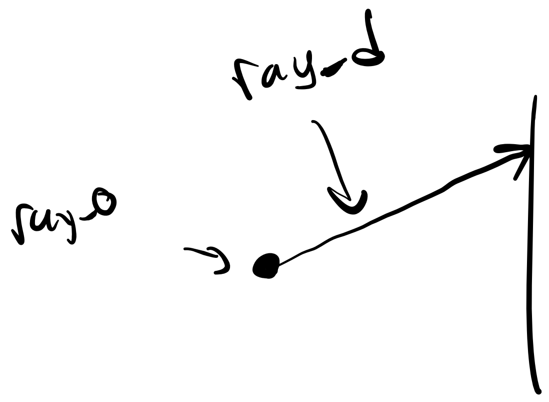
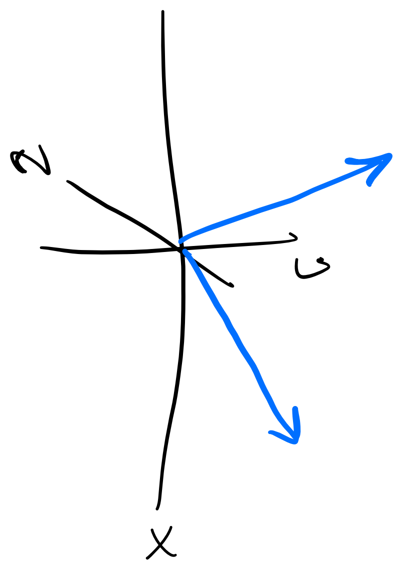
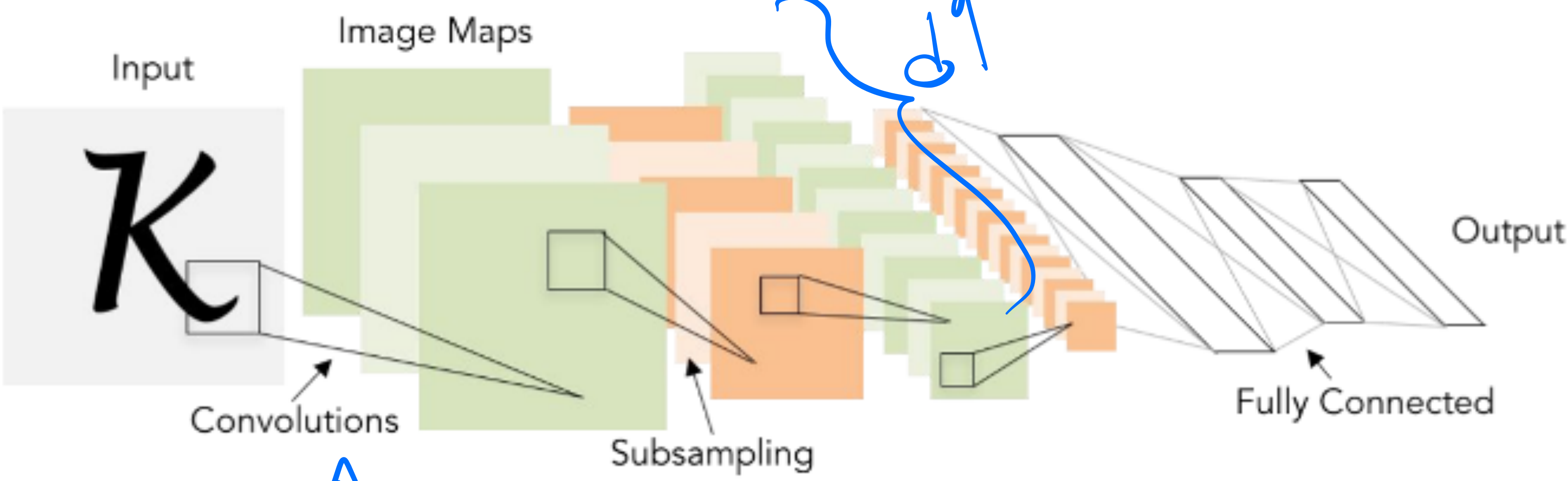
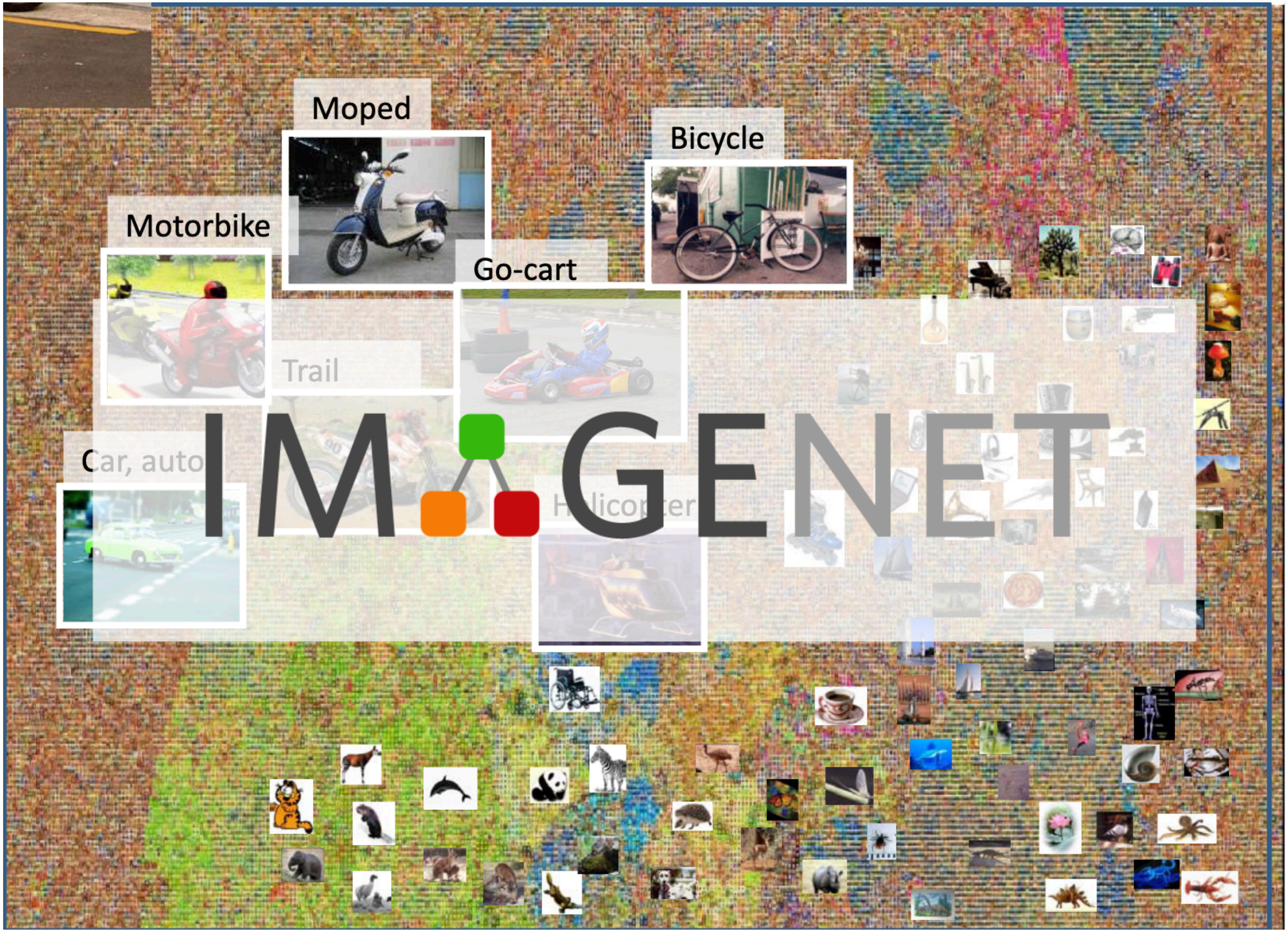

