Machine Learning for Design

Lecture 4

Machine Learning for Images. *Part 2*

How do humans see?

Hubel and Wiesel, 1959

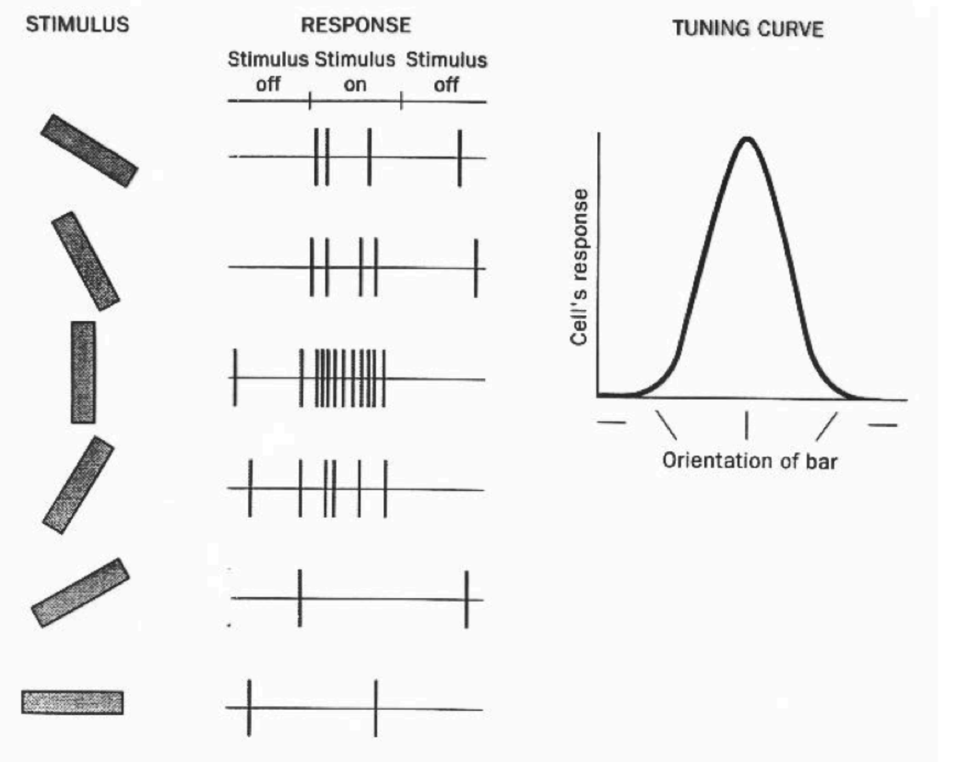
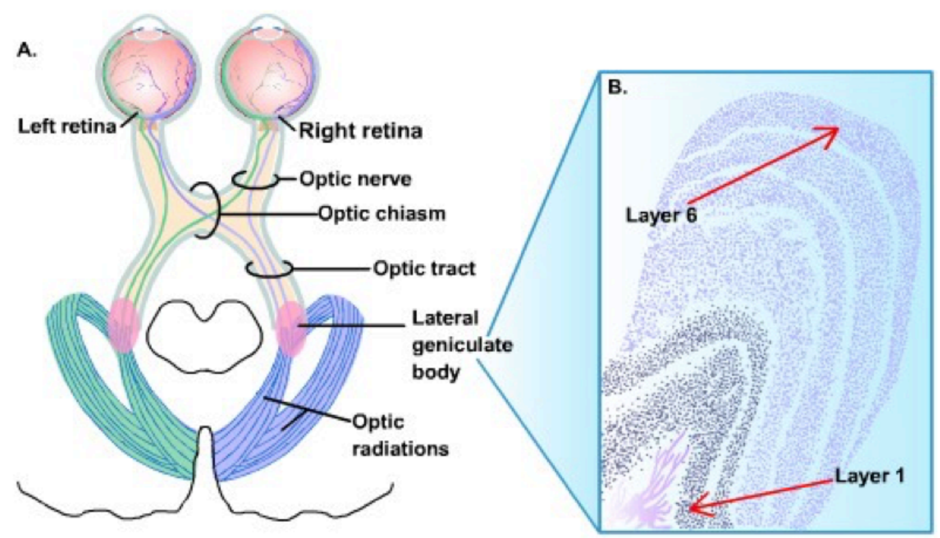


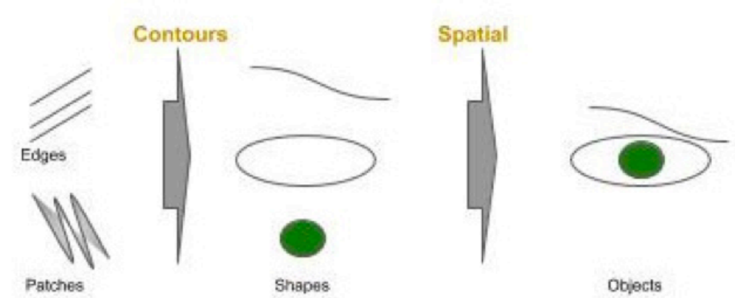
FIGURE 4.8 Response of a single cortical cell to bars presented at various orientations.

Neural Pathways



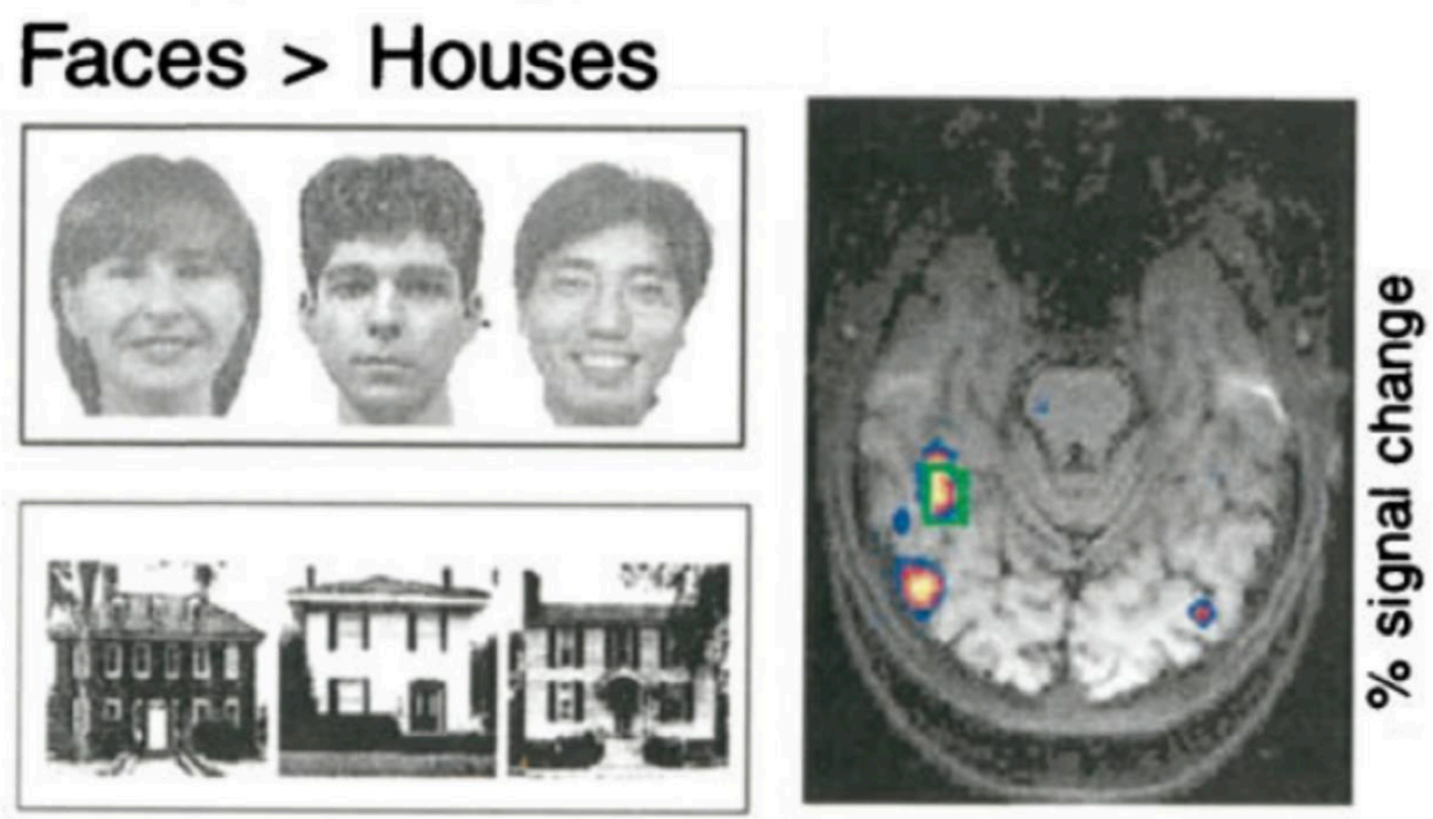
Edges → **Simple Shapes** → **Complex Shapes** → **Faces and Objects**

Lower layers   Upper layers

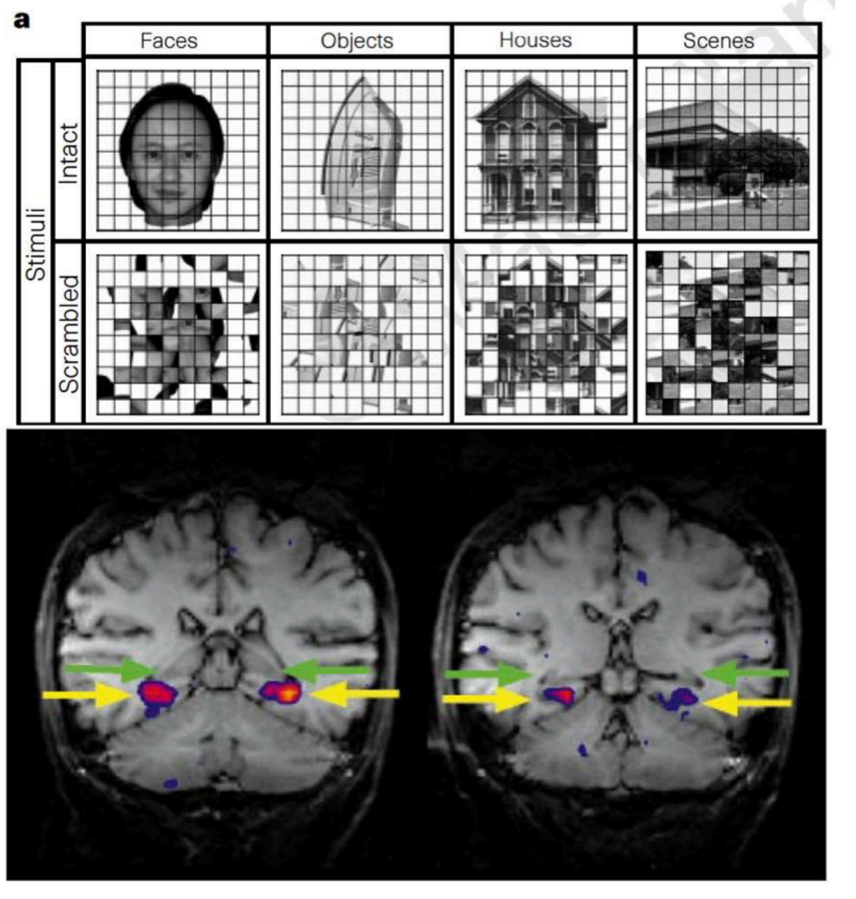


Neural Correlation

of Objects & Scene Recognition



Kanwisher et al. J. Neuro. 1997



Epstein & Kanwisher, Nature, 1998

Why is machine vision hard?

