MENG 3065 - MODULE 8

Artificial Intelligence: A Modern Approach
Chapter 25 Computer Vision
Image classification using CNN

These slides has been extracted, modified and updated from original slides of :
Artificial Intelligence: A Modern Approach, Fourth Edition.

HUMBER

WE ARE HUMBER

Outline

- Introduction
- Image Formation
- Simple Image Features
- Classifying Images
- Detecting Objects
- Using Computer Vision
- Image classification using CNN

Introduction

- Most agents that use vision use passive sensing
- A feature is a number obtained by applying simple computations to an image.
- The model-based approach to vision uses two kinds of models
  - Object model
    - Example: precise geometric model produced by computer aided design systems
  - Rendering model
    - Describes the physical, geometric, and statistical processes that produce the stimulus from the world

Two core problems of computer vision are

- **Reconstruction**, where an agent builds a model of the world from an image or a set of images,
- **Recognition**, where an agent draws distinctions among the objects it encounters based on visual and other information

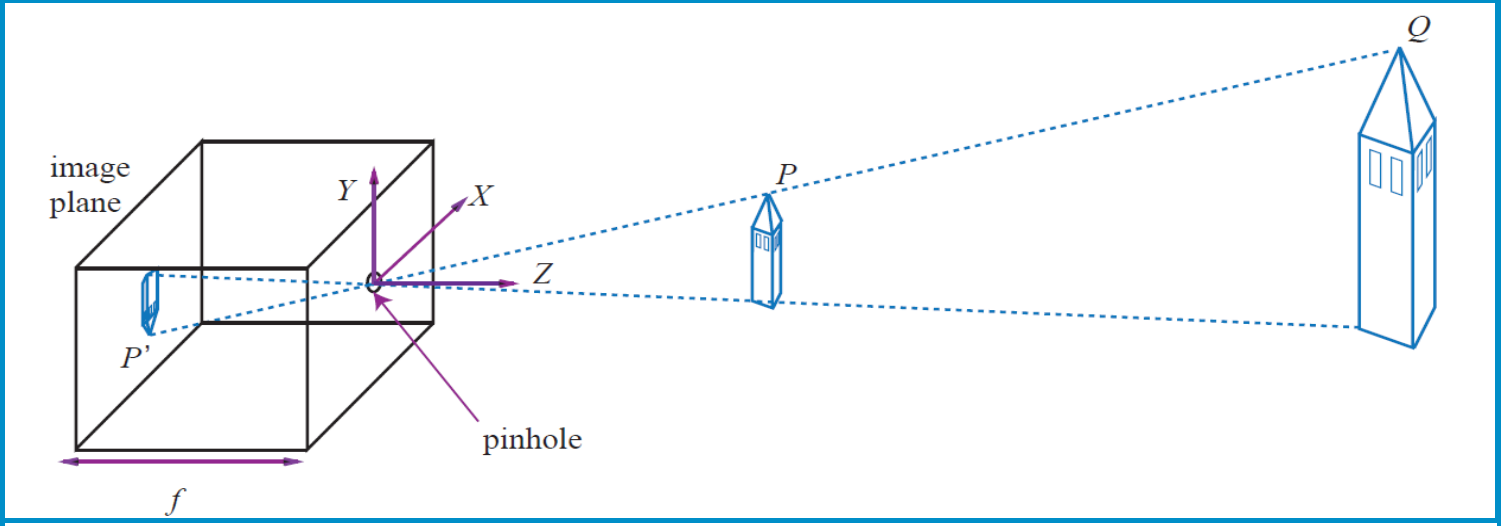
Image Formation

Images without lenses: The pinhole camera

- A pinhole camera captures images without traditional lenses, using a small opening to focus light onto an image plane.
- This design allows for the creation of a 2D image where each pixel acts as an individual sensor—usually a charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS)
- Proper focus results in a sharp image, while defocusing leads to motion blur.

- **Pinhole camera:**
  - This term describes the camera setup, consisting of a pinhole opening (aperture) at the front of a box and an image plane at the back.
  - Pinhole opening, O, at the front of a box, and an image plane at the back of the box.
  - The opening (O) is referred to as the aperture, and the distance from the pinhole to the image plane is known as the focal length (f).

Image Formation



Each light sensitive element at the back of a pinhole camera receives light that passes through the pinhole from a small range of directions. If the pinhole is small enough, the result is a focused image behind the pinhole. The process of projection means that large, distant objects look the same as smaller, nearby objects—the point P^t in the image plane could have come from a nearby toy tower at point P or from a distant real tower at point Q.

Image Formation

Light and shading

The brightness of a pixel in the image is a function of the brightness of the surface patch in the scene that projects to the pixel

The **ambiguity** occurs because there are three factors that contribute to the amount of light:

- the overall intensity of ambient light;
- whether the point is facing the light or is in shadow); and
- the amount of light reflected from the point.

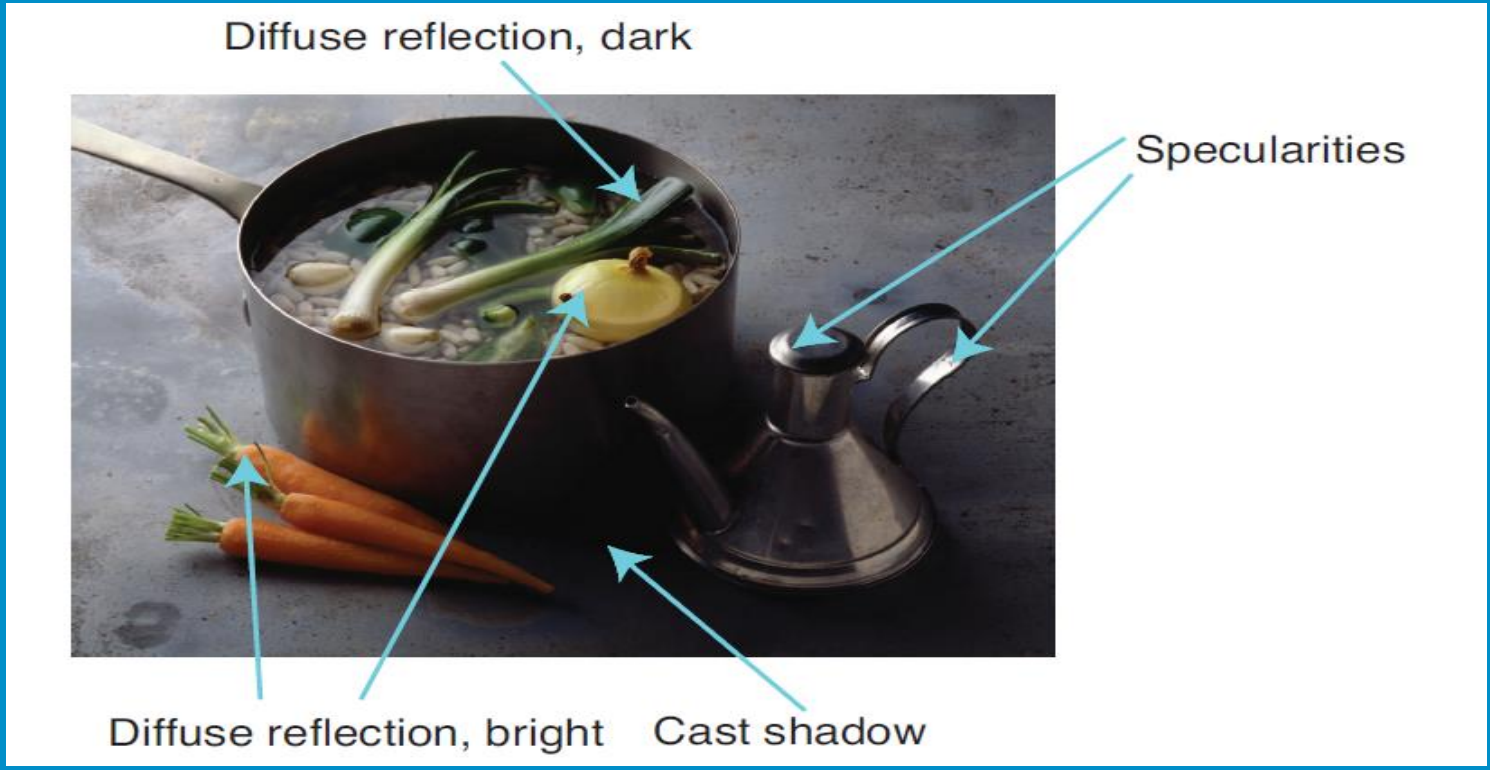
Diffuse **reflection scatters** light evenly across the directions leaving a surface, so the brightness of a diffuse surface doesn't depend on the viewing direction

