VECTOR INSTITUTE

Bias in AI:
Week #2: Introduction to Modern Computer Vision

Instructor:
Sayyed Nezhadi

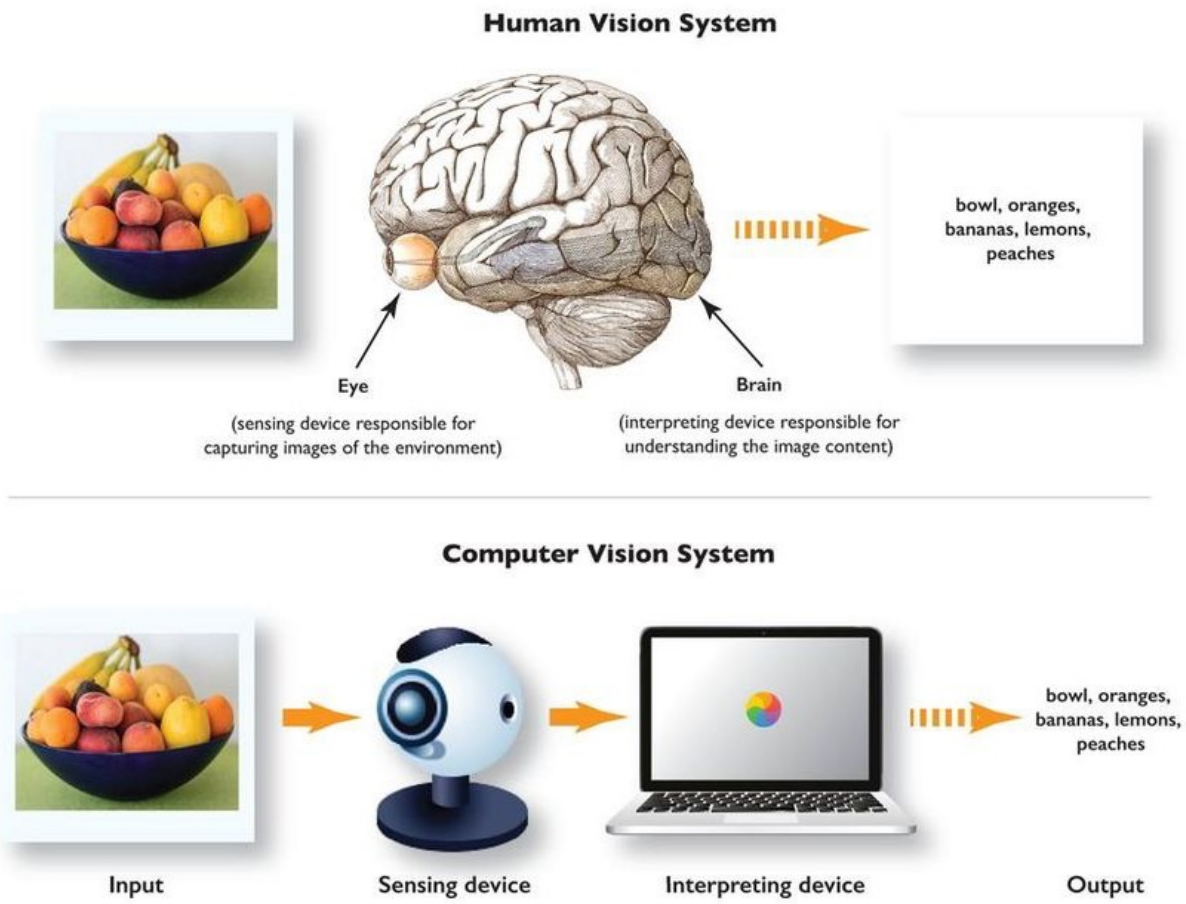
Summer 2022

Purpose of this Lecture

- This lecture is designed to give you a very quick overview of Computer Vision journey in the deep learning era.
- Rather than focusing on the state-of-the-art methods, I review the earlier and simpler methods to introduce you the thought process and basic concepts.
- For more modern architectures, you can refer to the reading materials.

Definition

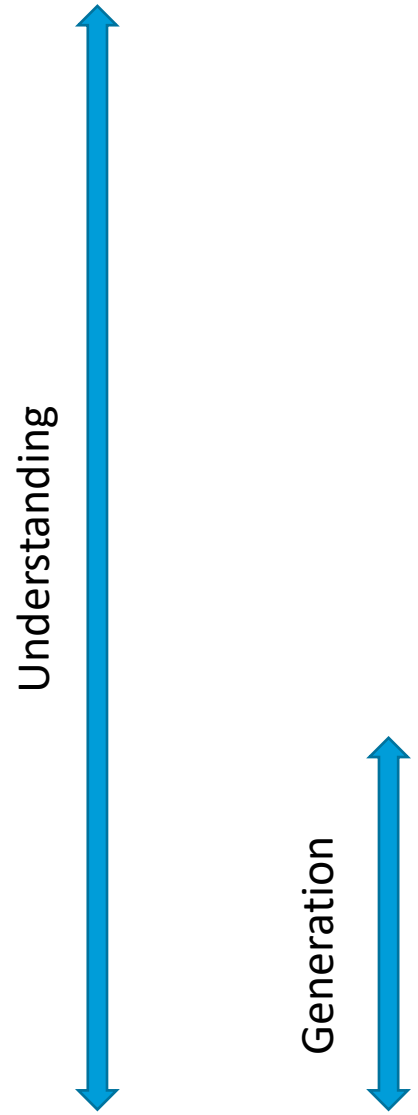
- **Computer Vision**, is the field of study surrounding how computers see and understand digital images and videos.
- **The goal:** to understand and automate tasks that the human visual system can do.
- The modern computer vision goes beyond that goal and works on generating images and videos too.



Source: Google image

Sample Computer Vision Applications

- Image Classification
- Object Detection
- Image Segmentation
- Autonomous Driving
- Face Detection
- Image Captioning
- Action Detection in Videos
- Object Tracking in Videos
- Visual Question and Answering
- Image/Video Generation
- DeepFake
- Super Resolution
- Image Repair
- Recolorization



Started with a Summer Project ☺

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

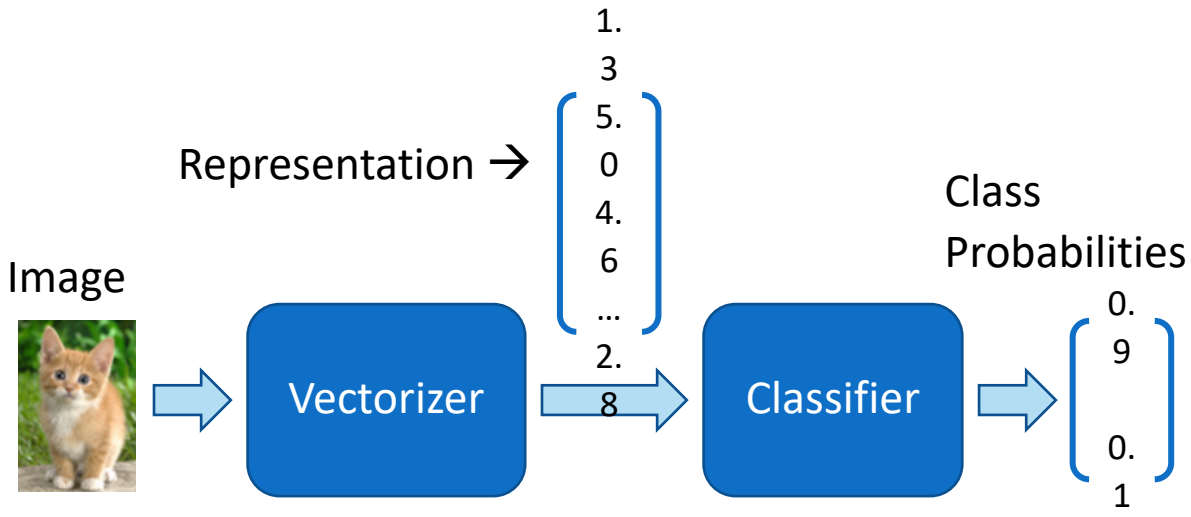
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Image Classification

To use a classifier, we need to represent the text by a vector (Vectorization):

- How to represent an image?



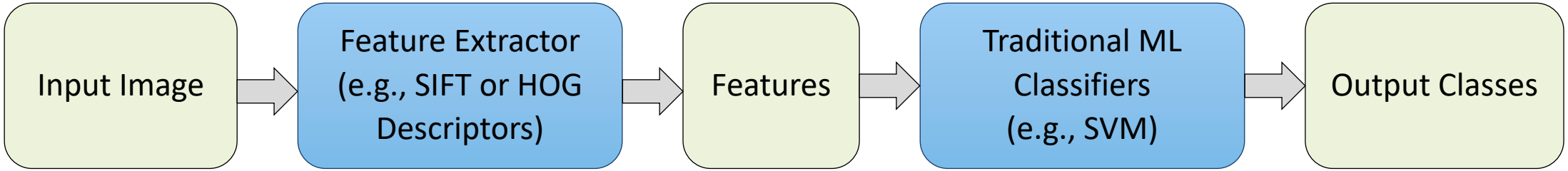
Dog: 96%
Cat: 29%
Duck: 2%
Bird: 0%

Dog: 36%
Cat: 94%
Duck: 2%
Bird: 1%

Source: https://www.oreilly.com/library/view/deep-learning-for/9781788295628/d9eef68b-4586-472c-bd5c-a244471a277f.xhtml

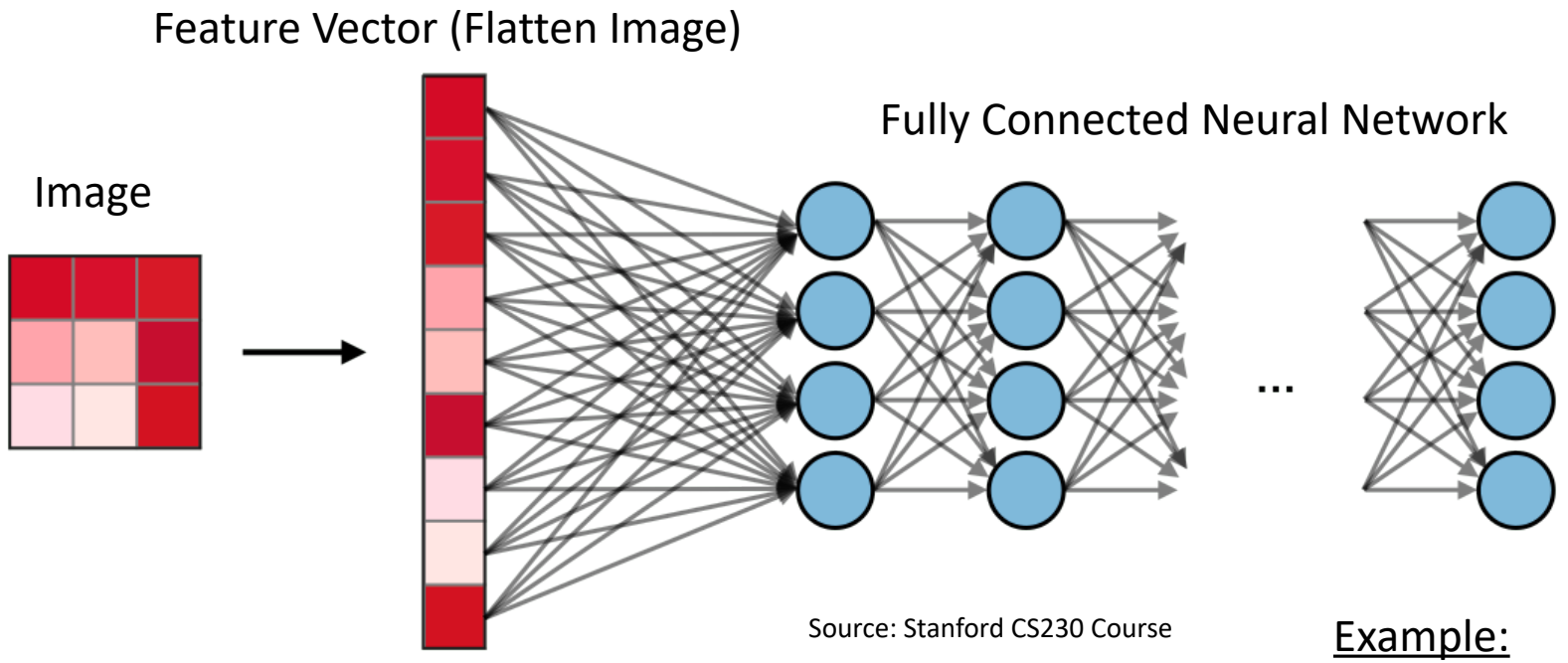
Traditional Computer Vision

- Using pre-set/generic image descriptors to extract features that had the following problems:
  - The accuracy of classification wasn't very high
  - One fixed way of feature extraction was used for every domain
  - Not able to use CPU accelerators for real-time processing
- In deep learning, we prefer to learn the feature extraction algorithm



Using Raw Image as the Feature

- The simplest idea is to flatten the raw image to use as the feature vector:



Source: Stanford CS230 Course

Problems:

- High number of inputs -> huge number of parameters to learn
- Spatial relationship is lost

Example:

Image size: 200x200x3

Hidden layer 1 : 4000 neurons

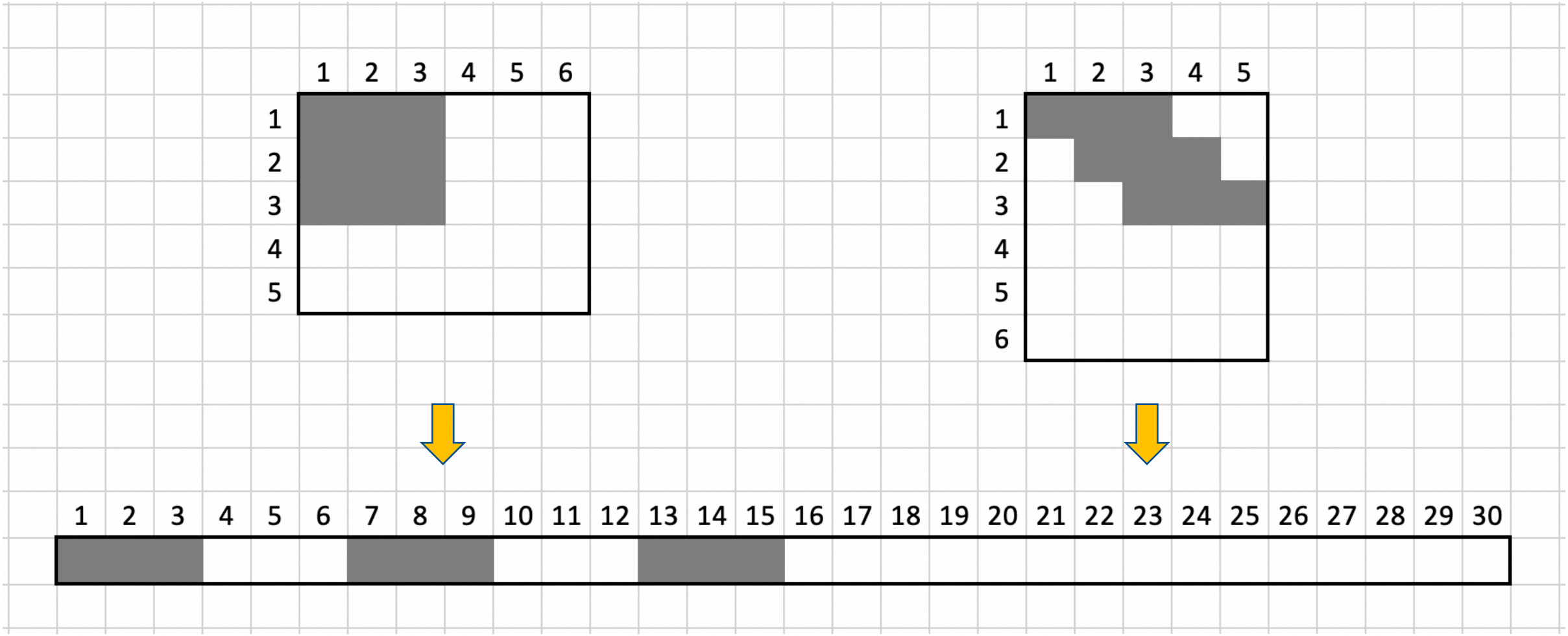
Hidden layer 2 : 1000 neurons

Output: 100 classes

# of params: ≈ 485 millions

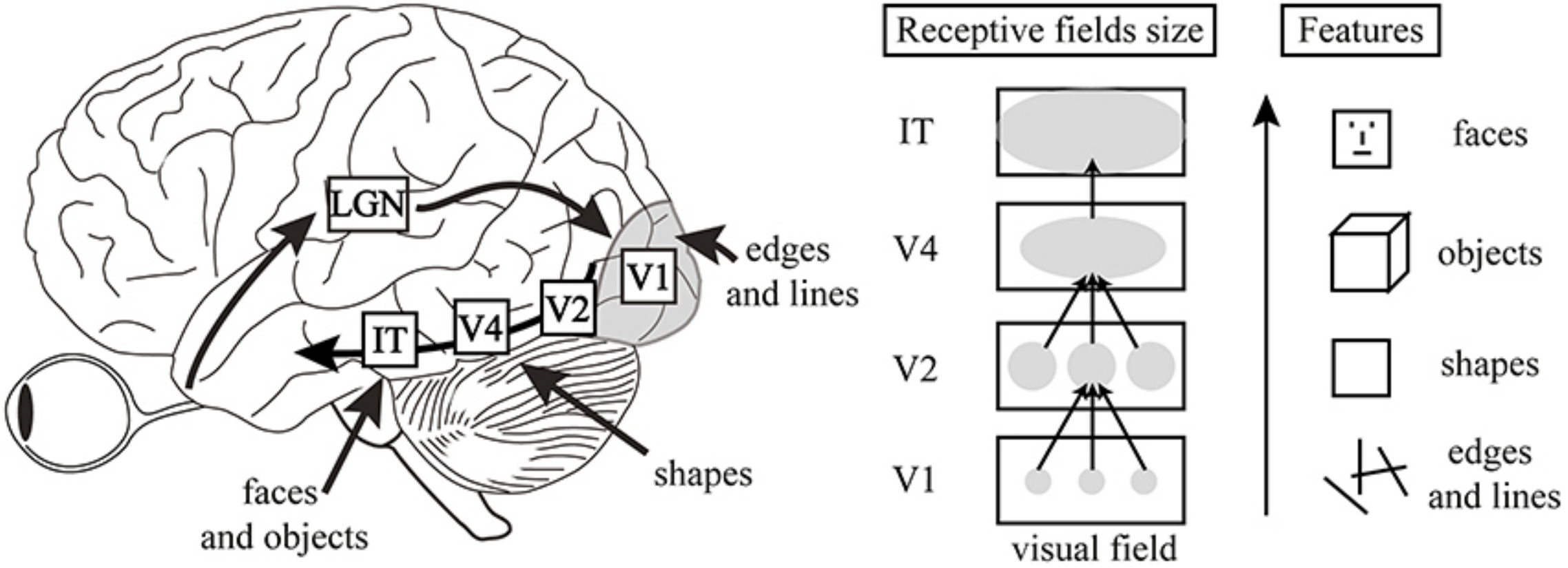
Spatial Relationship Matters

- Two different shapes but with the same flatten vector:



How Human's Brain Does It?

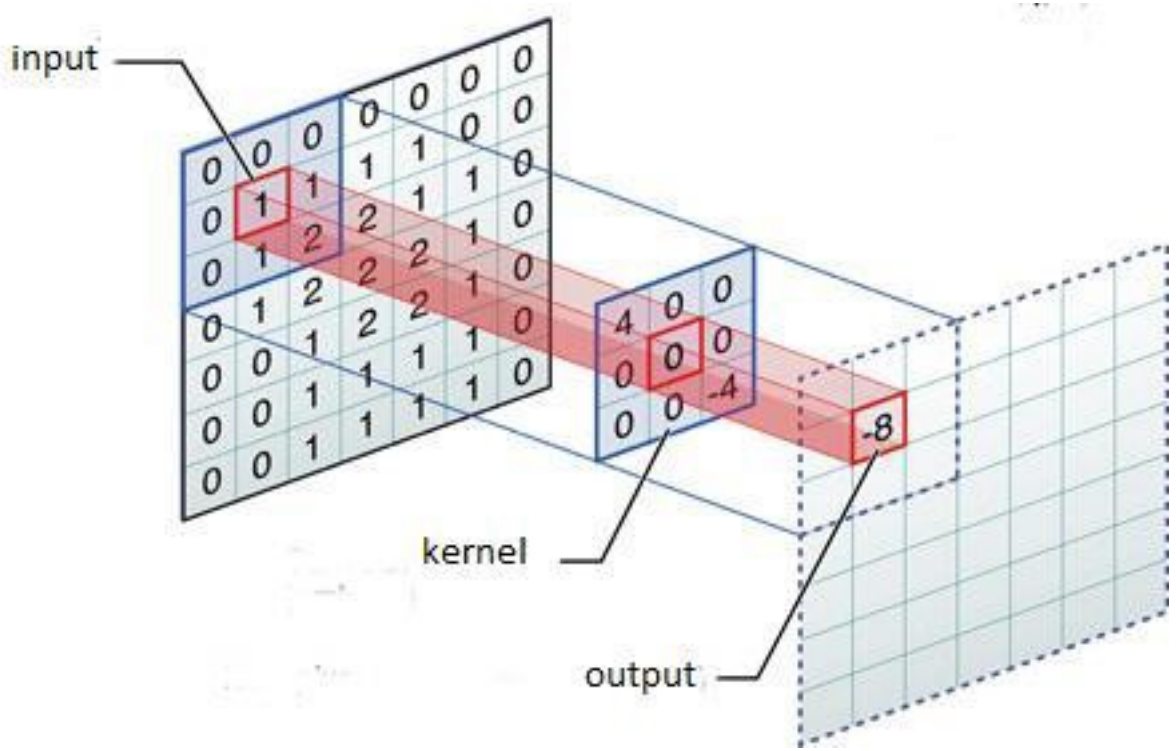
- The visual system has convolution property and multi-layer structure:

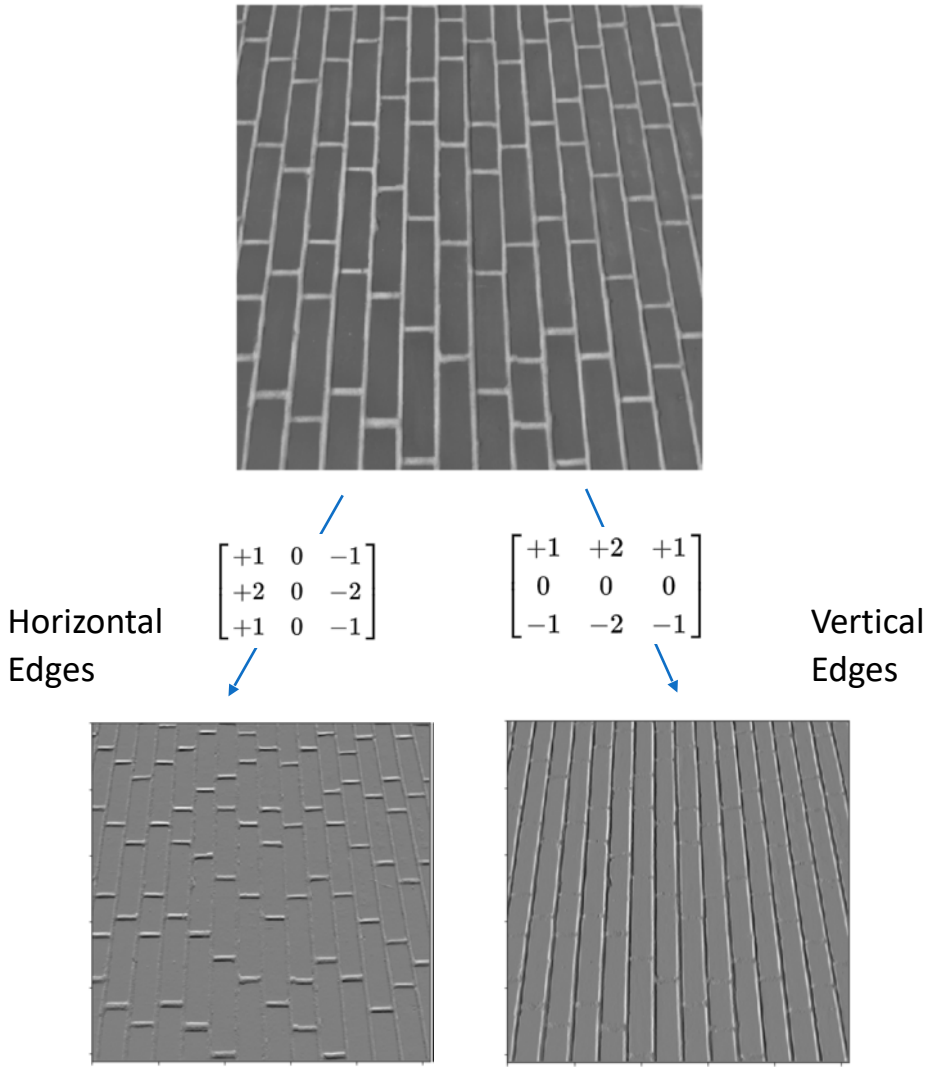


Source: Google image

Convolution & Filters

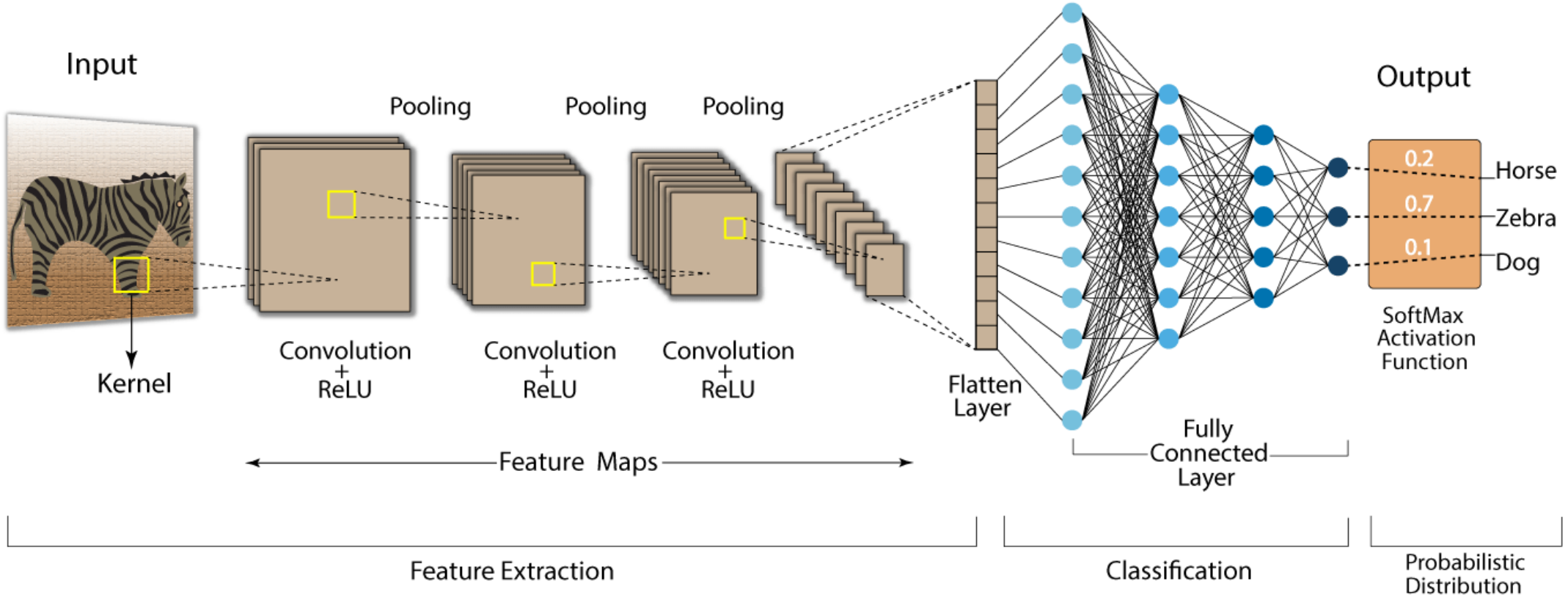
Applying a sliding 2D filter on the image in both directions:





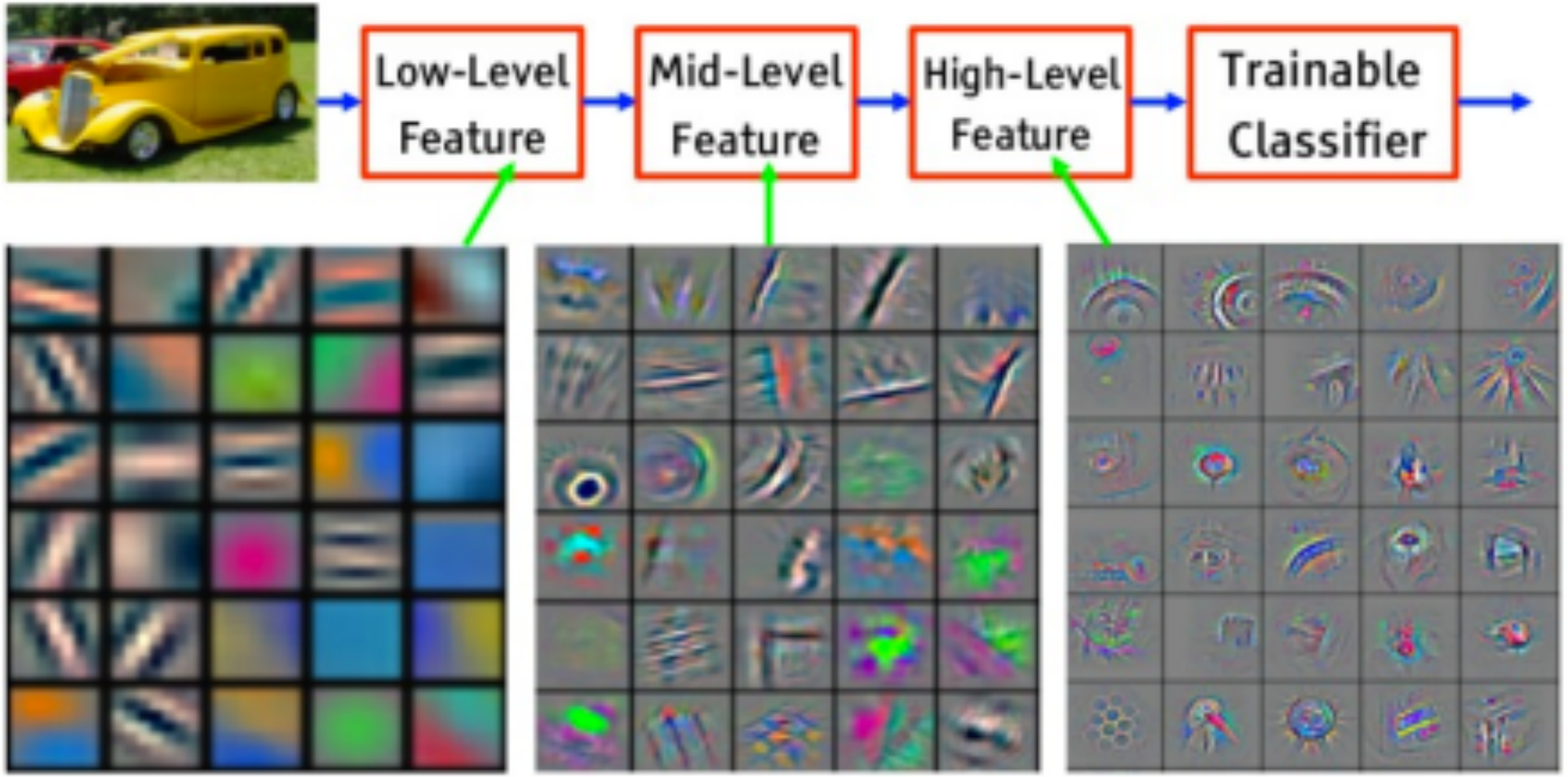
Convolutional Neural Networks (CNNs)

The idea: let's stack multiple layers of multiple filters and learn the weights of the filters. The last feature map can be flattened because every pixel has spatial information from the entire image



Source: developersbreach.com/convolution-neural-network-deep-learning/

CNN Filters (similarity to brain)



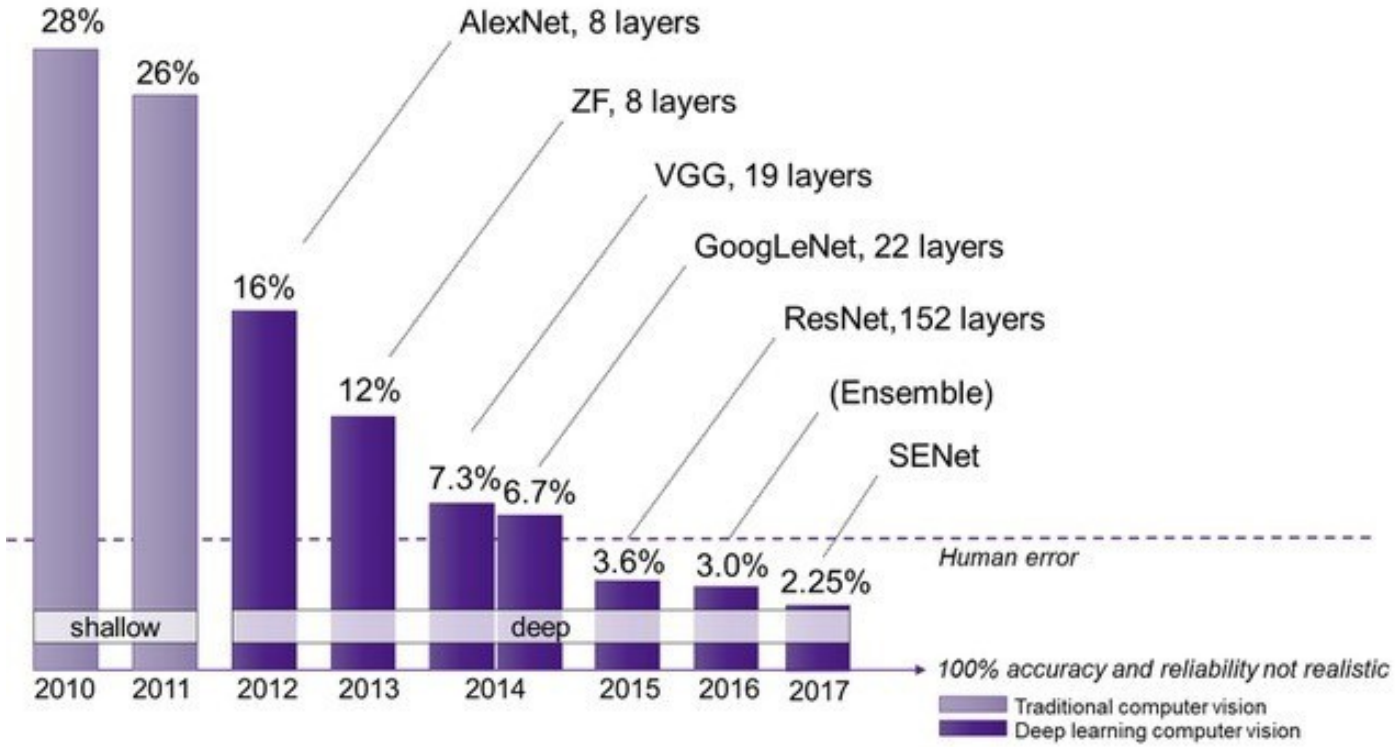
Source: wiki.tum.de/display/lfdv/Convolutional+Neural+Networks

CNN Performance on ImageNet

ImageNet:

14M+ Images, 20,000 Categories

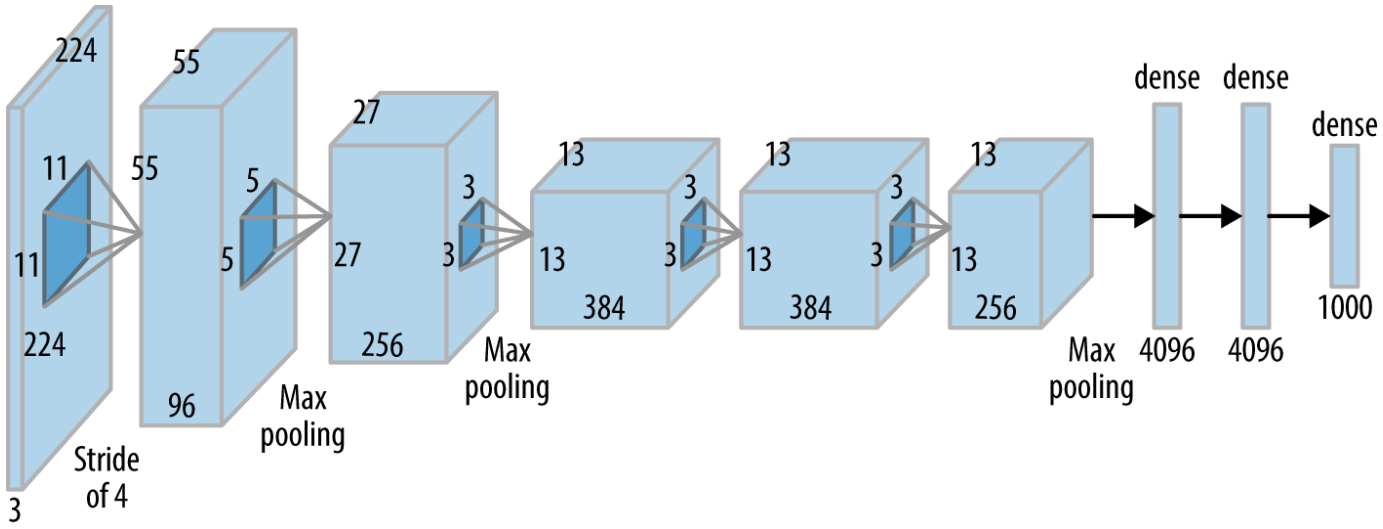
Classification Error on ImageNet



Source: semiengineering.com/new-vision-technologies-for-real-world-applications/

Example CNN Architectures

AlexNet



ResNet

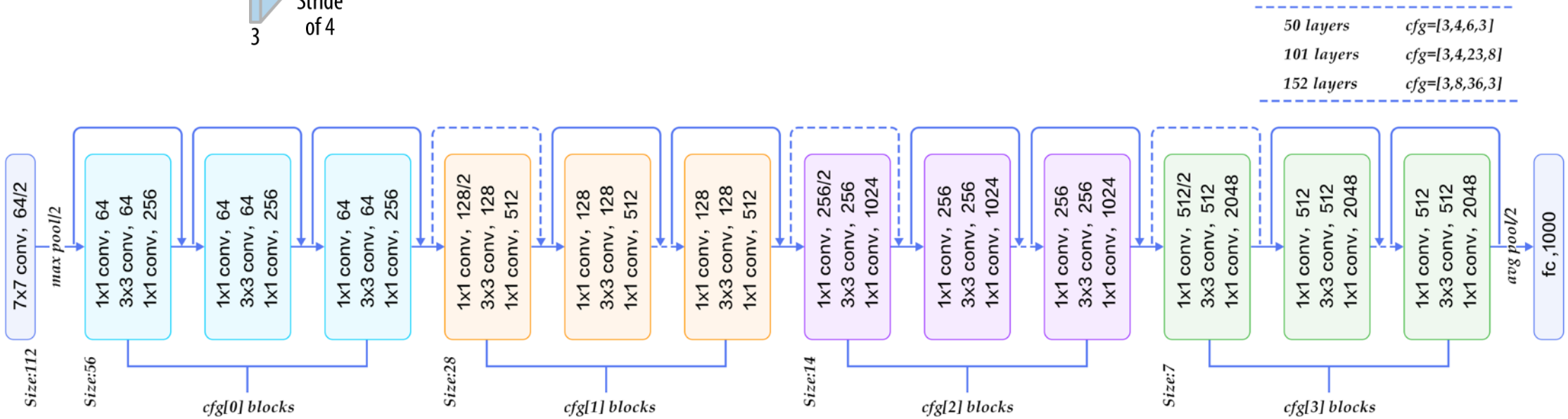
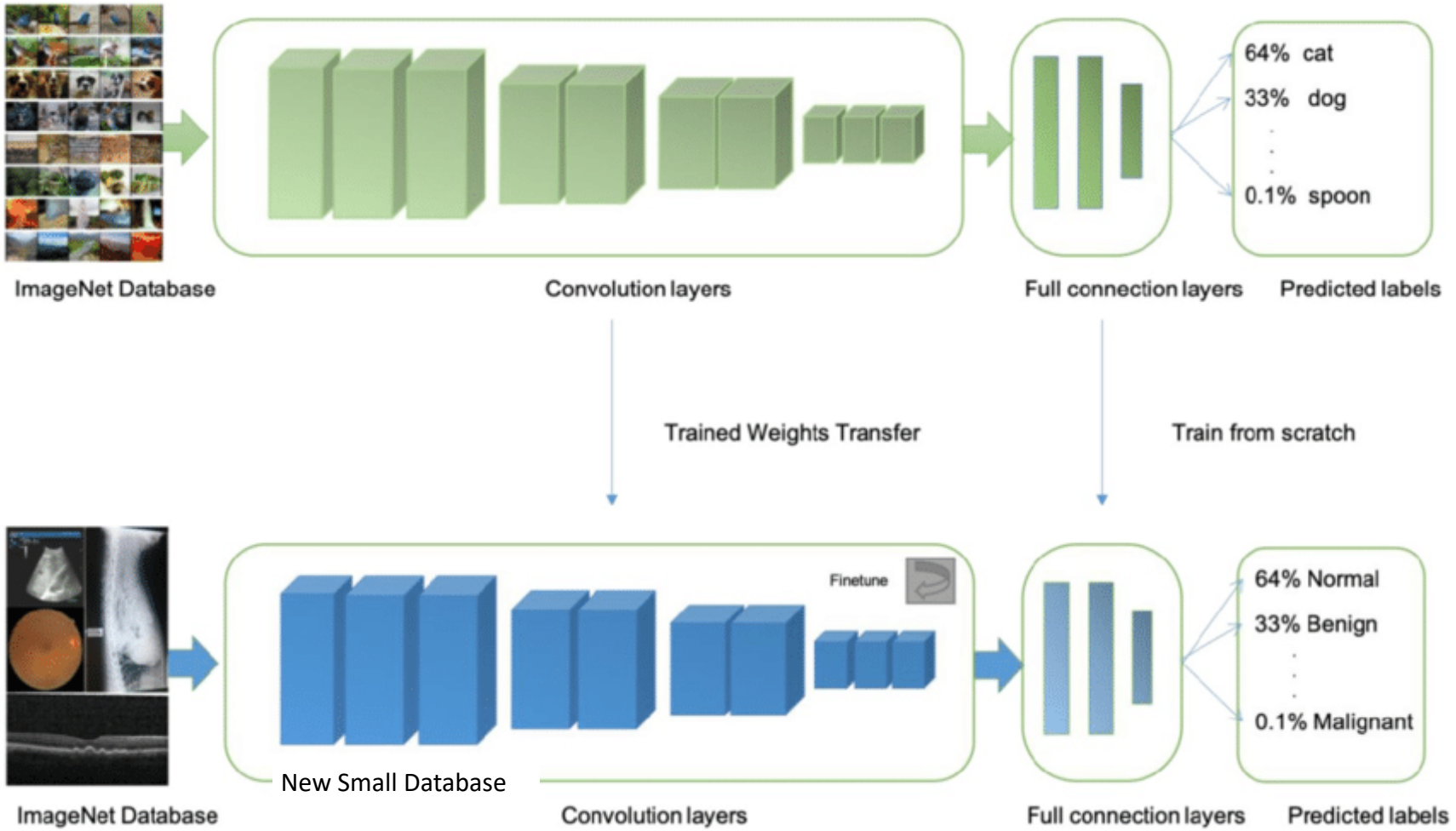


