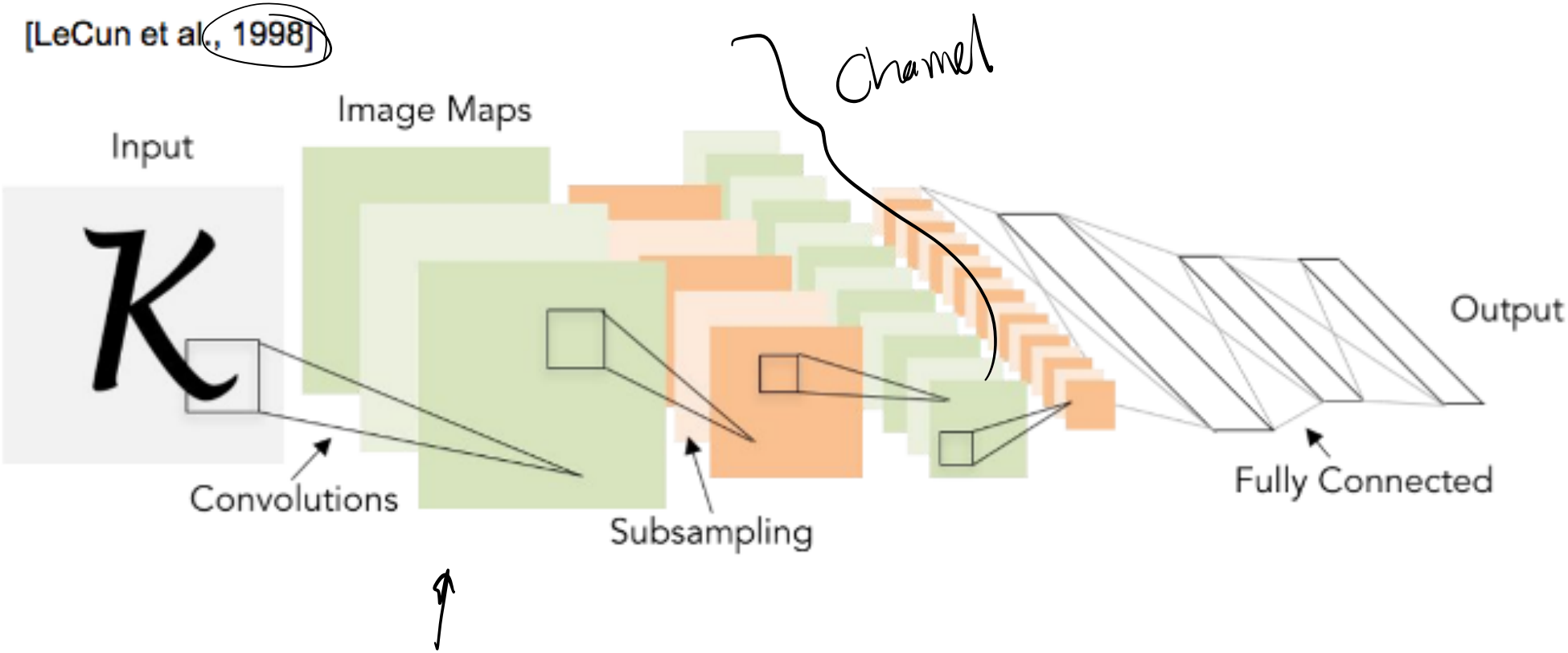
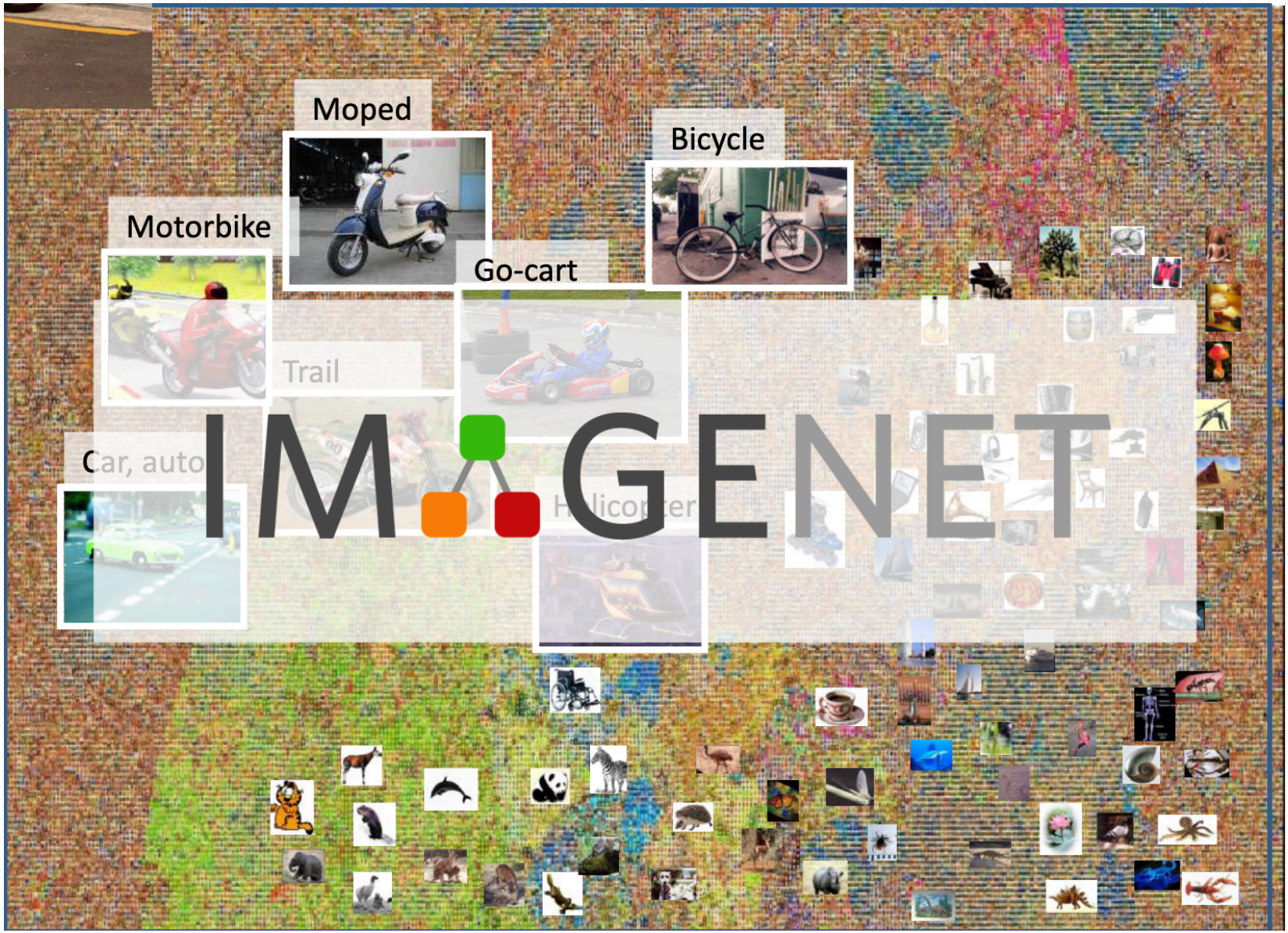
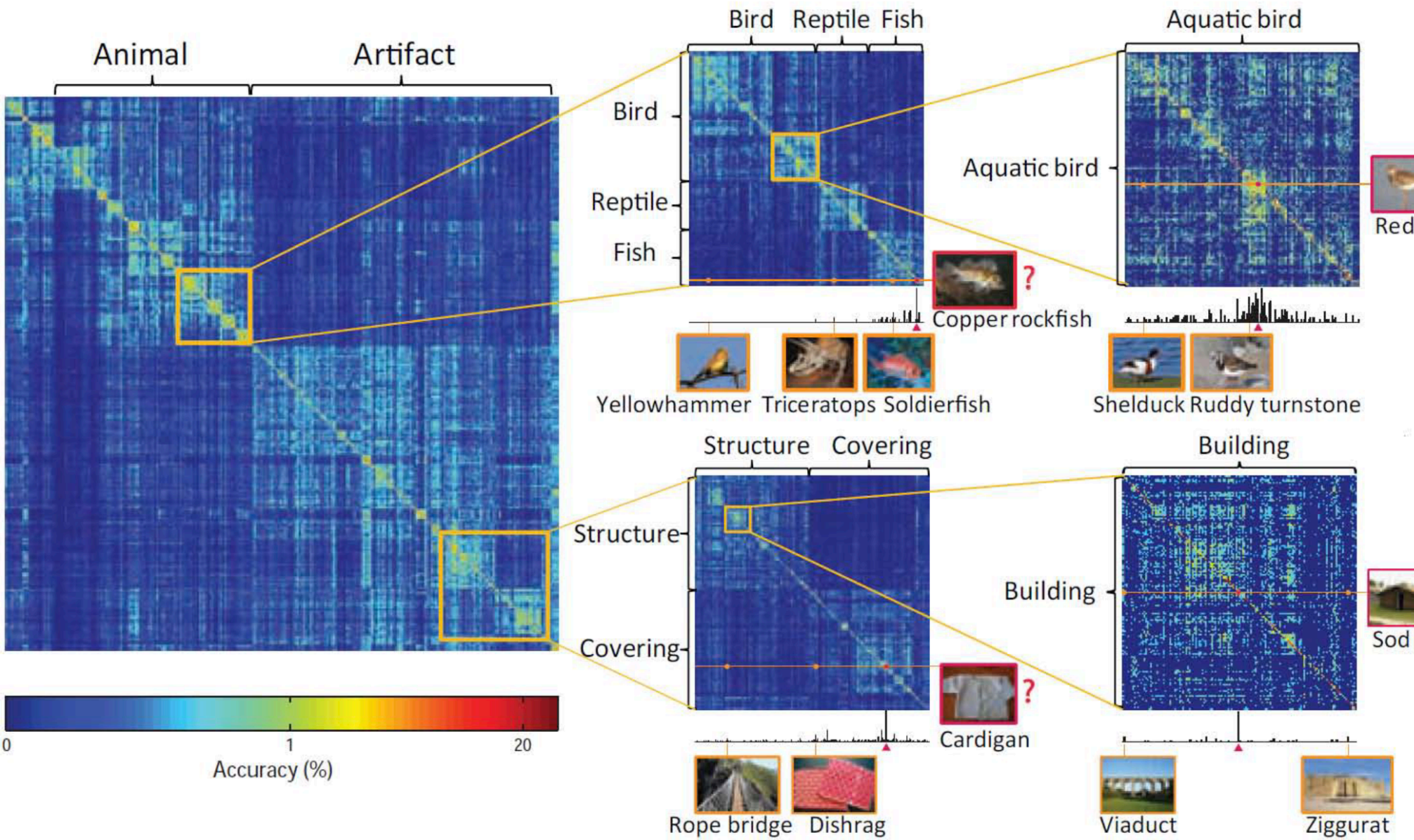
Review: LeNet-5