Specular reflection causes incoming light to leave a surface in a lobe of directions that is determined by the direction the light arrived from

Illumination Model

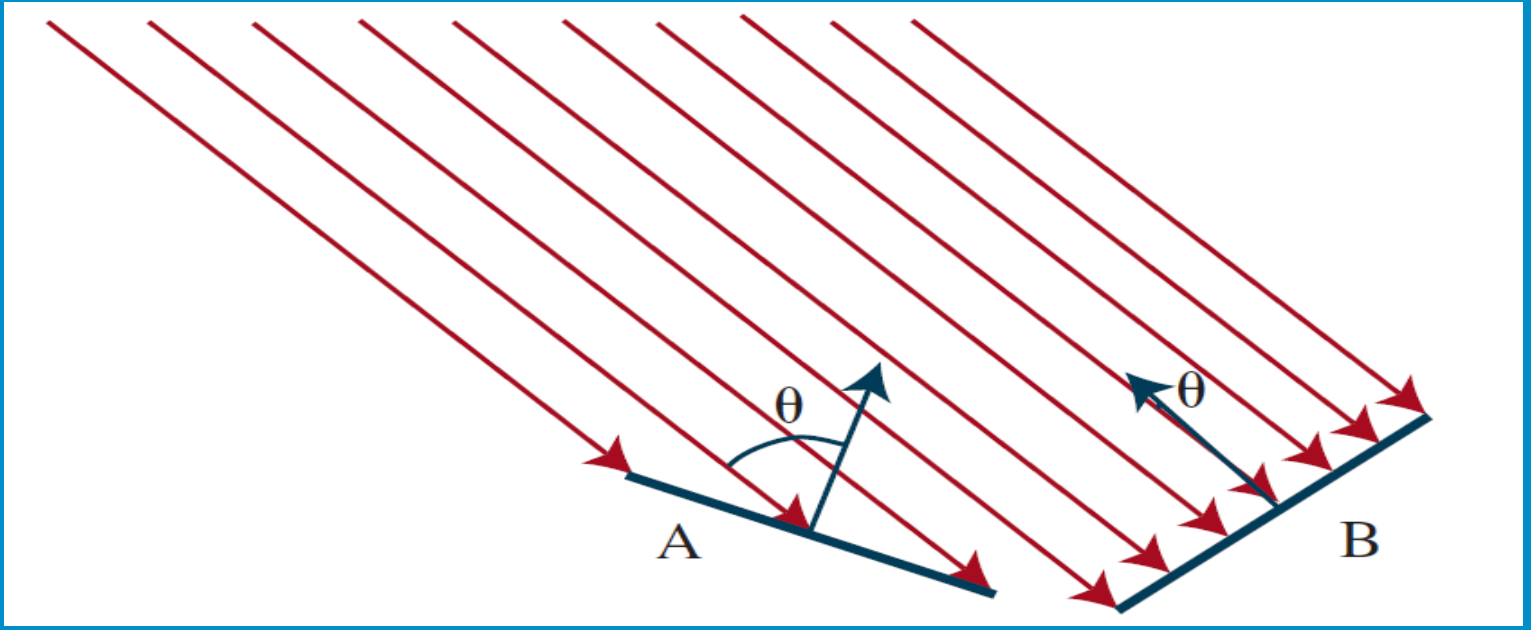
- Model behavior with a distant point light source.
- If the surface cannot see the source, it is in shadow.

Image Formation



This photograph illustrates a variety of illumination effects. There are specularities on the stainless steel cruet. The onions and carrots are bright diffuse surfaces because they face the light direction. The shadows appear at surface points that cannot see the light source at all. Inside the pot are some dark diffuse surfaces where the light strikes at a tangential angle. (There are also some shadows inside the pot.)

Image Formation



Two surface patches are illuminated by a distant point source, whose rays are shown as light arrows. Patch A is tilted away from the source (θ is close to 90°) and collects less energy, because it cuts fewer light rays per unit surface area. Patch B, facing the source (θ is close to 0°), collects more energy.

Image Formation

Color

- Principle of trichromacy
  - match the visual appearance of any spectral energy density, however complex, by mixing appropriate amounts of just three primaries.

- Three primaries RGB Color Model
  - Red, green, and blue primaries.
  - No mixture of two matches the third.

- Color constancy
  - estimate the color the surface would have under white light
  - Ignore the effects of different colored lights

Simple Image Features

- Edges
- Texture
- Optical flow
- Segmentation of natural images

Simple Image Features

Edges

Edges are straight lines or curves in the image plane across which there is a “significant” change in image brightness.

- Edge detection: abstract away from the messy, multi-megabyte image and towards a more compact, abstract representation