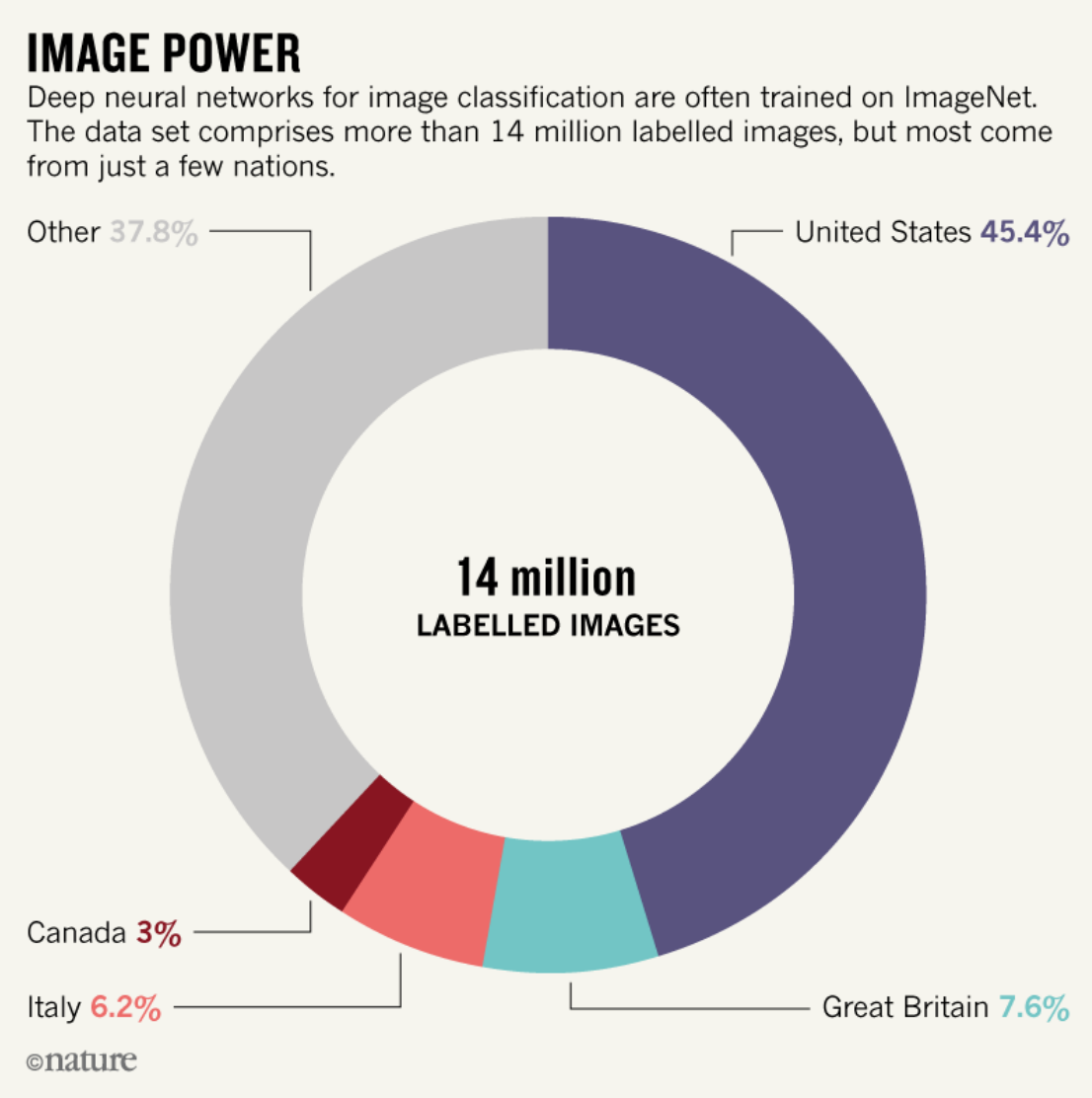
Image Representation & Transfer Learning

ImageNet features are being used successfully in other domains even with small datasets.



Source: Jie Xu et. al 2019

Bias in ImageNet Representation



Algorithms trained on biased data sets often recognize only the left-hand image as a bride.

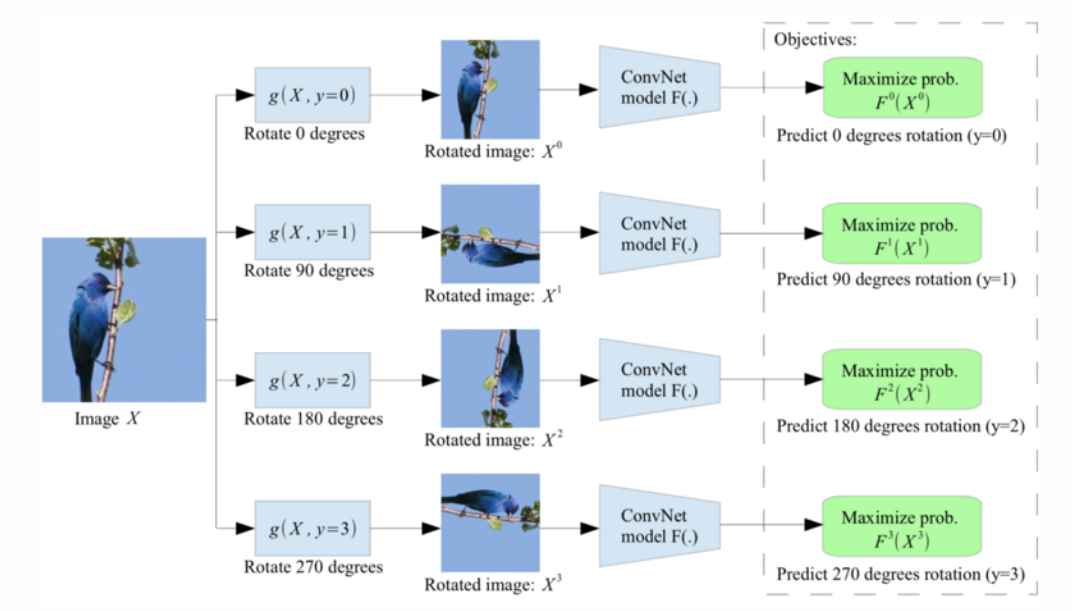
Credit: Left: iStock/Getty; Right: Prakash Singh/AFP/Getty

Source: www.nature.com/articles/d41586-018-05707-8

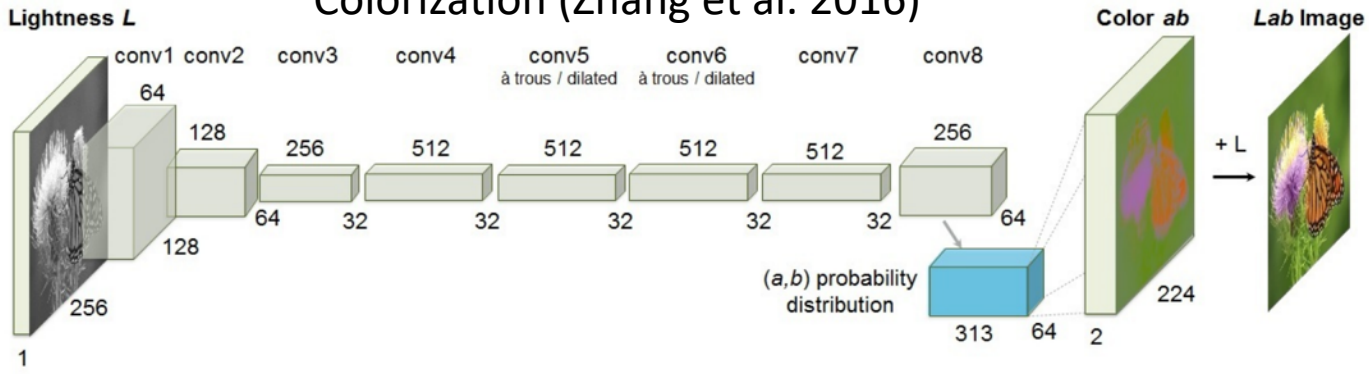
Self-Supervised Representation Learning

- Problem:
  - The domain is very different from ImageNet
  - We have large dataset but not labeled
- Solution:
  - automatically generate some kind of supervisory signal
- Examples: Colorization, Rotation, Patches, …

Rotation (Gidaris et al. 2018)



Colorization (Zhang et al. 2016)

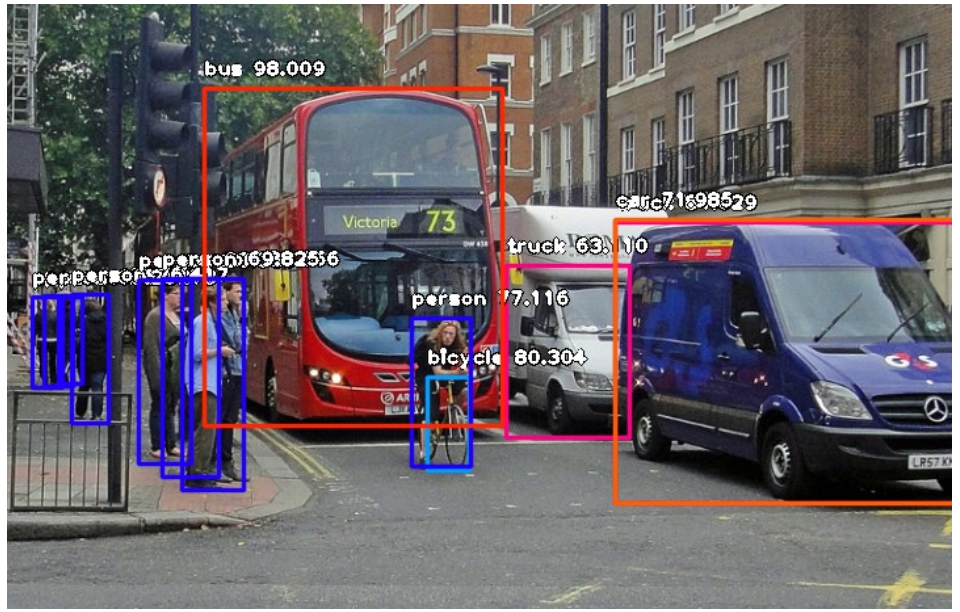


Beyond Image Classification: Object Detection

- Detecting instances of semantic objects of a certain class in digital images and videos:

Source: www.kaggle.com/dilavado/labeled-surgical-tools/version/1

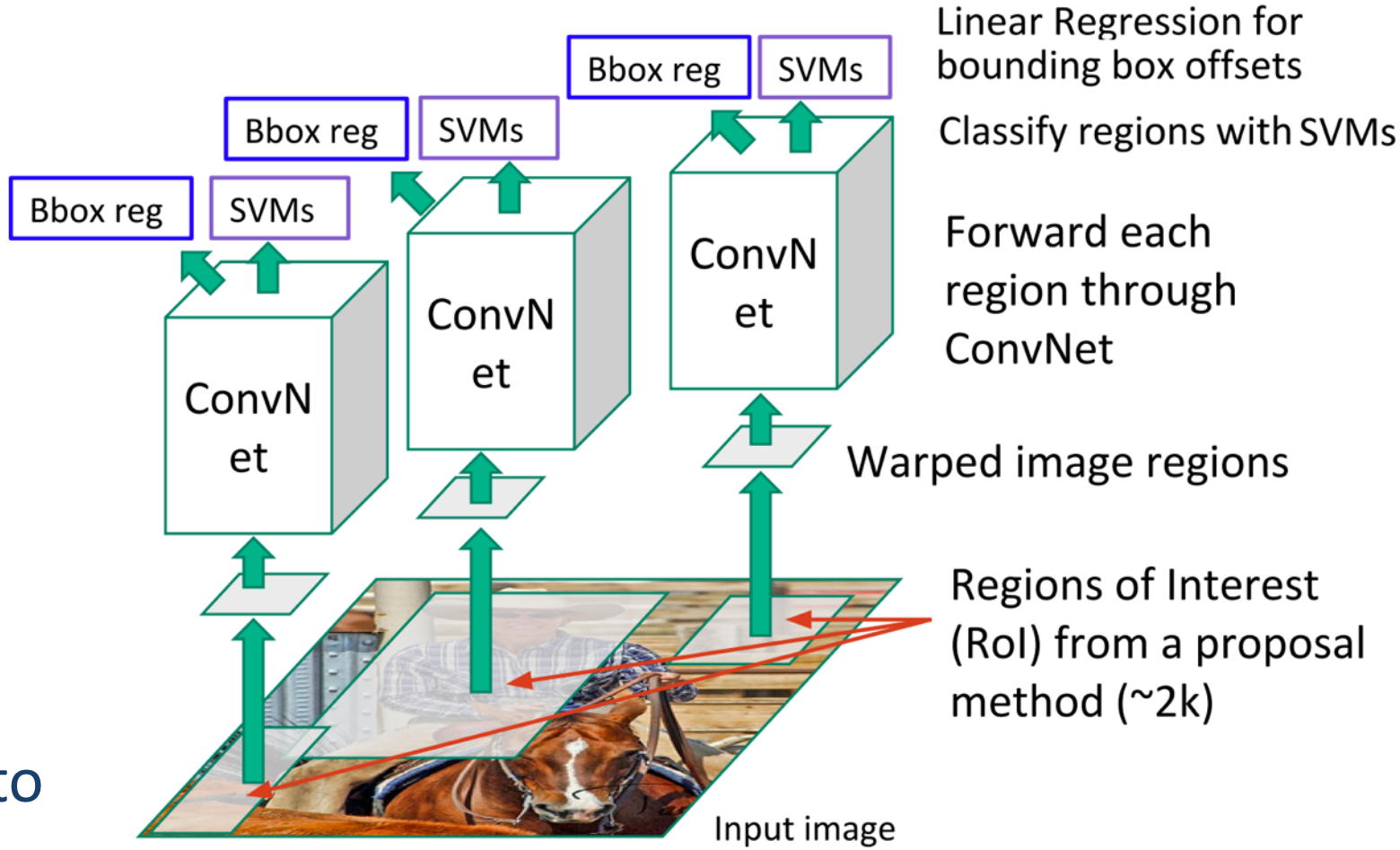
Source: towardsdatascience.com/building-a-distance-violation-detector-d-v-d-for-a-post-lockdown-era-5b9894f5a6b1