The deformable and truncated cat

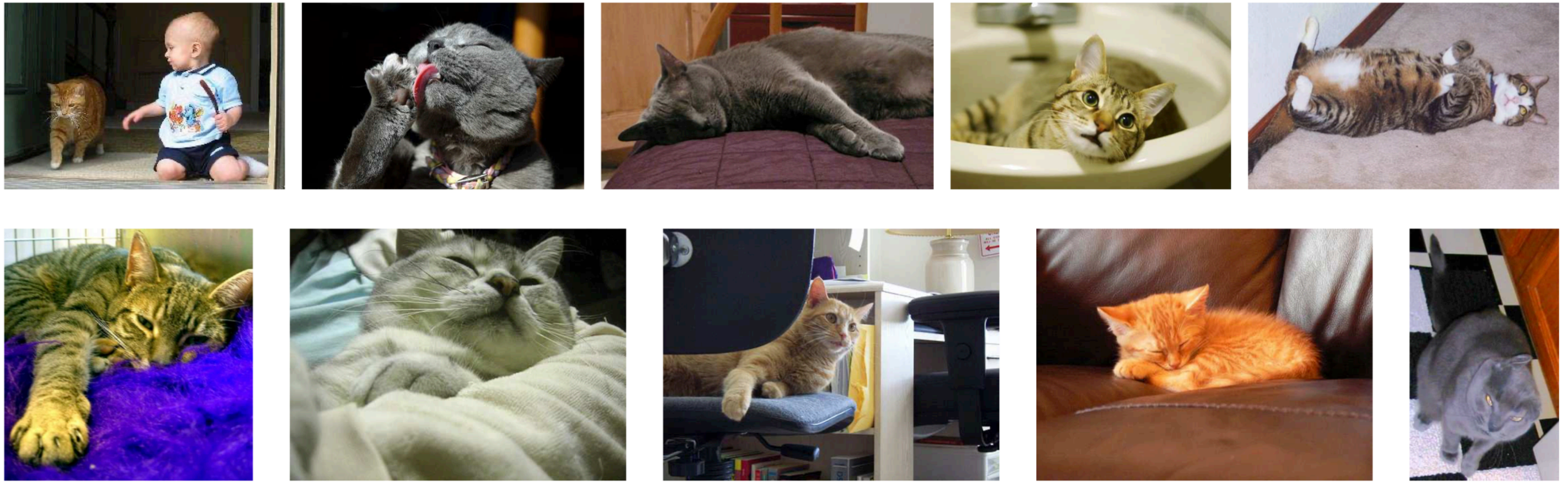
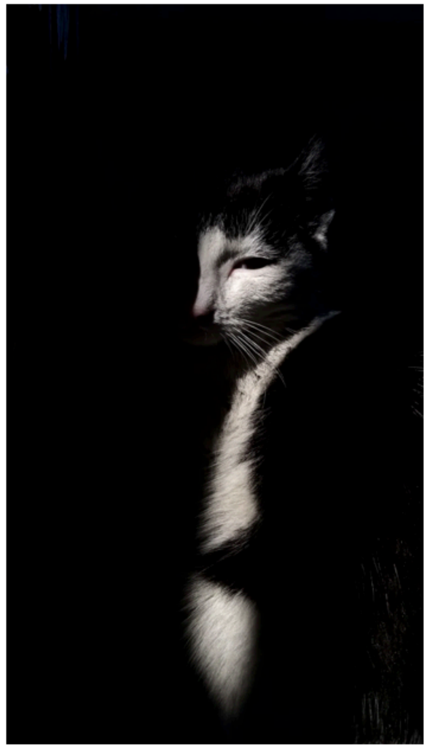
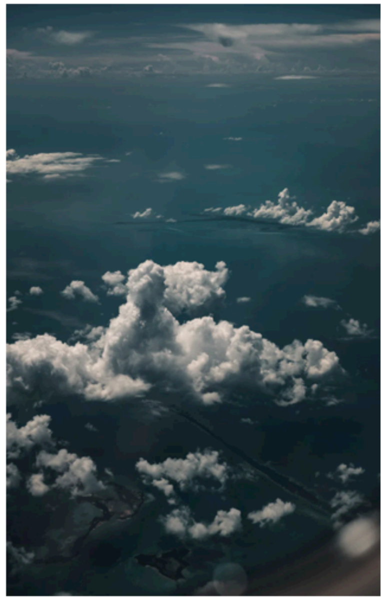
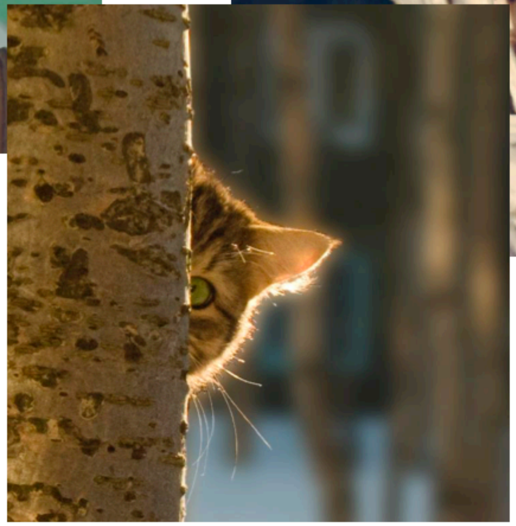


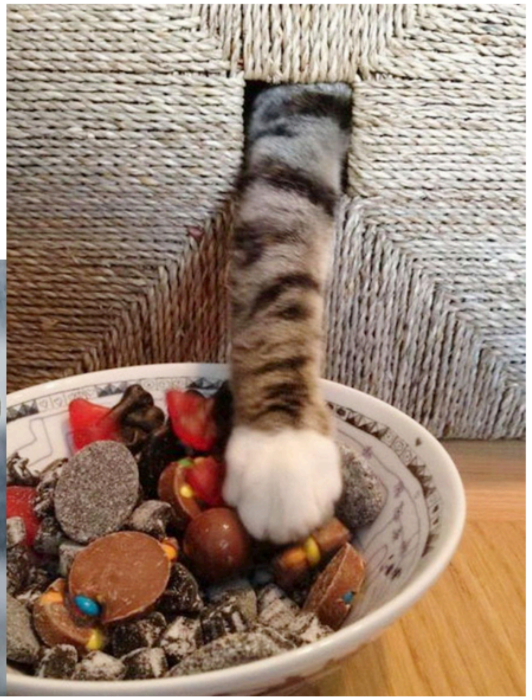
Figure 1. **The deformable and truncated cat.** Cats exhibit (almost) unconstrained variations in shape and layout.

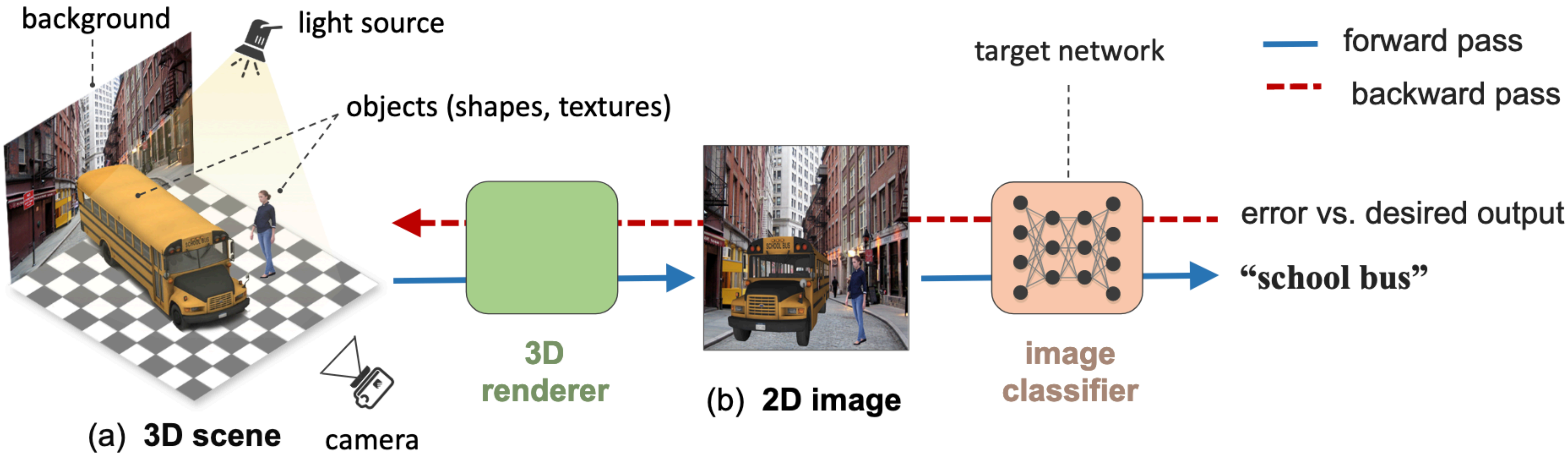
Parkhi et al. *The truth about cats and dogs*. 2011










background
light source
objects (shapes, textures)
(a) 3D scene
camera
3D
renderer
(b) 2D image
target network
image
classifier
forward pass
backward pass
error vs. desired output
"school bus"

Computer Vision Challenges

Viewpoint Variation
A single instance of an object can be oriented in many ways to the camera.

Deformation

Many objects of interest are not rigid bodies and can be deformed in extreme ways.

Occlusion

The objects of interest can be occluded. Sometimes only a tiny portion of an object (as few pixels) could be visible.

Illumination Condition

The effects of illumination can be drastic on the pixel level.

Scale variation

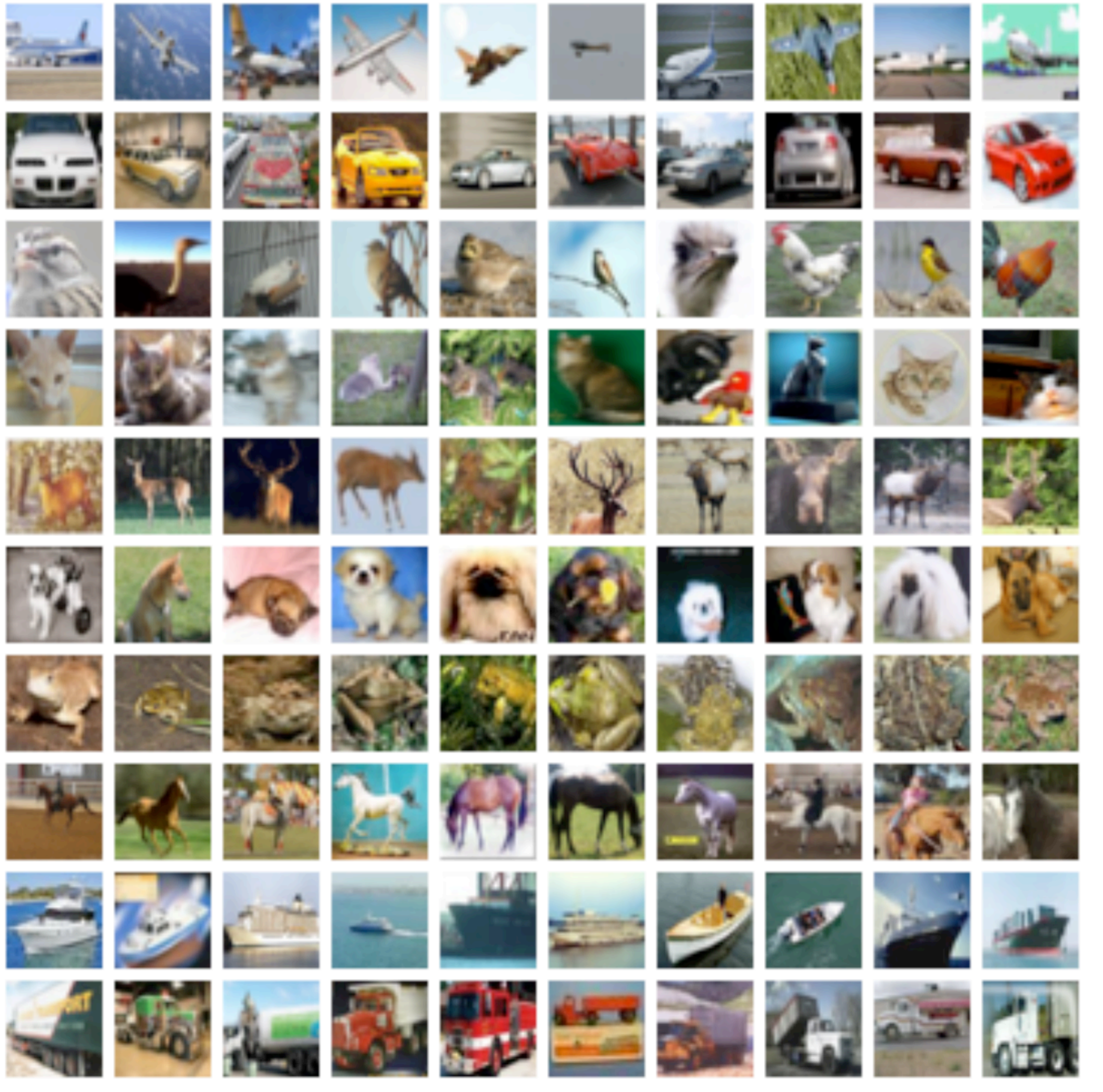
- Visual classes often exhibit variation in their size
  - Size in the real world
  - Size in the image

Background clutter

The objects of interest may blend into their environment, making them hard to identify.

Intra-class variation

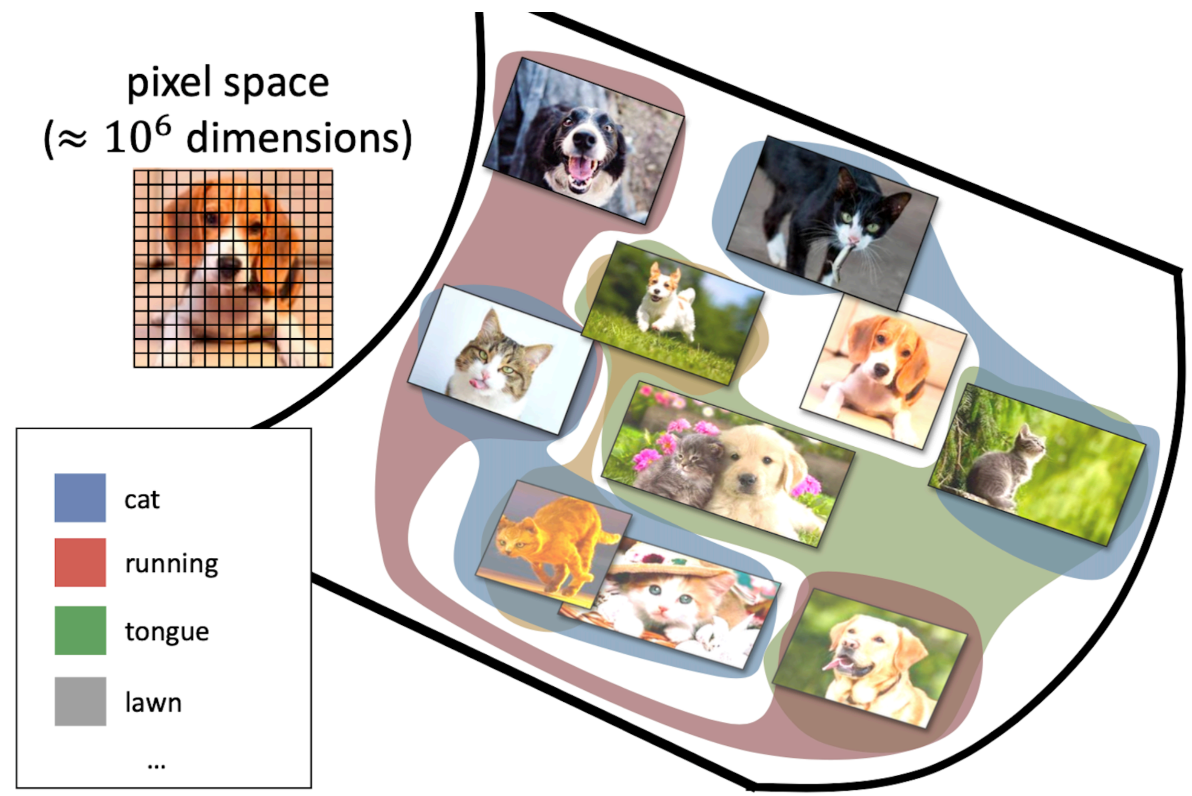
- The classes of interest can often be relatively broad, such as chairs.
- There are many different types of these objects, each with their appearance.