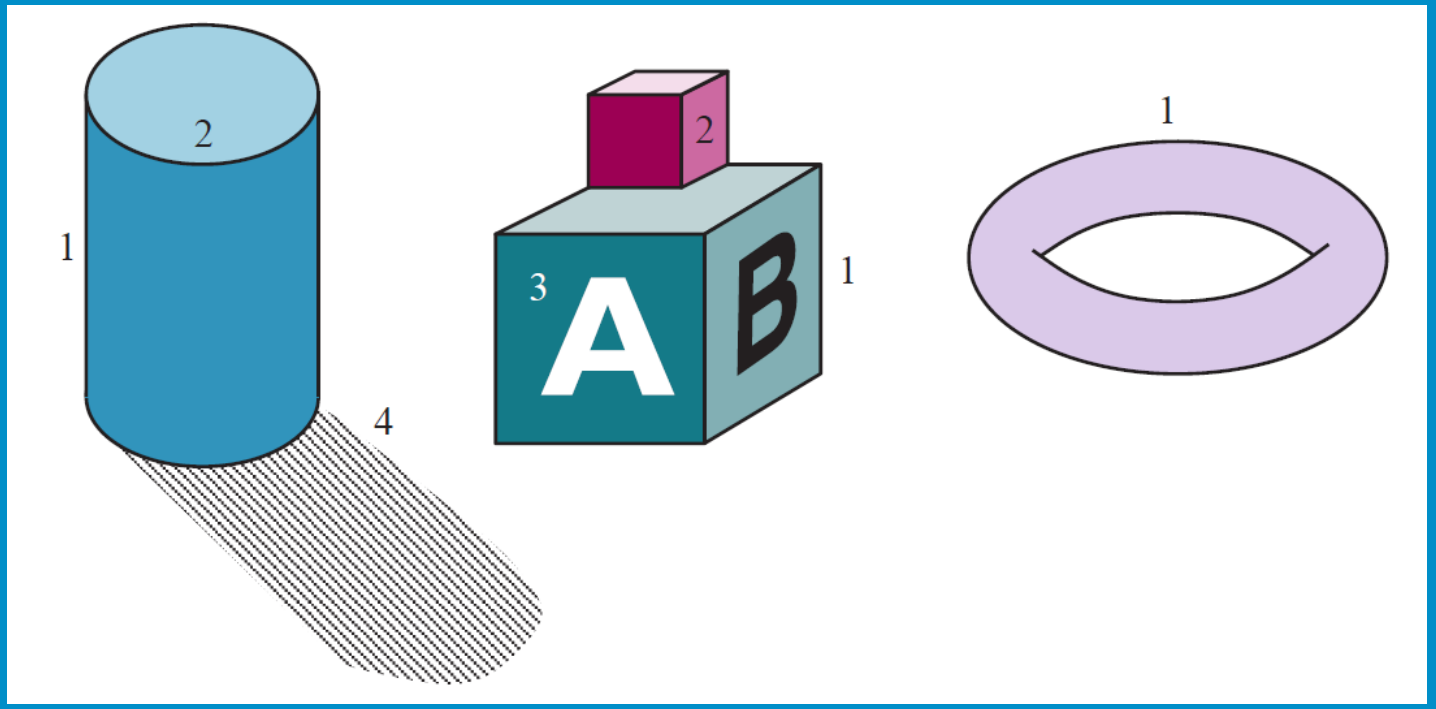
Effects in the scene very often result in large changes in image intensity producing edges

- Depth discontinuities can cause edges
- the surface normal changes
- surface reflectance
- shadow is a discontinuity in illumination

Finding edges requires care

- Intensity difference alone can give mistakes in identifying edges due to noise.
- **Noise**: changes to the value of a pixel that don’t have to do with an edge.

Simple Image Features



Different kinds of edges: (1) depth discontinuities; (2) surface orientation discontinuities; (3) reflectance discontinuities; (4) illumination discontinuities (shadows).

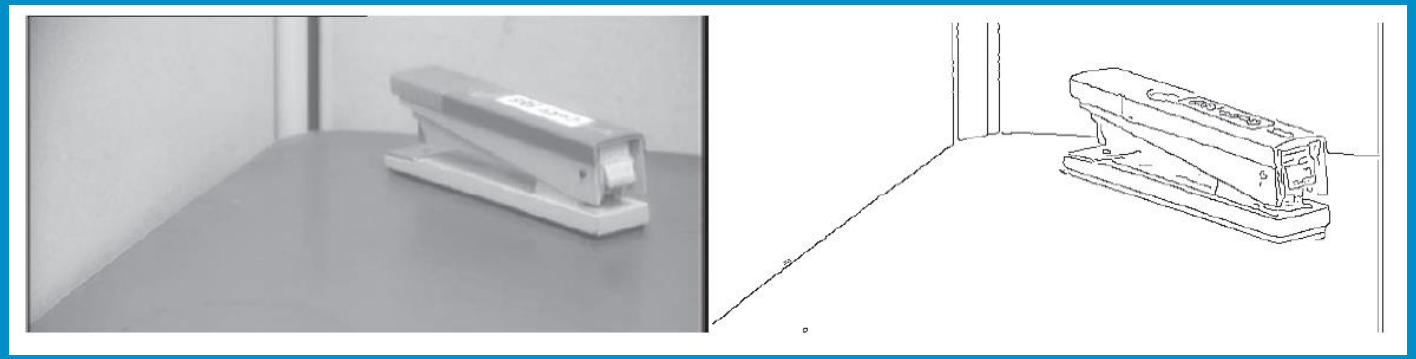
Simple Image Features

- Smoothing involves using surrounding pixels to suppress noise
- Predict value of pixel as a weighted sum of nearby pixels (more weight for closest)
- Gaussian filter: the convolution of two functions f and g (denoted as $h = f * g$)
- The Gaussian filter uses a Gaussian function to assign weights, with closer pixels having more influence, contributing to a smoother and less noisy image.
- Applying a Gaussian filter means replacing the intensity I(x0, y0) with the sum, over all (x, y) pixels, of I(x, y) Gσ(d), where d is the distance from (x0, y0) to (x, y).

$$h(x) = \sum_{u=-\infty}^{+\infty} f(u)\, g(x-u) \qquad \text{in one dimension, or}$$

$$h(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v)\, g(x-u, y-v) \qquad \text{in two dimensions.}$$

Simple Image Features



(a) Photograph of a stapler. (b) Edges computed from (a).

The output is not perfect: there are gaps where no edge appears, and there are "noise" edges that do not correspond to anything of significance in the scene. Later stages of processing will have to correct for these errors

Simple Image Features

Texture

- Texture refers to a **pattern** on a surface that can be sensed visually
- Usual rough model of texture is a repetitive pattern of elements, sometimes called **texels**
- Property of an image patch, rather than a pixel in isolation
- No change when the lighting changes
- Change in a sensible way when the patch rotates.
- Useful for identifying objects and matching patches
- Basic construction for a texture representation
  - compute the gradient orientation at each pixel in the patch
  - characterize the patch by a histogram of orientations

Simple Image Features

Optical flow

Optical flow: apparent motion whenever there is relative movement between the camera and one or more objects in the scene

Describes the direction and speed of motion of features in the image as a result of relative motion between the viewer and the scene.

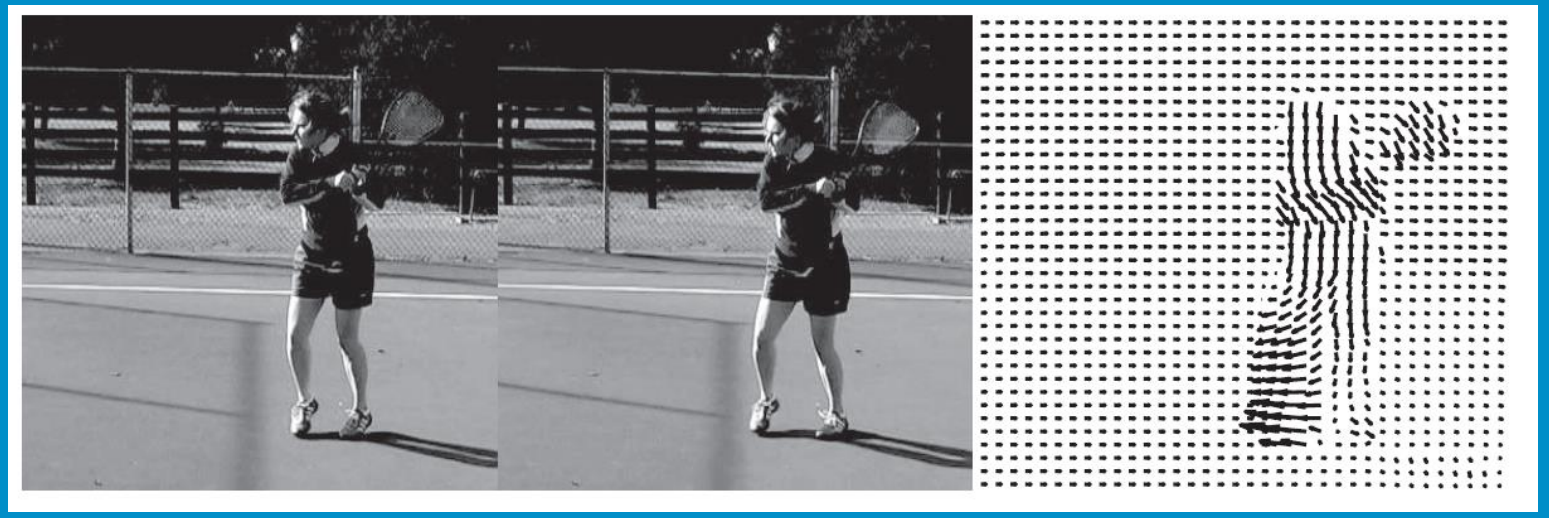
Measure of similarity, **sum of squared differences (SSD):**

$$\mathrm{SSD}(D_x, D_y) = \sum_{(x,y)} (I(x,y,t) - I(x+D_x, y+D_y, t+D_t))^2 .$$