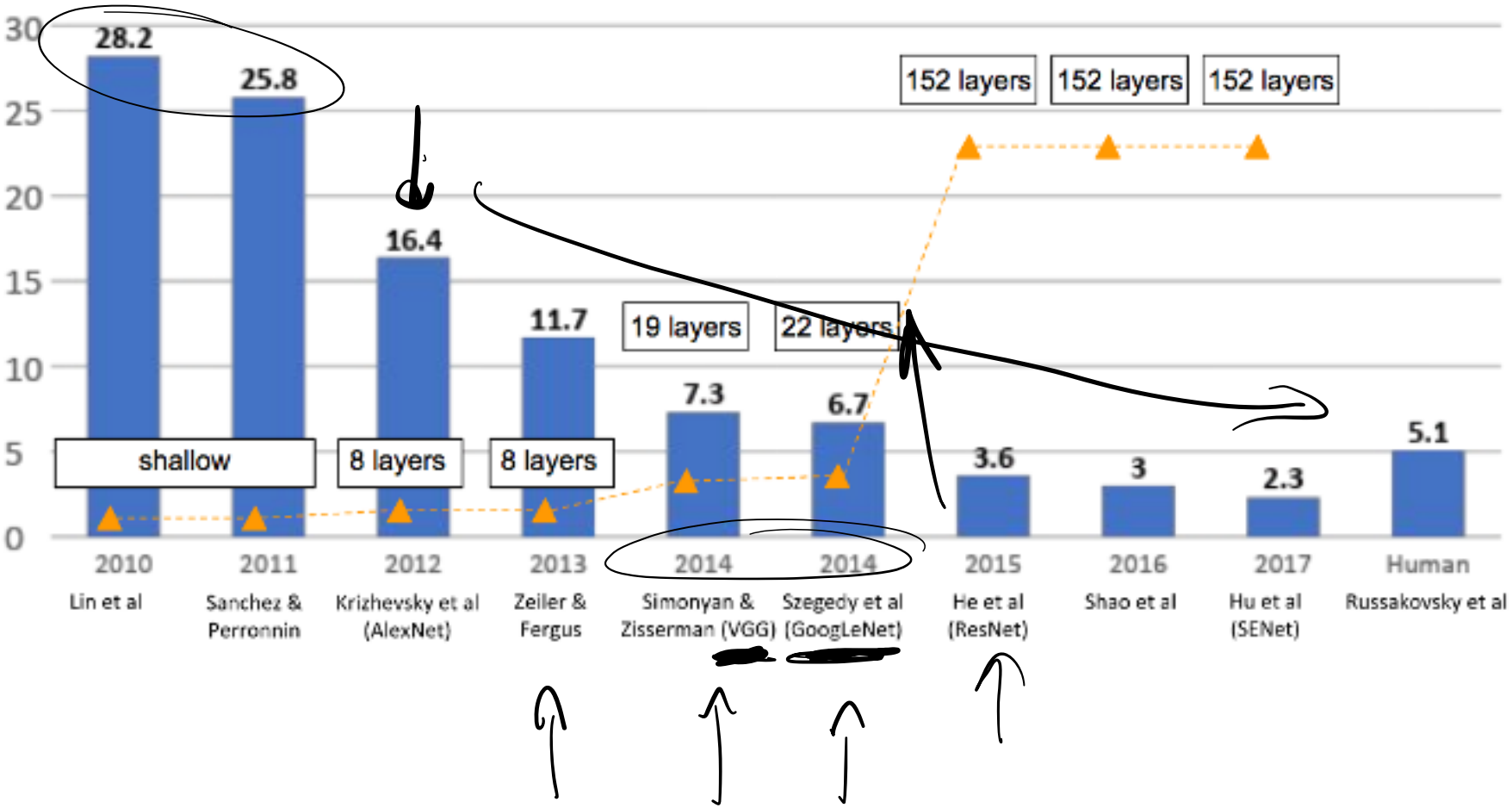
[LeCun et al., 1998]




Moped
Bicycle
Motorbike
Go-cart
Trail
IMAGENET

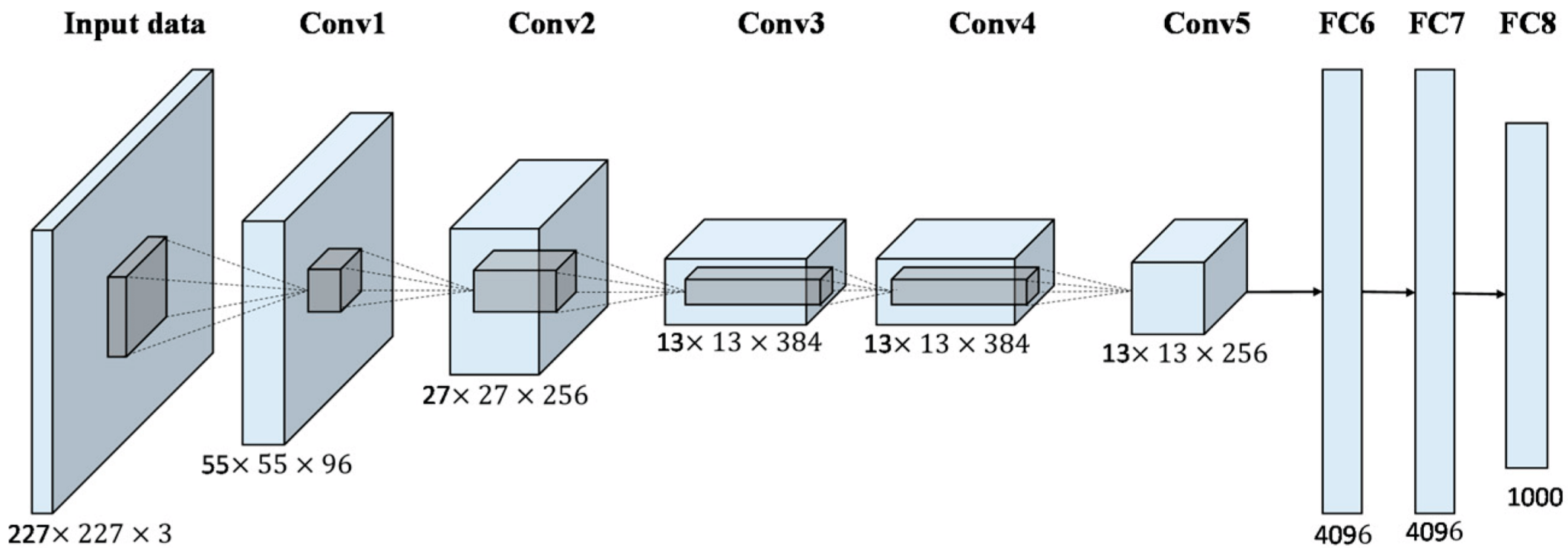

Animal
Artifact
0
1
20
Accuracy (%)
Bird
Reptile
Fish
Bird
Reptile
Fish
?
Copper rockfish
Yellowhammer
Triceratops
Soldierfish
Aquatic bird
Aquatic bird
Red
Shelduck
Ruddy turnstone
Structure
Covering
Structure
Covering
?
Cardigan
Rope bridge
Dishrag
Building
Building
Sod
Viaduct
Ziggurat

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

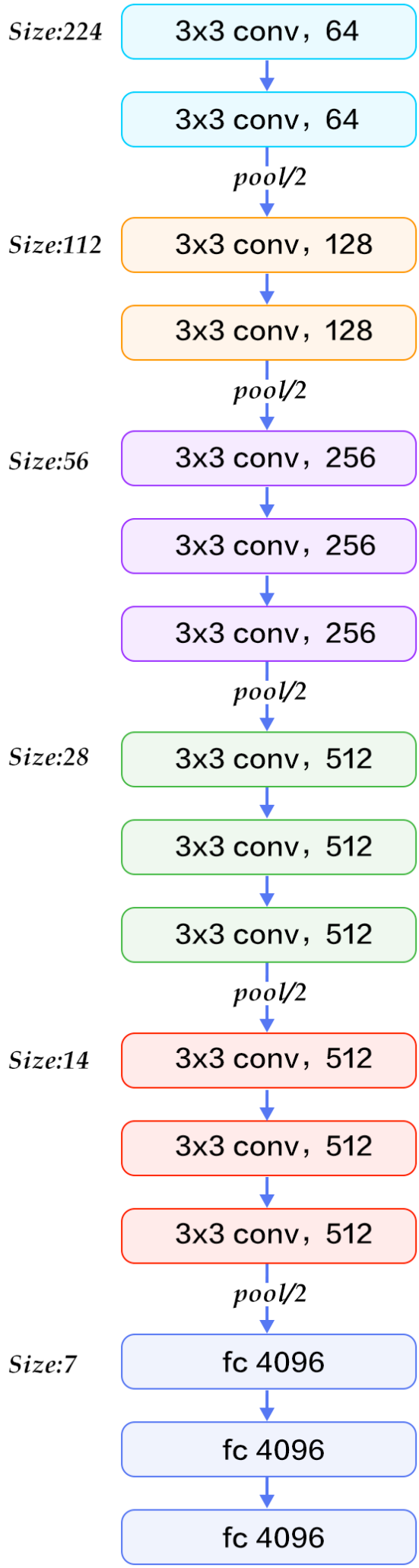



WE NEED TO GO
DEEPER

Alex Net


Input data
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
227× 227 × 3
55× 55 × 96
27× 27 × 256
13× 13 × 384
13× 13 × 384
13× 13 × 256
4096
4096
1000