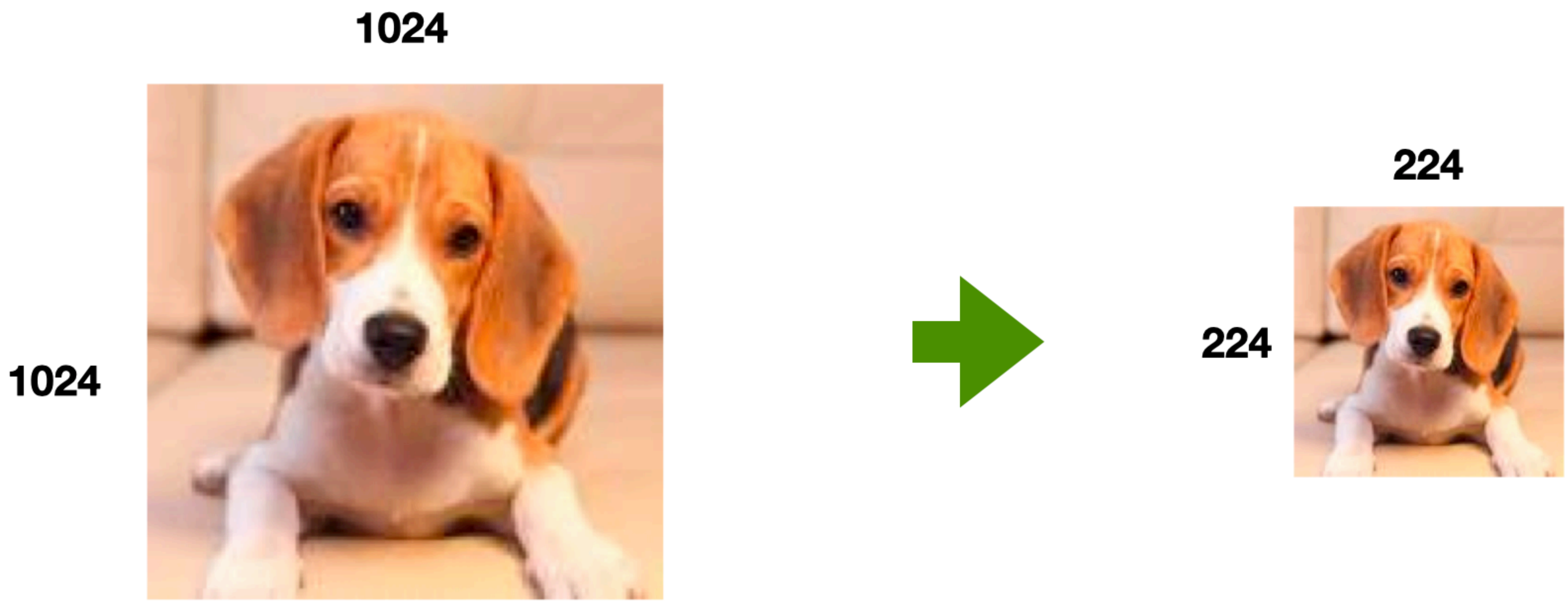
How Computer Vision models work?

Course of dimensionality

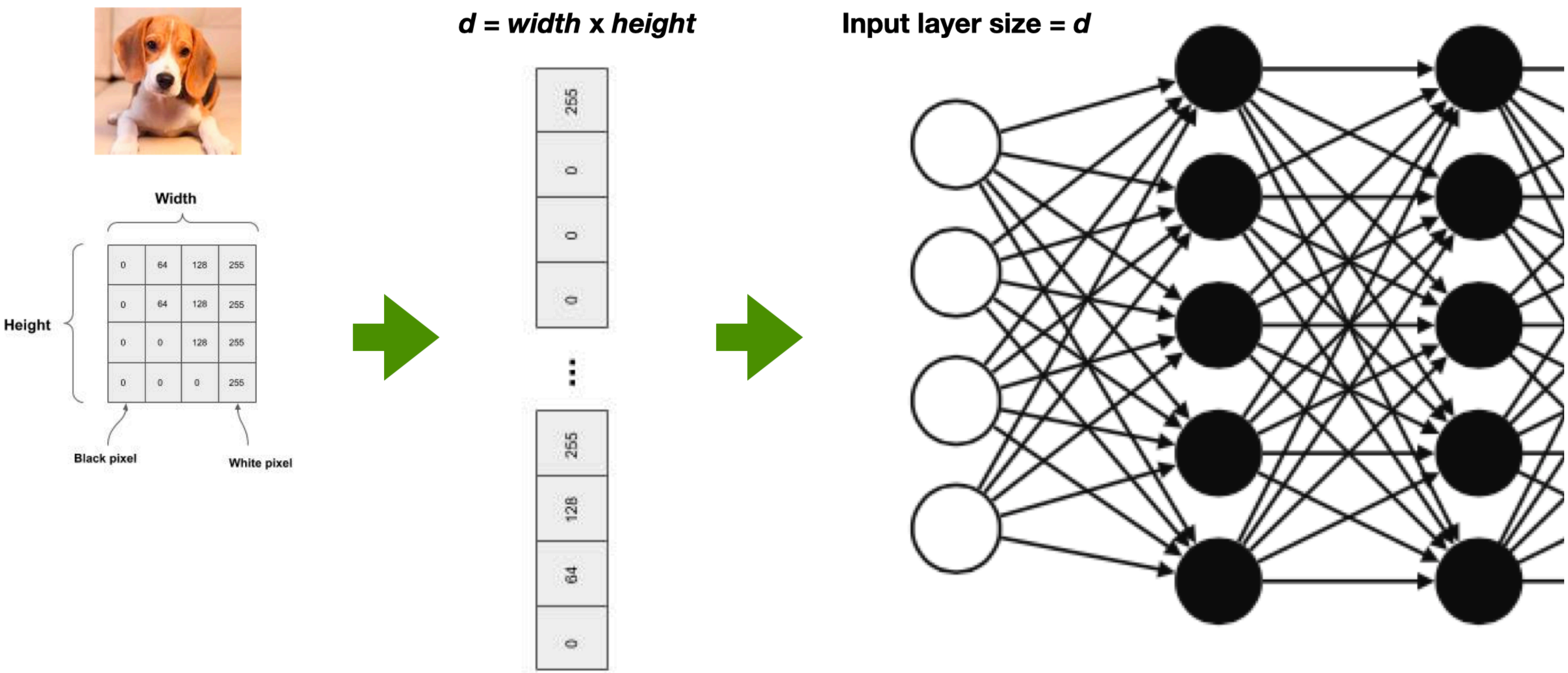
- High dimensionality
  - A 1024×768 image has $d = 786432$!
  - A tiny 32×32 image has $d = 1024$
- Decision boundaries in pixel space are extremely complex
- We will need "big" ML models with lots of parameters
  - For example, linear regressors need d parameters



Downsampling

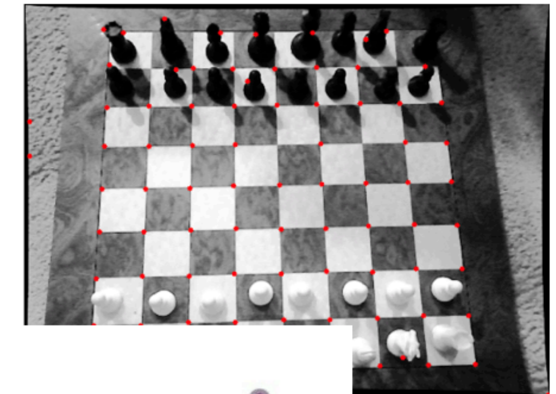


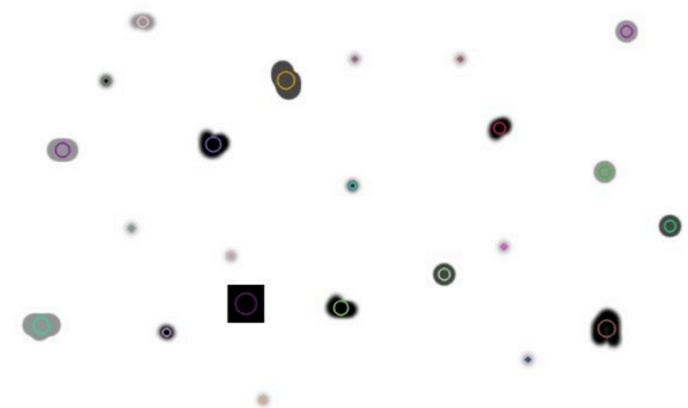
Flattening

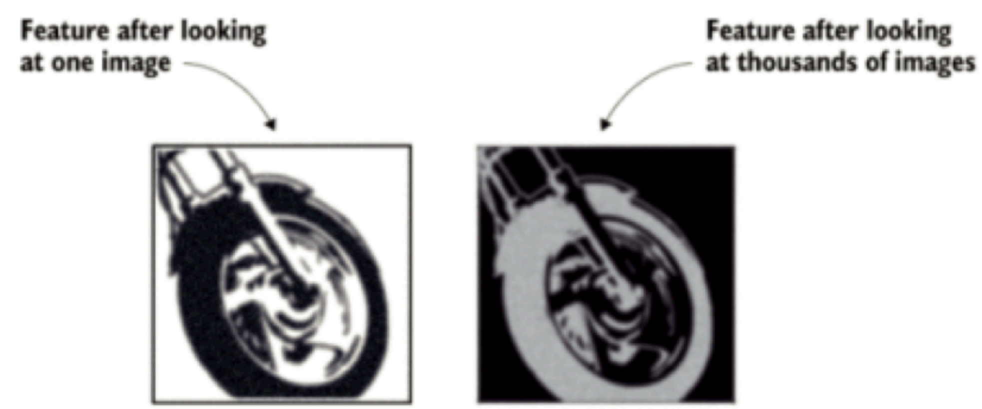

d = width x height
Input layer size = d
Width
Height
0 64 128 255
0 64 128 255
0 0 128 255
0 0 0 255
Black pixel
White pixel
255
0
0
0
...
255
128
64
0

The “old days”: Feature Extraction

- **Feature**
  - A relevant piece of information about the content of an image
  - e.g., edges, corners, blobs (regions), ridges
- **A good feature**
  - Repeatable
  - Identifiable
  - Can be easily tracked and compared
  - Consistent across different scales, lighting conditions, and viewing angles
  - Visible in noisy images or when only part of an object is visible
  - Can distinguish objects from one another

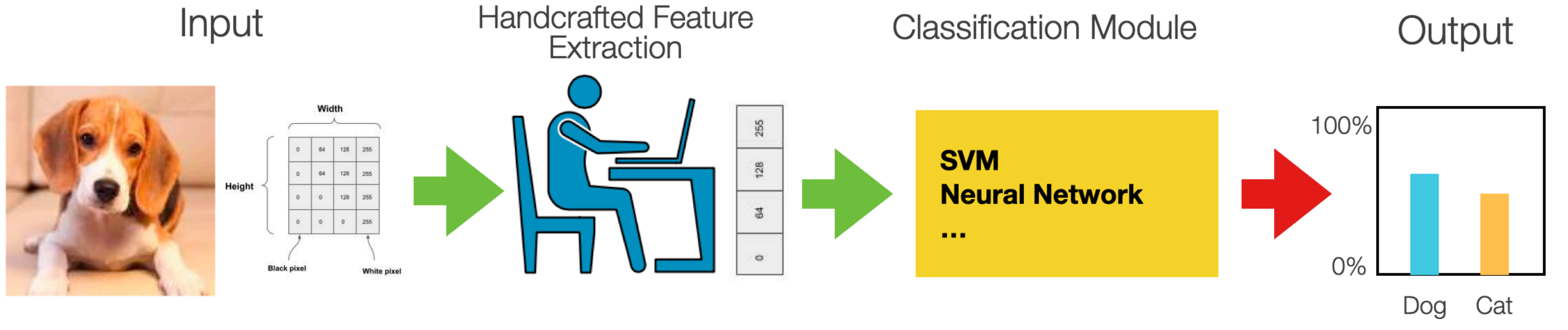






The “old days”: Feature Engineering

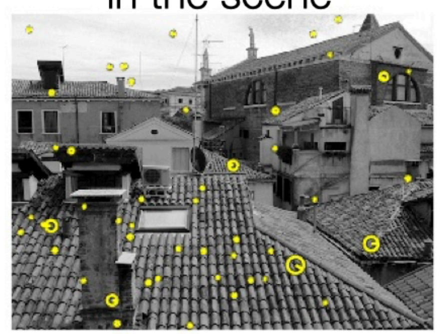
- Machine learning models are only as good as the features you provide
- To figure out which features you should use for a specific problem
- Rely on domain knowledge (or partner with domain experts)
- Experiment to create features that make machine learning algorithms work better



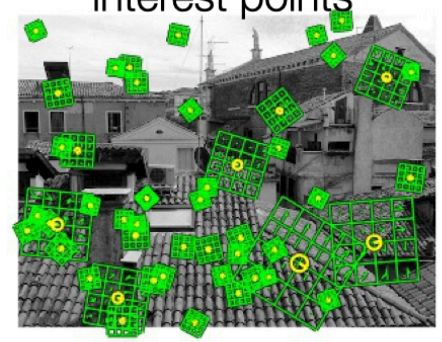
Feature Extraction Techniques

Scale-Invariant Feature Transform (SIFT)

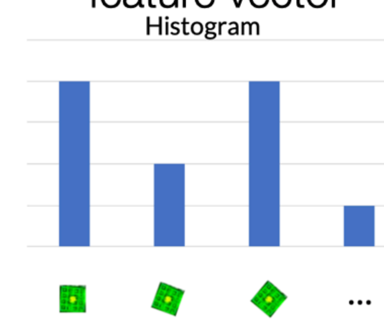
Find "interest points" in the scene

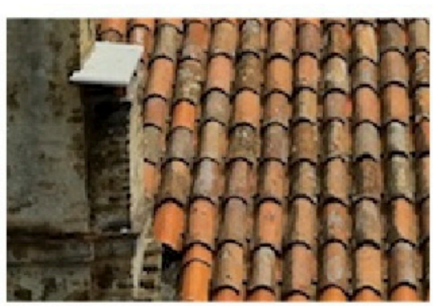


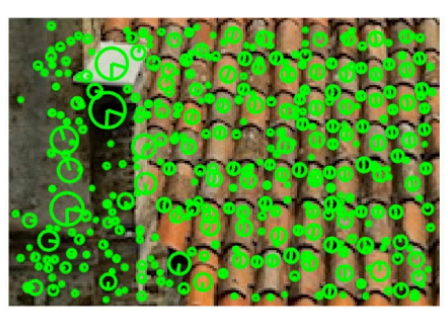
Compute features at interest points



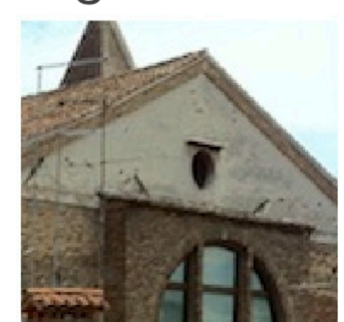
Convert to fixed-dimensional feature vector