Here, (x, y) ranges over pixels in the block centered at (x_0, y_0). We find the (D_x, D_y) that minimizes the SSD. The optical flow at (x_0, y_0) is then $(v_x, v_y) = (D_x/D_t, D_y/D_t)$.

Requires some texture in the scene, resulting in windows containing a significant variation in brightness among the pixels

Simple Image Features



Two frames of a video sequence and the optical flow field corresponding to the displacement from one frame to the other. Note how the movement of the tennis racket and the front leg is captured by the directions of the arrows.

Simple Image Features

Segmentation of natural images

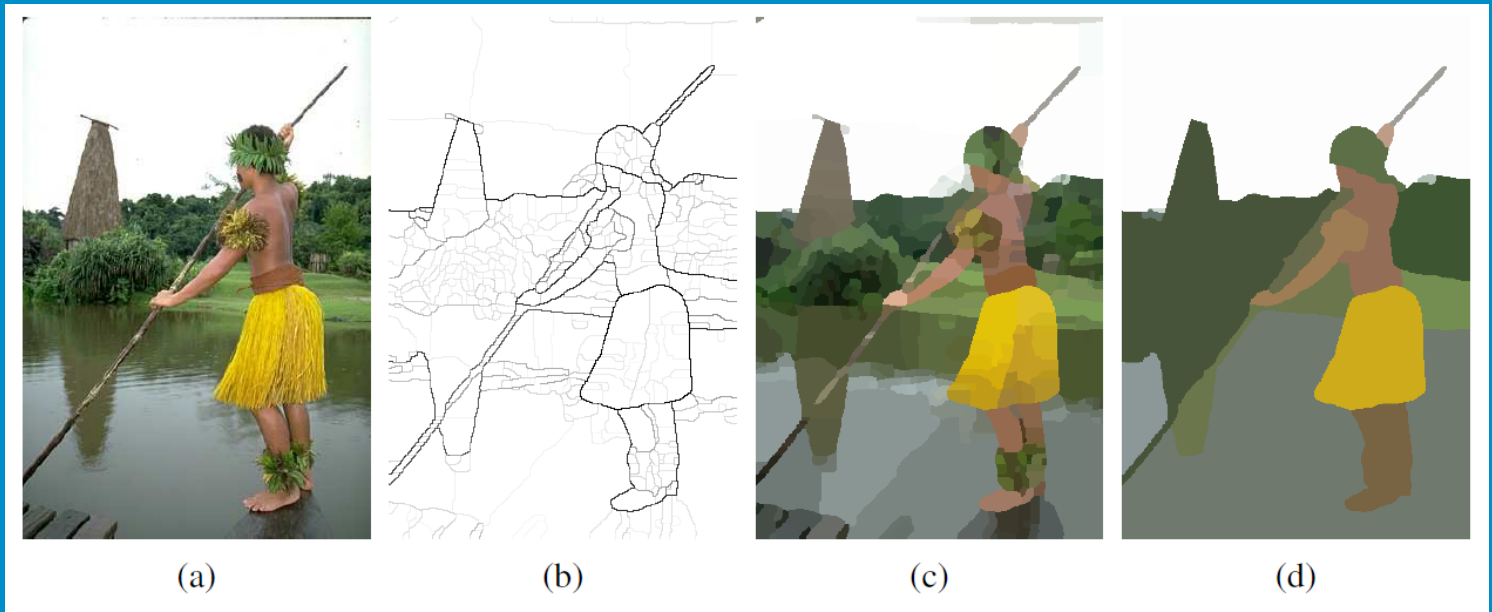
Segmentation is the process of breaking an image into groups of similar pixels.

Focus: detecting the boundaries or group themselves (regions)

Classification problem

- Boundary curve at pixel location (x, y), an orientation θ. An image neighborhood centered at (x, y) looks roughly like a disk, cut into two halves by a diameter oriented at θ.
- Compute the probability $P_b(x, y, \theta)$ that there is a boundary curve at that pixel along that orientation by comparing features in the two halves.
- To train a machine learning classifier using a data set of natural images in which humans have marked the ground truth boundaries
- The goal of the classifier is to mark exactly those boundaries marked by humans and no other
- Alternative approach is based on trying to “cluster” the pixels

Simple Image Features



(a) Original image.
(b) Boundary contours, where the higher the P_b value, the darker the contour.
(c) Segmentation into regions, corresponding to a fine partition of the image. Regions are rendered in their mean colors.
(d) Segmentation into regions, corresponding to a coarser partition of the image, resulting in fewer regions.

Classifying Images

Modern systems classify images using appearance (i.e., color and texture, as opposed to geometry)

There are two difficulties

- Different instances of the same class could look different—some cats (black and orange)
- Look different at different times depending on several effects
  - Lighting
  - Foreshortening
  - Aspect
  - Occlusion
  - Deformation
- Deal with these problems by learning representations and classifiers from very large quantities of training data using a convolutional neural network.

Classifying Images

Image classification with convolutional neural networks (CNN)
With enough training data and enough training ingenuity, CNNs produce very successful classification systems

Images can have small alterations without changing the identity

local patterns can be quite informative.

spatial relations between local patterns are informative

Convolution followed by a ReLU activation function—as a local pattern detector
- convolution measures how much each local window of the image looks like the kernel pattern
- ReLU sets low-scoring windows to zero, and emphasizes high-scoring window

convolution with multiple kernels finds multiple patterns
composite patterns can be detected by applying another layer to the output of the first layer.