Source: towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606

Regional CNN (R-CNN)

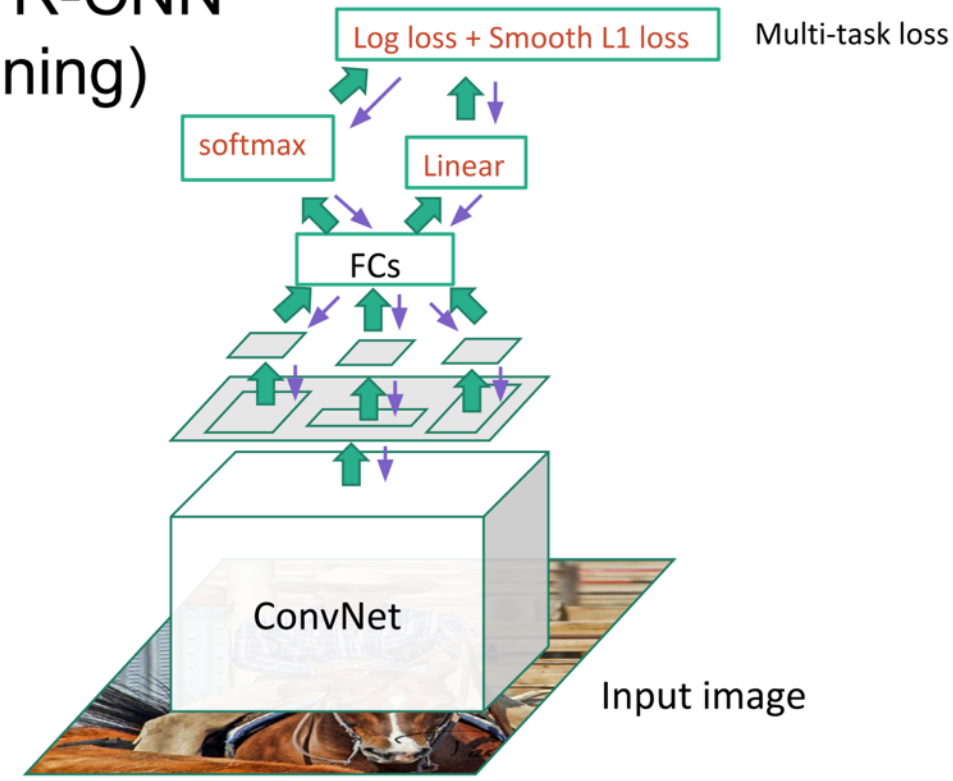
- Find the Regions of Interests (ROI) using Selective Search (Region Proposals)
- Fine tune a CNN for the desired objects to generate
- Use a classifier (e.g., SVM) for the object
- Use a linear regression to adjust bounding boxes



Source: Fei-Fei Li, J. Johnson, s. Yeung, Convolutional Neural Networks for Visual Recognition, Spring 2018

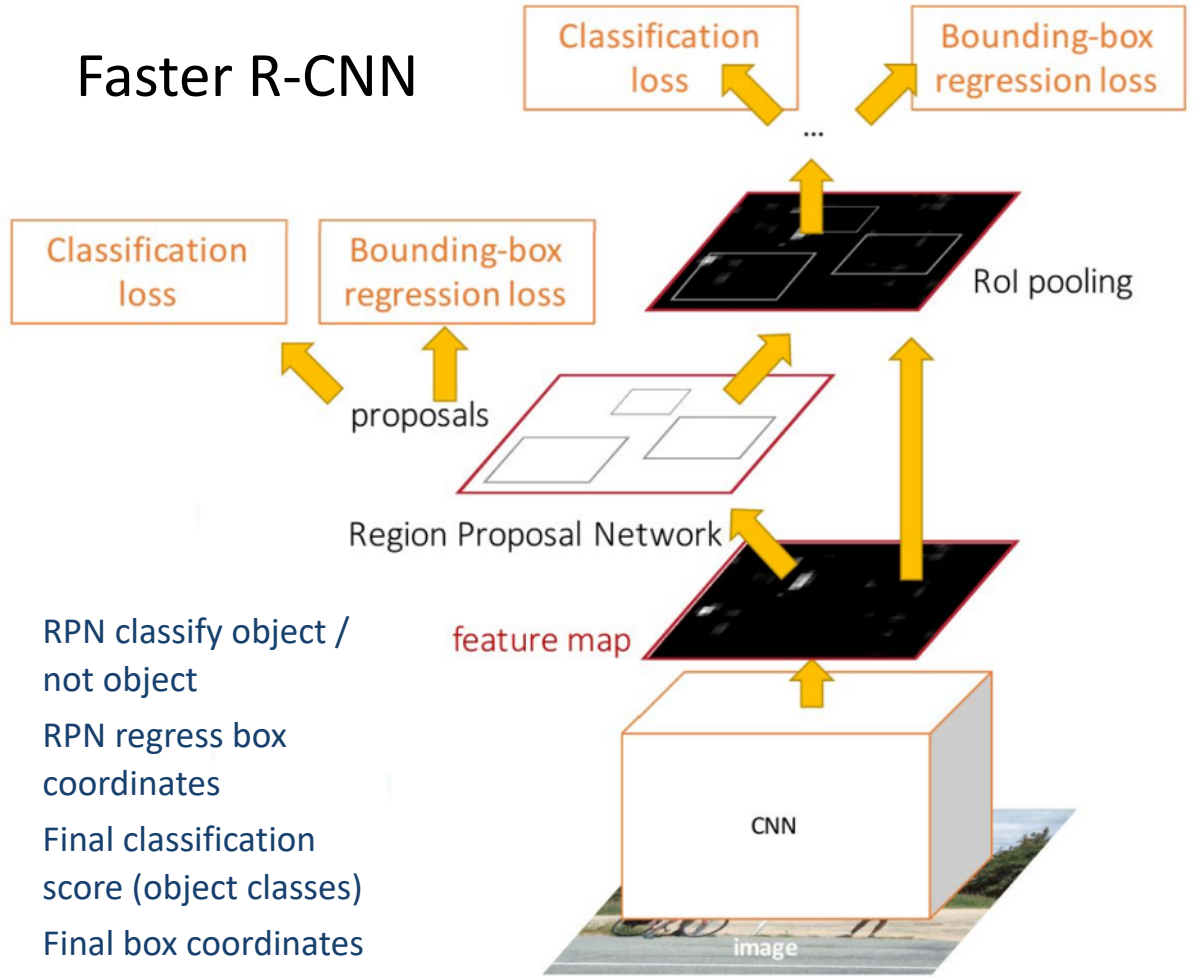
Fast and Faster R-CNN

Fast R-CNN (Training)



Source: Fei-Fei Li, J. Johnson, s. Yeung, Convolutional Neural Networks for Visual Recognition, Spring 2018

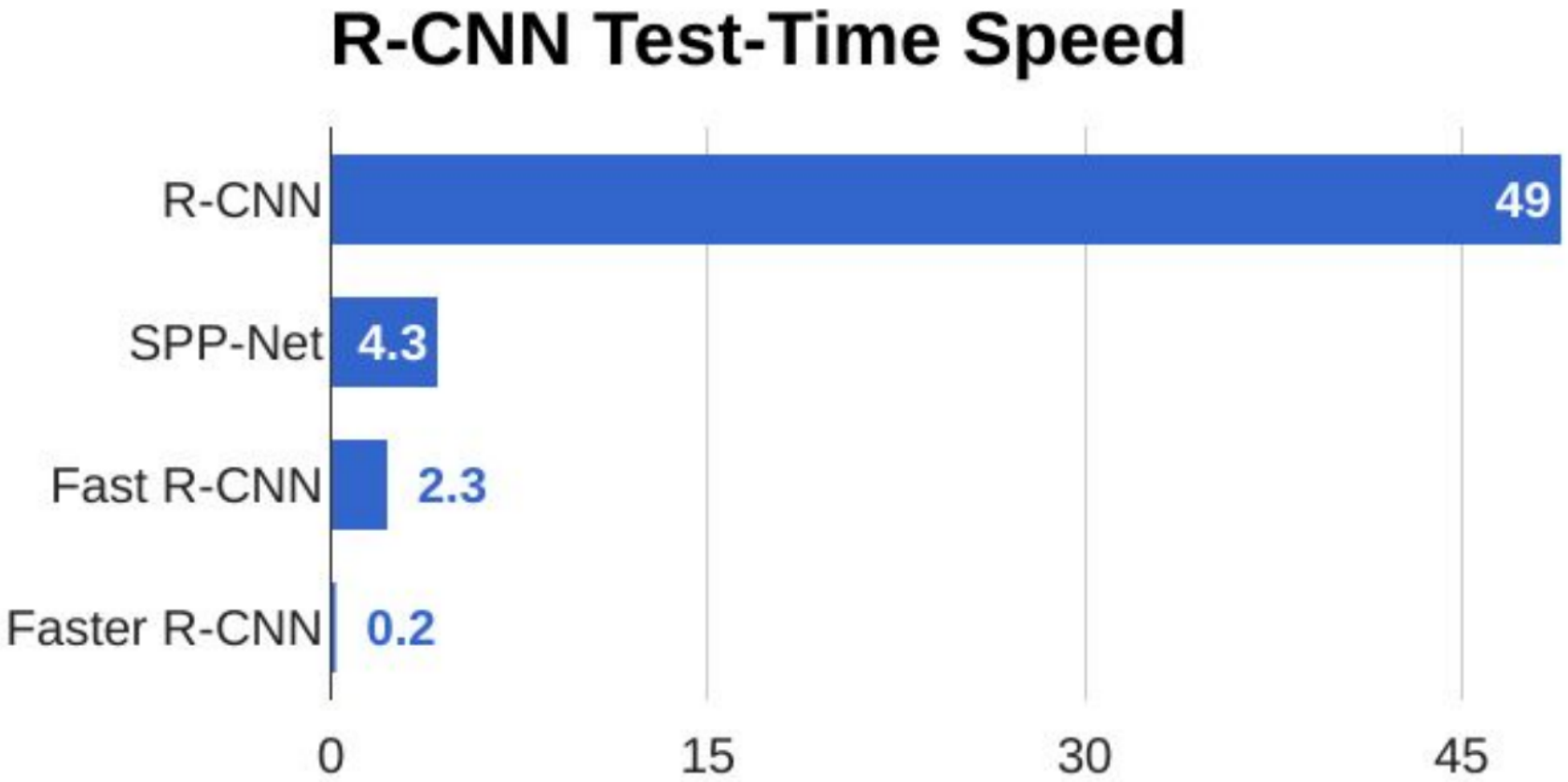
Faster R-CNN



1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

Source: Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

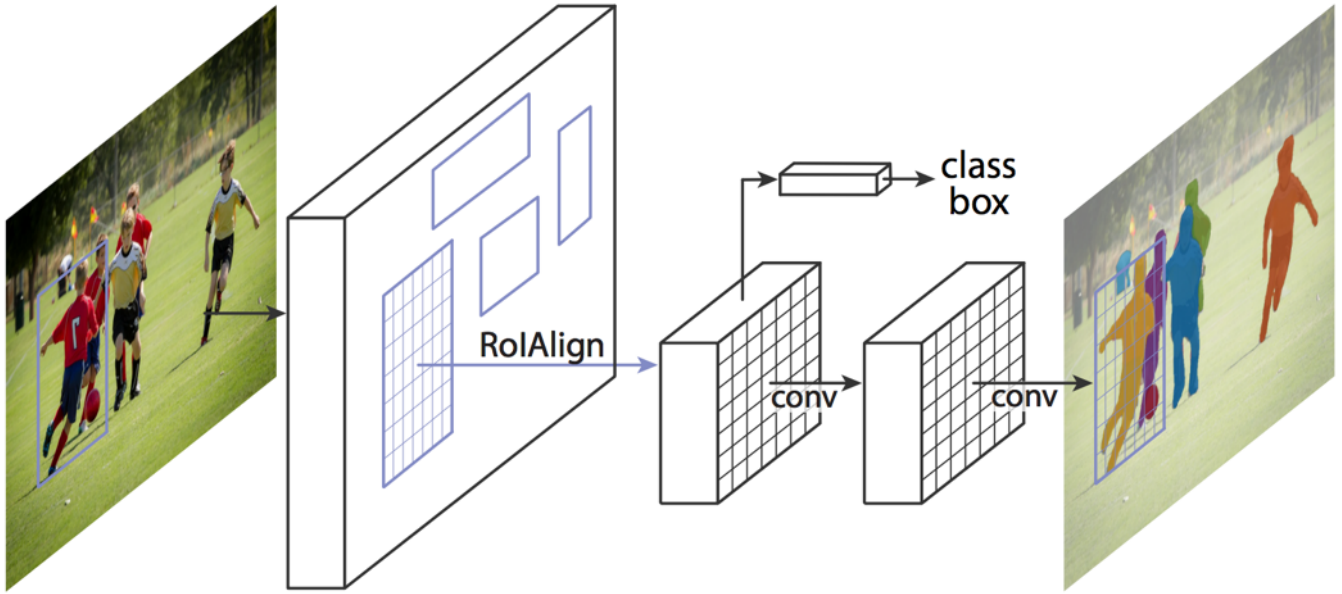
R-CNN Family Speed Comparison



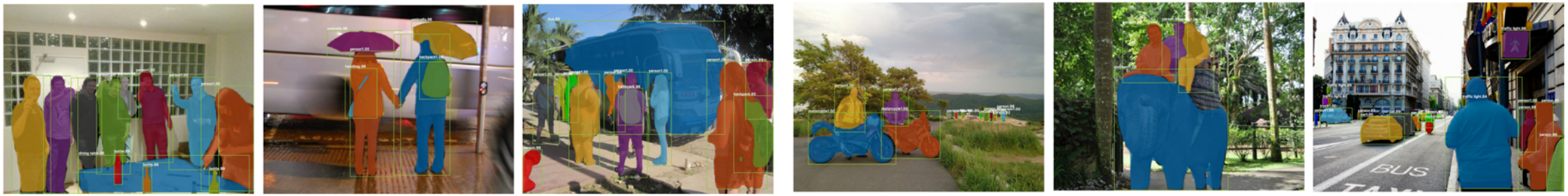
Source: Fei-Fei Li, J. Johnson, s. Yeung, Convolutional Neural Networks for Visual Recognition, Spring 2018