ray_d
ray_o

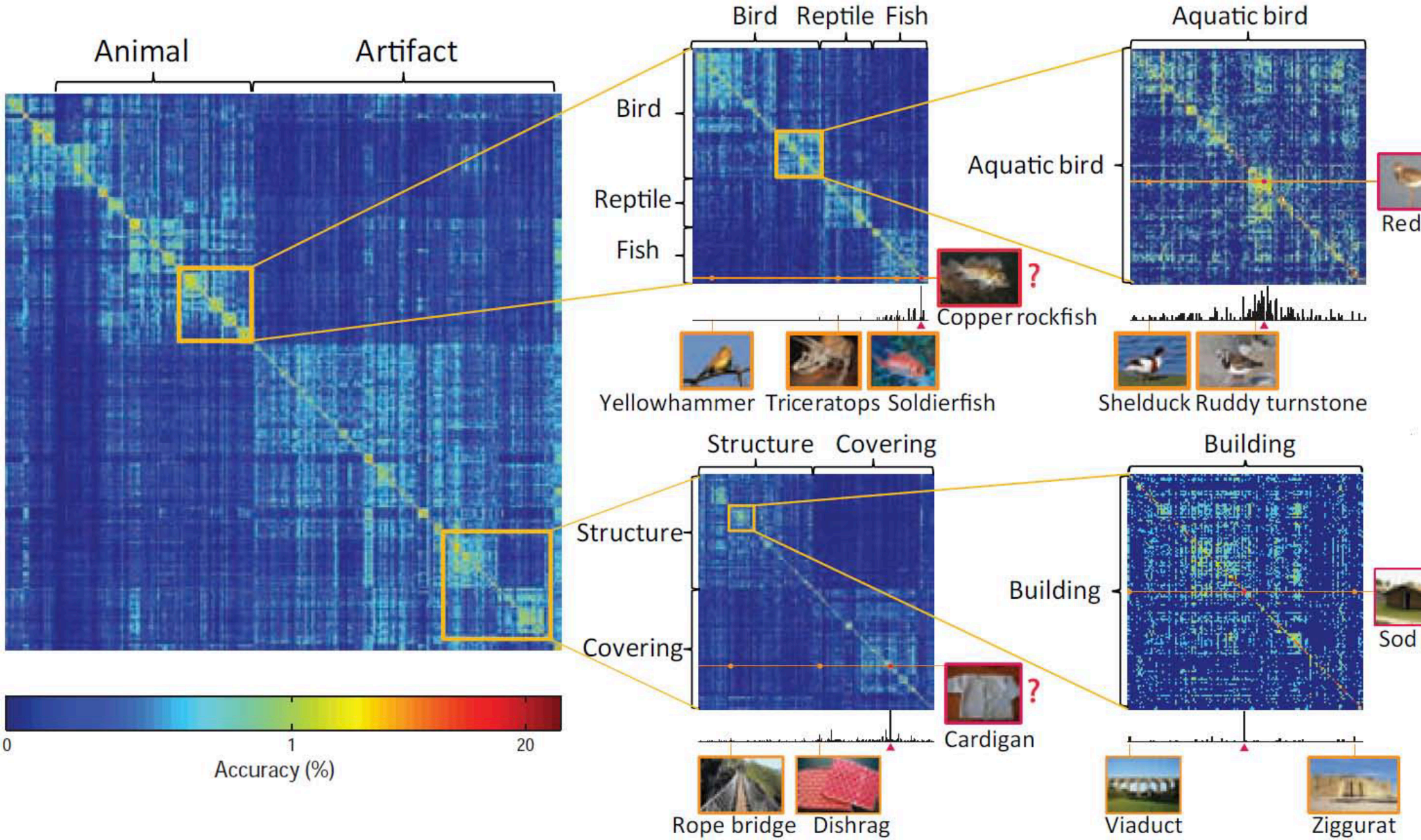

z
y
x

Review: LeNet-5

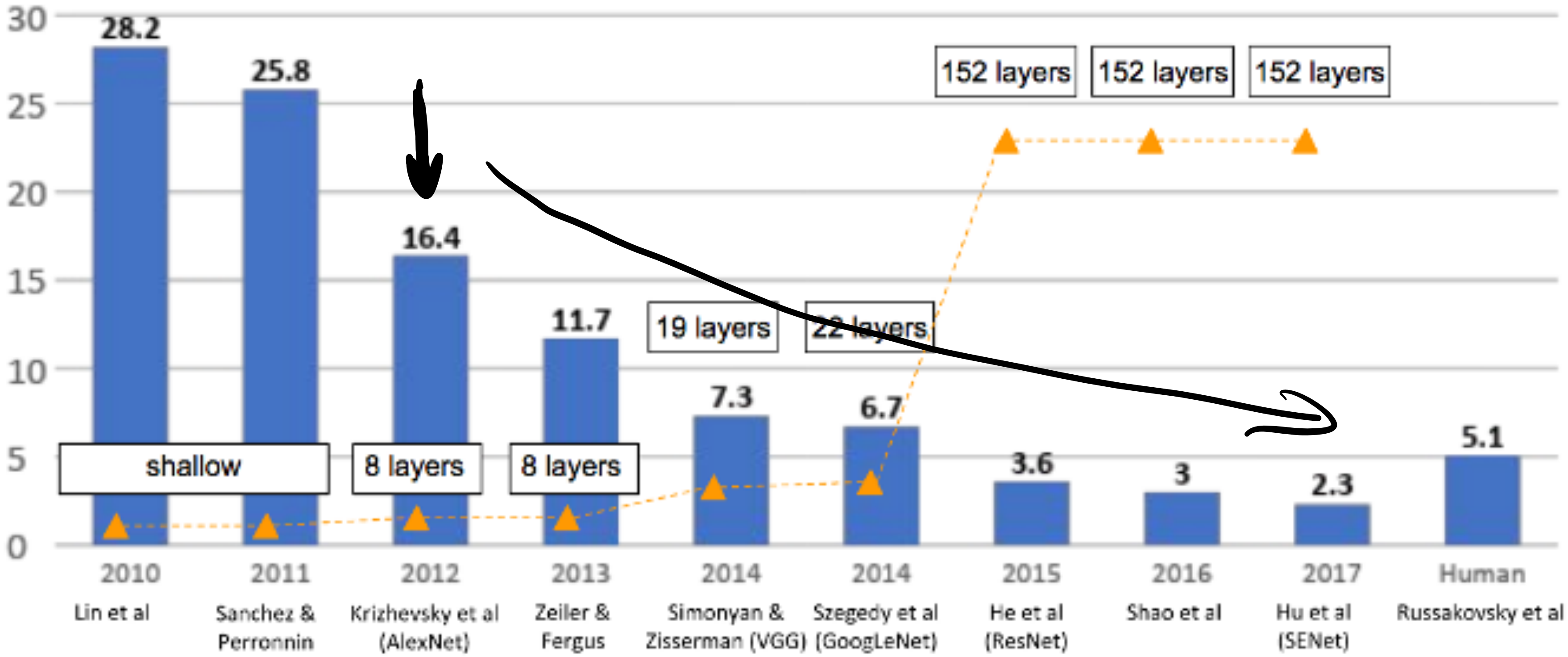
[LeCun et al., 1998]




Moped
Bicycle
Motorbike
Go-cart
Trail
Car, auto
Helicopter
IMAGENET


Animal
Artifact
0
1
20
Accuracy (%)
Bird
Reptile
Fish
Bird
Reptile
Fish
Copper rockfish
?
Yellowhammer
Triceratops
Soldierfish
Aquatic bird
Aquatic bird
Red
Shelduck
Ruddy turnstone
Structure
Covering
Structure
Covering
Cardigan
?
Rope bridge
Dishrag
Building
Building
Sod
Viaduct
Ziggurat