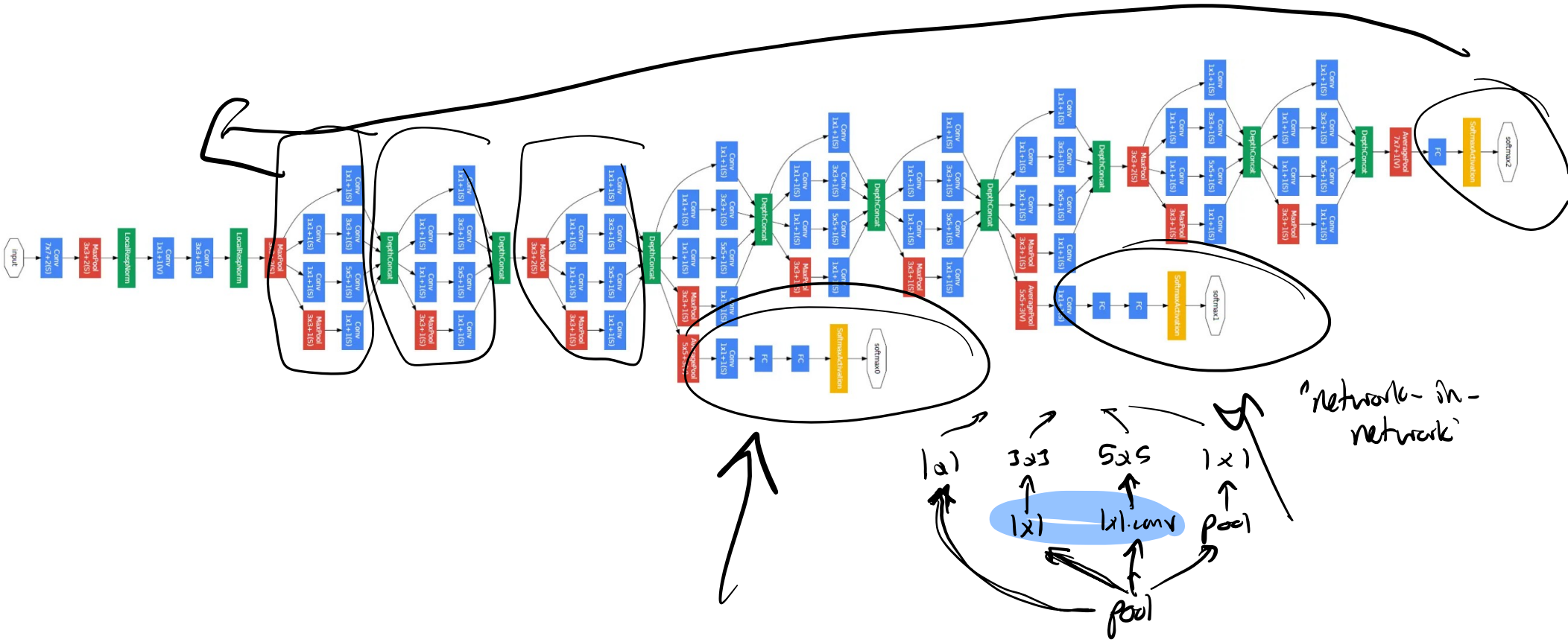

WE NEED TO GO
DEEPER


Size:224
3x3 conv，64
3x3 conv，64
pool/2
Size:112
3x3 conv，128
3x3 conv，128
pool/2
Size:56
3x3 conv，256
3x3 conv，256
3x3 conv，256
pool/2
Size:28
3x3 conv，512
3x3 conv，512
3x3 conv，512
pool/2
Size:14
3x3 conv，512
3x3 conv，512
3x3 conv，512
pool/2
Size:7
fc 4096
fc 4096
fc 4096

VGG Net

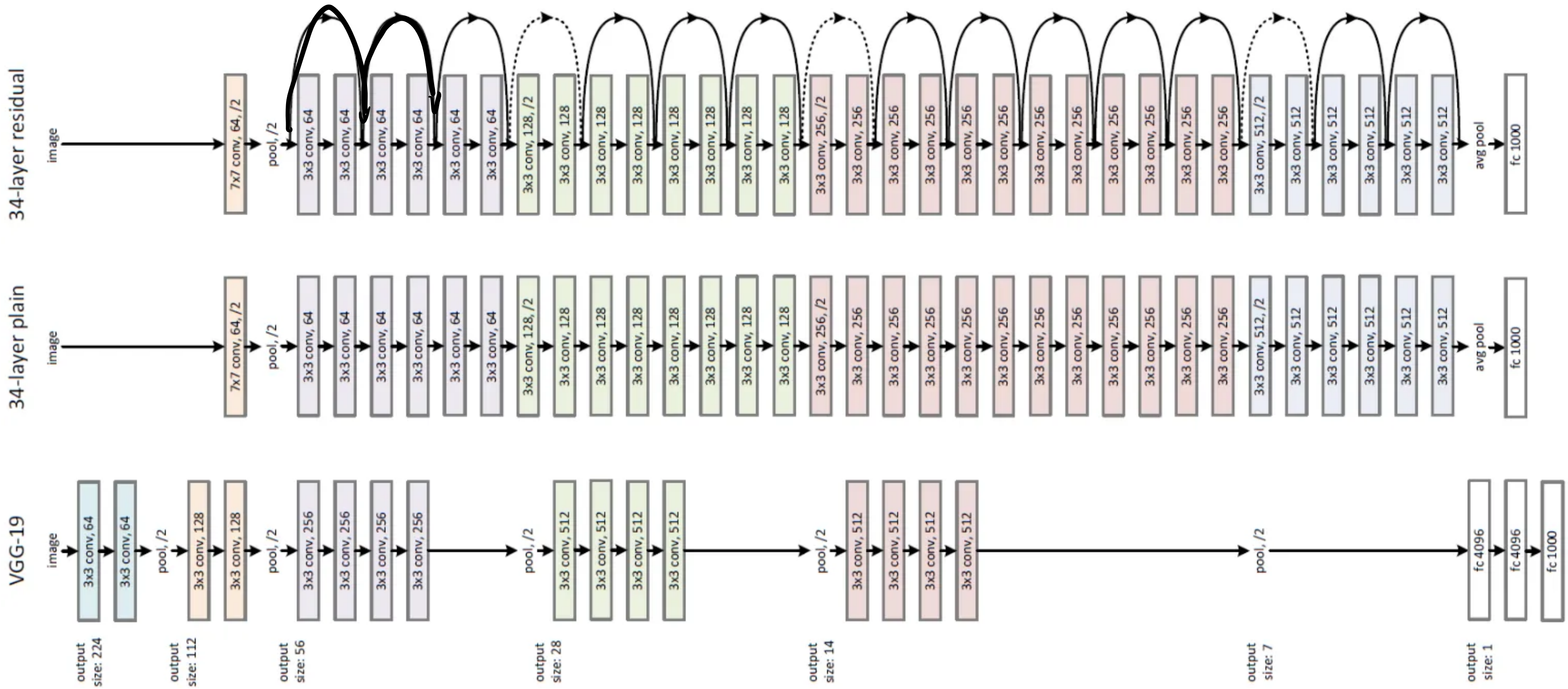

WE NEED TO GO
DEEPER

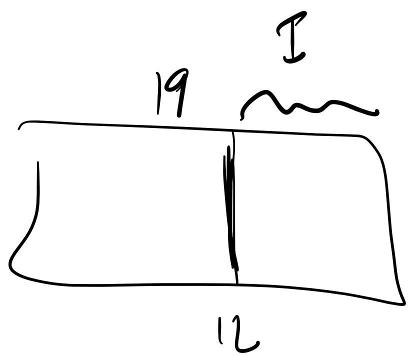
GoogLeNet (Inception)

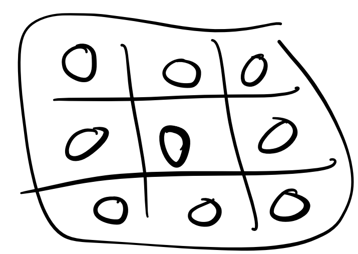


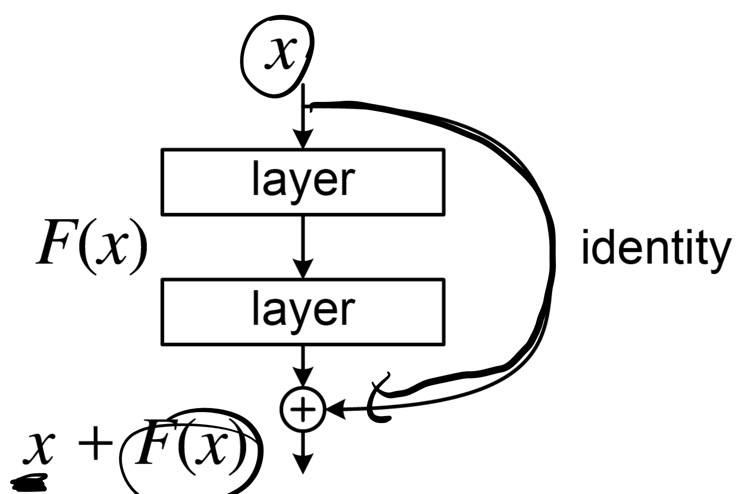

AFP


WE NEED TO GO
DEEPER


34-layer residual
34-layer plain
VGG-19
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
fc 4096
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1