Mask R-CNN: Object Detection + Instance Segmentation

- Instance Segmentation is an extension of object detection, that associates a binary mask (1 = object, 0 = background) with every bounding box.
- Example: autonomous driving



Source: K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, 2017



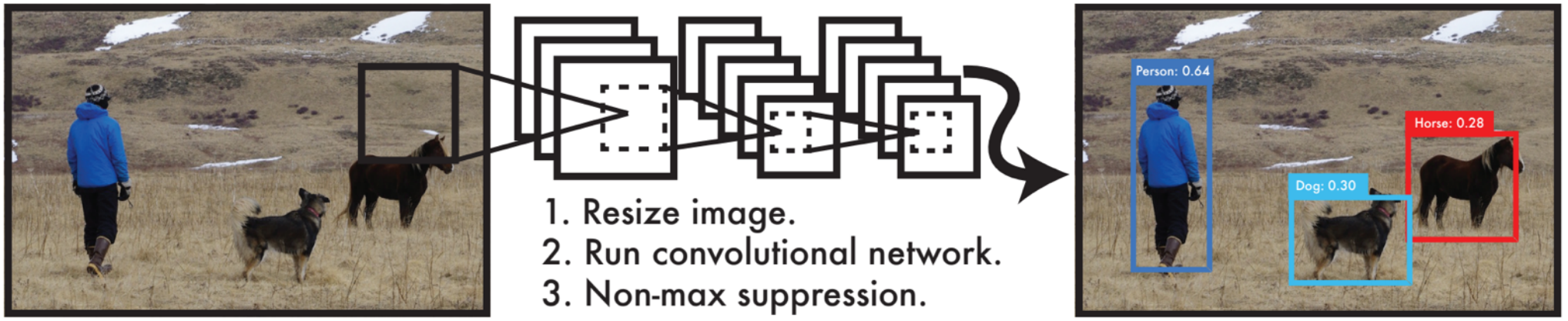
Human Pose Estimation

- Humen Pose Estimation: is a set of techniques in computer vision with the purpose of detecting human figures in images and/or videos.
- Mask R-CNN can easily be extended to human pose estimation. It models a keypoint's location as a one-hot mask and adopt Mask R-CNN to predict K masks, one for each of K keypoint types (e.g., left shoulder, right elbow).

Source: K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, 2017

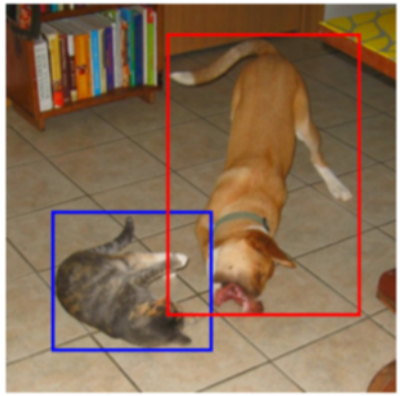
Single Shot Detectors

- R-CNN methods use region proposals followed by a high-quality classifier to classify these proposals. These methods are accurate but come at a big computational cost (low frame-rate), in other words they are not fit to be used on embedded devices and for real-time applications.
- Another types of object detectors are Single Shot Detectors that combine the above two tasks into one network. They use a set of pre-defined boxes to look for objects instead of having a network produce proposal.

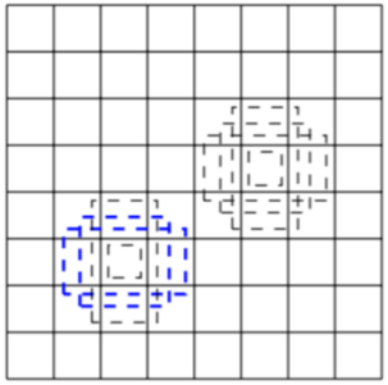


Source: J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, 2015

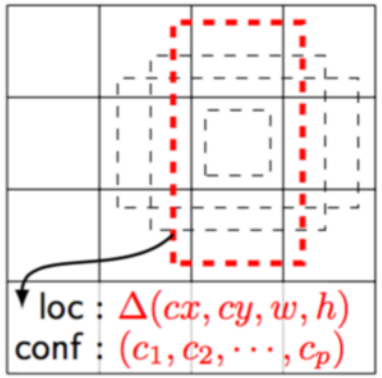
Single Shot Detectors



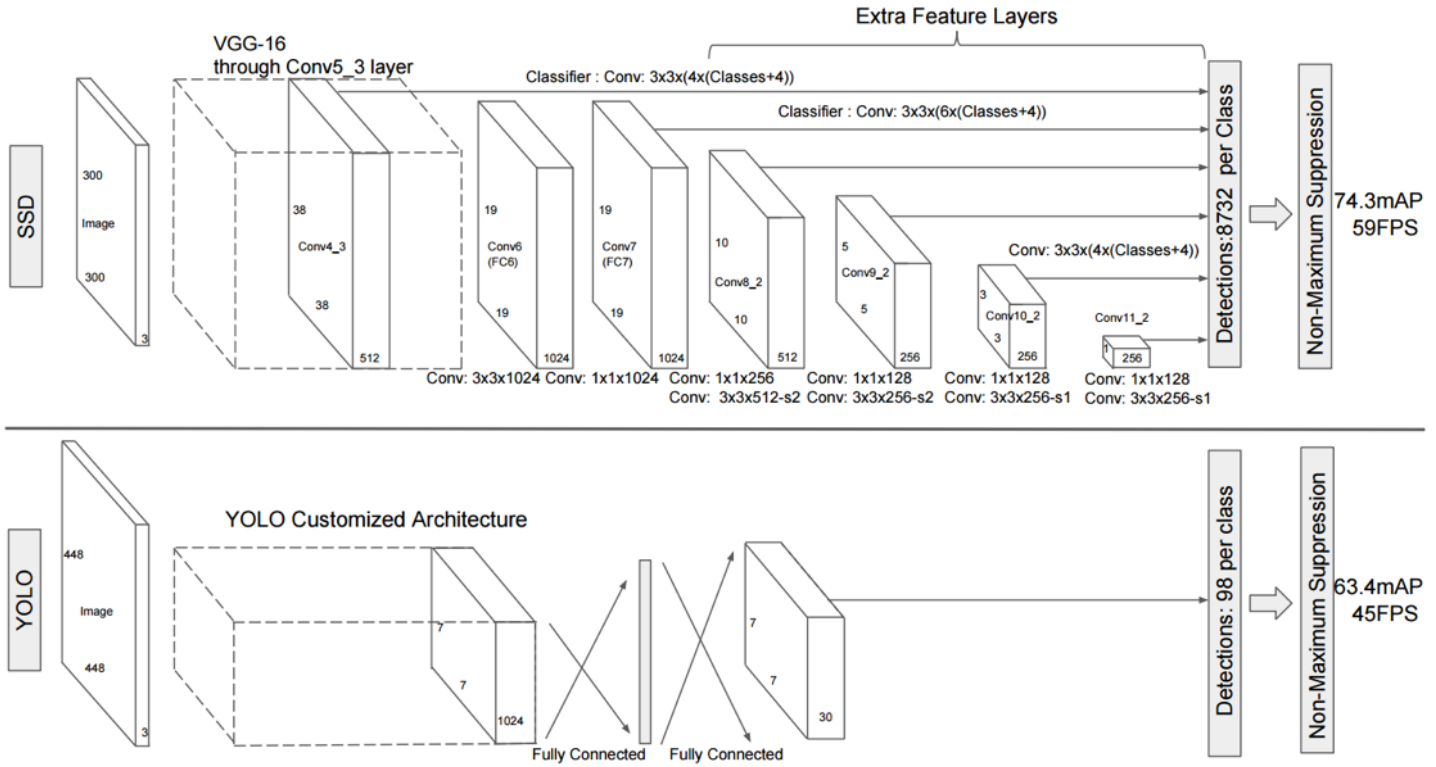
(a) Image with GT boxes



(b) 8×8 feature map



(c) 4×4 feature map



Source: W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, 2015

End-to-End Object Detection with Transformers

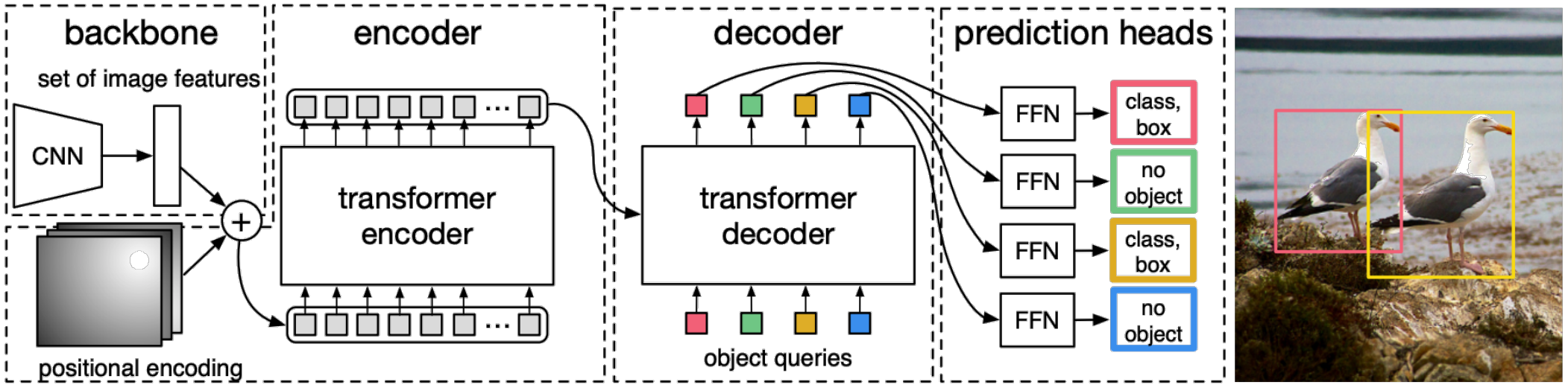
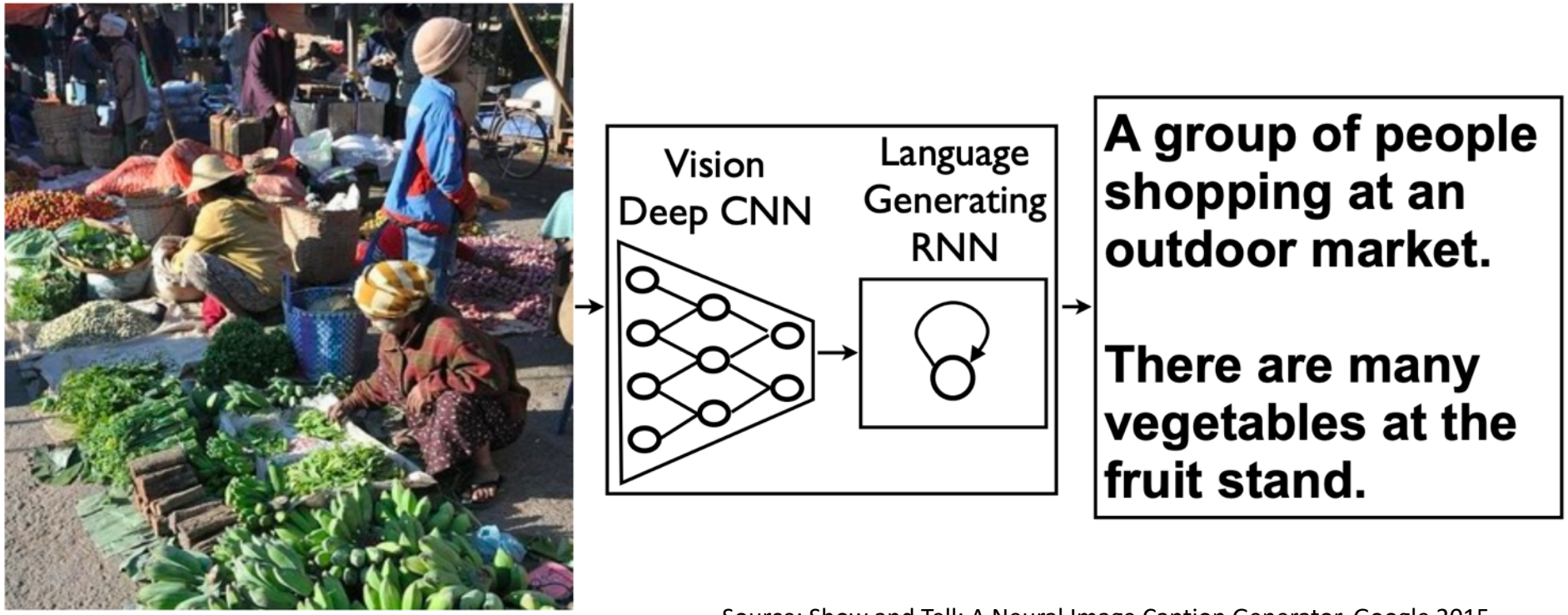


Fig. 2: DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "`no object`" class.