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

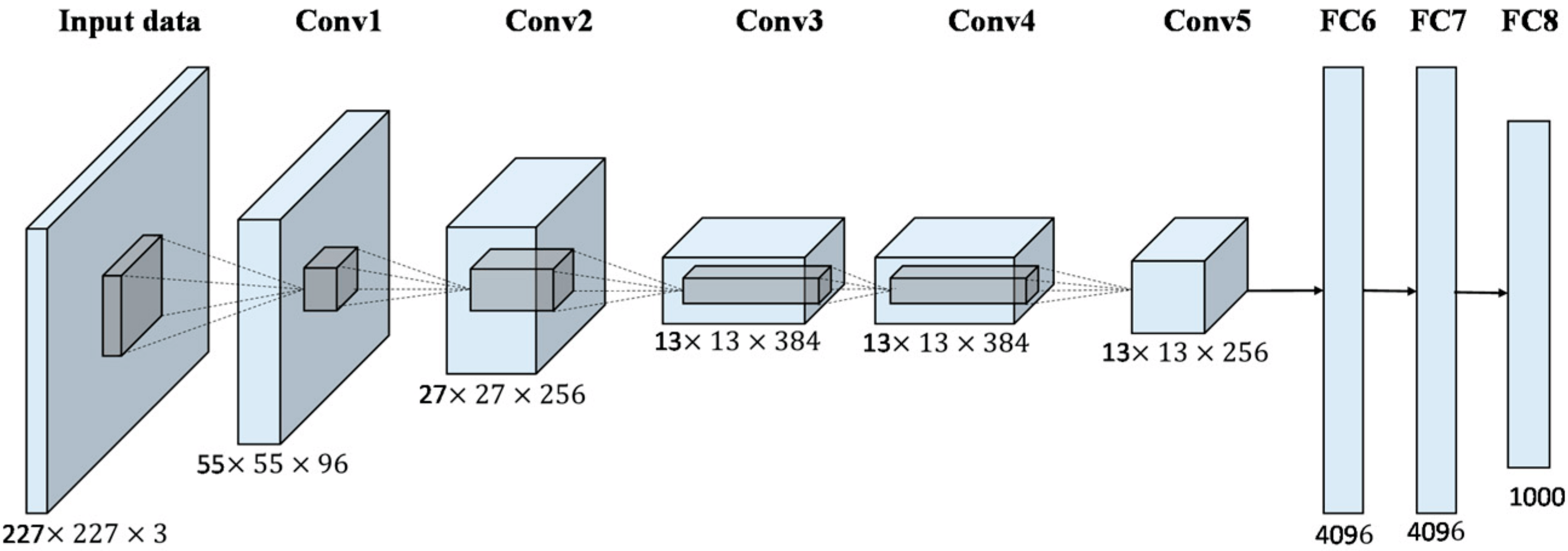


WE NEED TO GO
DEEPER

Alex Net

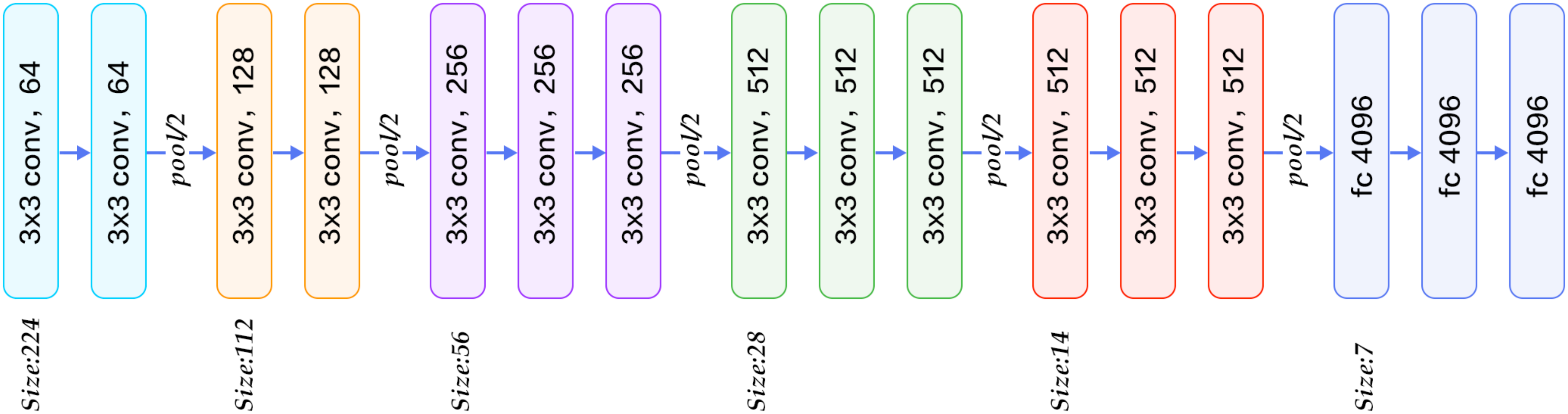
11x11 x3

5x5x96



WE NEED TO GO
DEEPER

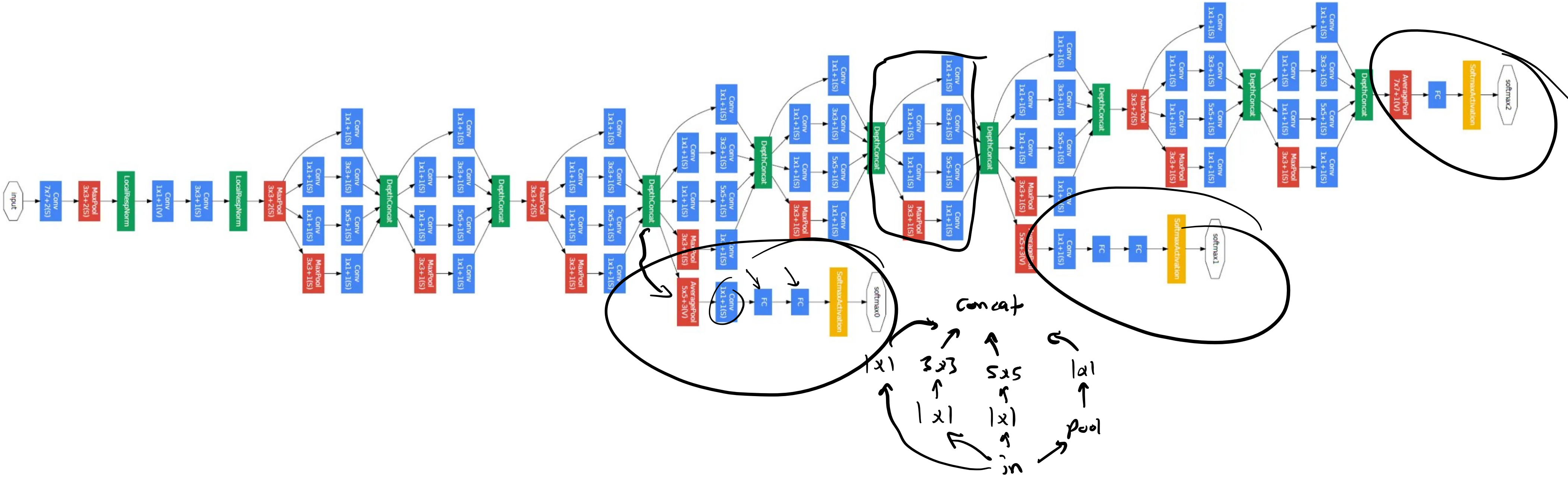
VGG (2014) VGG19



WE NEED TO GO DEEPER

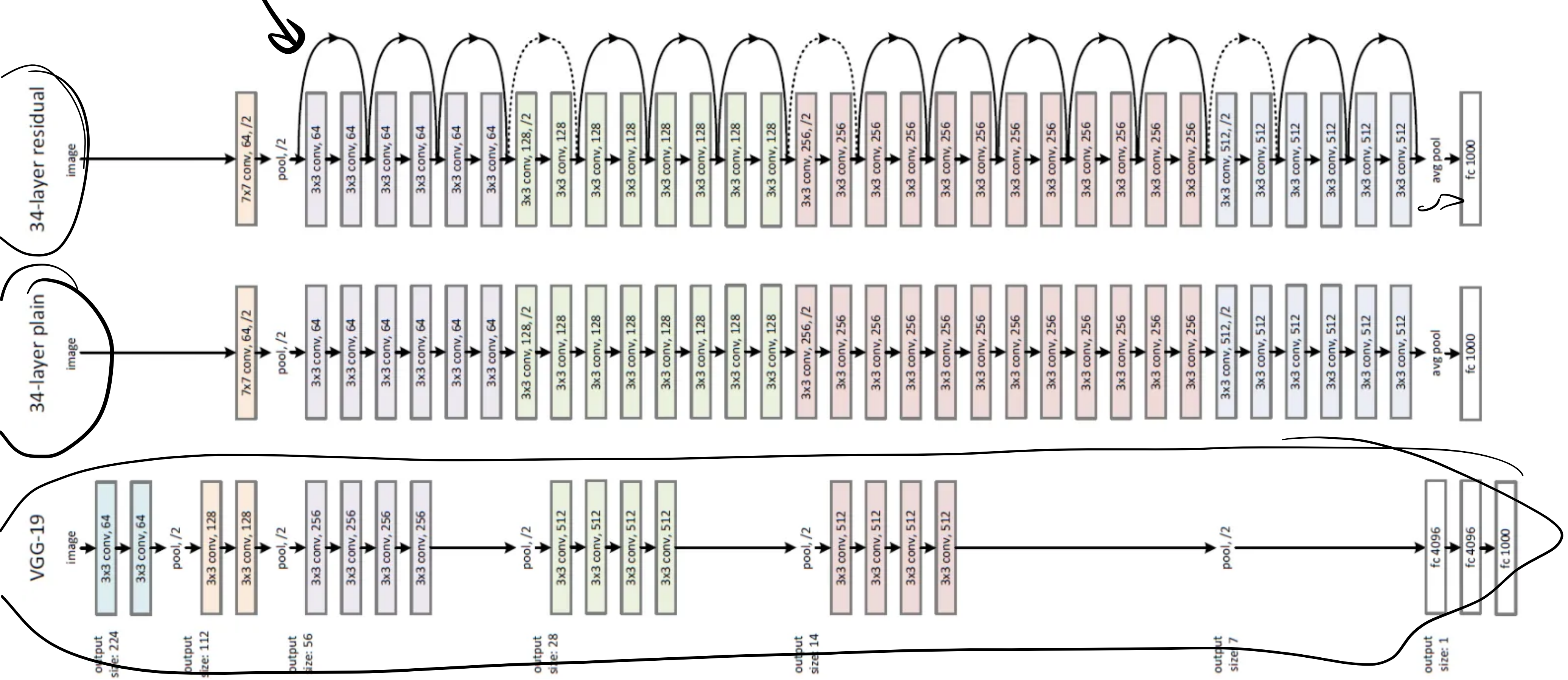
GoogLeNet (2014)