Classifying Images

Data set augmentation: training examples are copied and modified slightly
Images can have small alterations without changing the identity

- randomly shift, rotate,or stretch an image by a small amount, or randomly shift the hue of the pixels by a small amount local patterns
- CNN-based classifiers are good at ignoring patterns that aren't discriminative
- **Context:** patterns that lie off the object might be discriminative
    - e.g., a cat toy, a collar with a little bell, or a dish of cat food might actually help tell that we are looking at a cat

Detecting Objects

Object detectors find multiple objects in an image, **report** what **class** each object is and also **report where** each object is by giving a **bounding box** around the object

Building an object detector:

- looking at a small sliding window onto the larger image—a rectangle.
- At each spot, we classify what we see in the window, using a CNN classifier

Details:

- Decide on a window shape
- Build a classifier for windows
- Decide which windows to look at
- Choose which windows to report
- Report precise locations of objects using these windows

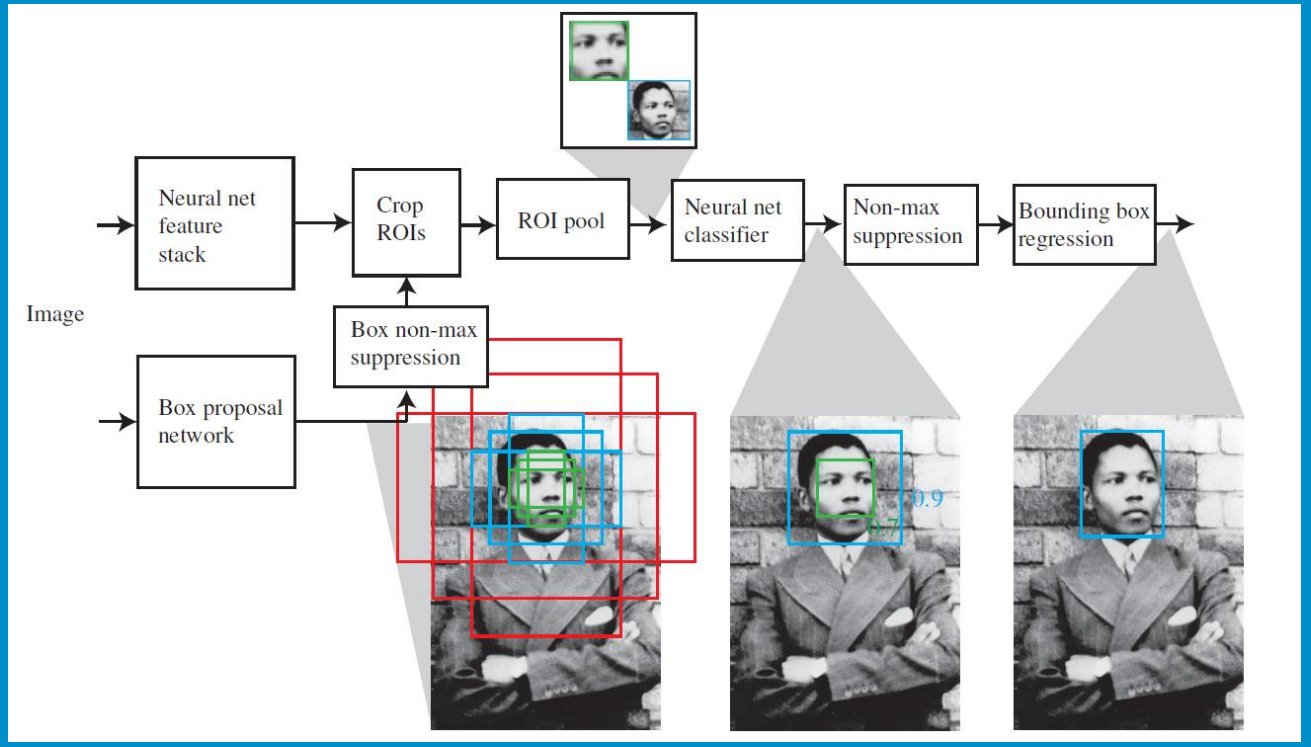
A network that finds regions with objects is called a **regional proposal network** (RPN)

- Faster RCNN encodes a large collection of bounding boxes as a map of fixed size

Detecting Objects

- Construct a 3D block where each spatial location in the block has two dimensions for the center point and one dimension for the type of box
- Any box with a good enough objectness score is called a region of interest (ROI)
- Make them have the same number of features by sampling the pixels to extract features, a process called ROI pooling
- Greedy algorithm for deciding windows to report called **non-maximum suppression:** sort based scores over threshold, choose highest score, discared overlapping
- **Bounding box regression:** trim the window down to a proper bounding box

Detecting Objects



Faster RCNN (Region-based Convolutional Neural Network) uses two networks

- first network computes “objectness” scores of candidate image boxes, called “anchor boxes,” centered at a grid point.
- second network is a feature stack that computes a representation of the image suitable for classification

Using Computer Vision

Understanding what people are doing

Difficulty of how to link observations of the body and the objects nearby to the goals and intentions of the moving people.

Another difficulty is caused by time scale. What someone is doing depends quite strongly on the time scale

Unrelated behaviors are going on at the same time

Learned classifiers are guaranteed to behave well only if the training and test data come from the same distribution.

For activity data, the relationship between training and test data is more untrustworthy people do so many things in so many context

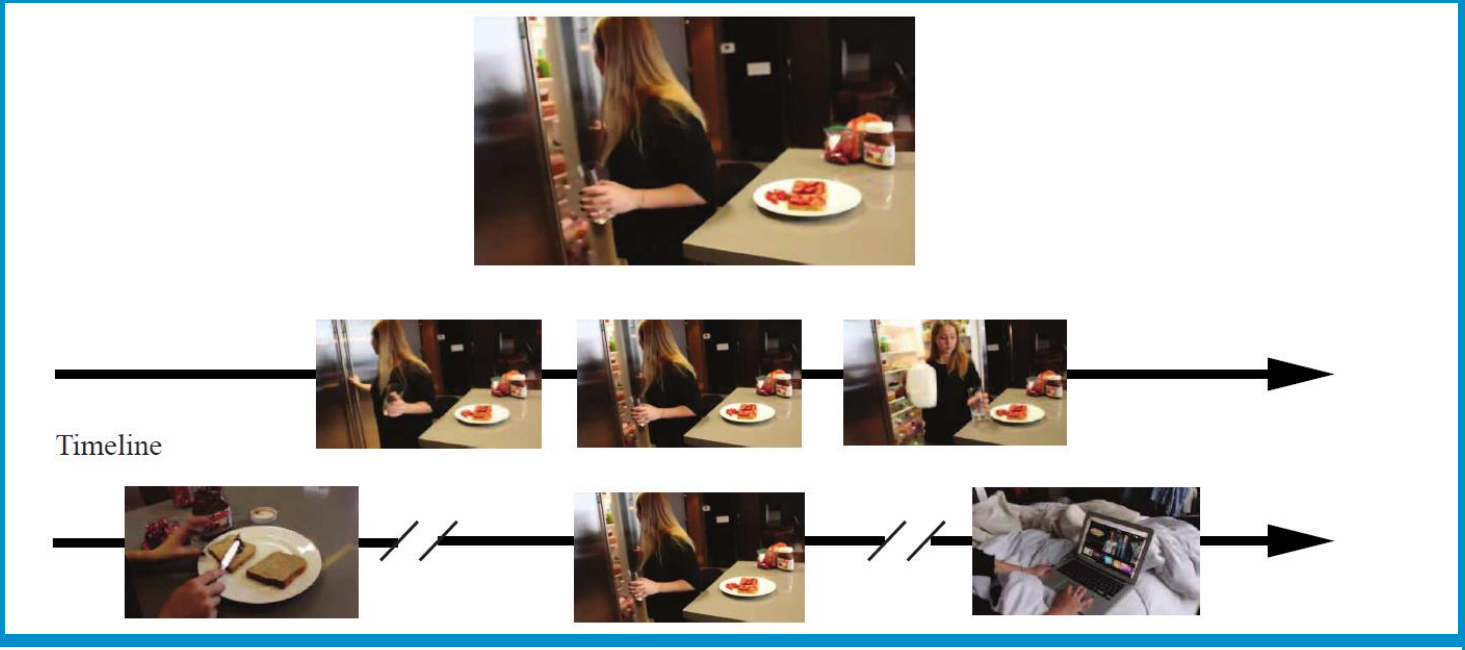
Using Computer Vision

Reconstructing humans from a single image is now practical. Each row shows a reconstruction of 3D body shape obtained using a single image

Using Computer Vision

The same action can look very different; and different actions can look similar

Using Computer Vision



What you call an action depends on the time scale. The single frame at the top is best described as opening the fridge (you don't gaze at the contents when you close a fridge). But if you look at a short clip of video (indicated by the frames in the center row), the action is best described as getting milk from the fridge. If you look at a long clip (the frames in the bottom row), the action is best described as fixing a snack.