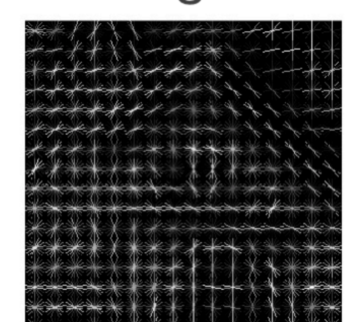






Histogram and oriented gradients



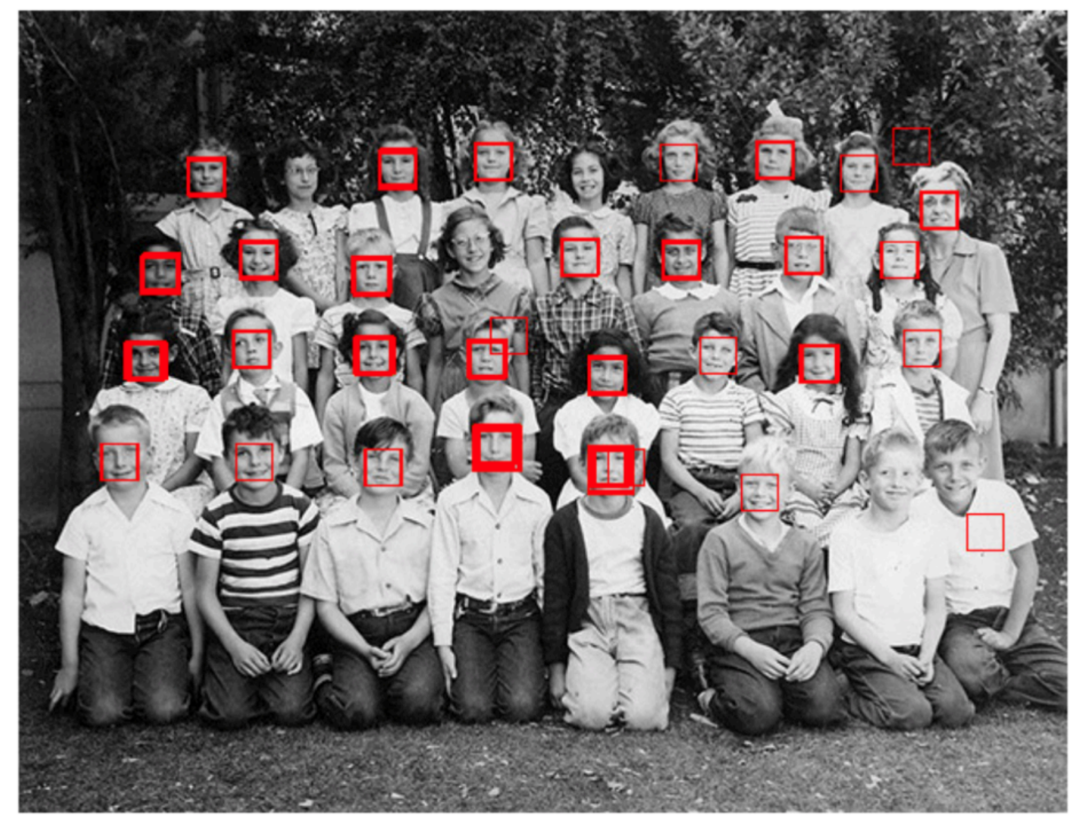


Performance

Object Detection (~2007)

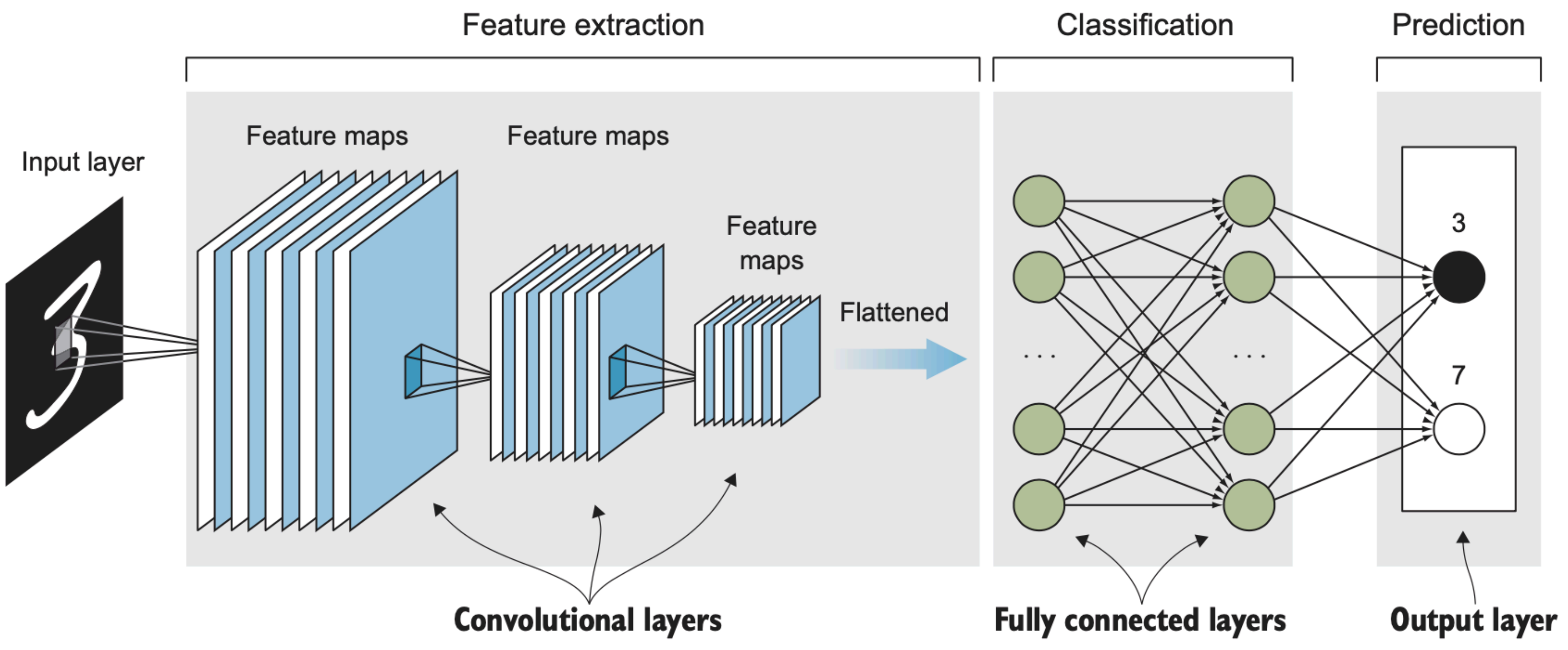
Felzenszwalb, Ramanan, McAllester. A Discriminatively Trained, Multiscale, Deformable Part Model. CVPR 2008 (DPM v1)

Face Detection (~2013)



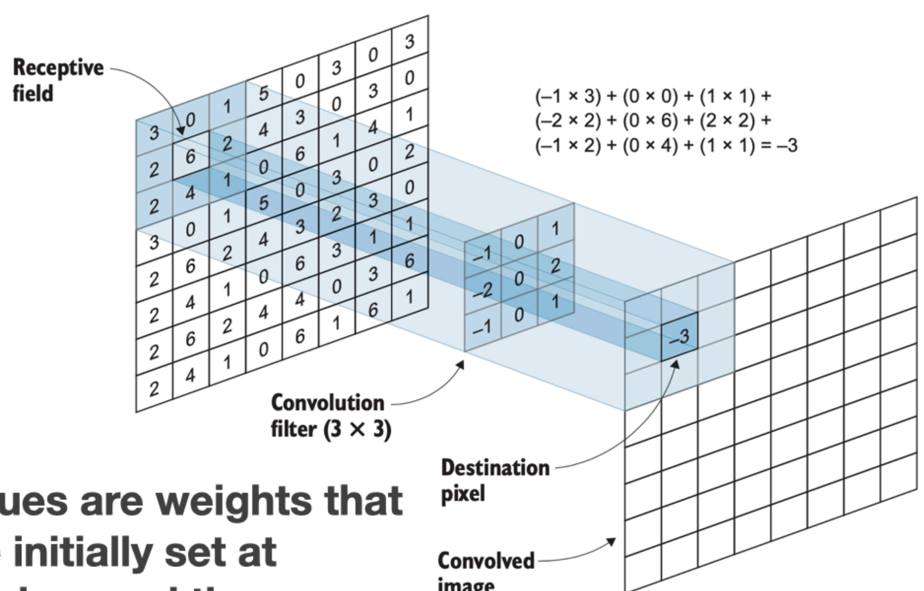
https://github.com/alexdemartos/ViolaAndJones

Convolutional Neural Networks

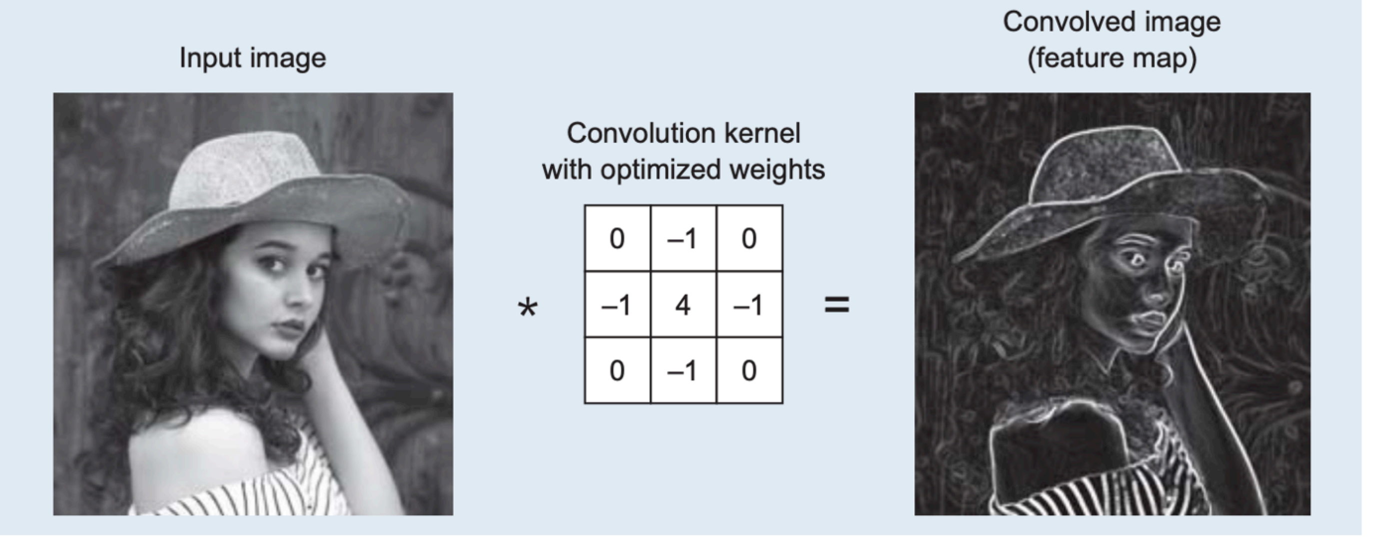


- CNNs exploit image properties to reduce the number of model parameters drastically
- Feature maps
  - Automatically extracted hierarchical features
  - Retain spatial association between pixels
- Local interactions
  - All processing happens within tiny image windows
  - Within each layer, far-away pixels cannot influence nearby pixels
- Translation invariance
  - A dog is a dog even if its image is shifted by a few pixels