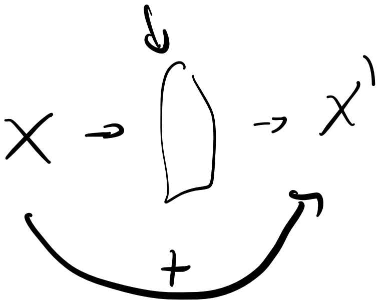
Inception "network-in-network"

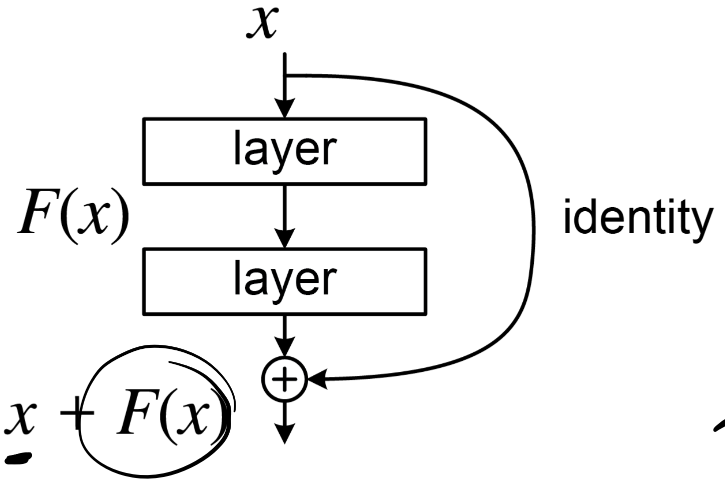


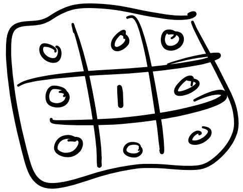
AFP

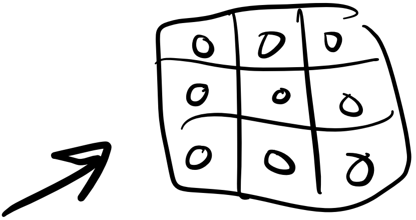
WE NEED TO GO
DEEPER


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000
34-layer plain
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000
VGG-19
image
3x3 conv, 64
pool, /2
3x3 conv, 128
3x3 conv, 256
3x3 conv, 512
fc 4096
fc 1000
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1


X
X'
+


x
layer
layer
F(x)
identity
x + F(x)

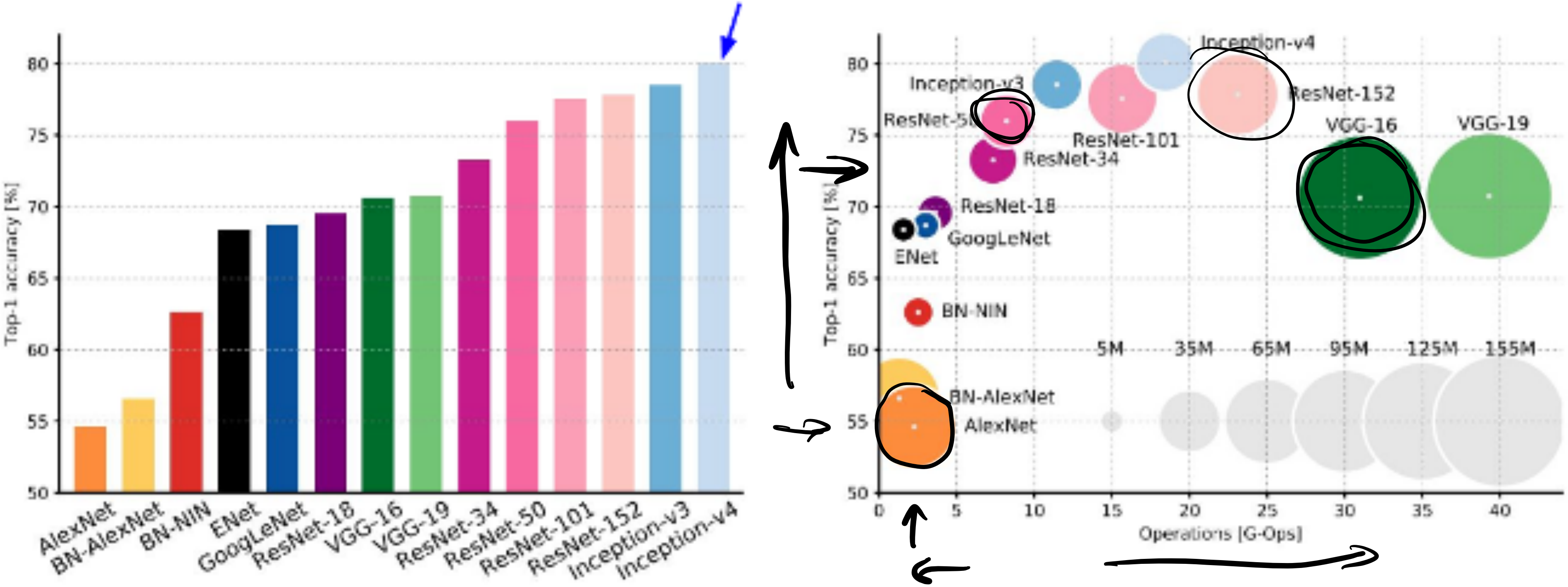

0 0 0
0 1 0
0 0 0


0 0 0
0 0 0
0 0 0

Do we?

Comparing complexity...

Inception-v4: Resnet + Inception!

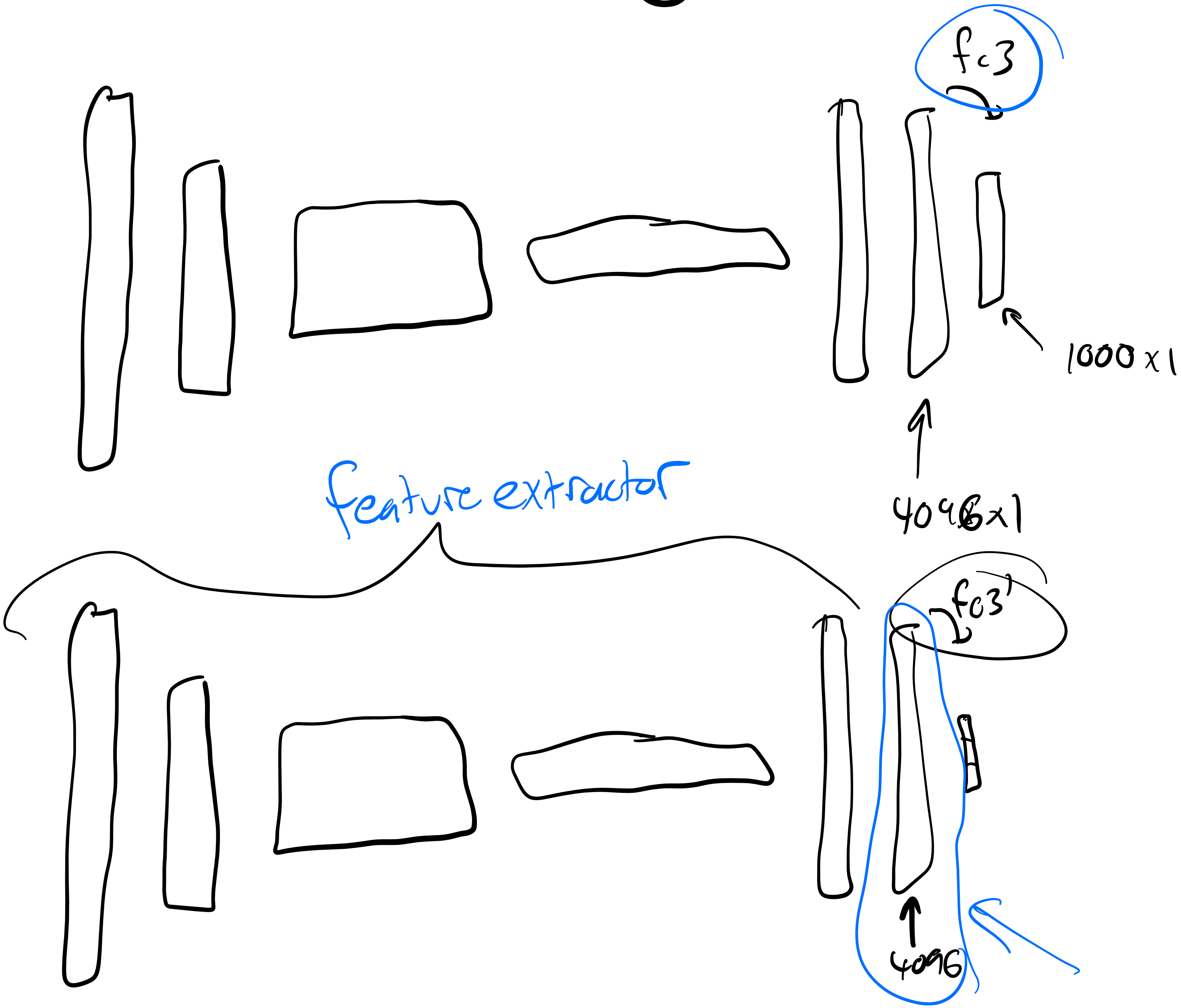


An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Figures copyright Alfredo Canziani, Adam Paszke, Eugenio Culurciello, 2017. Reproduced with permission.