Using Computer Vision

Linking pictures and words

- Tagging systems that tag images with relevant words.
- Tags aren’t a comprehensive description of what is happening in an image
- Captioning systems—systems that write a caption of one or more sentences describing the image.
- Current methods for captioning use detectors to find a set of words that describe the image, and provide those words to a sequence model that is trained to generate a sentence
- To establish whether a system has a good representation of what is happening in an image is visual question answering or VQA system and a visual dialog system

Using Computer Vision

A baby eating a piece of food in his mouth

A young boy eating a piece of cake

A small bird is perched on a branch

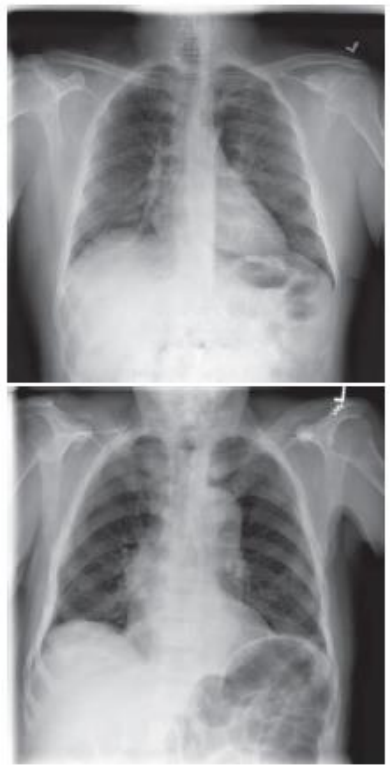
A small brown bear is sitting in the grass

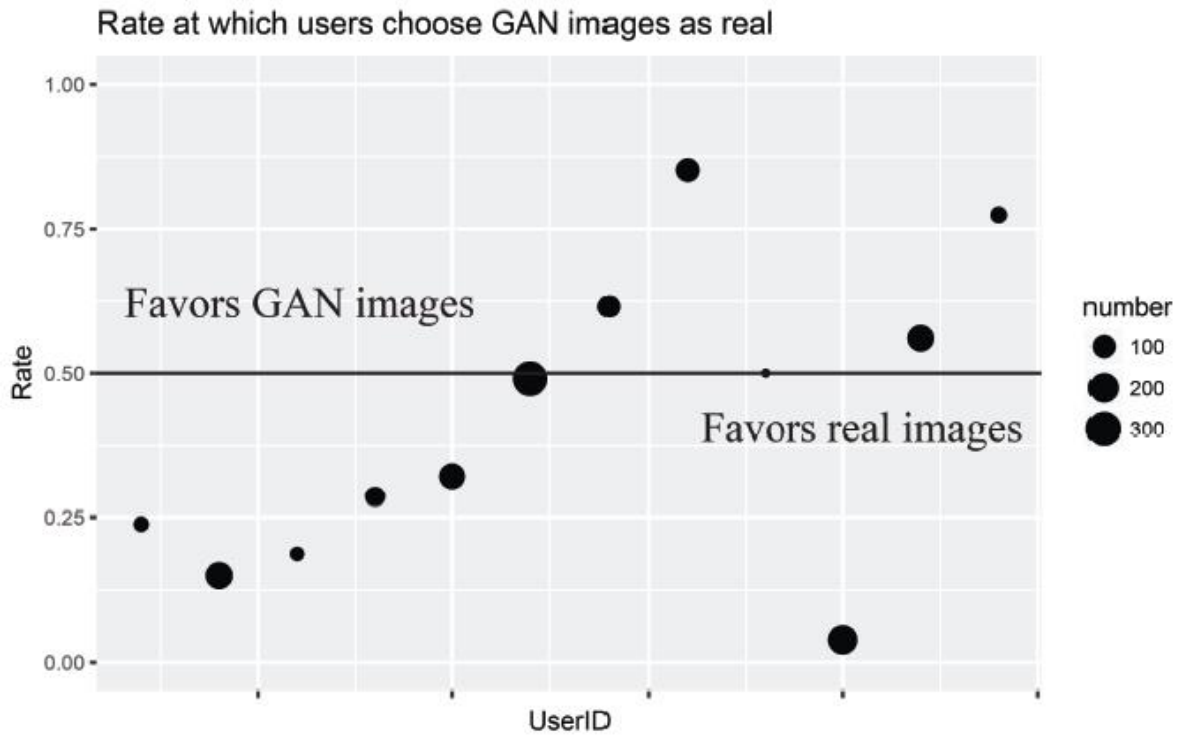
Automated image captioning systems produce some good results and some failures.

Using Computer Vision

Visual question-answering systems produce answers (typically chosen from a multiple-choice set) to natural-language questions about images

Using Computer Vision



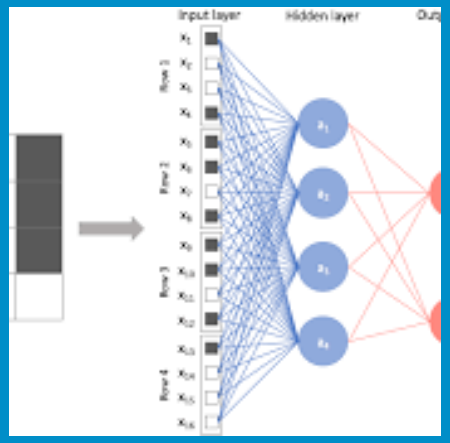


GAN generated images of lung X-rays. results of a test asking radiologists, given a pair of X-rays as seen on the left, to tell which is the real X-ray

Summary

- Representations of images capture edges, texture, optical flow, and regions
- Convolutional neural networks produce accurate image classifiers that use learned features.
- Image classifiers can be turned into object detectors
- With more than one view of a scene, it is possible to recover the 3D structure of the scene and the relationship between views

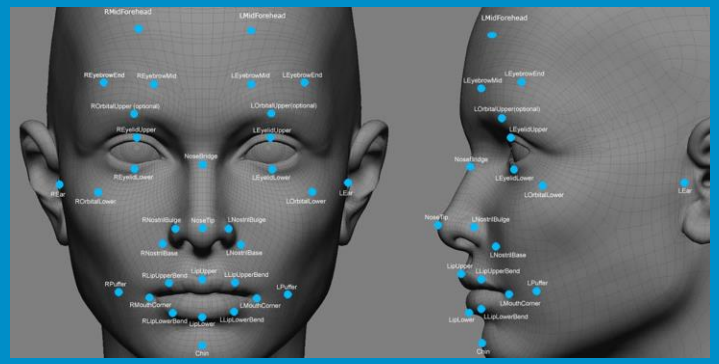
Convolutional Neural Network (CNN)

