the process reversible?

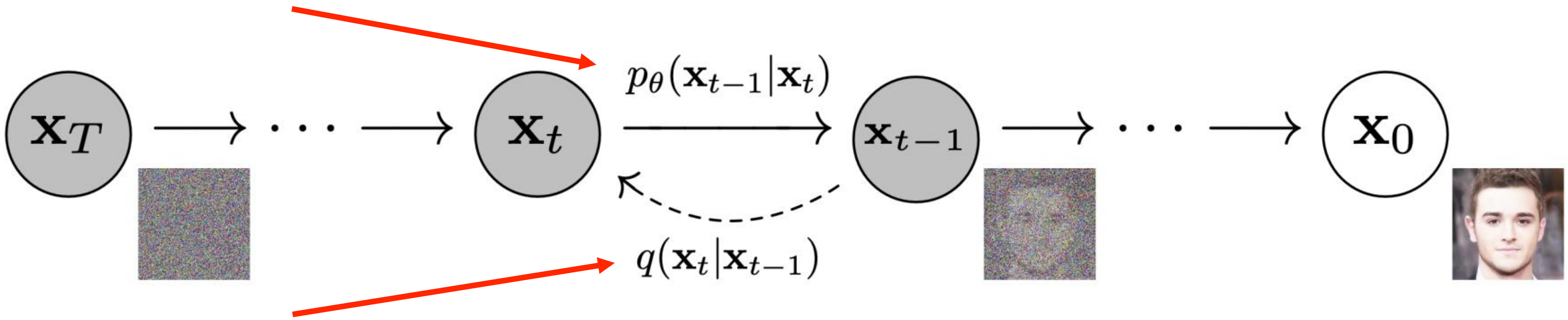
Diffusion models

	Particle diffusion	Image diffusion
		
Stochastic	✓	✓
Entropy increases	✓	✓
Partially Reversible	$\frac{\partial u}{\partial t} = D\nabla^2 u$	$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon$

Diffusion models

- Train a network to reverse the diffusion process

Reverse of the diffusion process to generate original data from the noise



Markov chain of diffusion steps in which we slowly and randomly add noise t

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon$$

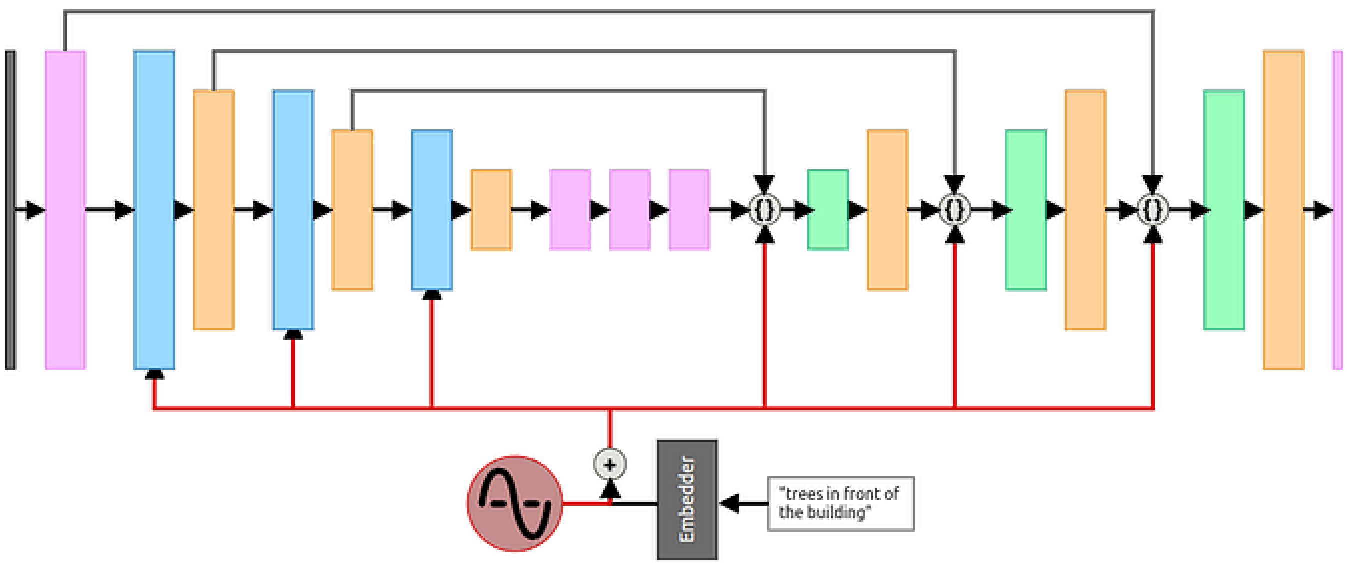
If noise is small the backward step has also “almost” gaussian distribution