Source: End-to-End Object Detection with Transformers, Facebook AI 2020

Image Captioning

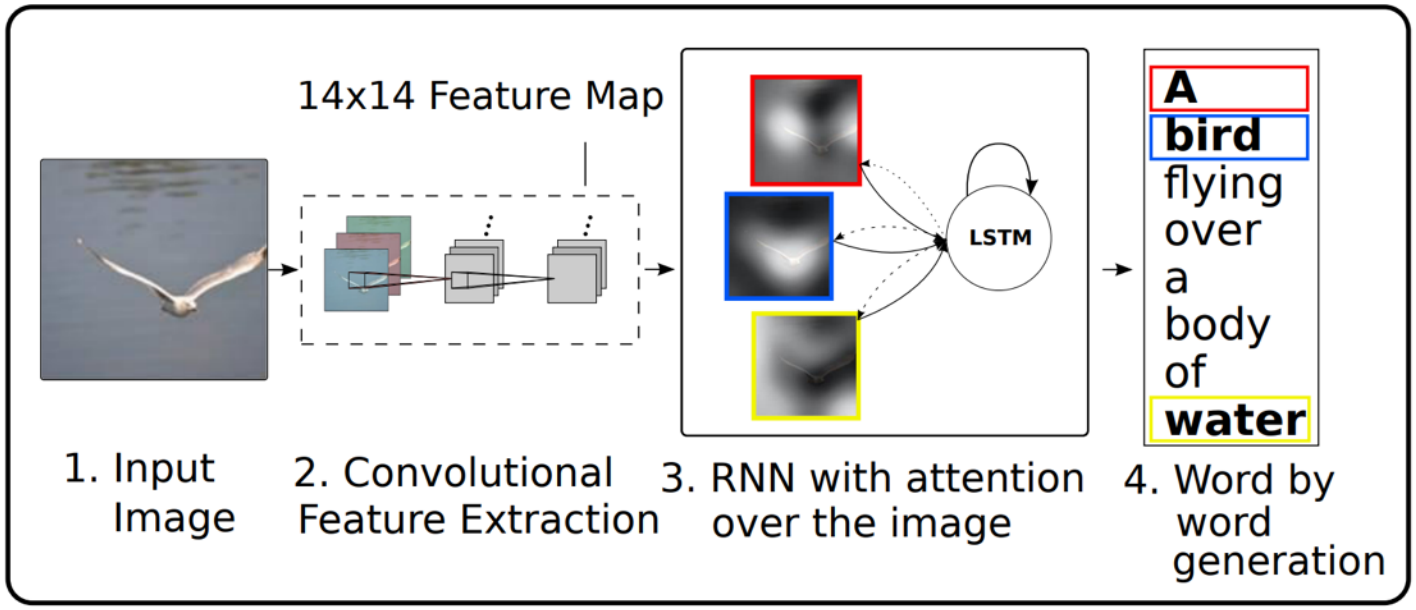
- The model consists of two parts:
  - A CNN to understand image and create a representation vector
  - A RNN to generate the text



Source: Show and Tell: A Neural Image Caption Generator, Google 2015

Image Captioning with Attention

- An attention-based model that automatically learns to describe the content of images.



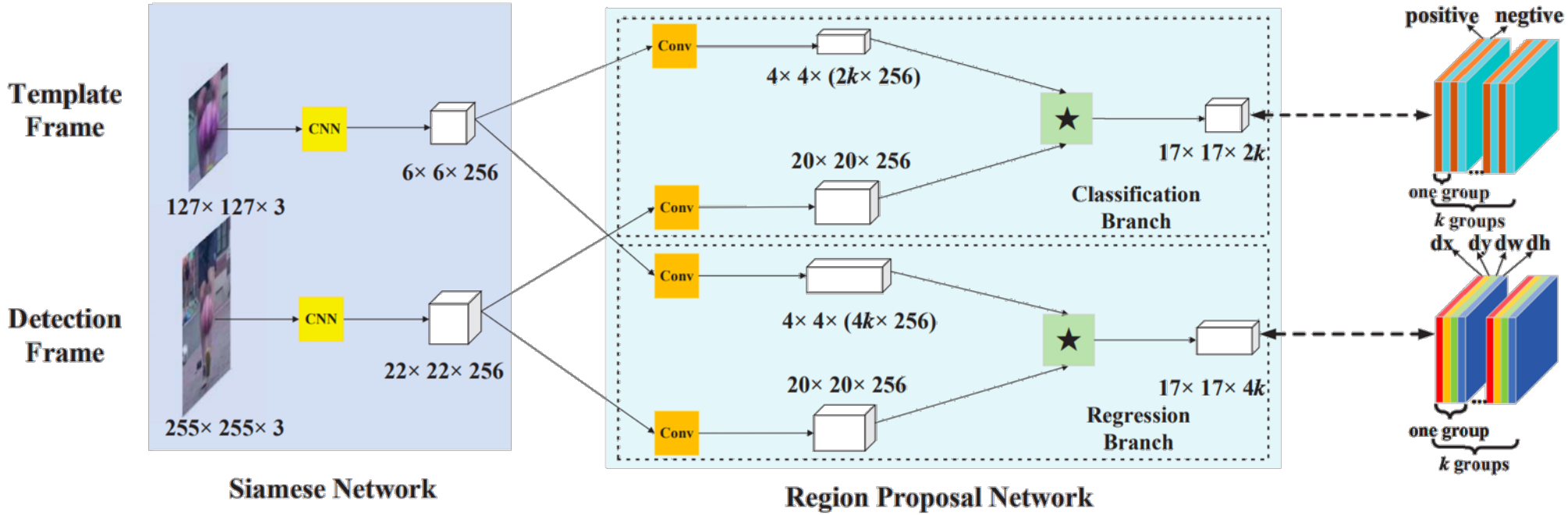
A stop sign is on a road with a mountain in the background.

A dog is standing on a hardwood floor.

Source: Kelvin Xu et al., Show, Attend and Tell: Neural Image Caption Generation with Visual Attention 2016

Object Tracking in Videos

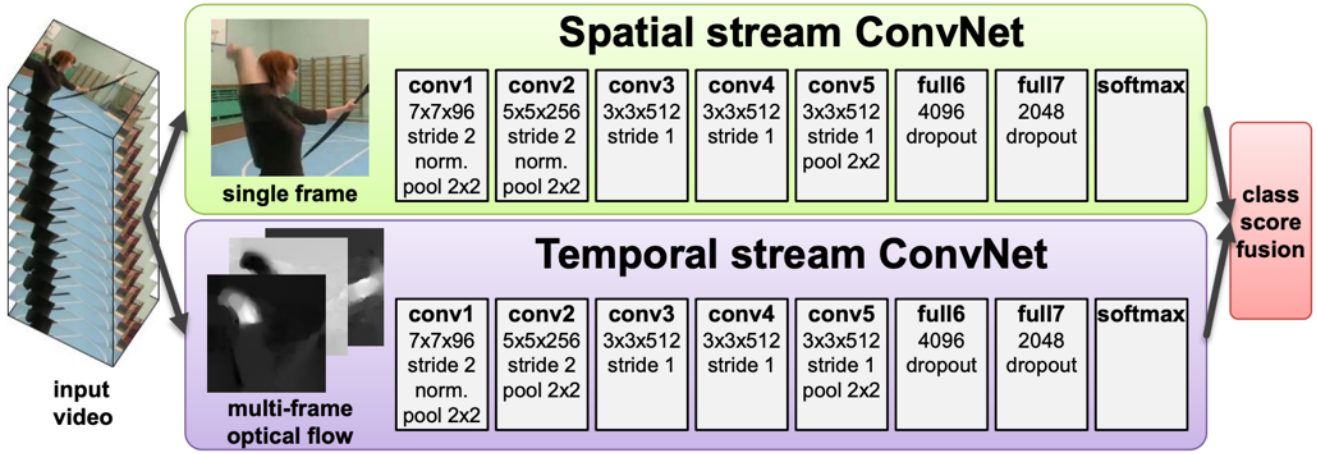
- Example: Siamese-RPN for single object tracking :
  - Siamese subnetwork for visual feature extraction
  - Region proposal subnetwork (pair-wise correlation is adopted to obtain the output of two branches)



Source: High Performance Visual Tracking with Siamese Region Proposal Network, Bo Li et al, 2018

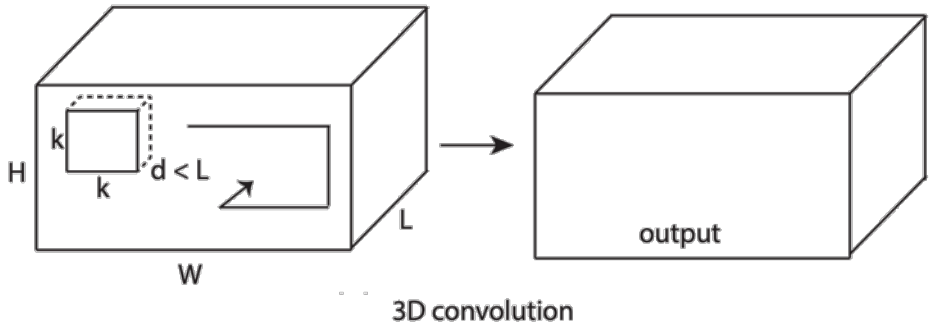
Action Recognition in Videos

- Two-stream architecture for video classification

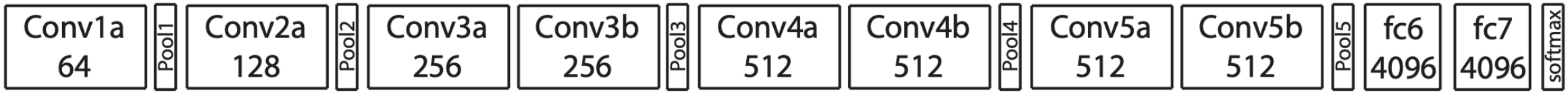


Source: Two-Stream Convolutional Networks for Action Recognition in Videos. K. Simonyan 2014

- Learning Spatiotemporal Features with 3D Convolutional Networks



Source: Learning Spatiotemporal Features with 3D Convolutional Networks. Facebook AI 2015



This was just the tip of the iceberg.

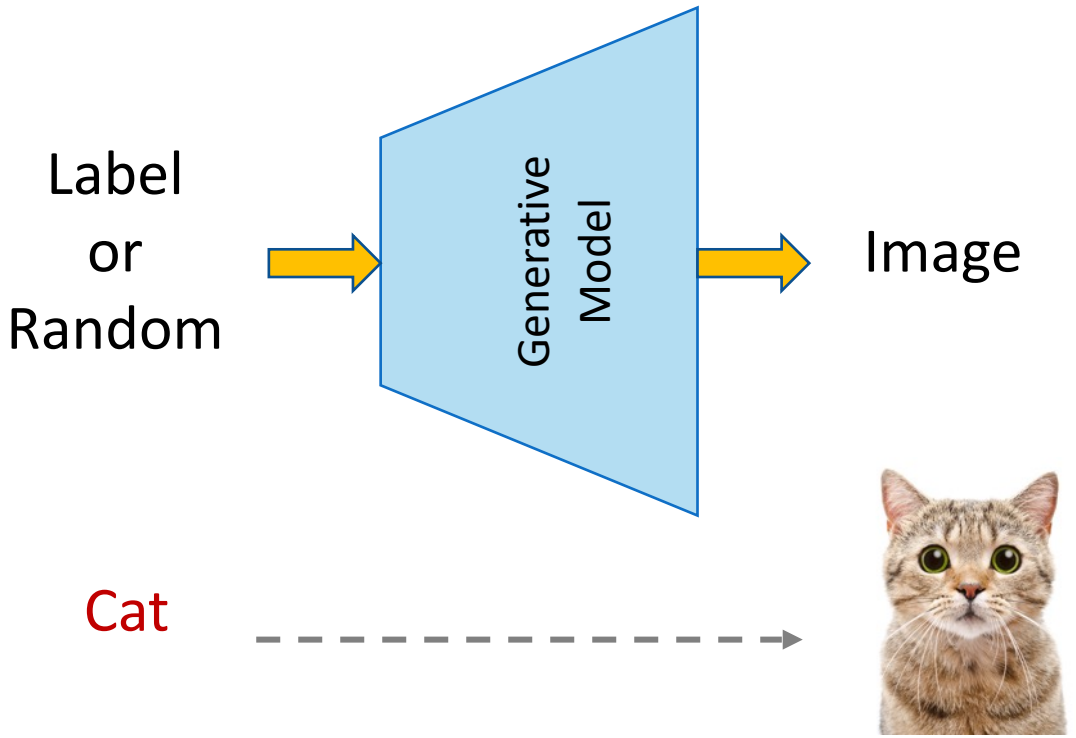
Now, a very brief review of :

Image Generation

Generative Model

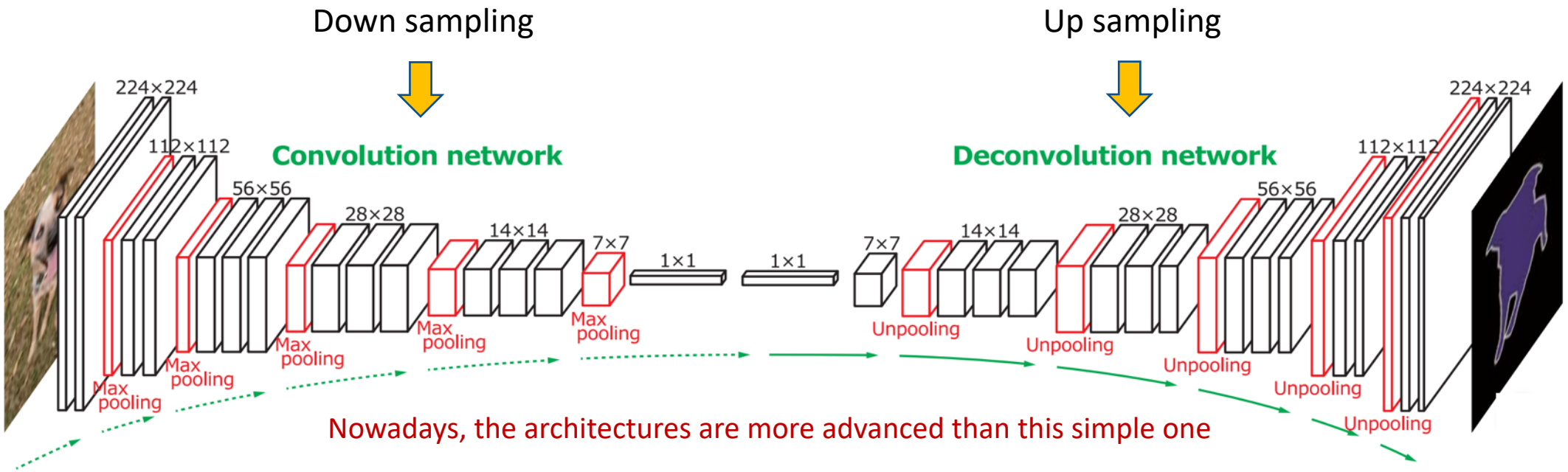
- Generative models can generate new data instances, by estimating the probability distribution of the data “P(X)” or the joint probability distribution of the data and the label “P(X, y)”.

- The input can be:
  - A random vector
  - A label/class
  - A representation vector for a text, another image, and so on.