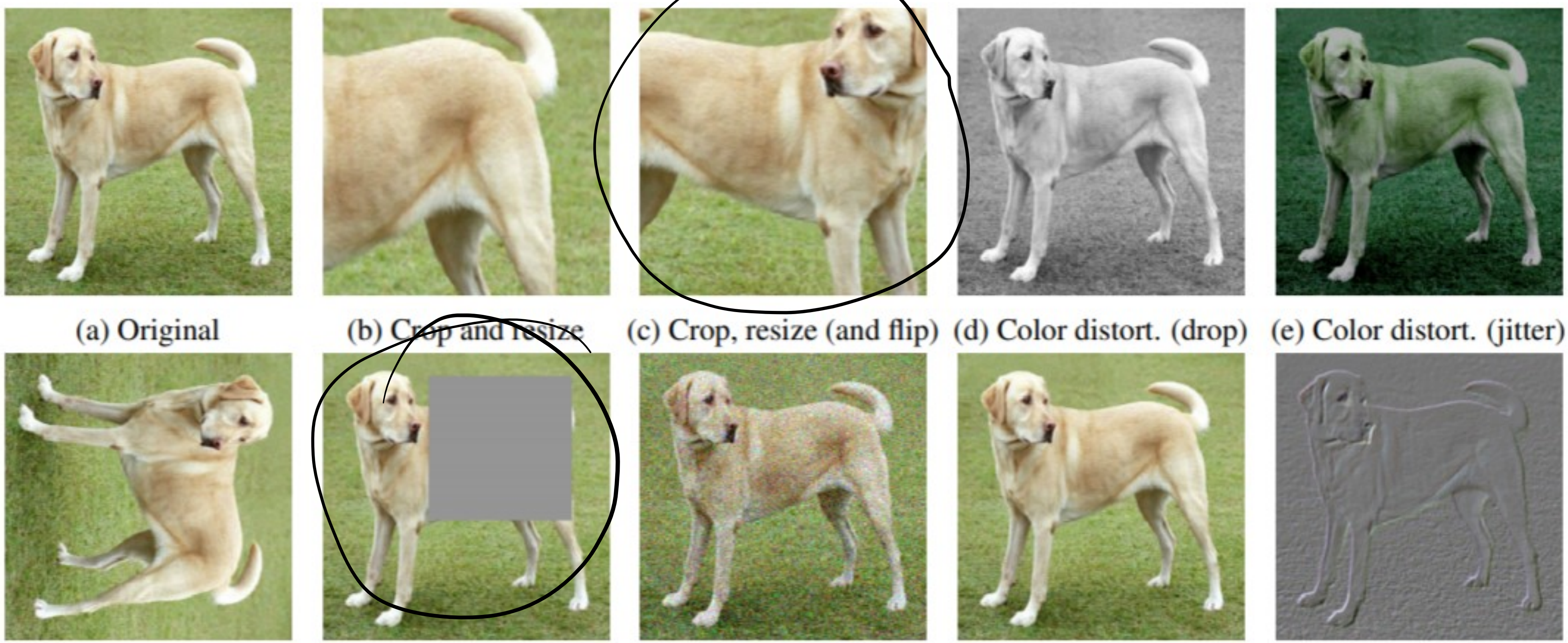
Okay but the data…

Transfer Learning / finetuning



Unsupervised / self-supervised learning case study: SimCLR

A Simple Framework for Contrastive Learning of Visual Representations

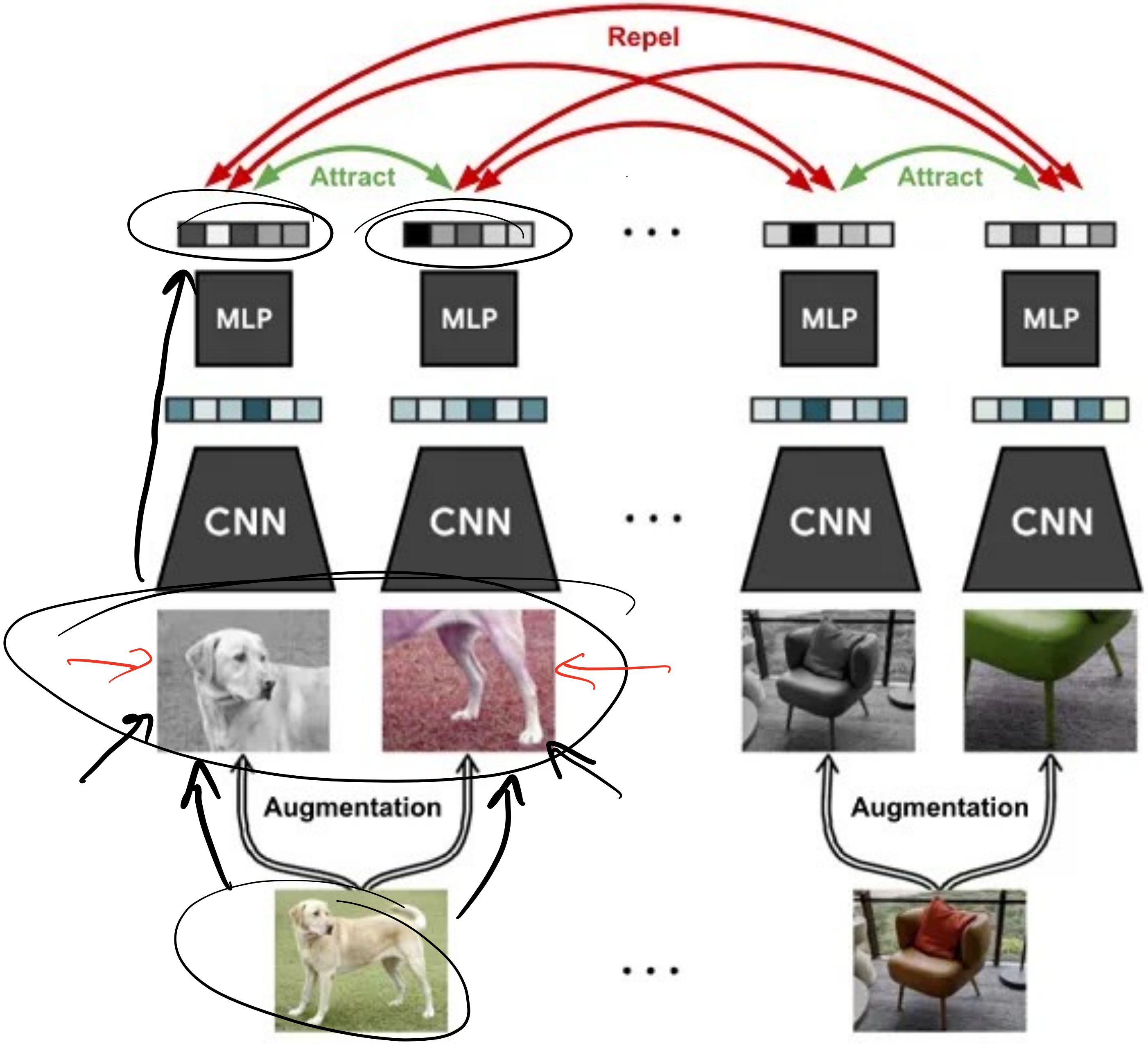


(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