Convolution & Feature Maps

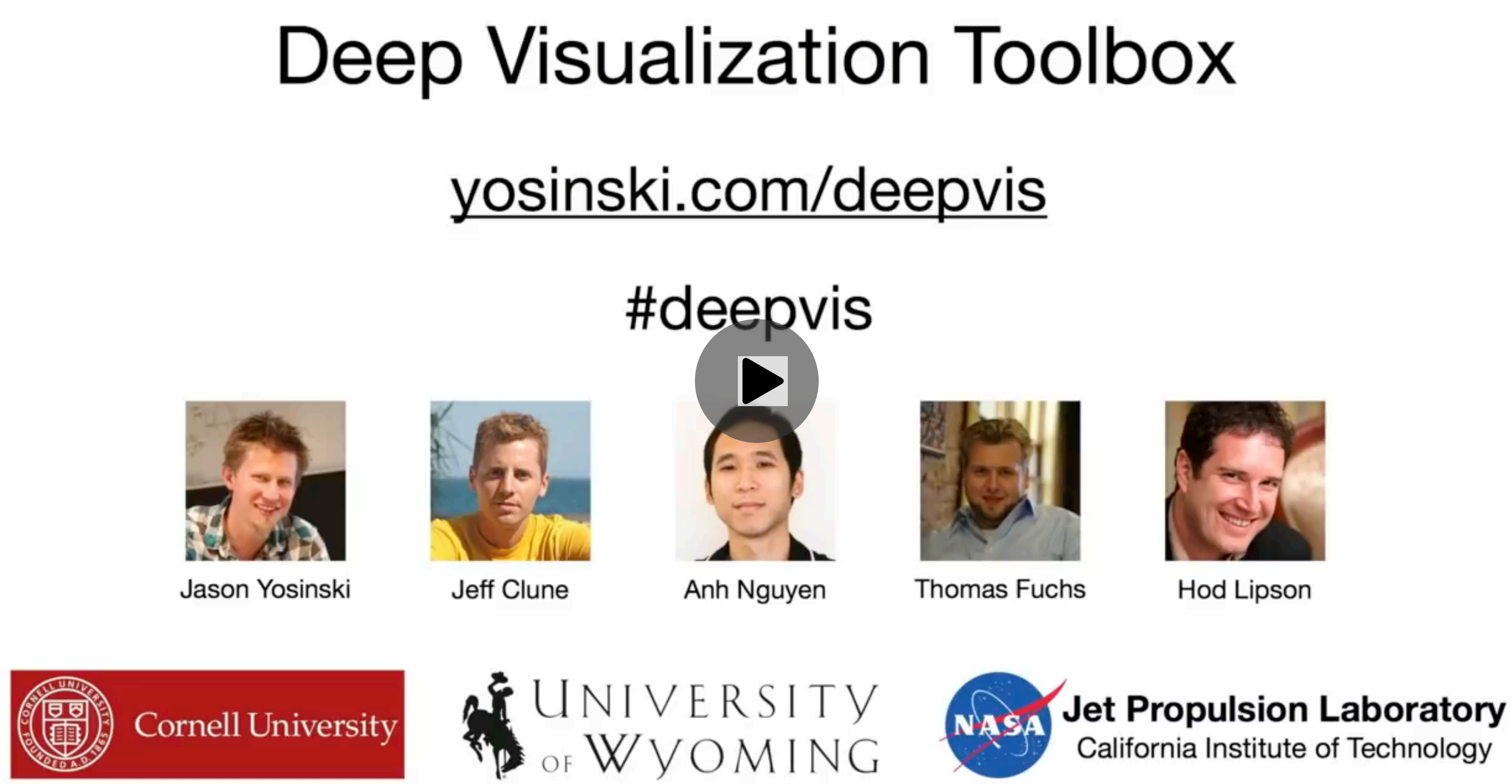


Values are weights that are initially set at random and then learned

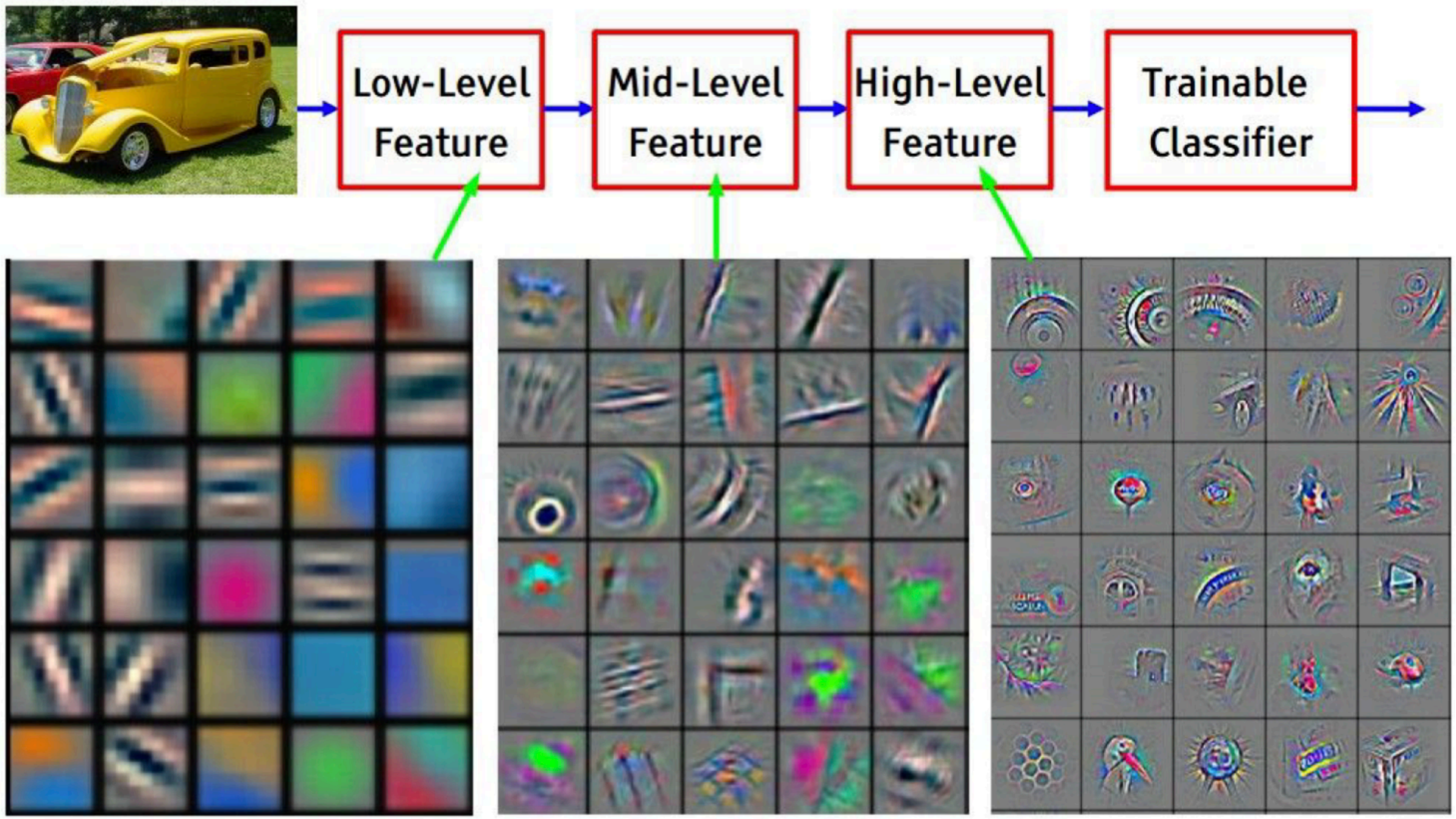


What CNNs learn?