19
12

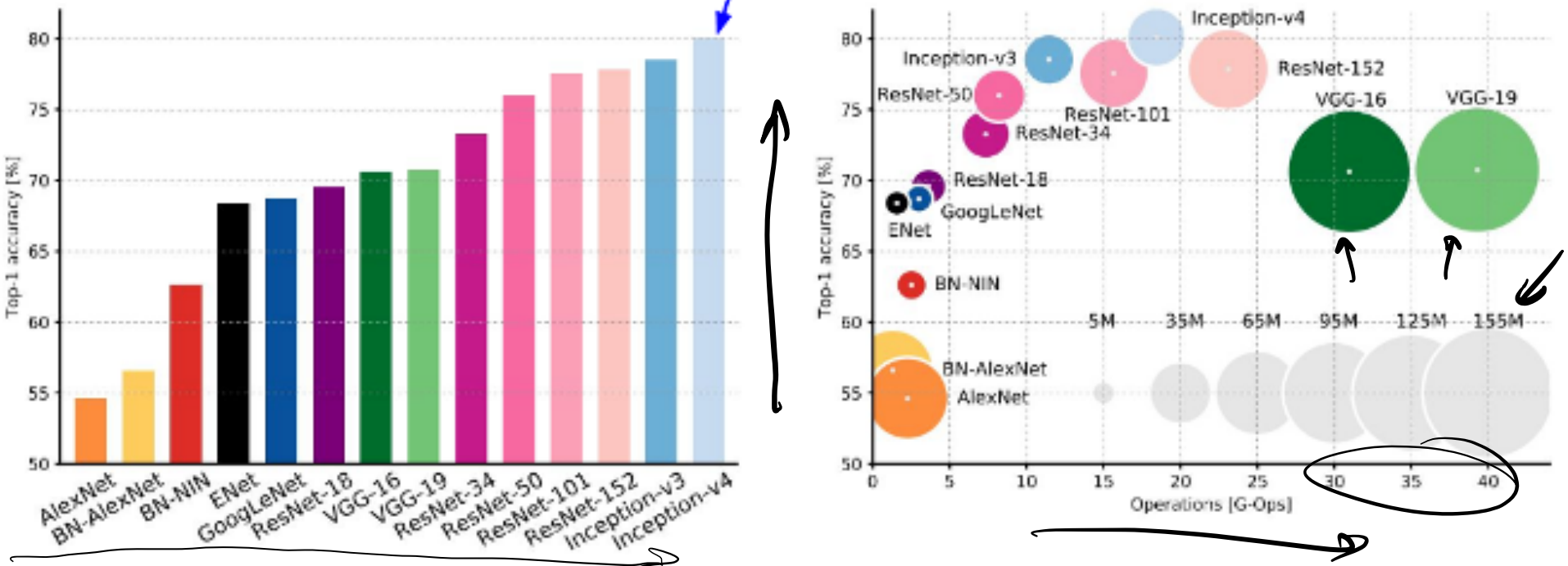



x
layer
layer
F(x)
identity
x + F(x)

Do we?

Comparing complexity…

Inception-v4: Resnet + Inception!

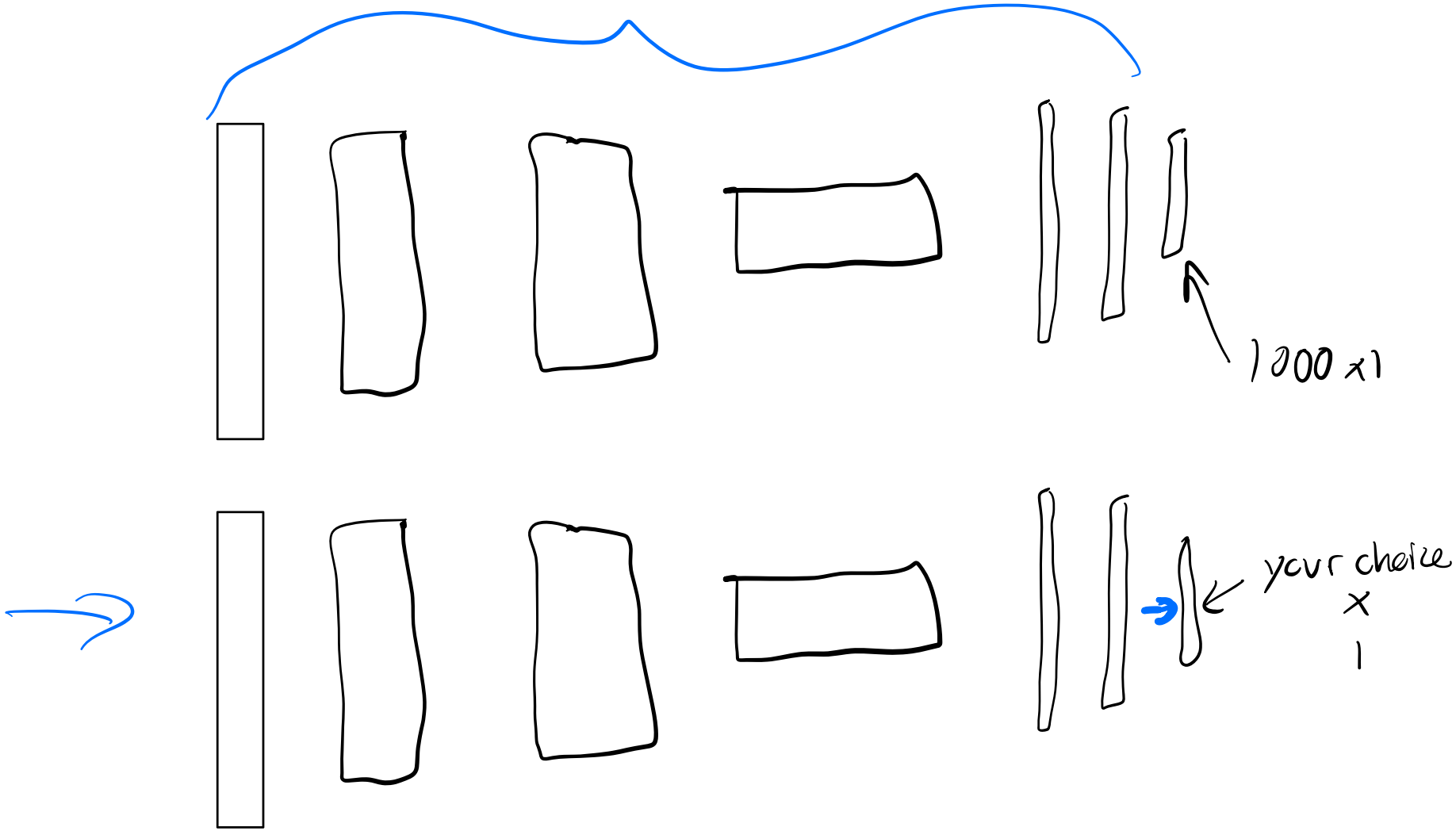


An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Okay but the data...

is expensive!