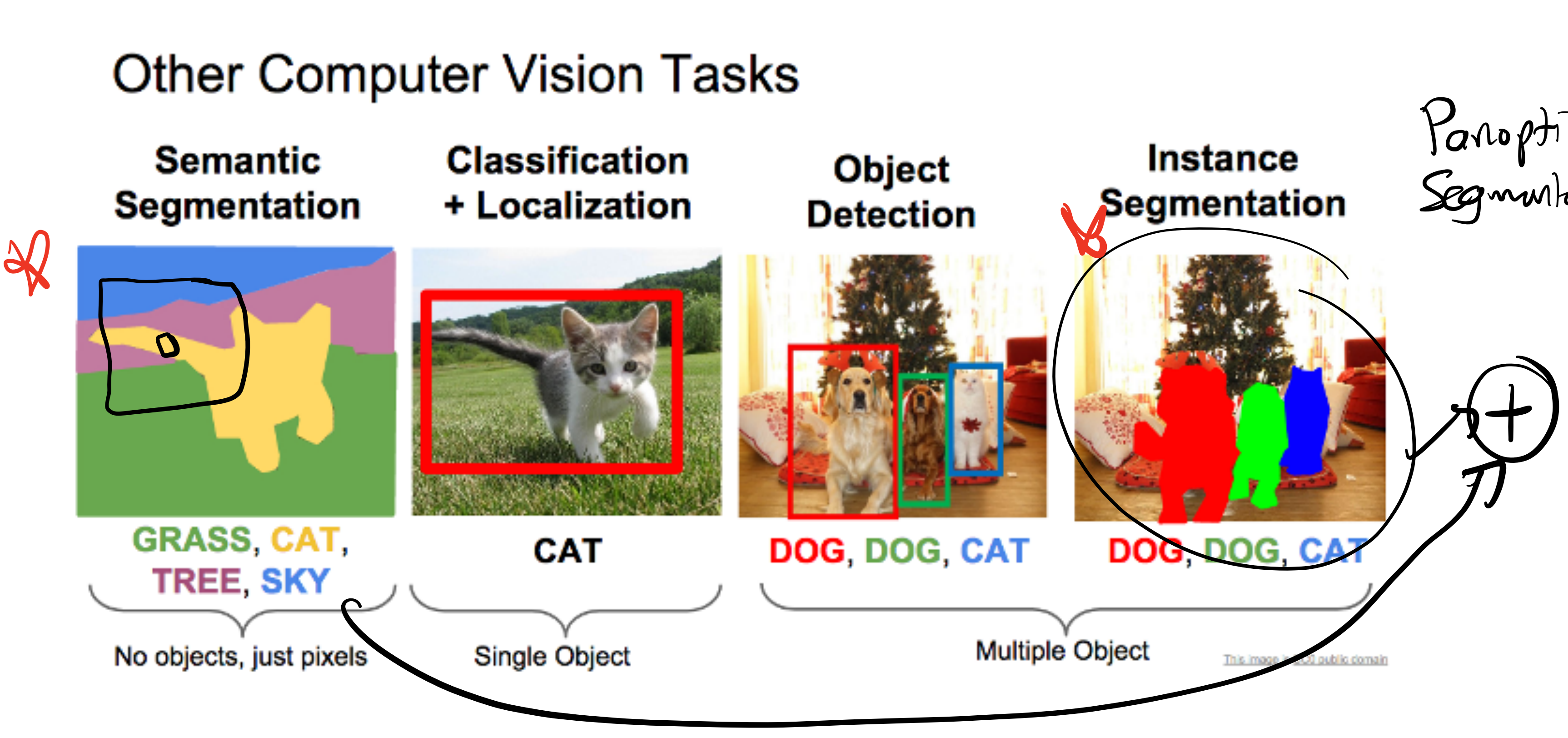
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize), color distortion,* and *Gaussian blur.* (Original image cc-by: Von.grzanka)

Unsupervised / self-supervised learning case study: SimCLR



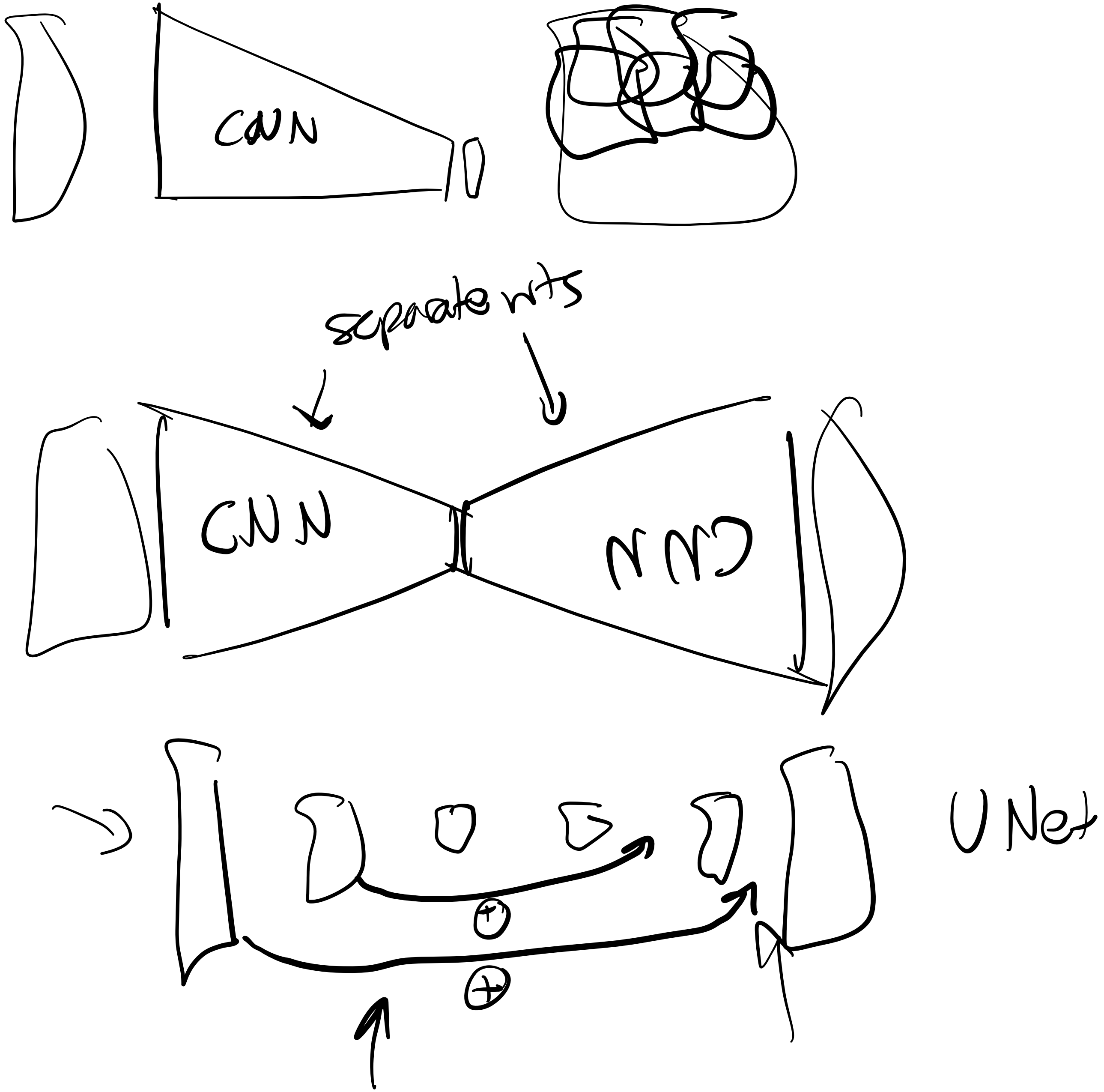
What about not image recognition?