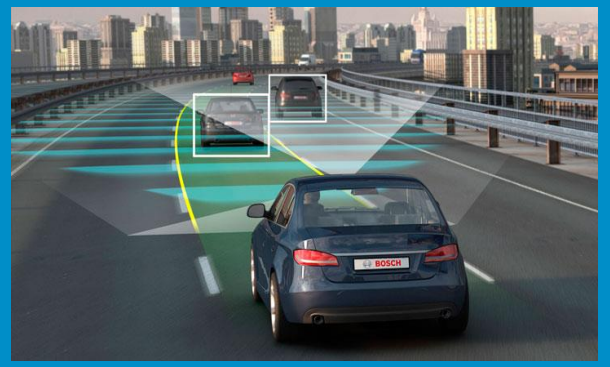


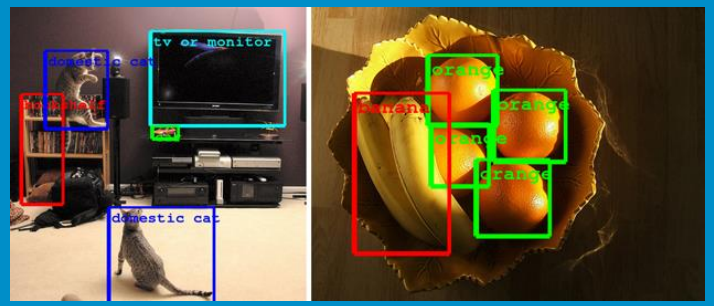
HUMBER

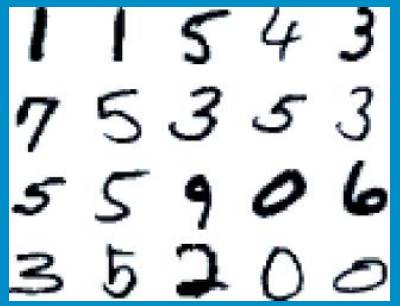
WE ARE HUMBER

Convolutional Neural Network for Image/Video Recognition









Introduction to Neural networks

- Neural networks are a class of machine learning models inspired by the structure and function of the human brain.
- These computational models consist of interconnected nodes, or artificial neurons, organized into layers.
- The fundamental building blocks are the neurons, which process information and transmit signals to create complex patterns and representations.
- Neural networks excel at learning from data, making them particularly powerful for tasks such as pattern recognition, classification, regression, and decision-making.

Neural Network Structure

- *Layers:*

  1. **Input Layer:** Receives input data.
  2. **Hidden Layers:** Process information hierarchically.
  3. **Output Layer:** Produces the network's output.

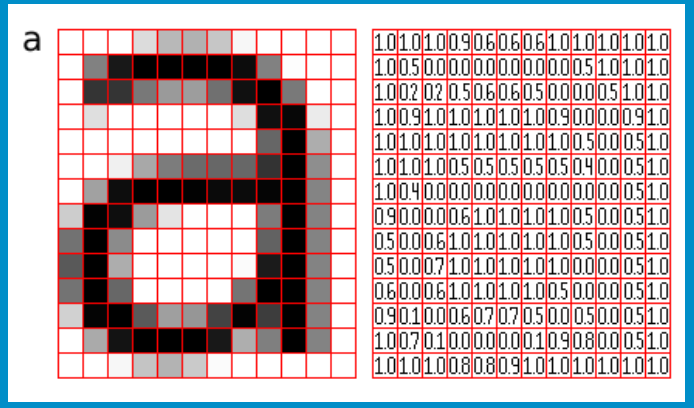
- *Connections:*
  - Neurons in different layers are connected with weights.

Types of Neural Networks

1. **Feedforward Neural Networks (FNN):**
    1. Information flows in one direction.
    2. Suitable for various tasks.
2. **Recurrent Neural Networks (RNN):**
    1. Contains loops to process sequential data.
    2. Effective in tasks with temporal dependencies.
3. **Convolutional Neural Networks (CNN):**
    1. Specialized for image and grid data.
    2. Utilizes convolutional layers for feature extraction.

Convolutional Neural Networks

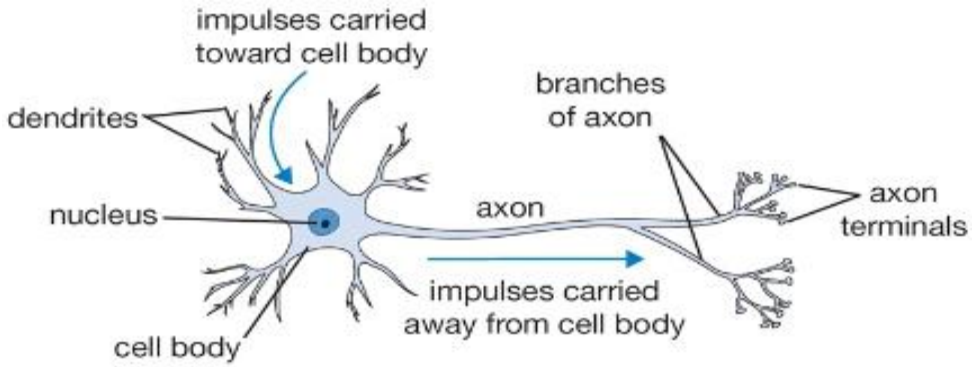
- CNN or ConvNet
- A class of neural networks that specializes in processing data that has a grid-like topology, such as an image.
- A digital image is a binary representation of visual data. It consists of a series of pixels organized as a grid that contains pixel values to signify the brightness and color each pixel should be.