Transfer Learning / finetuning



Unsupervised / self-supervised learning case study: SimCLR

A Simple Framework for Contrastive Learning of Visual Representations

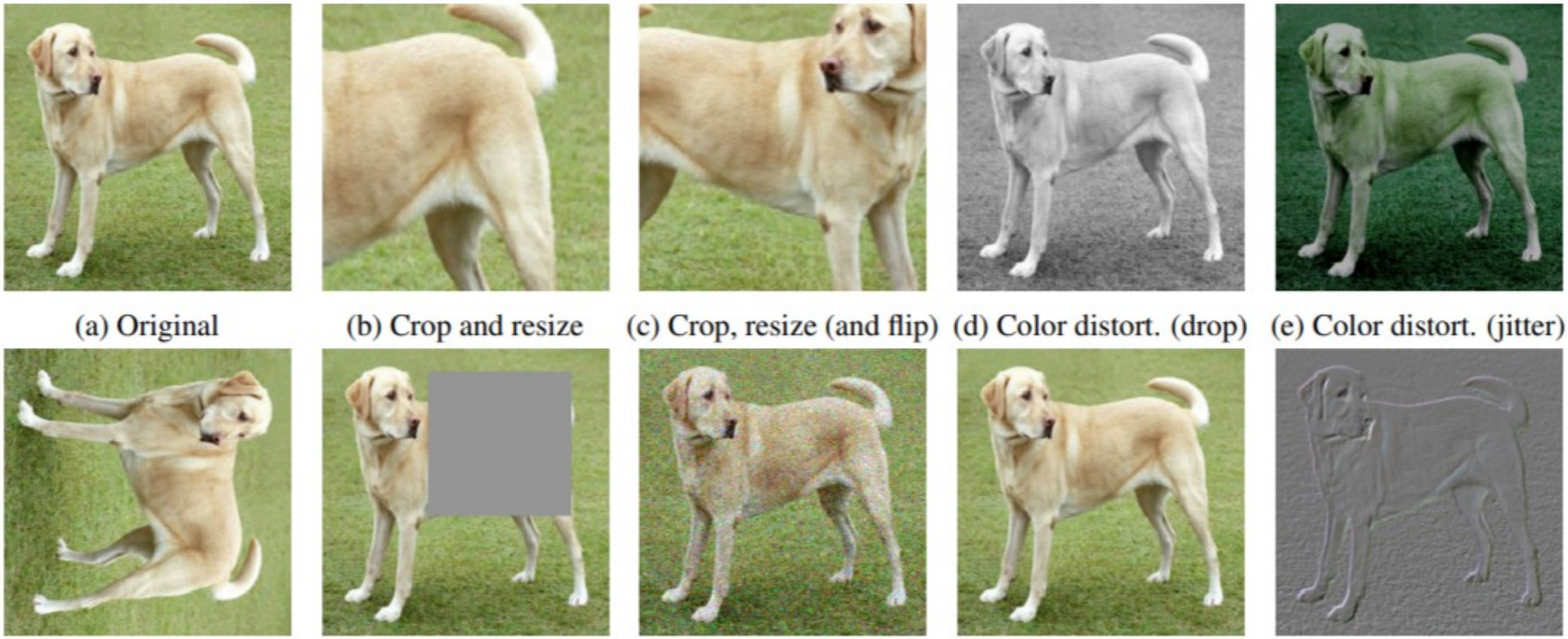
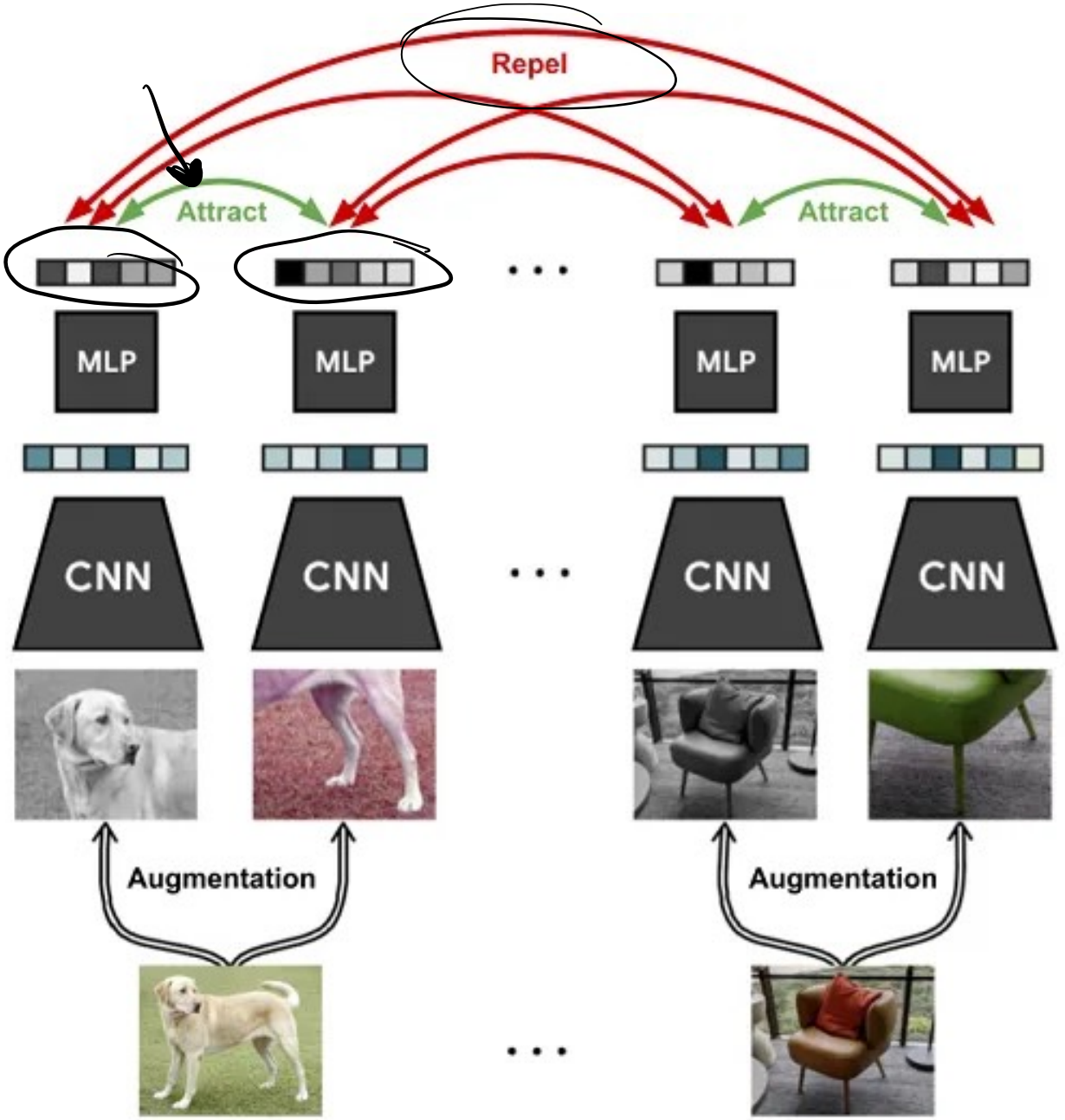
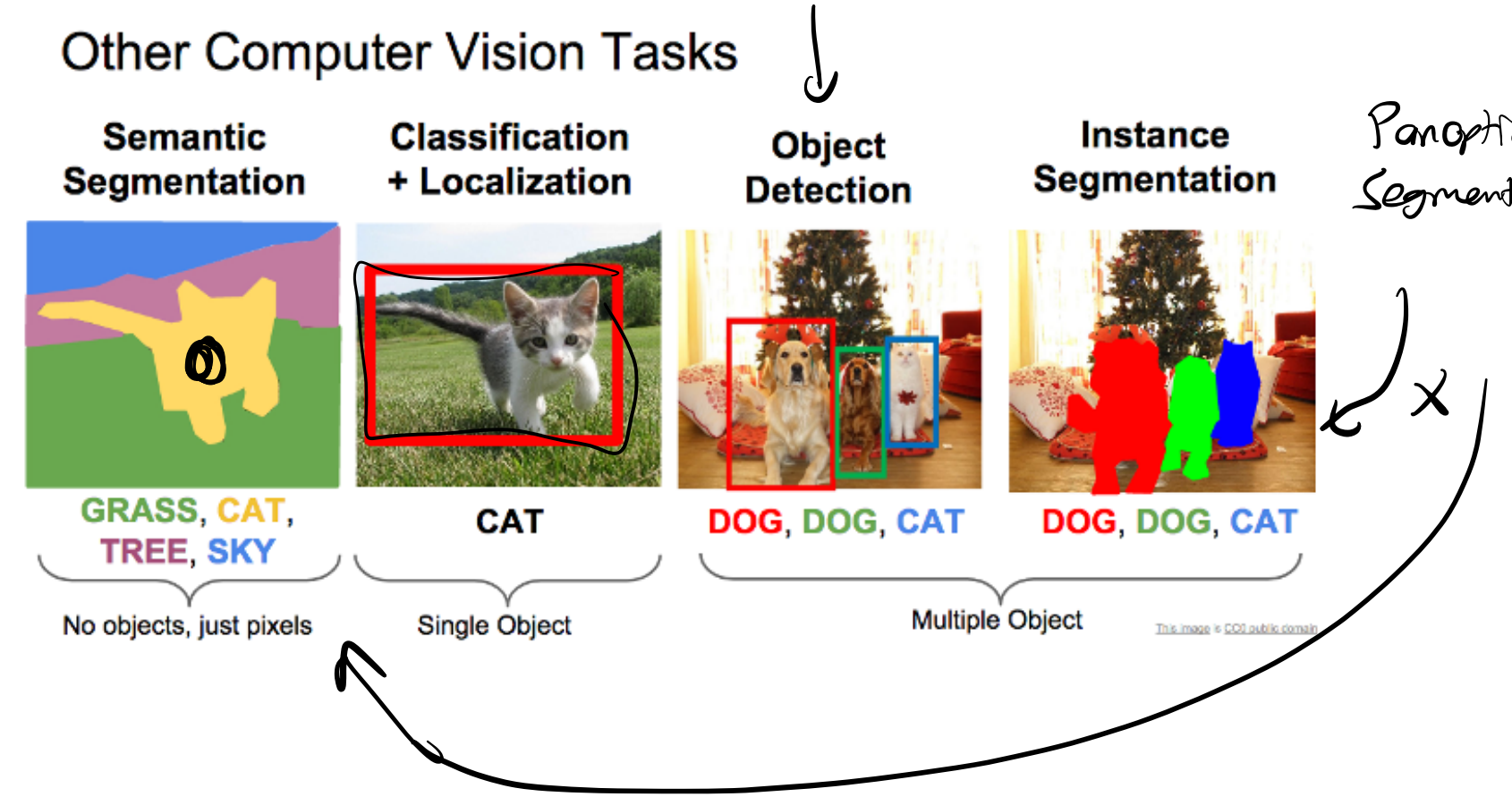


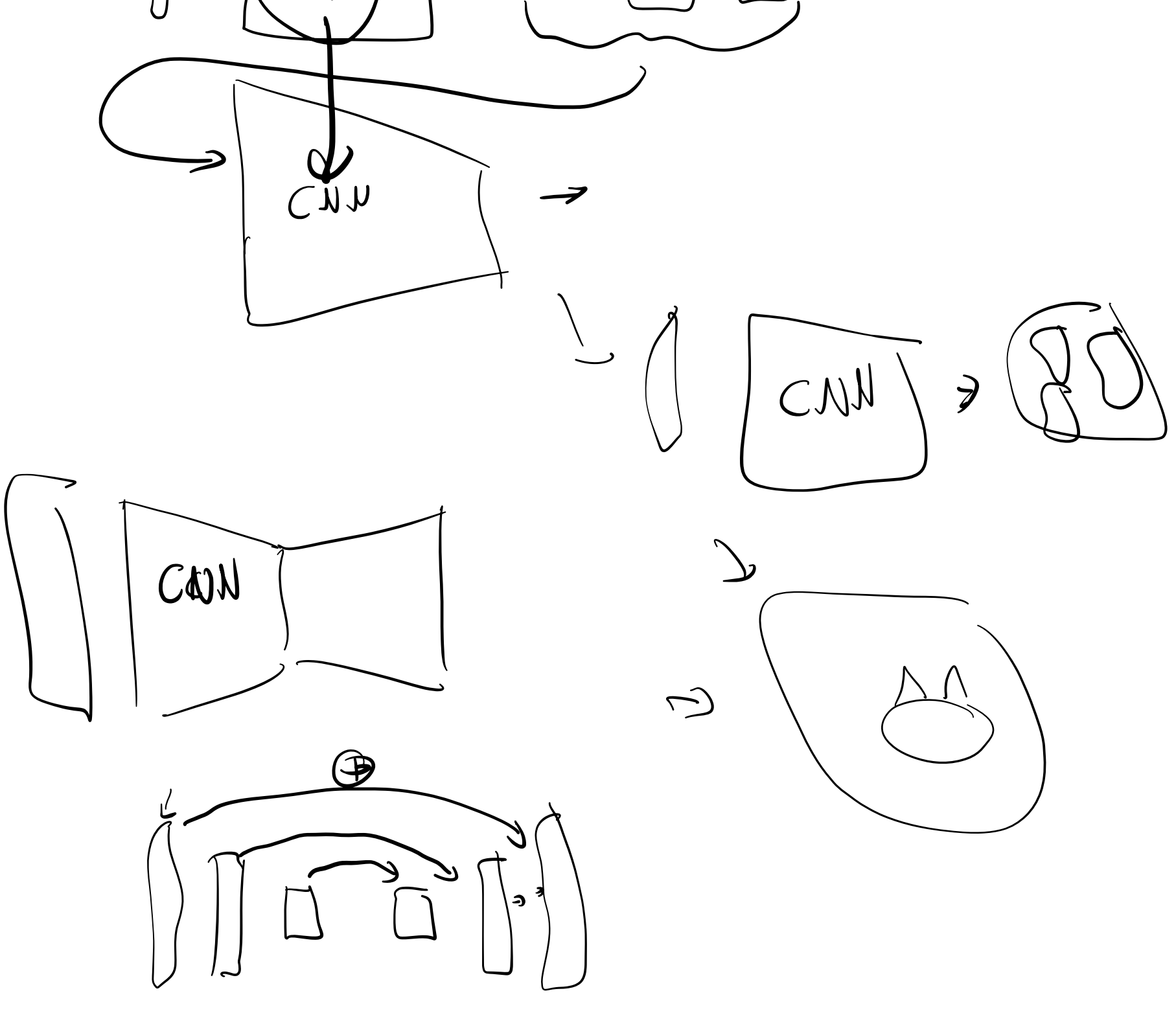
Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes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