Other Computer Vision Tasks

This image is CC0 public domain

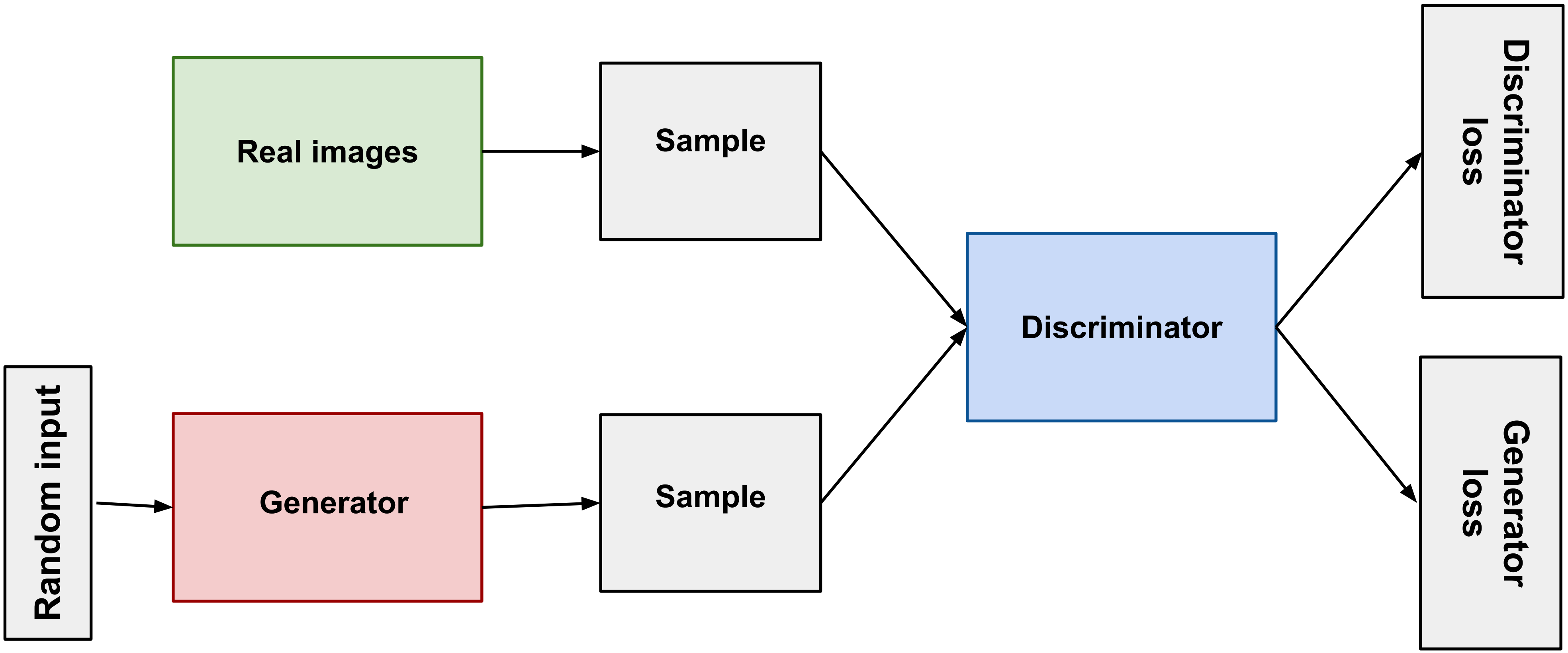
Panoptic Segmentation


CNN
separate wts
CNN
CNN
UNet
+
+

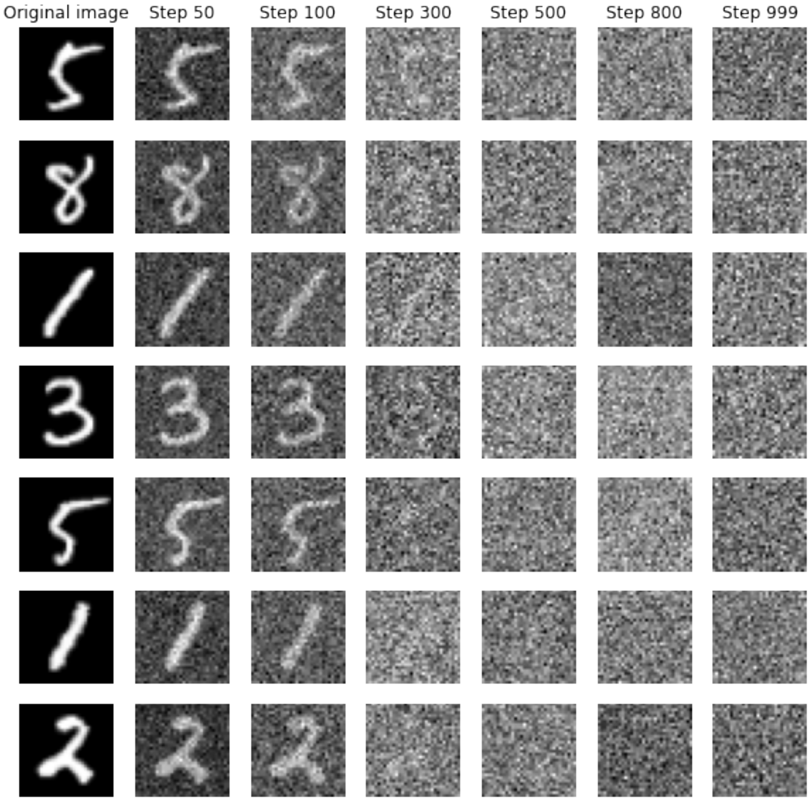
(Sharp?) left turn:
Embeddings, Manifold Learning, and Autoencoders