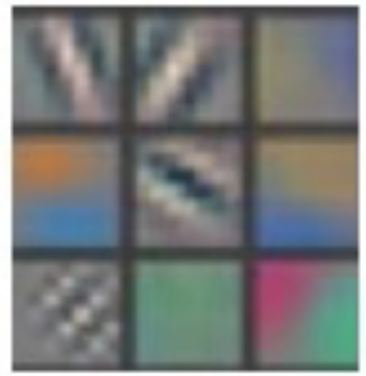
Deep Visualization Toolbox



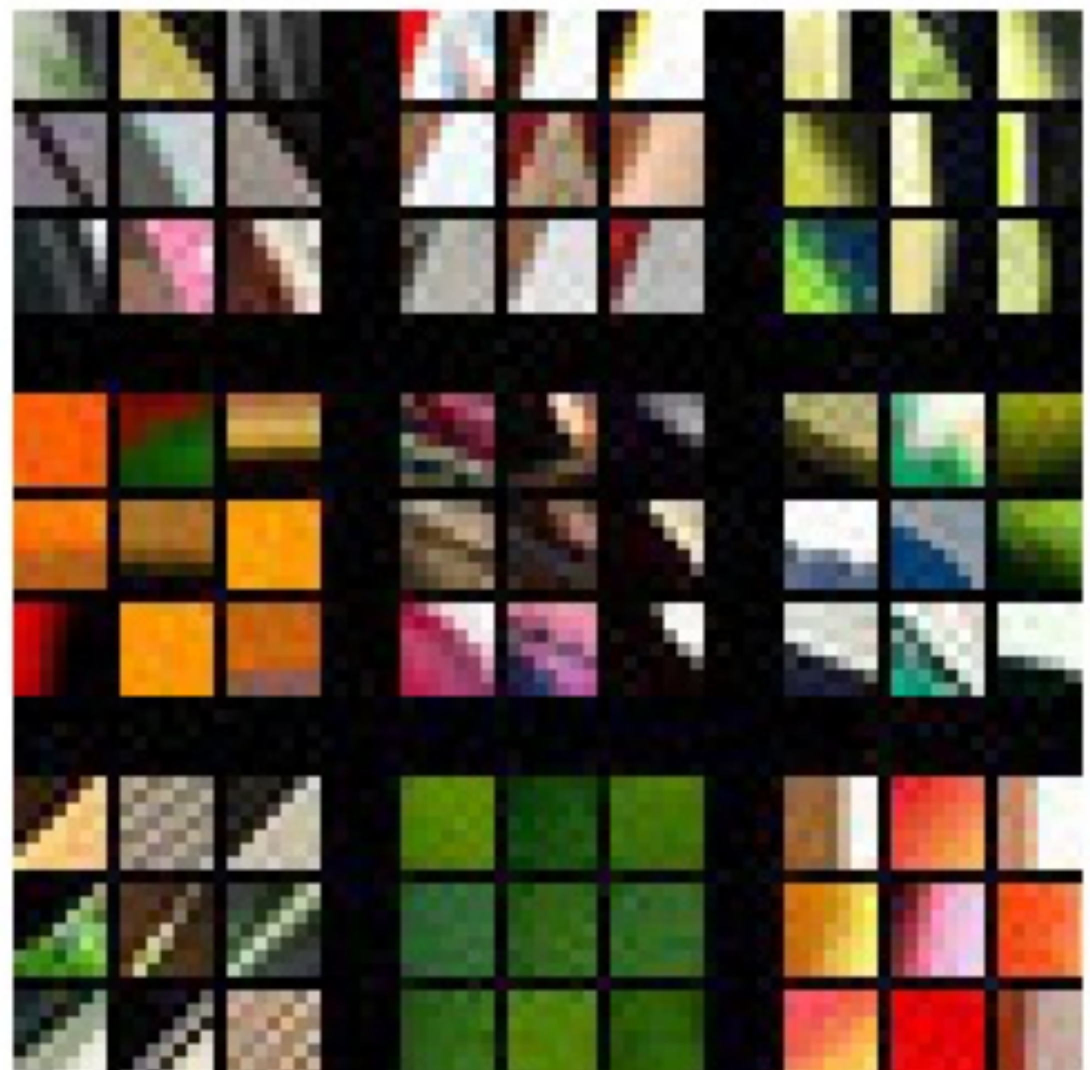
Feature Visualisation

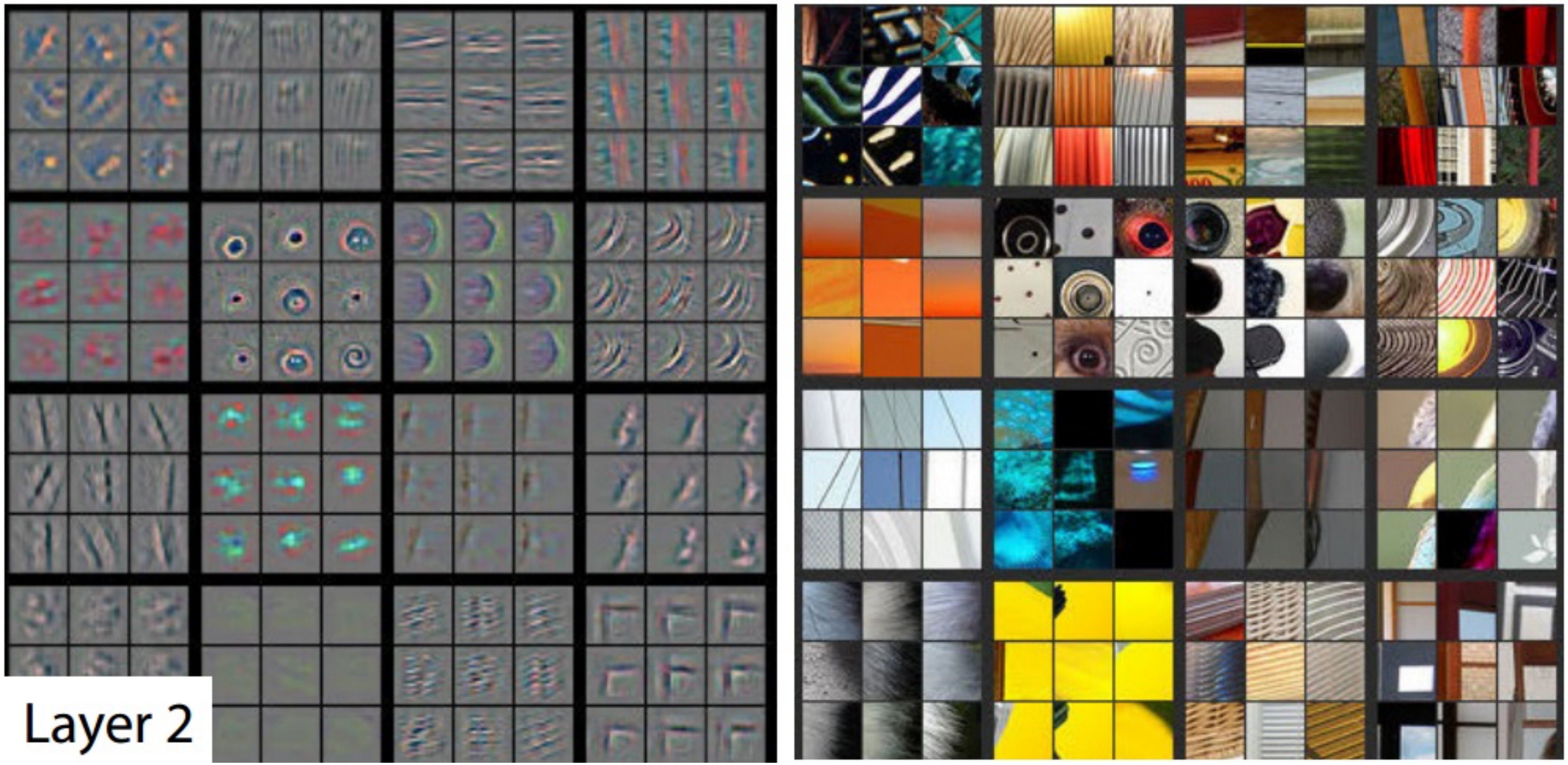


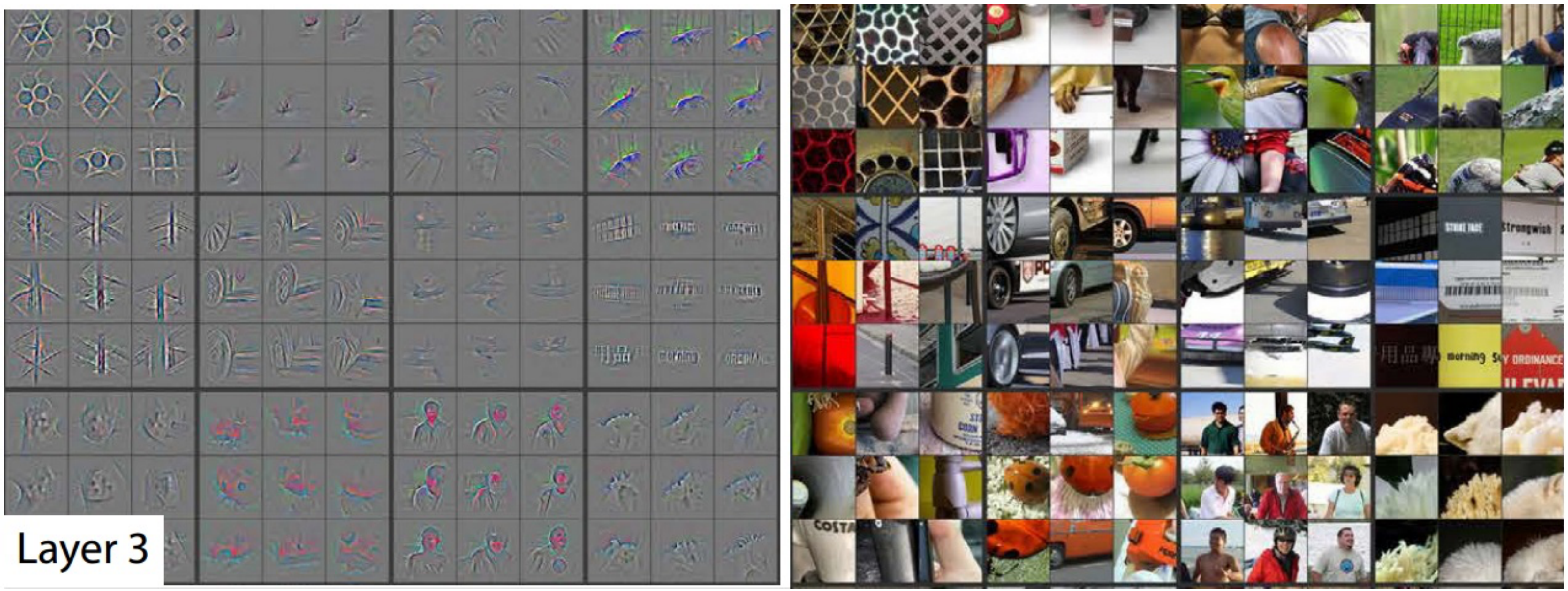
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



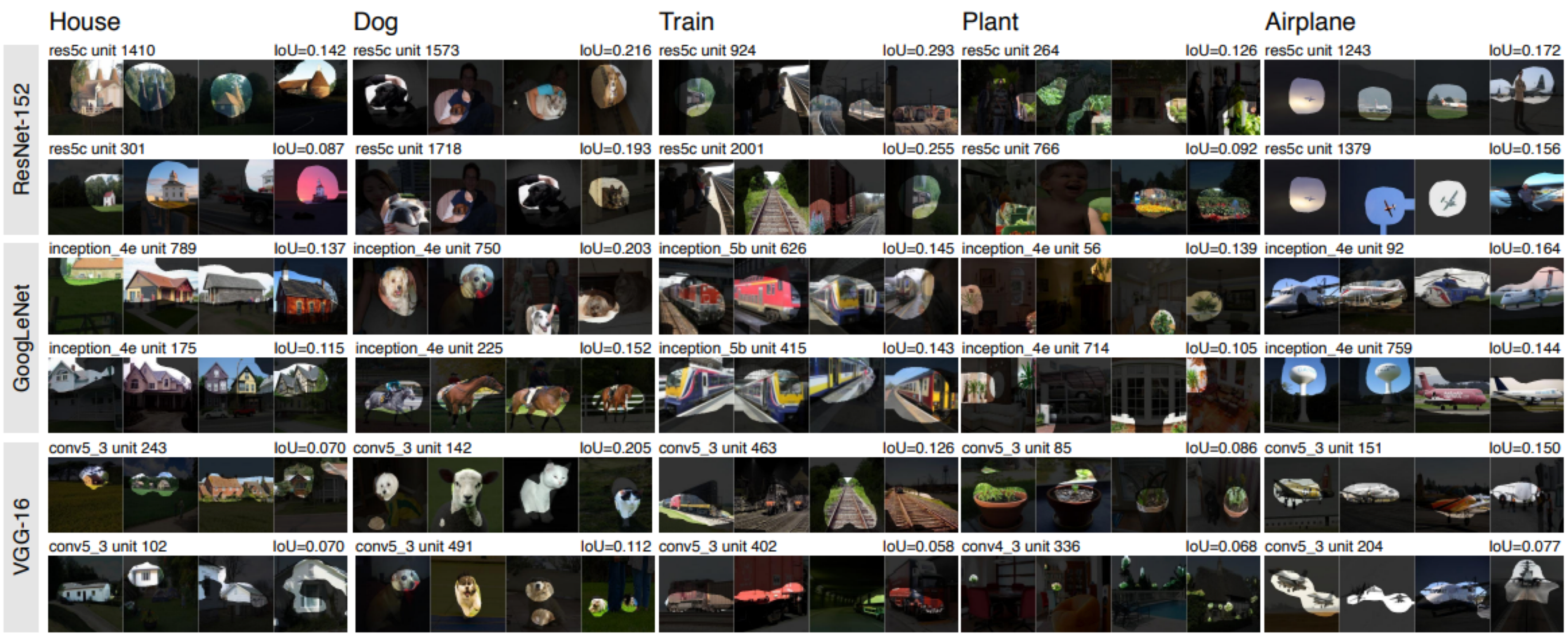
Layer 1



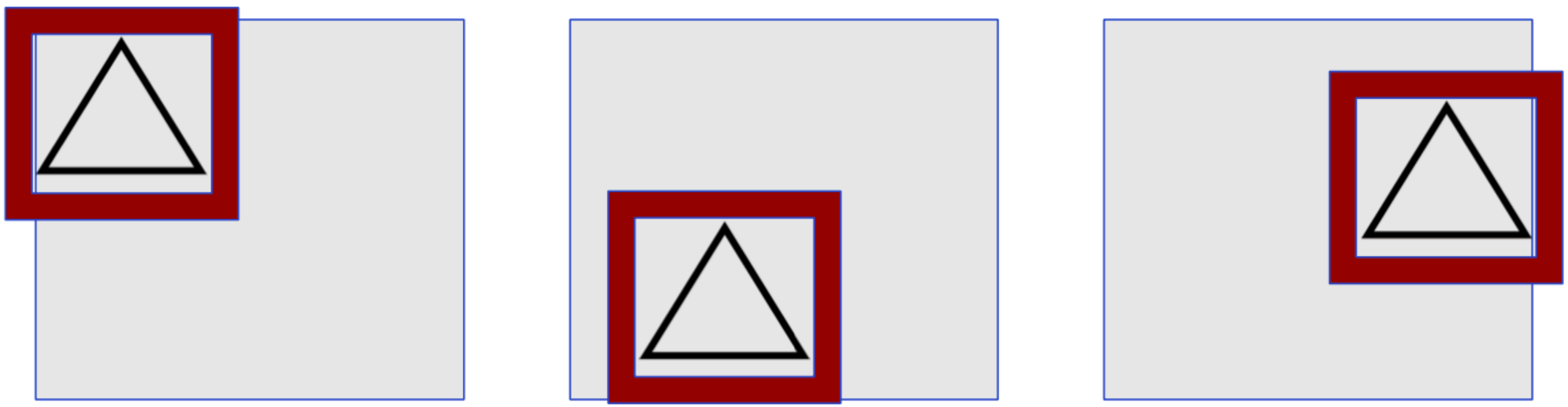

Layer 2


Layer 3

Network Dissection

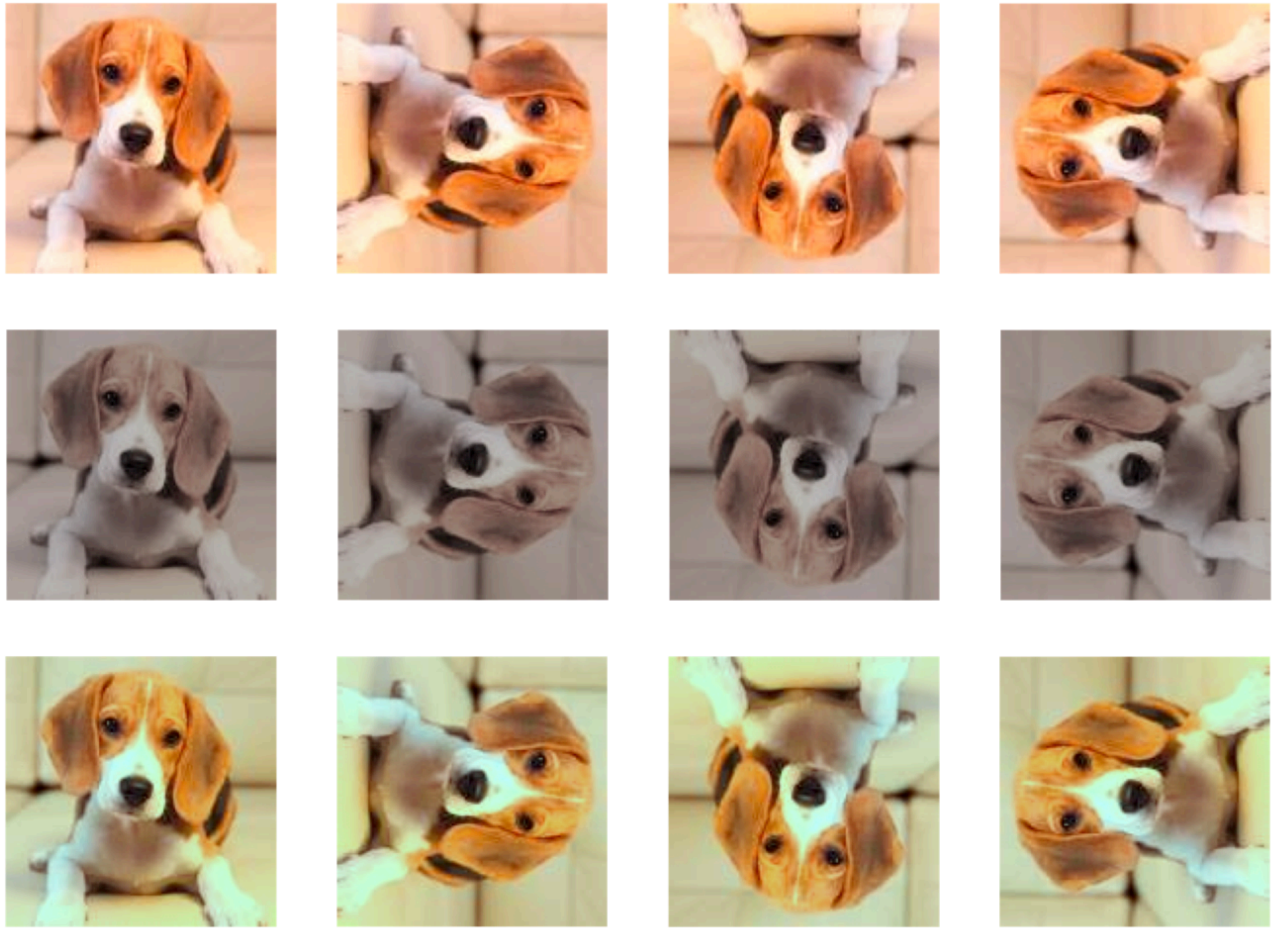


Translation Invariance



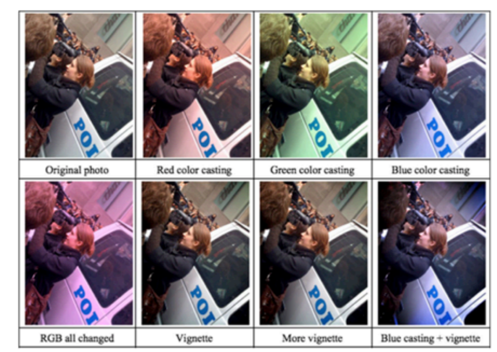
But not rotation and scaling invariance!

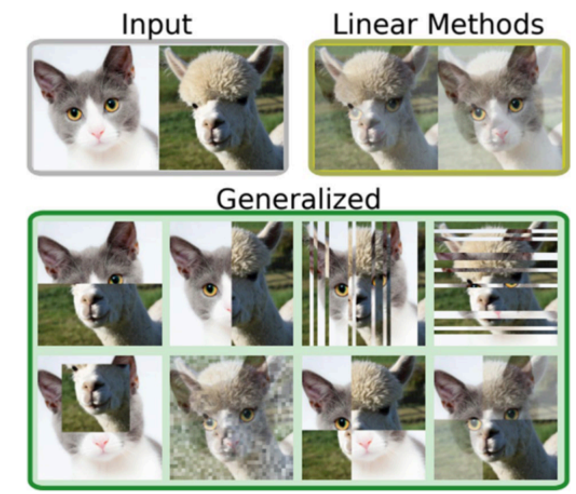
What about generalisation?