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

$\rightarrow$ train de-noising networks through L2-norm with the original image

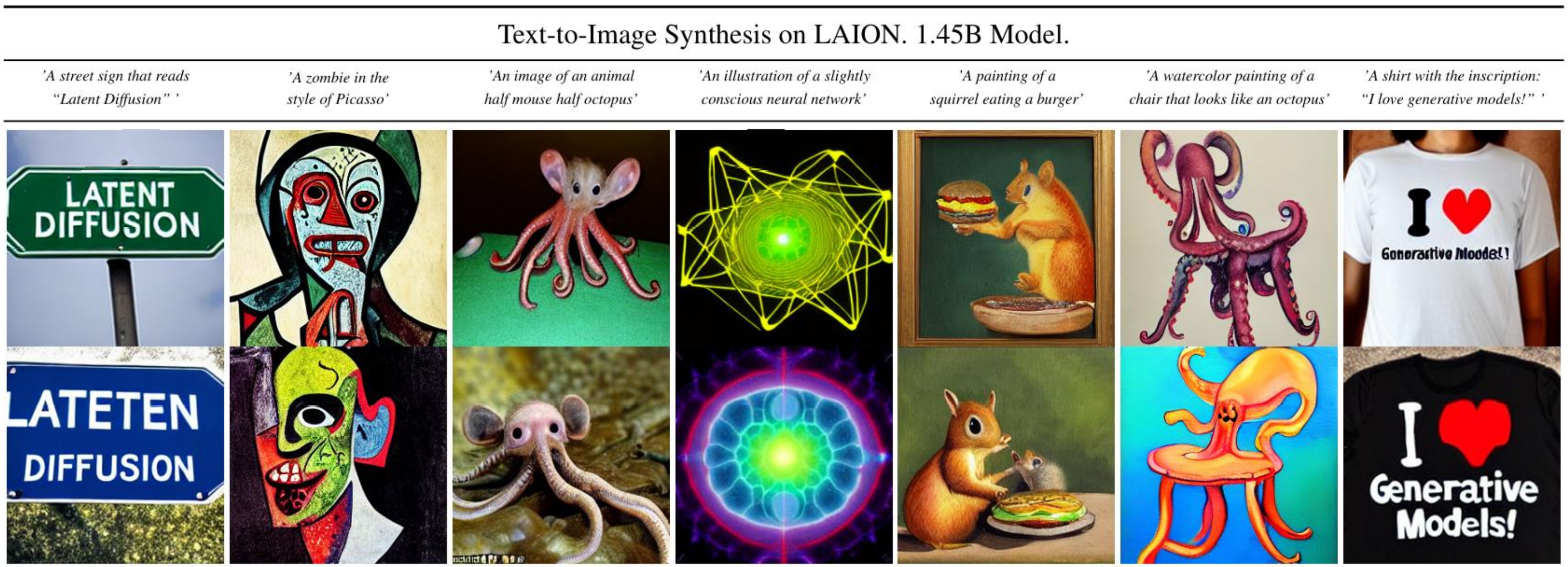
Diffusion models

- Train a network to reverse the diffusion process
- An adapted U-Net architecture for denoising allows customizing the embedding
- Pros
  - High-quality imagery
- Cons
  - **<u>Slow</u>** (the generation process is iterative, thousands of steps needed)



Diffusion models

- Stable diffusion



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." CVPR 2022

Competencies gained for the test

- Semantic segmentation
- Object detection, evaluation (Precision, Recall, AP)
- Generative models (GANs, Diffusion models)