Unsupervised / self-supervised learning case study: SimCLR

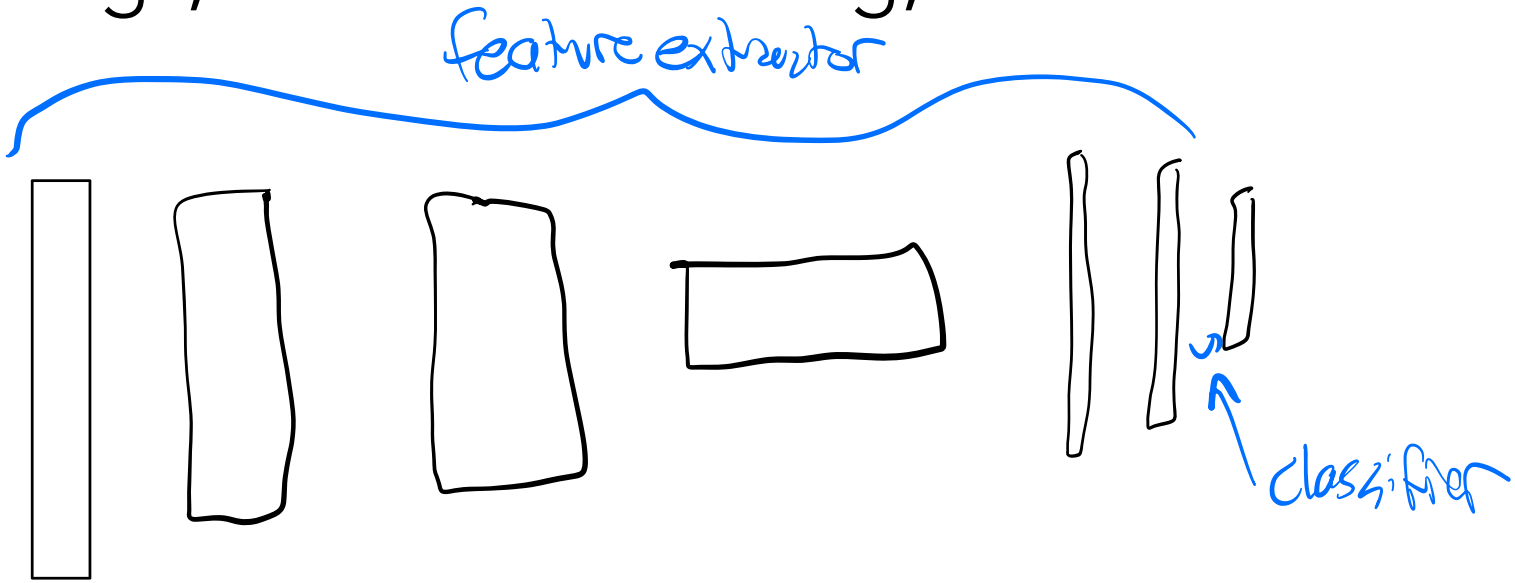


What about not image recognition?