Generative Modeling

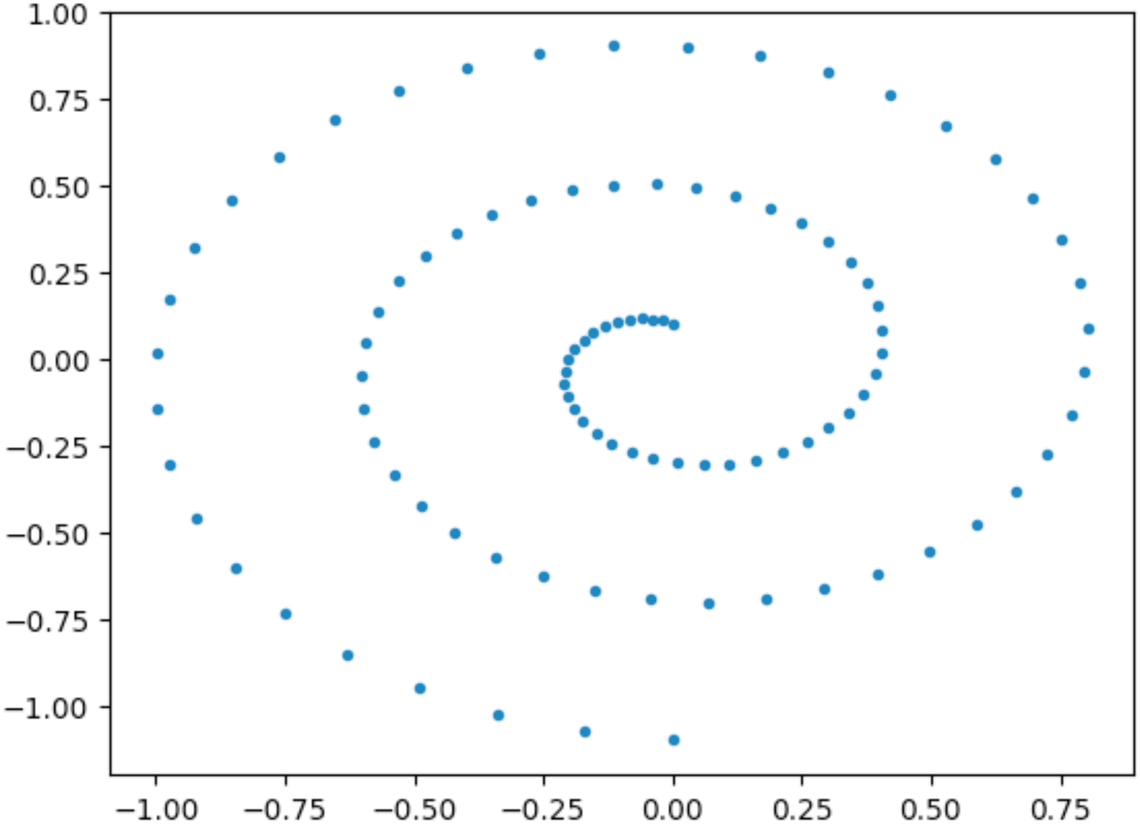
Generative Adversarial Networks

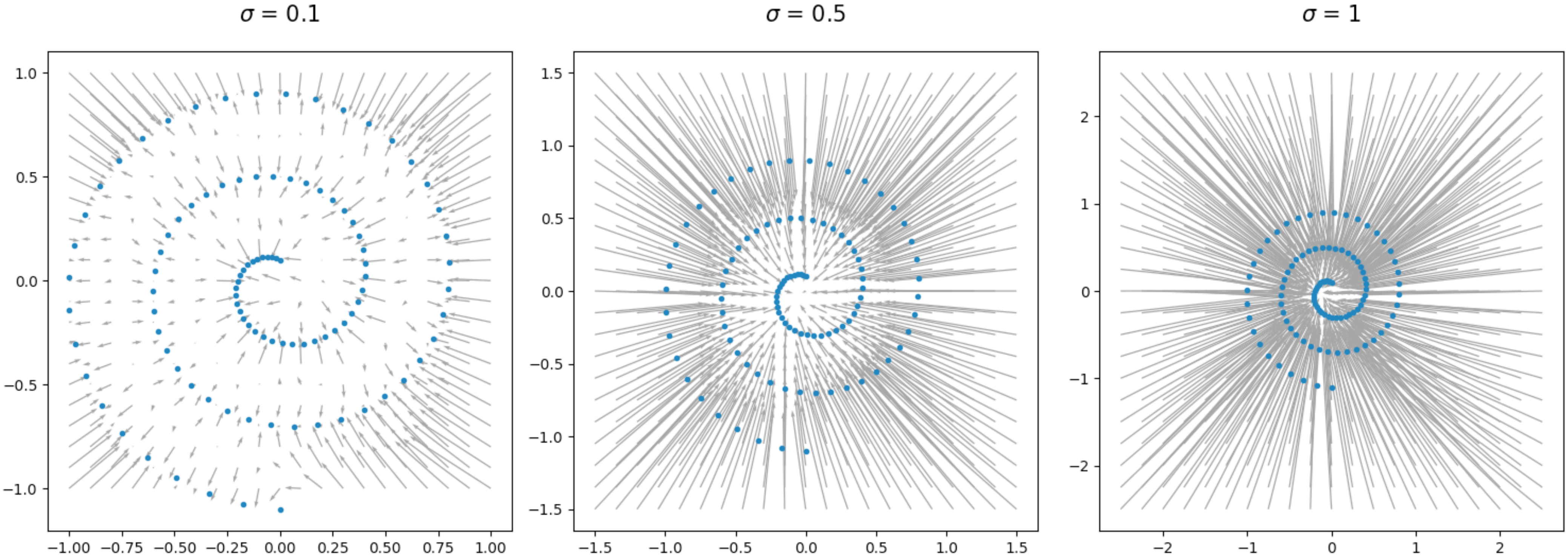


Diffusion Models



Some other good visuals: https://www.chenyang.co/diffusion.html


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75

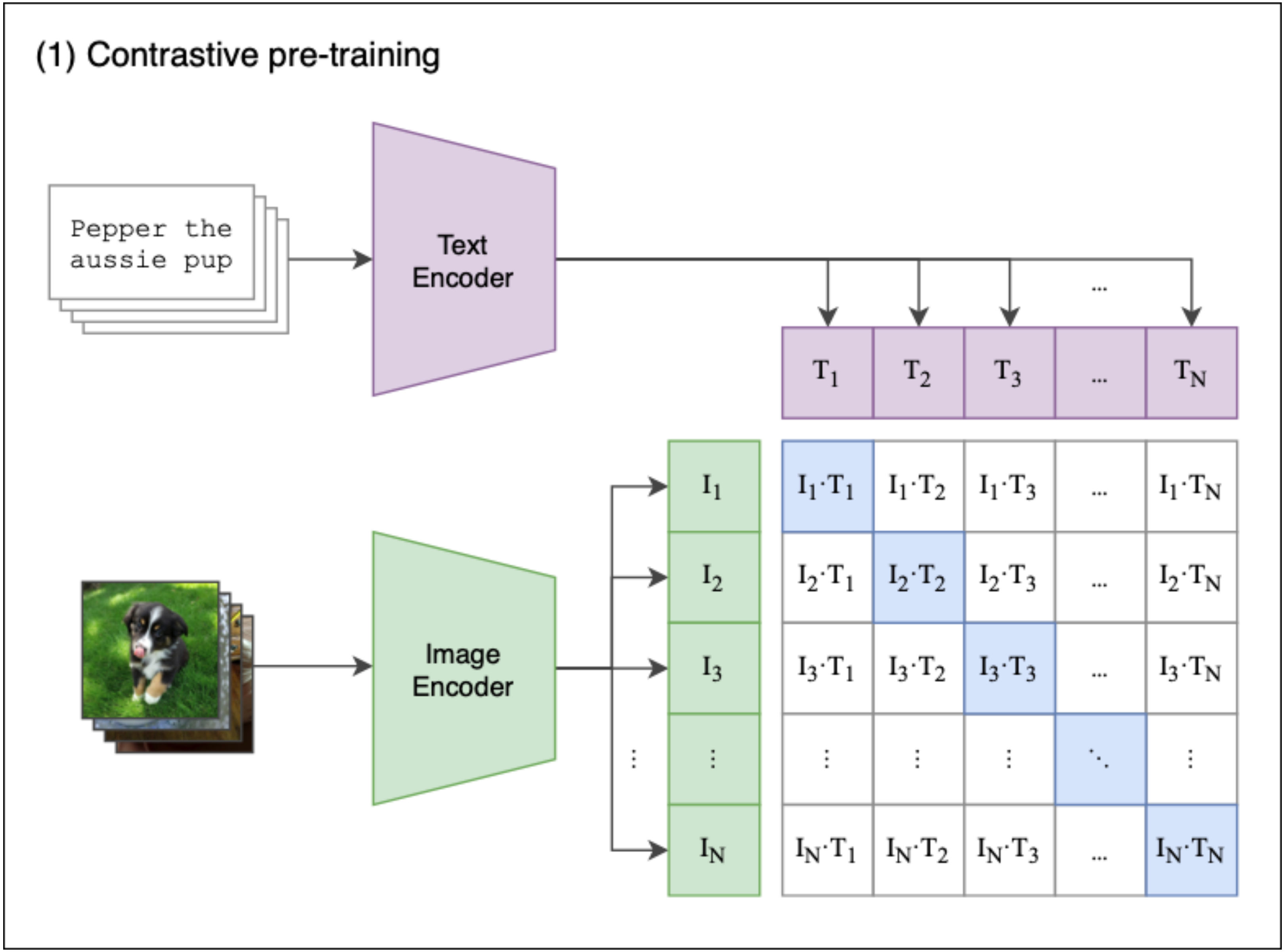

$\sigma = 0.1$
$\sigma = 0.5$
$\sigma = 1$
1.0
0.5
0.0
−0.5
−1.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2
1
0
−1
−2
−2
−1
0
1
2

© JIAN WEI

Stable Diffusion
(without the text-conditioning)

Vision and Language

Case study: CLIP



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

unCLIP aka DALL-E 2

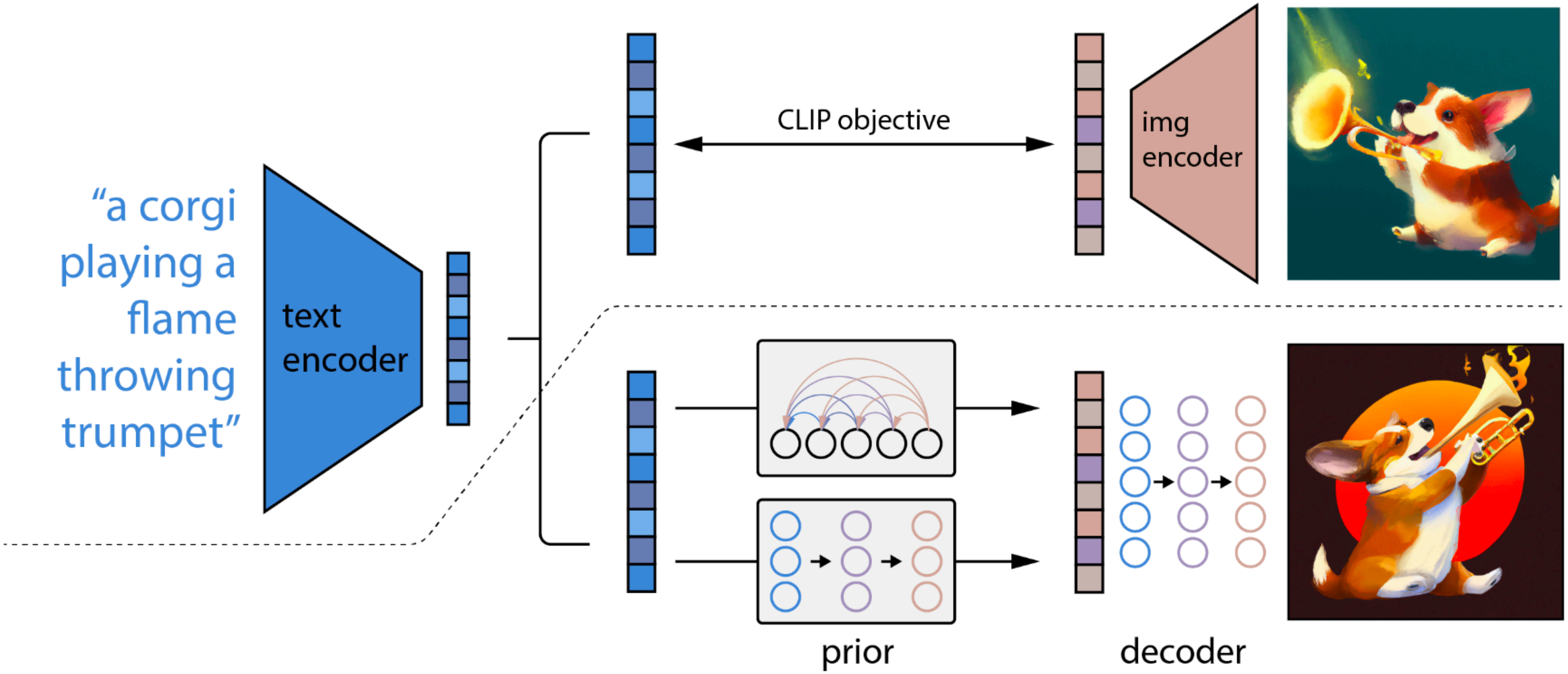


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

Stable Diffusion
(with the text-conditioning)