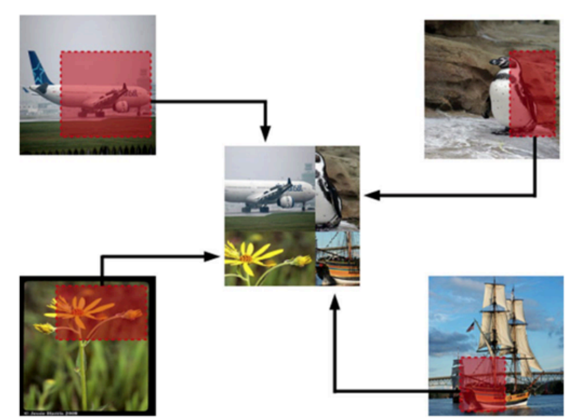


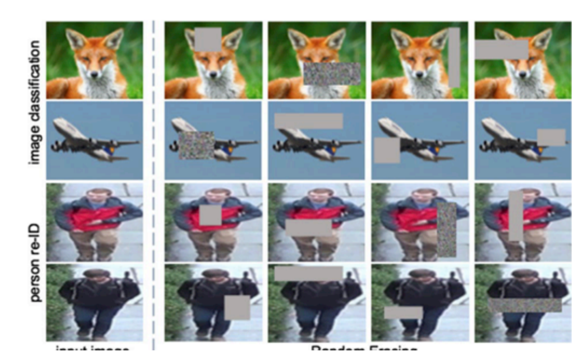
Data Augmentation

- Generate variations of the input data
  - To improve generalisability (out-of-distribution inputs)
  - Improve invariance (rotation, scaling, distortion)



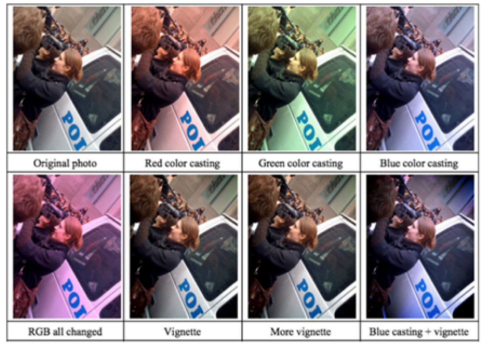


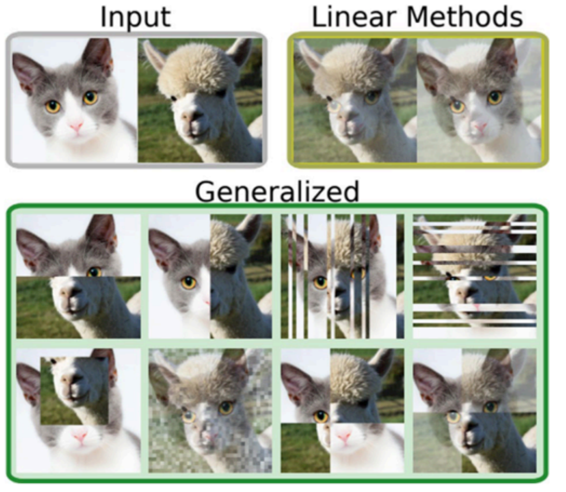


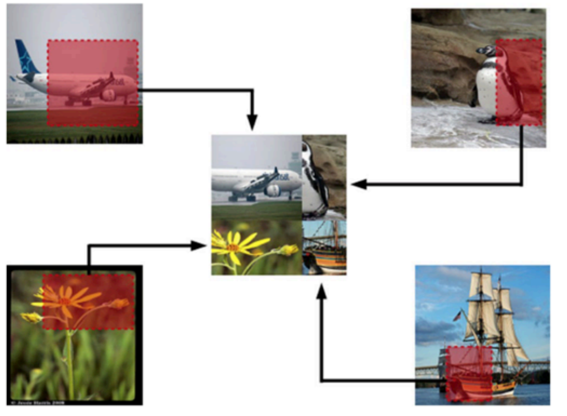


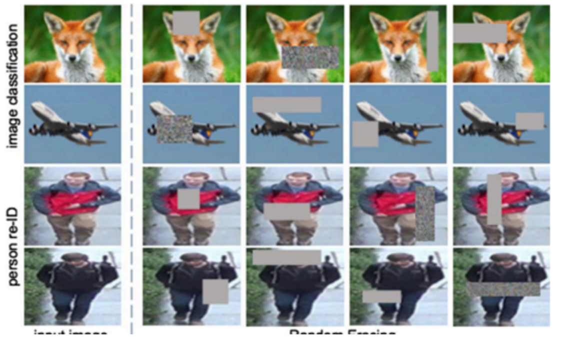
Data Augmentation

- Geometric
  - Flipping, Cropping, Rotation, Translation,
- Noise Injection
- Color space transformation
- Mixing Images
- Random erasing
- Adversarial training
- GAN-based image generation





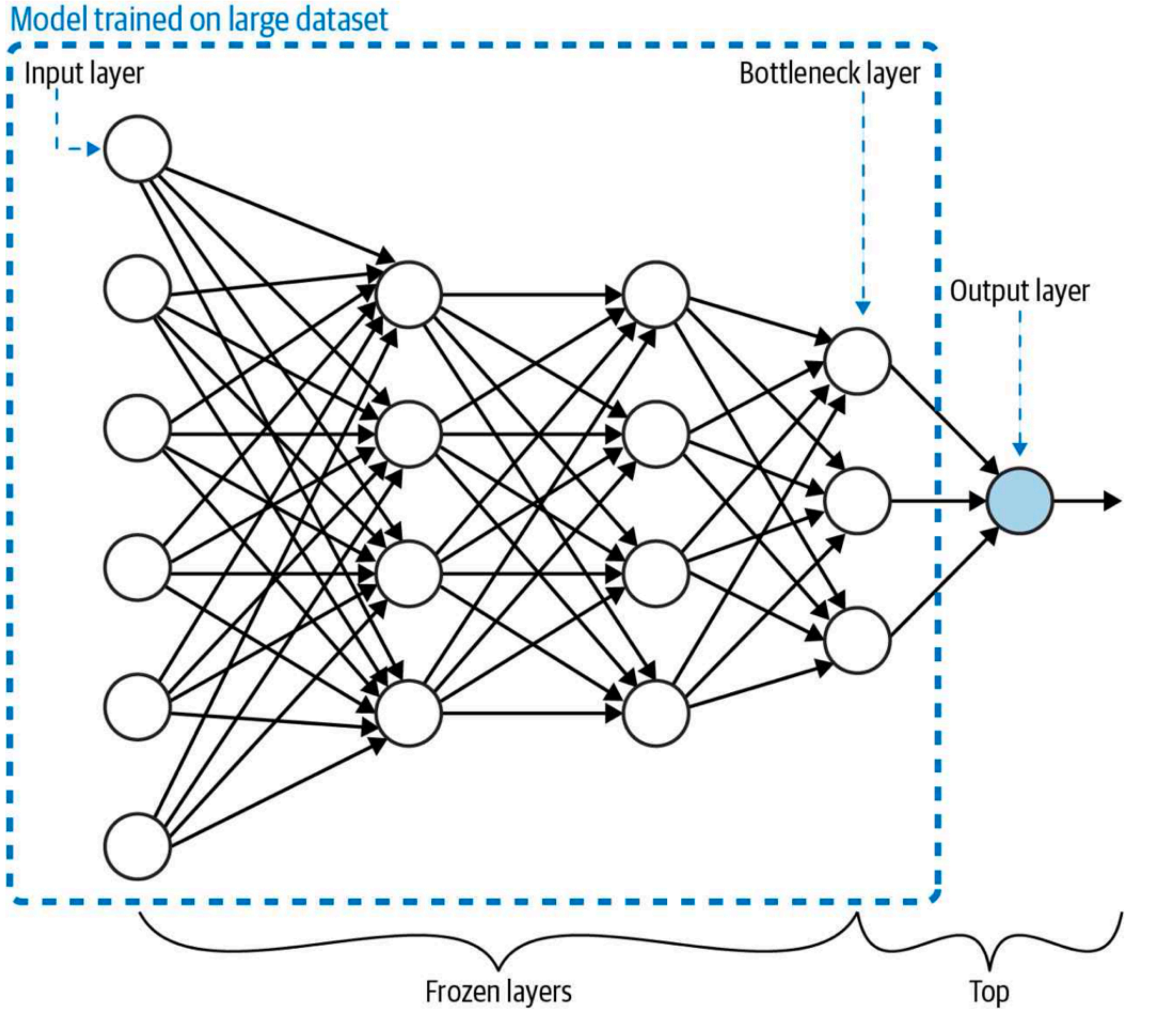




Robustness to input variation

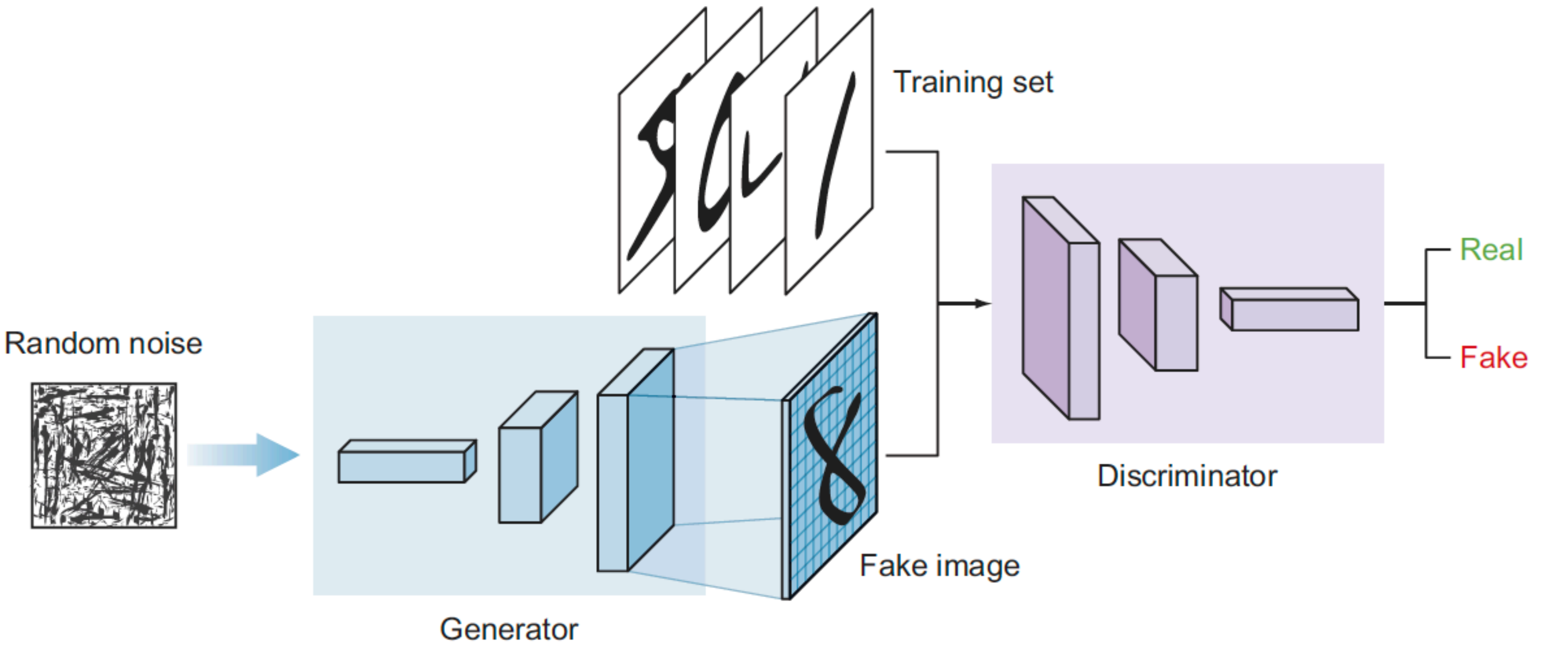
Transfer Learning

- **Problem**: training custom ML models requires huge datasets
- **Transfer learning**: take a model trained on the same data type for a similar task and apply it to a specialised task using our custom data.
  - **Same data**: same data modality. same types of images (e.g., professional pictures vs. Social media pictures)
  - **Similar tasks**: if you need a new object classification model, use a model pre-trained for object classification



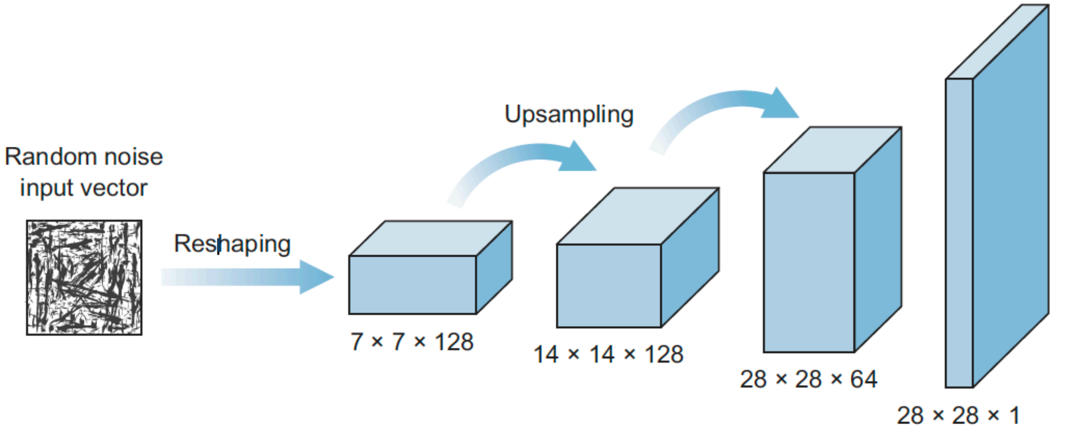
Advanced Computer Vision Techniques

Generative Adversarial Networks



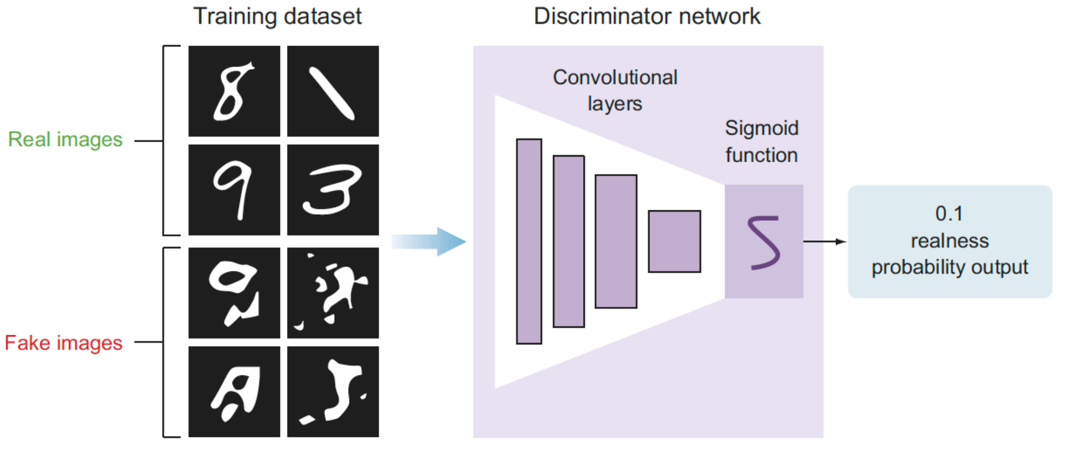
- Learn patterns from the training dataset and create new images that have a similar distribution of the training set
- Two deep neural networks that compete with each other
  - The **generator** tries to convert random noise into observations that look as if they have been sampled from the original dataset
  - The **discriminator** tries to predict whether an observation comes from the original dataset or is one of the generator’s forgeries

- The **generator**'s architecture looks like an inverted CNN that starts with a narrow input and is upsampled a few times until it reaches the desired size



- The **discriminator**

's model is a typical classification neural network that aims to classify images generated by the generator as real or fake



Which face is real?