CNN
CNN
CNN

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

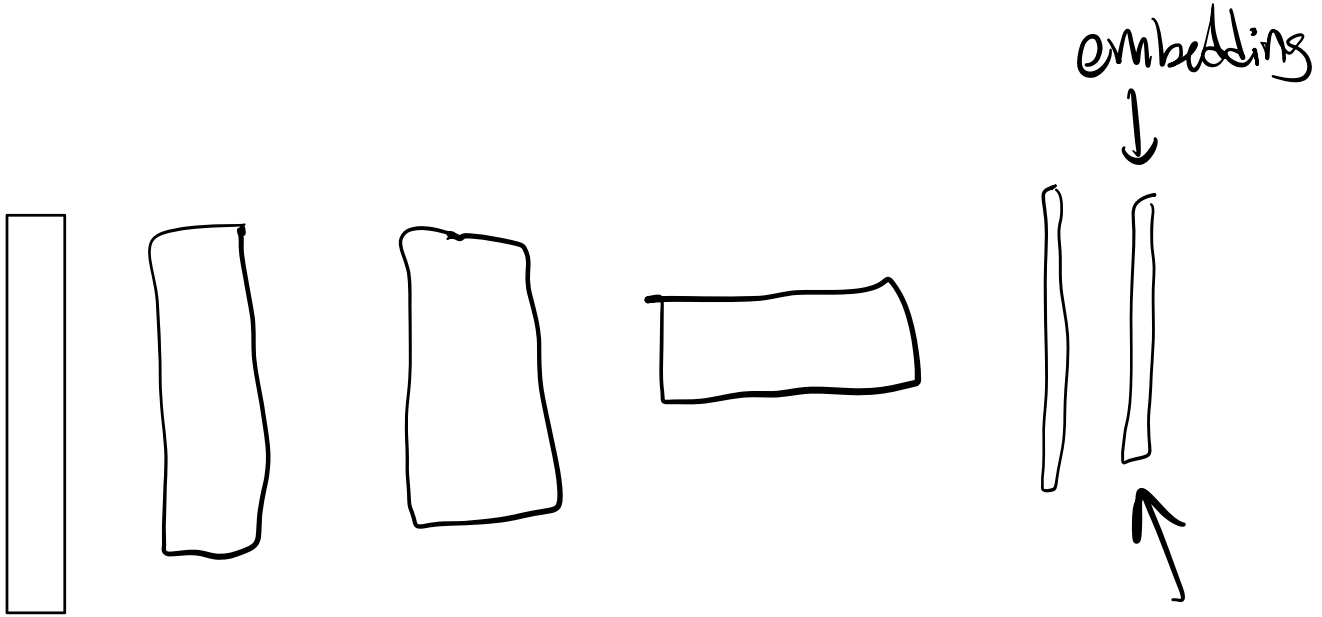


feature extraction → classify

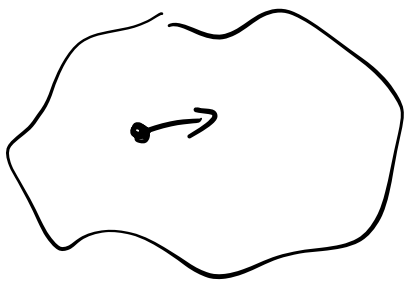
Img → canny → group edges into contours

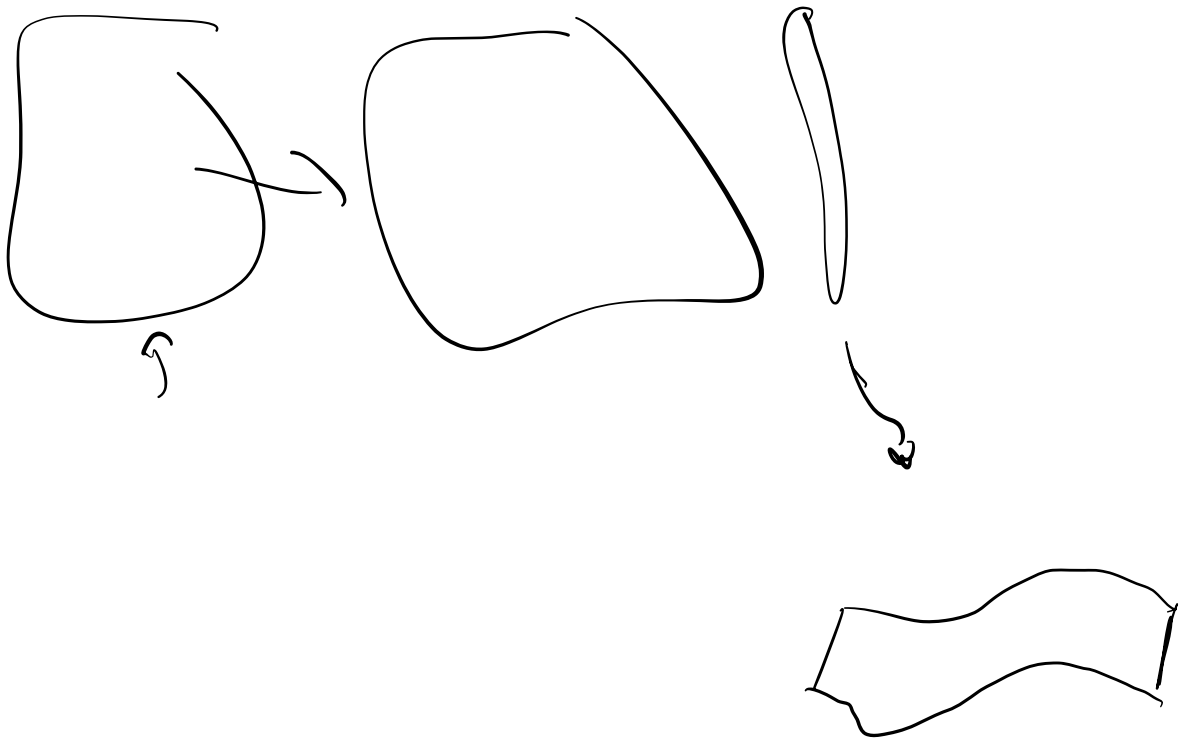
group into obj. parts

sum



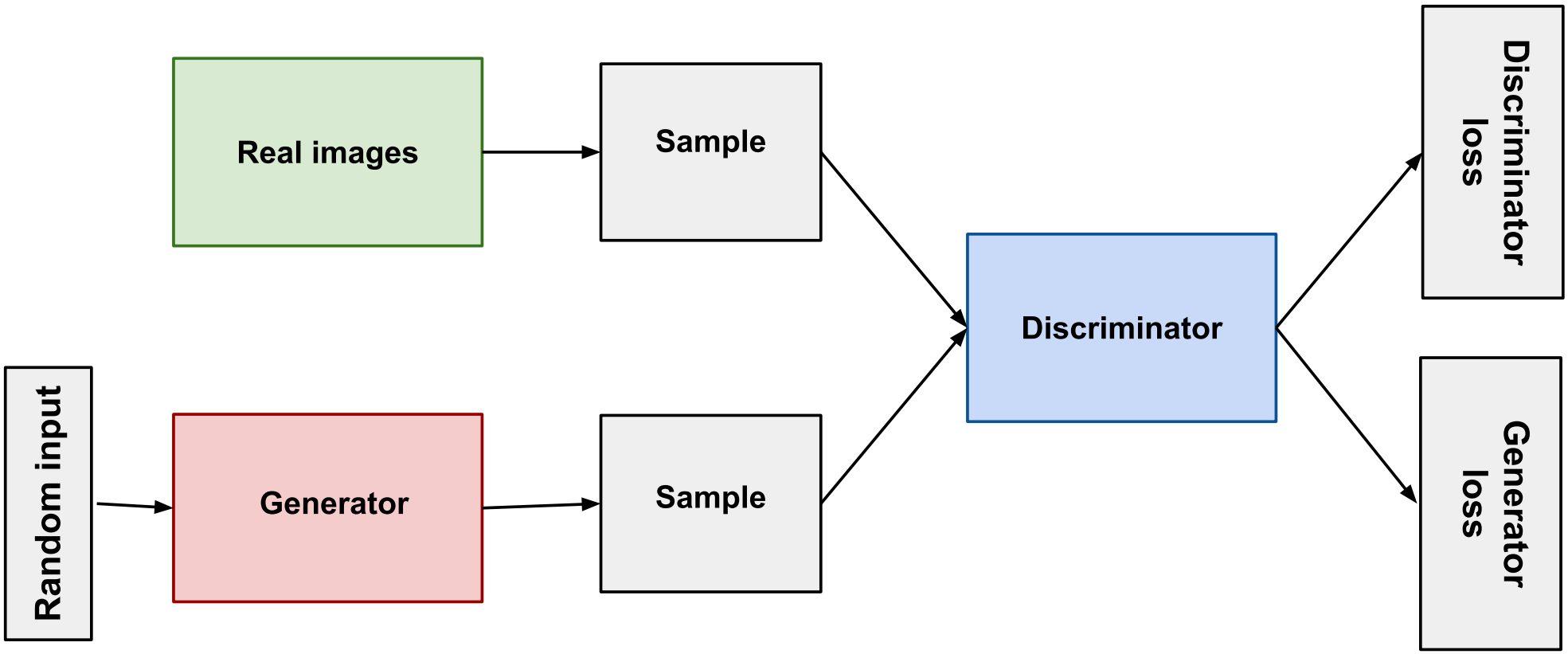
"King" - "Woman" → "Queen"