Convolutional Autoencoders

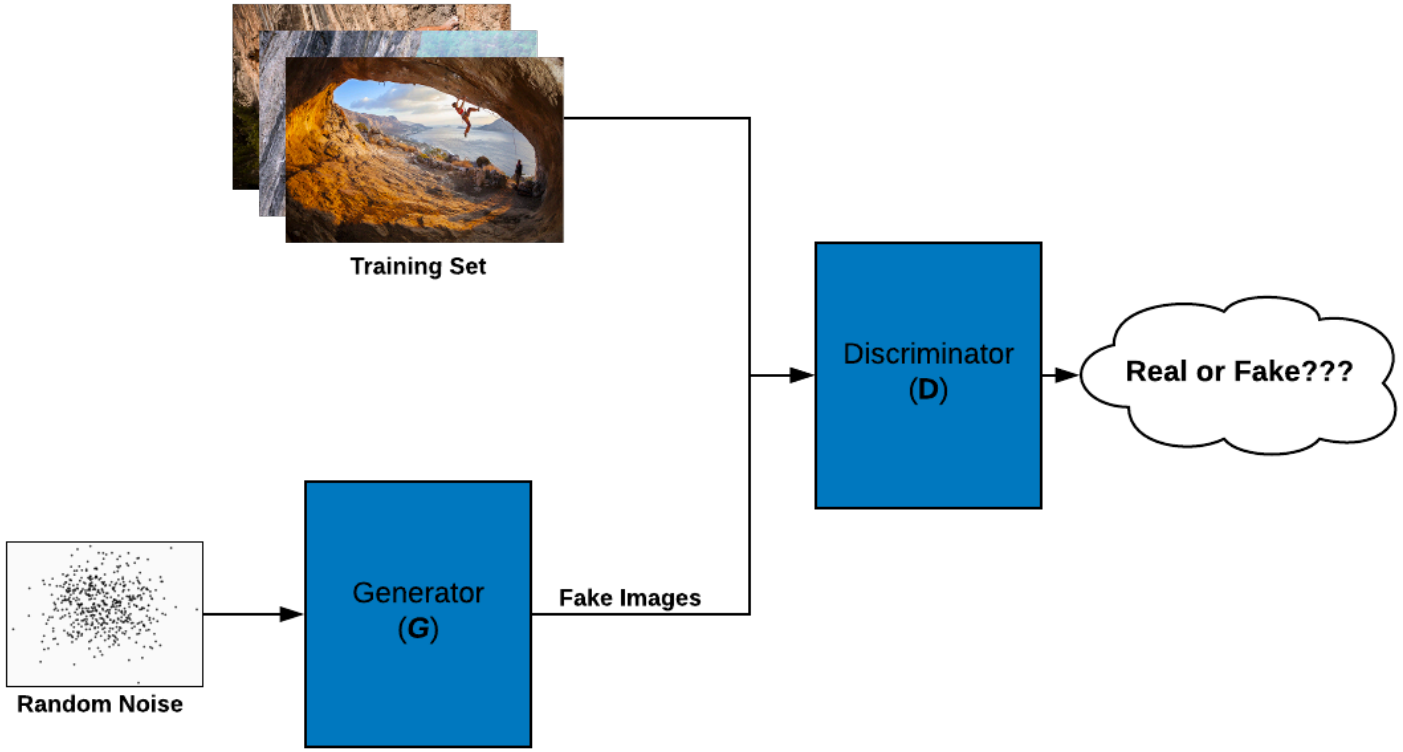
- Sample applications:
  - Semantic Segmentation
  - Image Denoising
  - Image Repair
  - Super-resolution



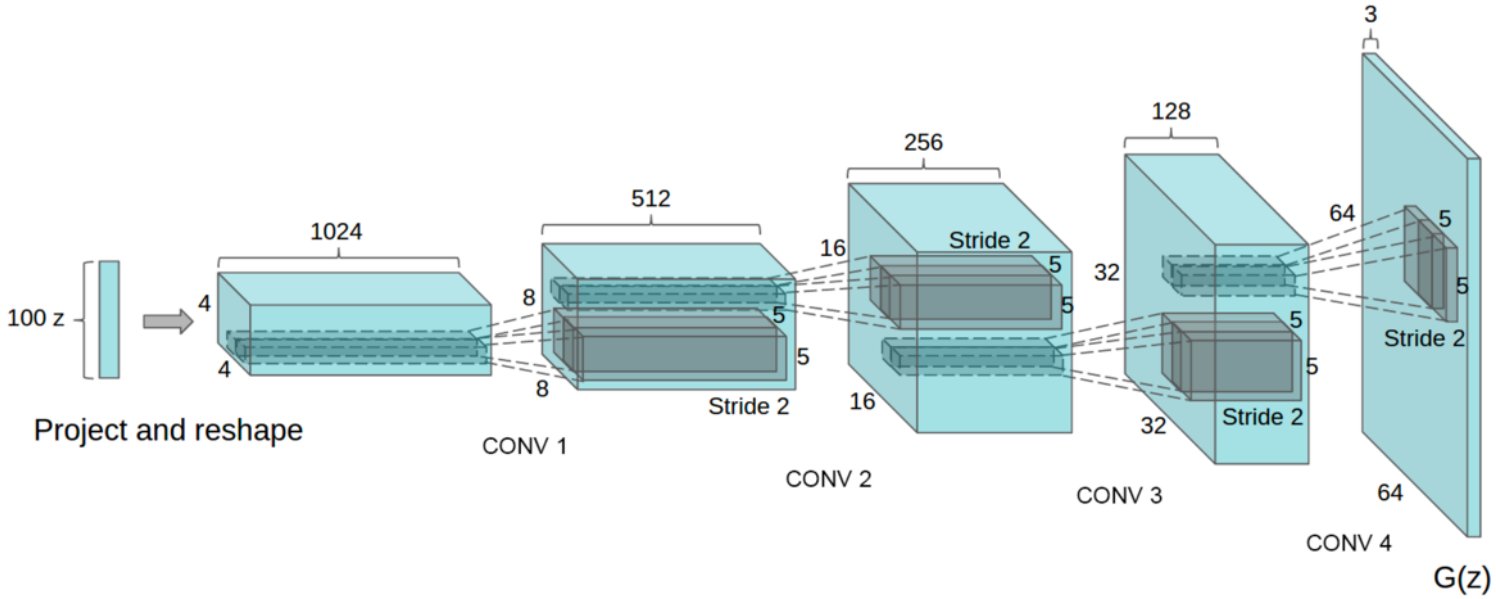
Source: H. Noh, S. Hong, B. Han, Learning Deconvolution Network for Semantic Segmentation, 2015

Generative Adversarial Network (GAN)

- The idea of Generative Adversarial Network (GAN) is train two networks together:
  1. The Generator network (G), tries to fool the discriminator in thinking that the generated images are real, meaning that they are taken from the given image distribution.
  2. The Discriminator network (D), tries to differentiate between real and fake images



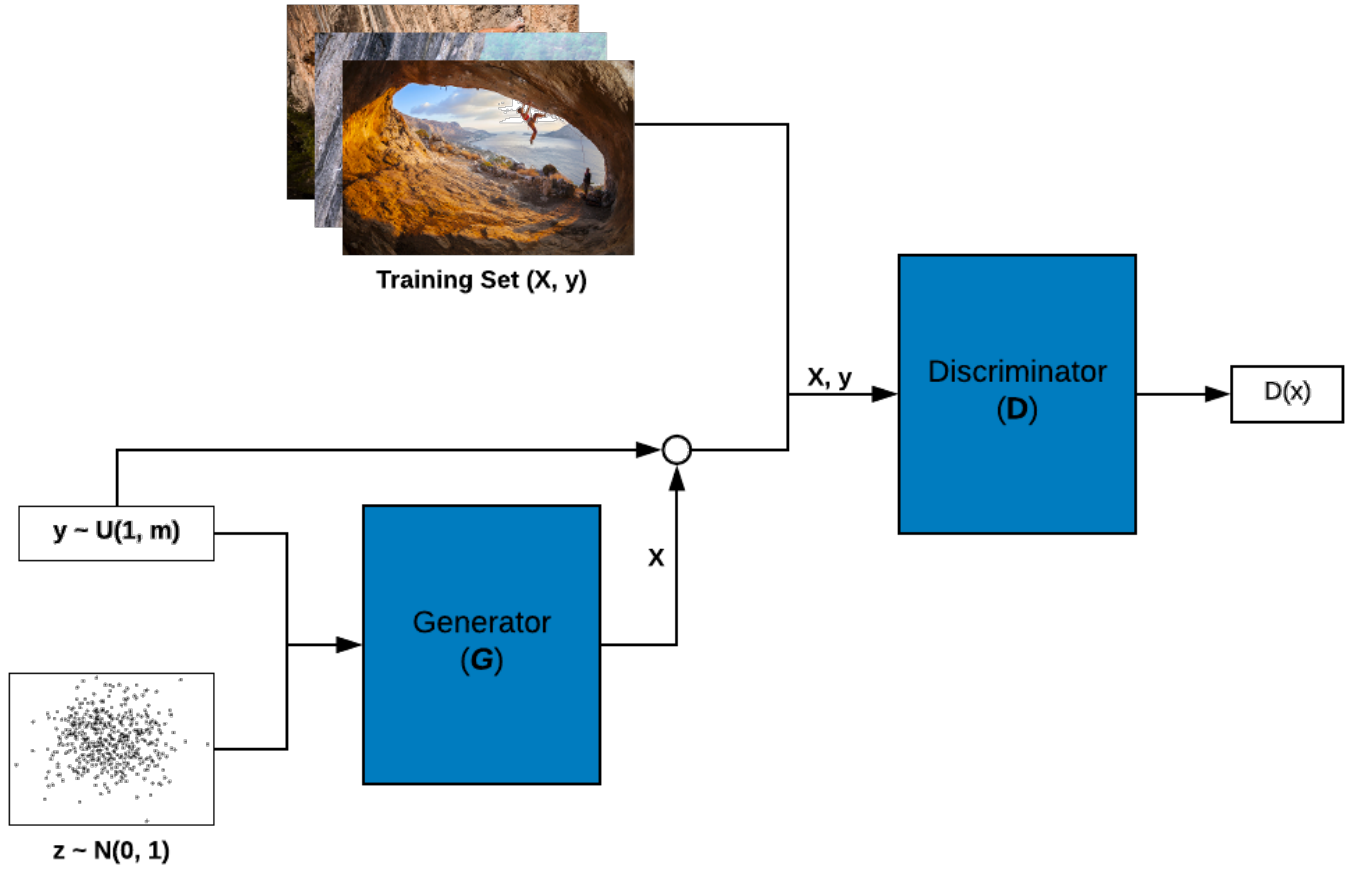
Deep convolutional generative adversarial networks (DCGAN)



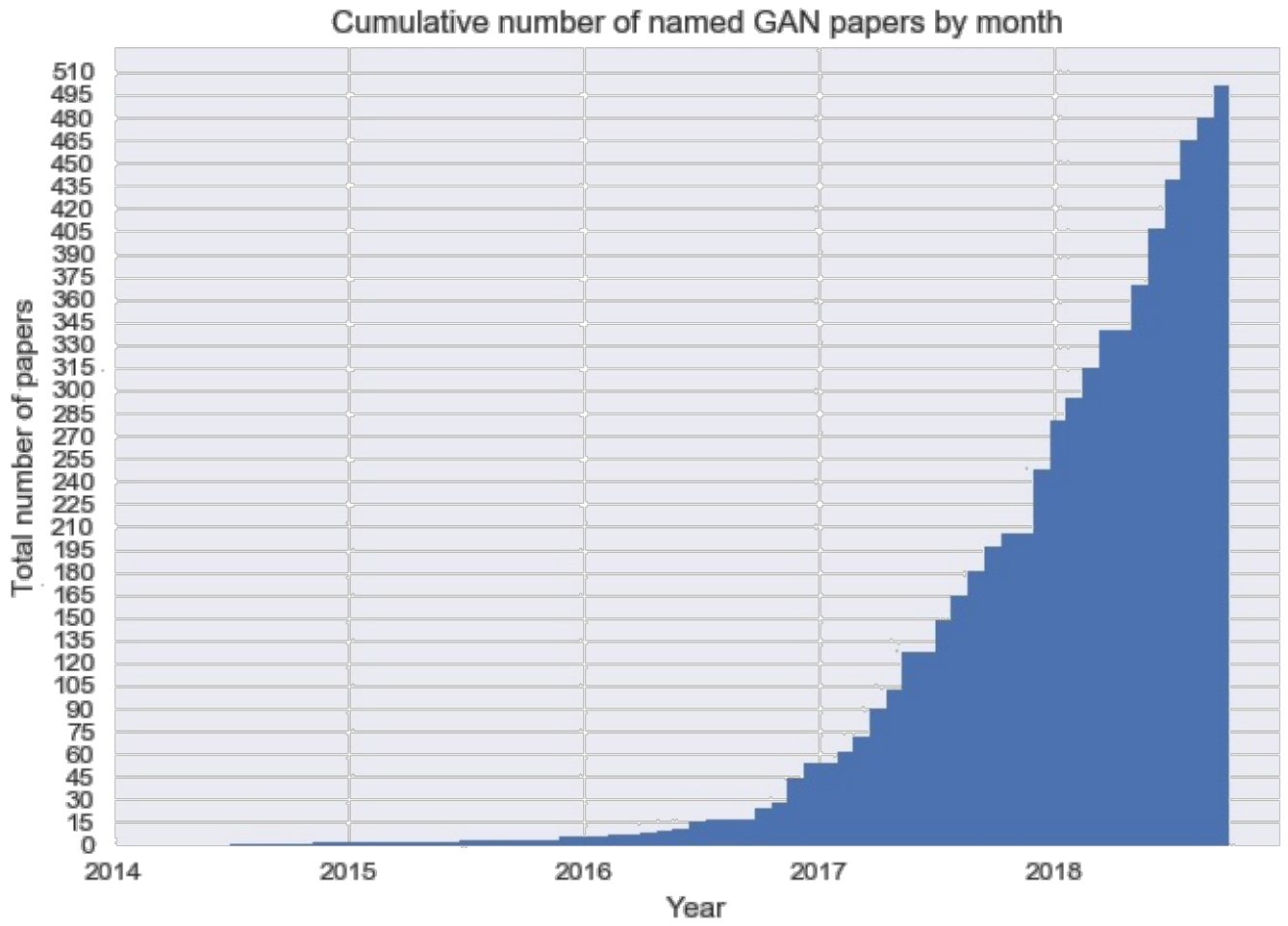
Radford, A., Metz, L., & Chintala, S. (2015).Unsupervised representation learning with deep convolutional generative adversarial networks.

Conditional GAN

- Conditional GANs brings the labels into the system. The labels act as a helper to the discriminator, so that it can have a more guided decision to make.



The GAN Zoo



https://github.com/hindupuravinash/the-gan-zoo

VECTOR
INSTITUTE