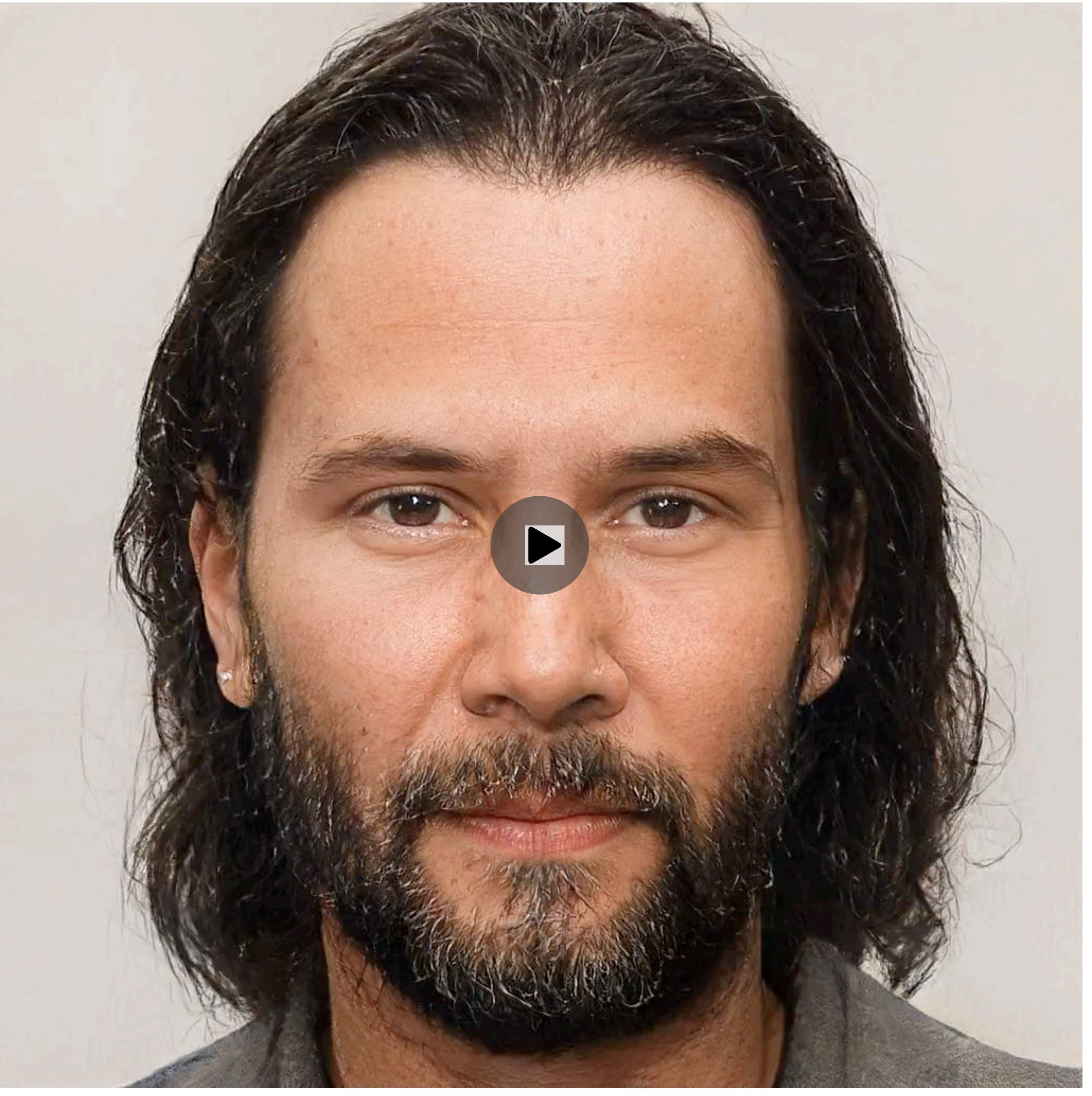
Image super-resolution GAN

- A good technical summary

- ML-generated painting sold for $432,500
- The network trained on a dataset of 15,000 portraits painted between the fourteenth and twentieth centuries
- Network “learned” the style and generated a new painting

Neural Style Transfer

Content Image Style Image Stylized Result

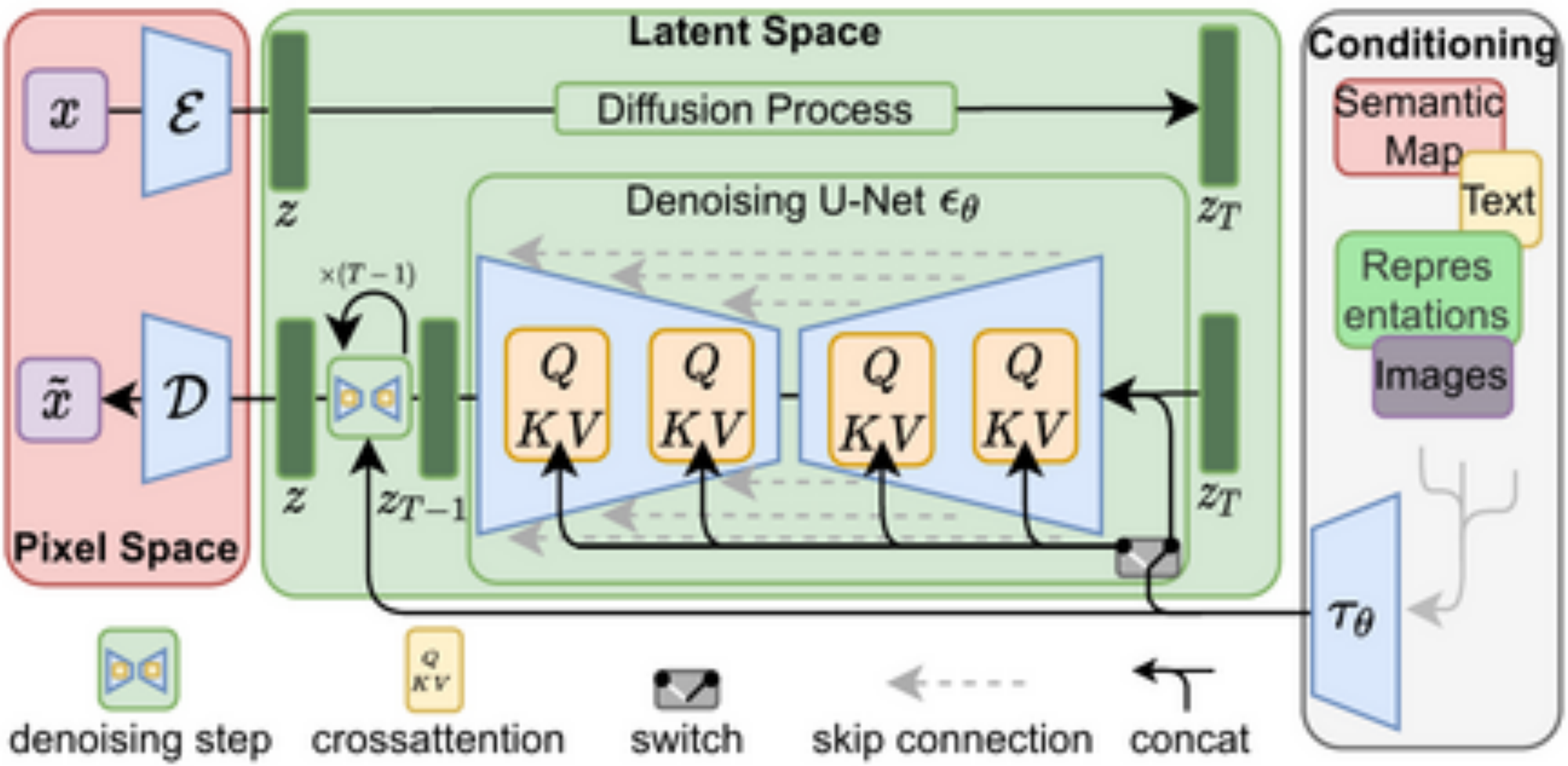


Text-To-Image Generation

'A street sign that reads "Latent Diffusion" ' | *'A zombie in the style of Picasso'* | *'An image of an animal half mouse half octopus'* | *'An illustration of a slightly conscious neural network'* | *'A painting of a squirrel eating a burger'* | *'A watercolor painting of a chair that looks like an octopus'* | *'A shirt with the inscription: "I love generative models!" '*

Design

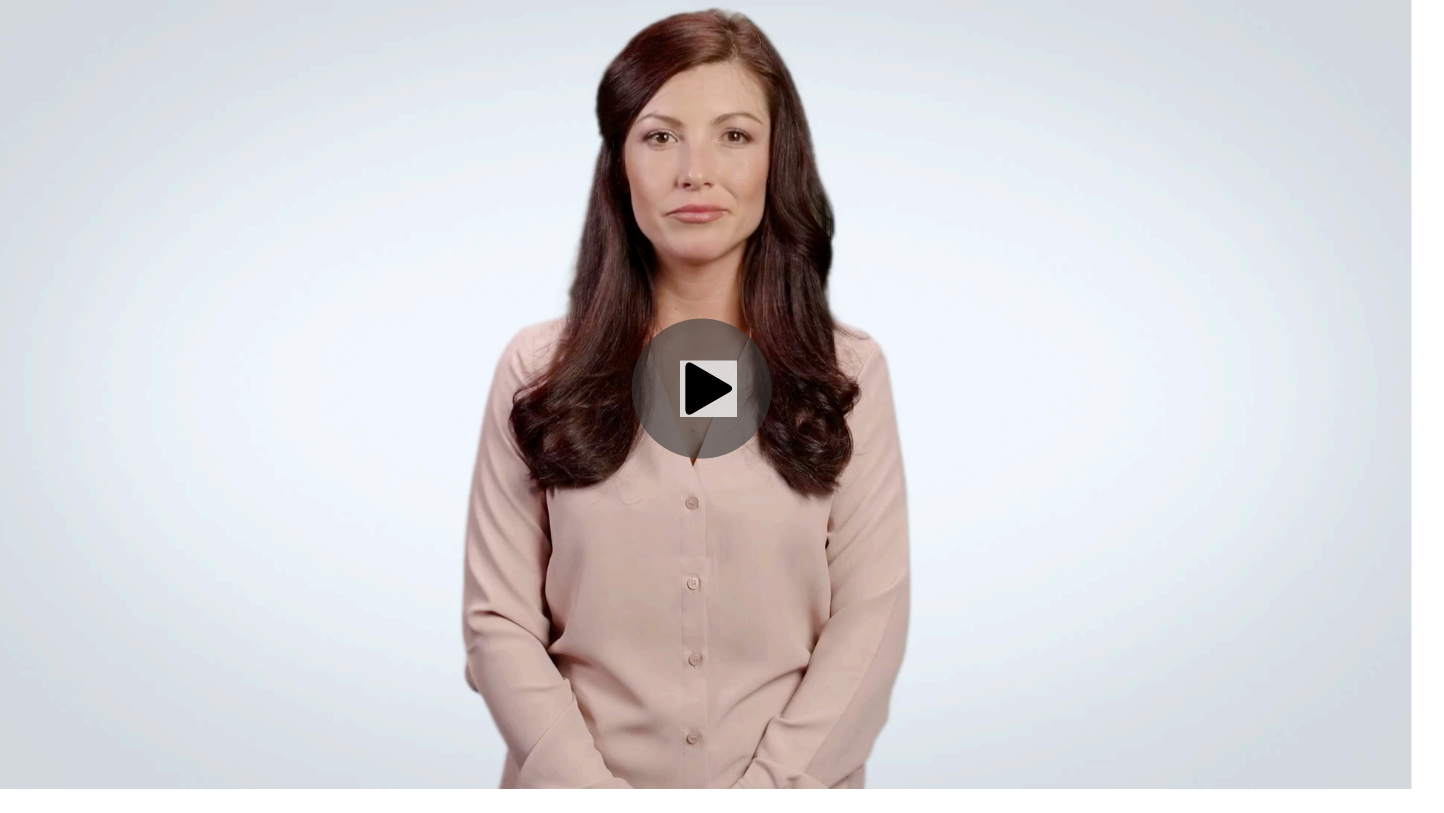
Computer Science



Latent Space
Conditioning
x
$\mathcal{E}$
Diffusion Process
Semantic Map
Text
Representations
Images
z
z_T
Denoising U-Net ϵ_θ
$\times(T-1)$
$\tilde{x}$
$\mathcal{D}$
z
z_{T-1}
Q $K\,V$
z_T
Pixel Space
τ_θ
denoising step
crossattention
switch
skip connection
concat

Image-to-Image Generation

Synthetic Video Generation



Generated from Synthesia.io

Deep Fakes

Very realistic Tom Cruise Deepfake

Machine Learning for Design

Lecture 4

Machine Learning for Images. *Part 2*

Credits

CMU Computer Vision course - Matthew O'Toole.

Grokking Machine Learning. Luis G. Serrano. Manning, 2021

[CIS 419/519 Applied Machine Learning]. Eric Eaton, Dinesh Jayaraman.

Deep Learning Patterns and Practices - Andrew Ferlitsch, Maanning, 2021

Machine Learning Design Patterns - Lakshmanan, Robinson, Munn, 2020

Deep Learning for Vision Systems. Mohamed Elgendy. Manning, 2020