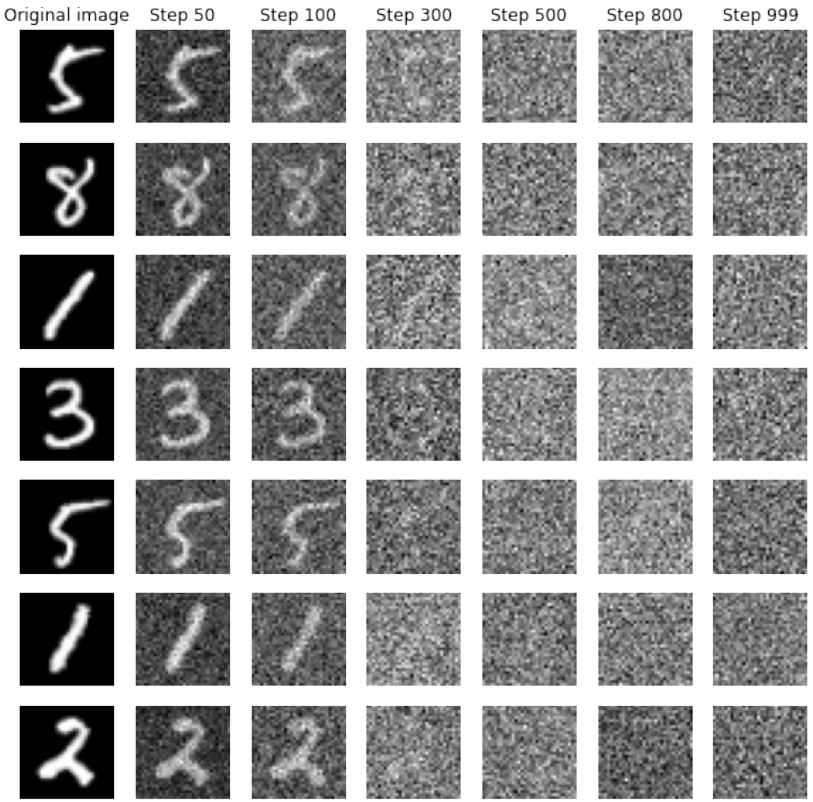
Generative Modeling

Generative Adversarial Networks

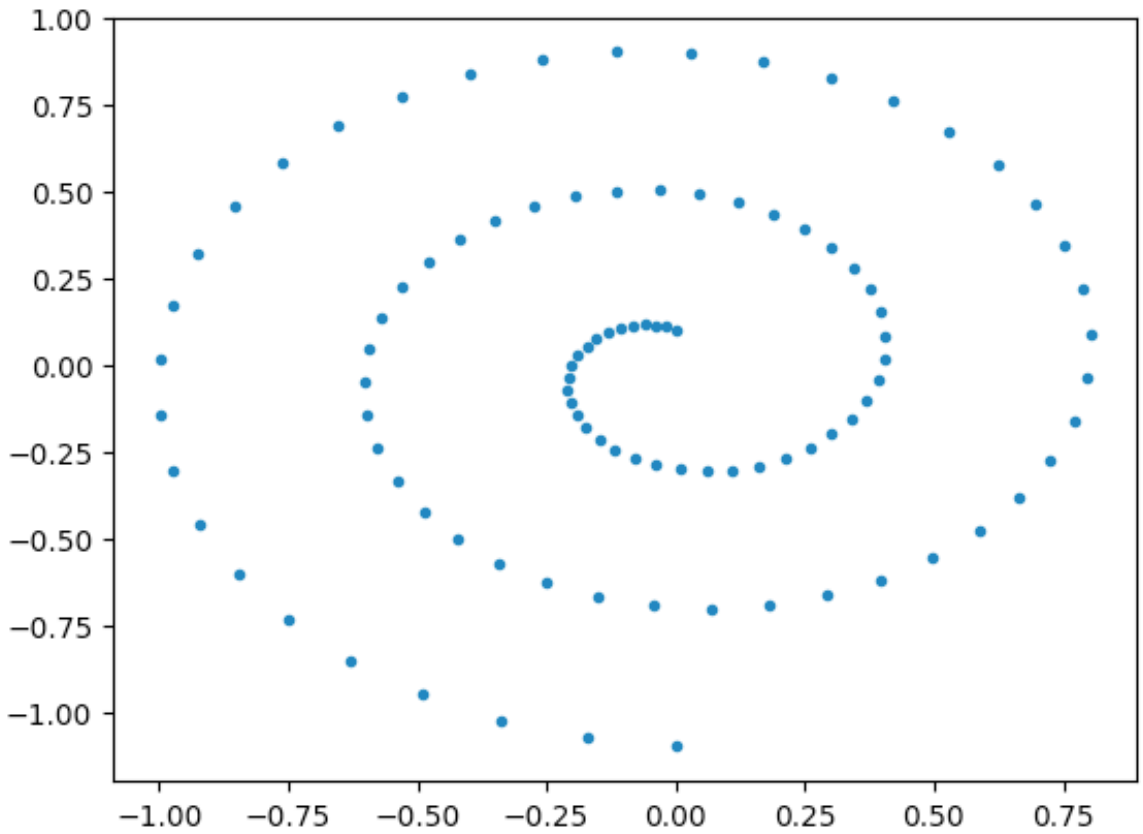


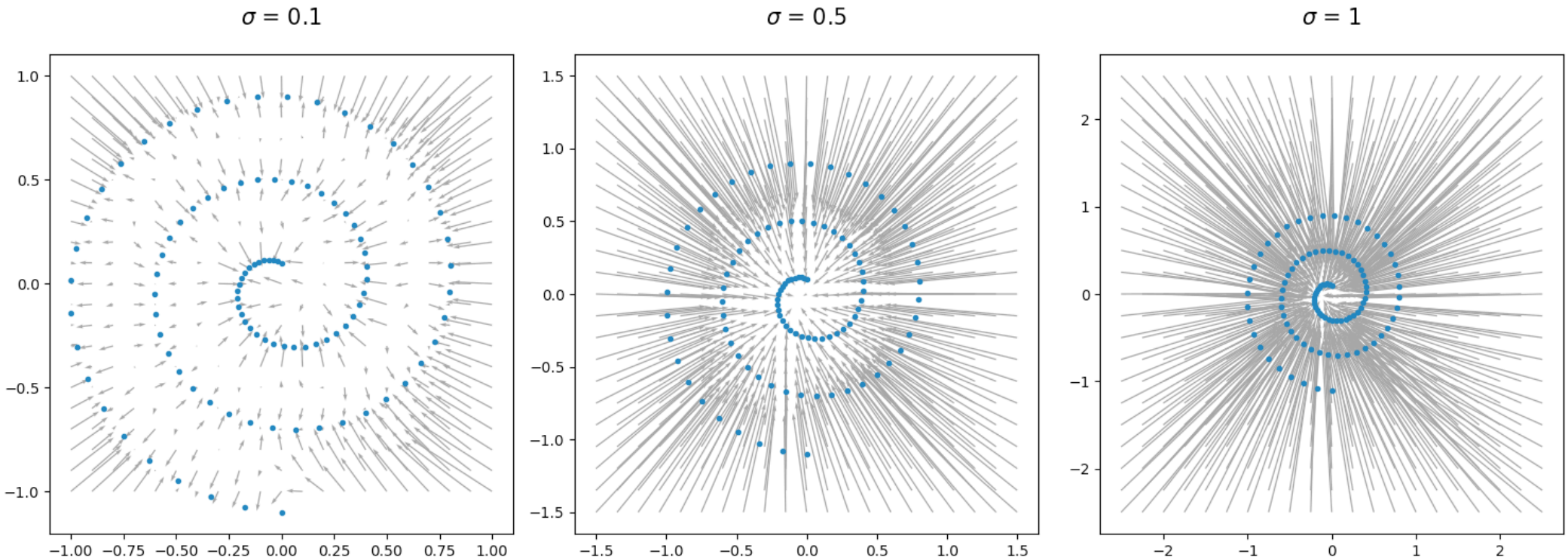

pixers

Diffusion Models



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

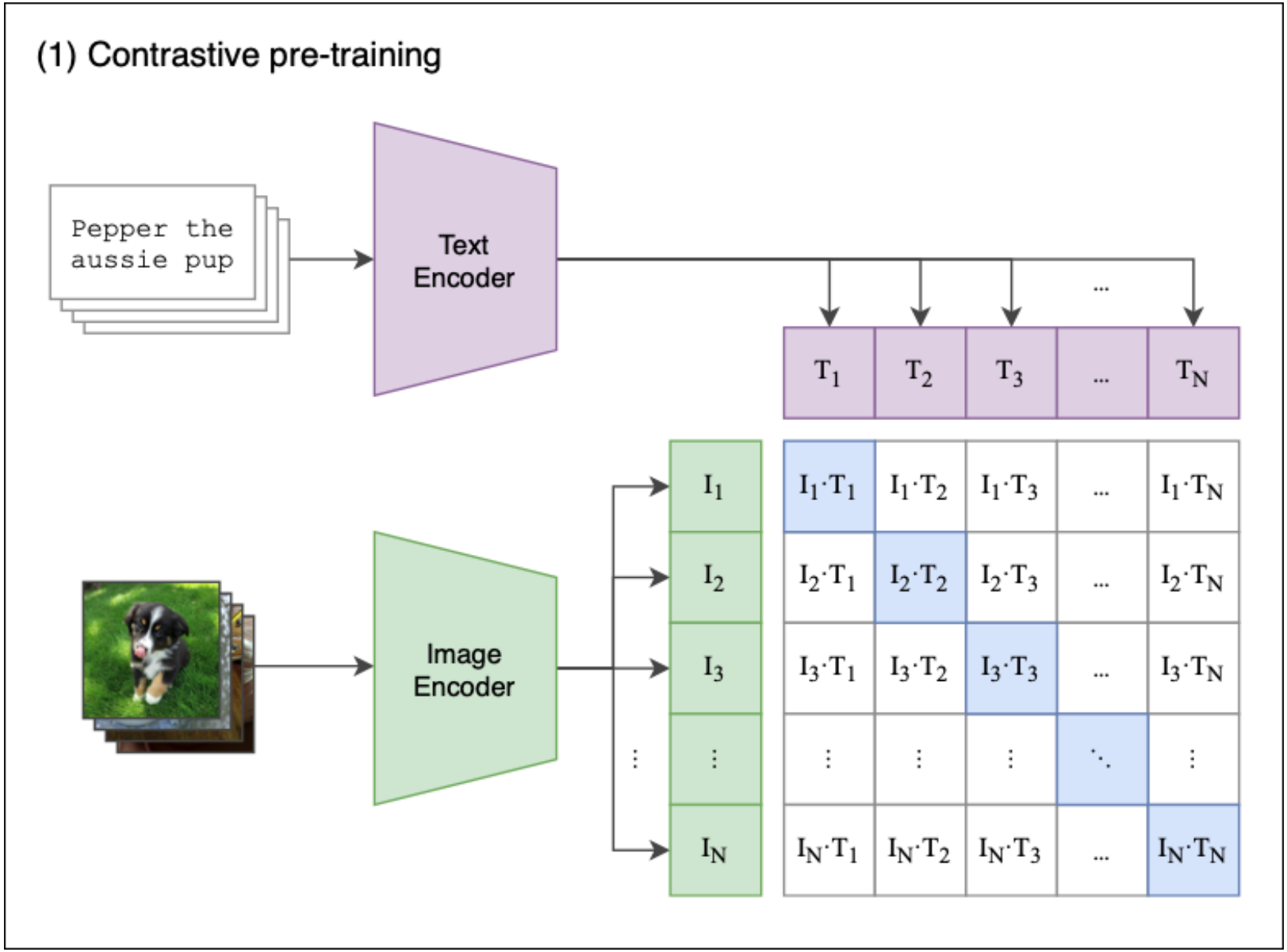

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75


$\sigma = 0.1$
$\sigma = 0.5$
$\sigma = 1$
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

Stable Diffusion
(without the text-conditioning)

Vision and Language

Case study: CLIP



unCLIP aka DALL-E 2

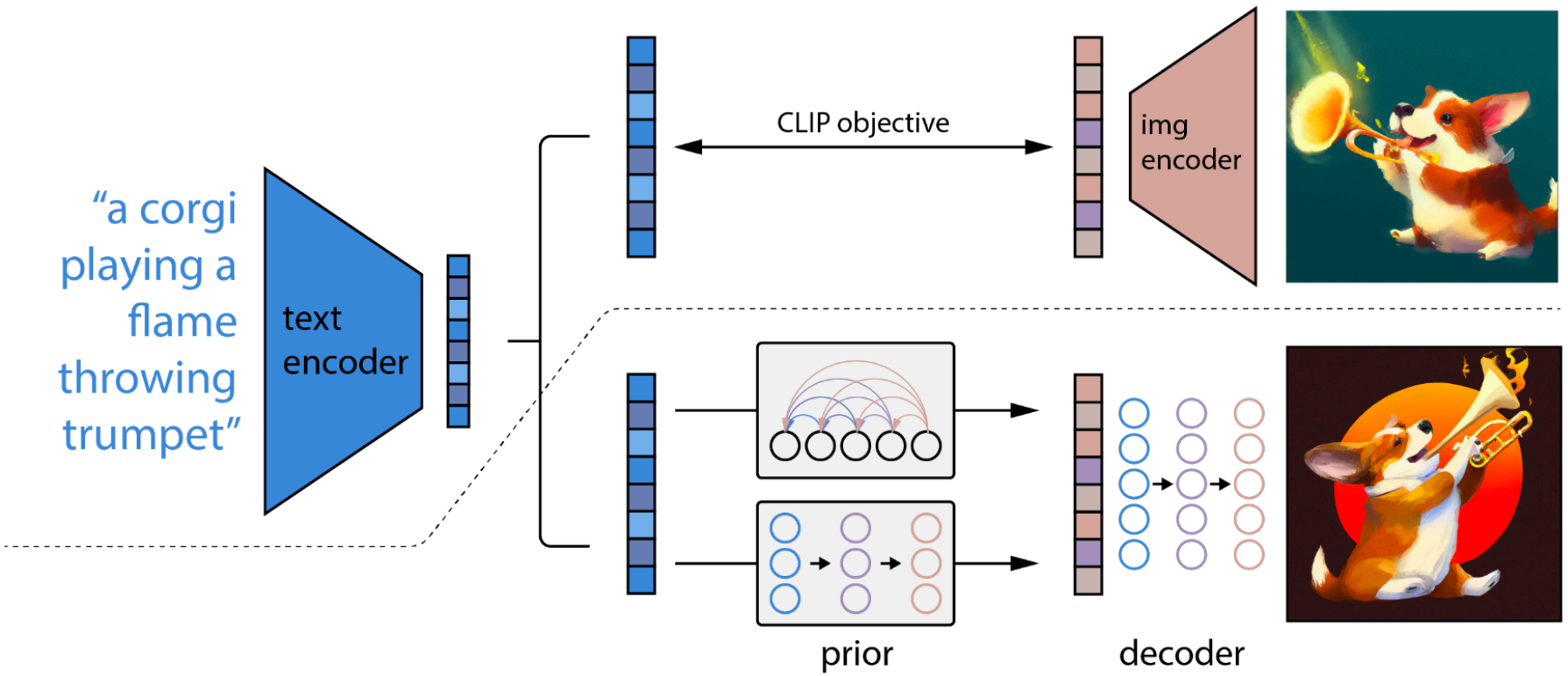


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive 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)