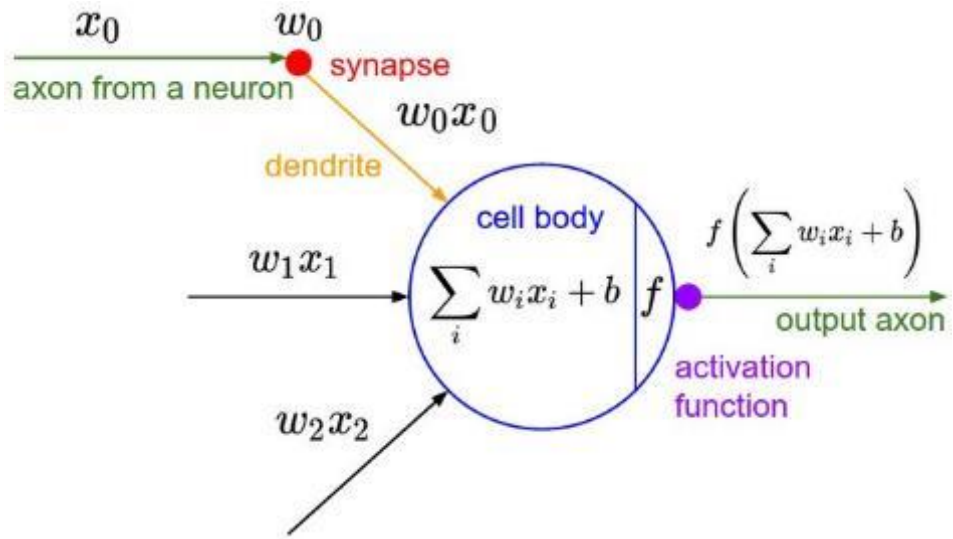


Perceptron

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

Rosenblatt, 1950

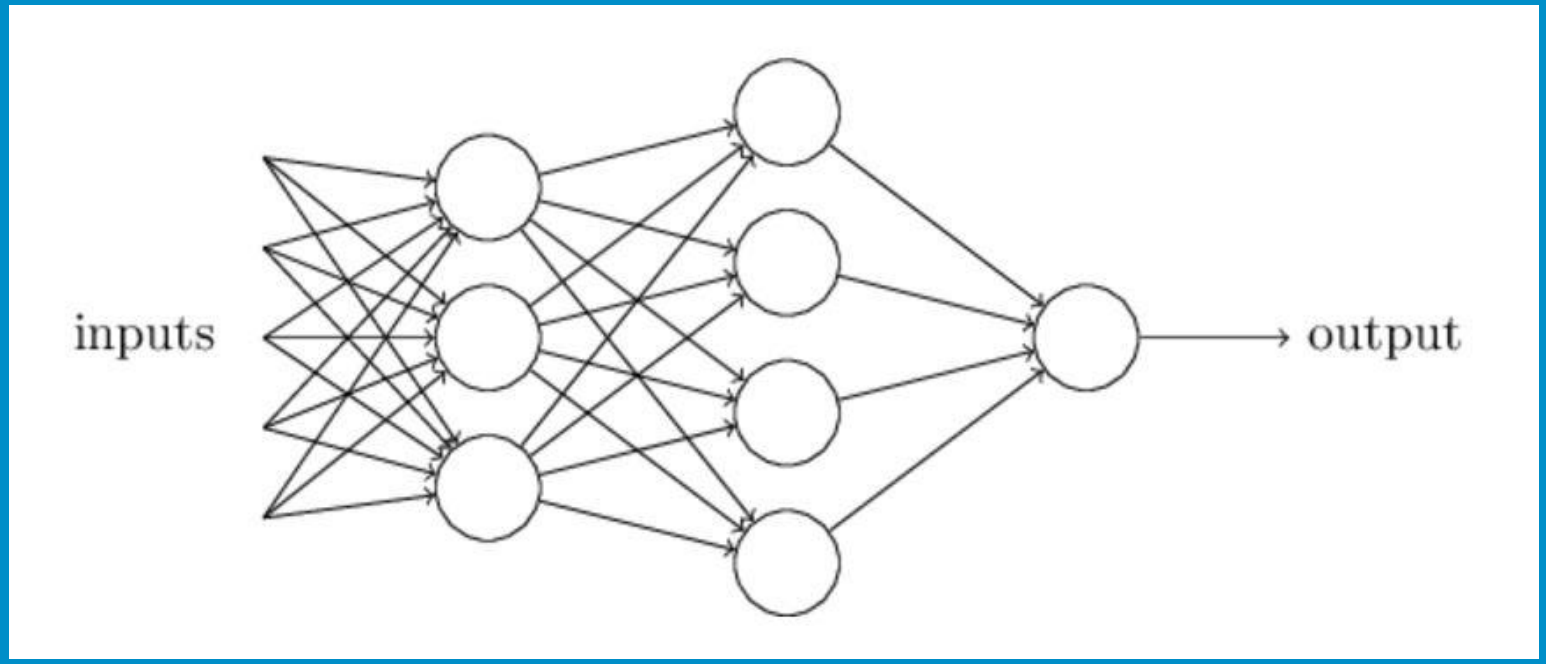




- A single neuron is a linear classifier, where the decision function is the activation function
- w x + b, a linear classifier, a neuron,
  - It's a dot product of two vectors, scalar product
  - A template matching, a correlation, the template w and the input vector x (or a matched filter)
  - Also, an algebraic distance which is non-linear (therefore the solution is usually nonlinear!)

A biological neuron and its mathematical model.

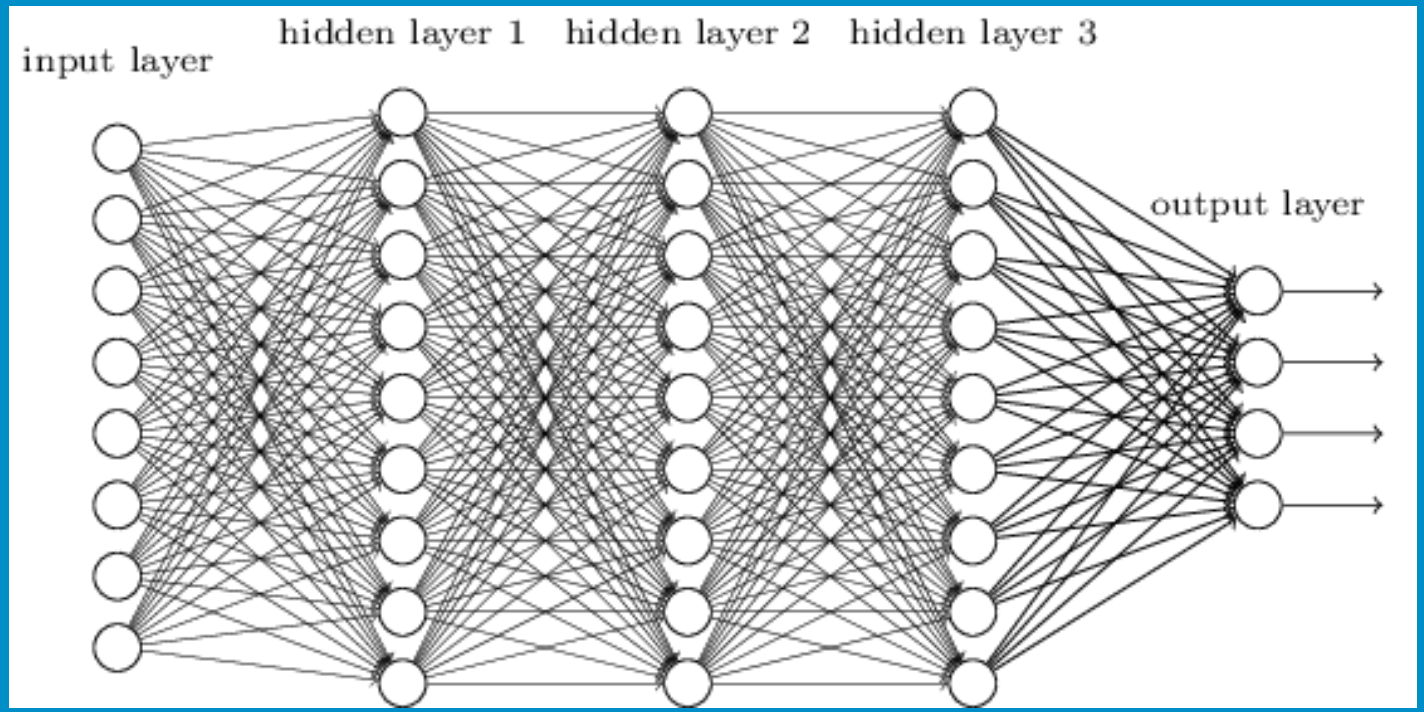
Multi-layer perceptron



Michael Nielsen, 2016

CNN: Smaller Network

- From this fully connected model, do we really need all the edges?
- We know it is good to learn a small model.
- Can some of these be shared?

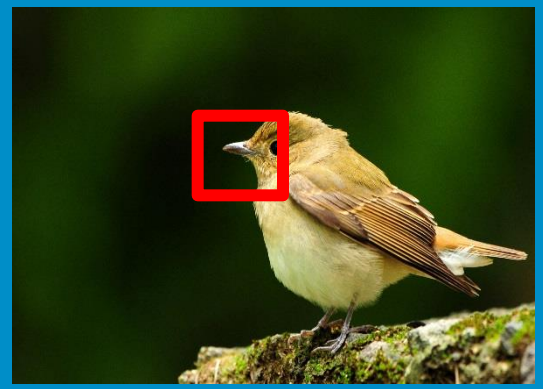


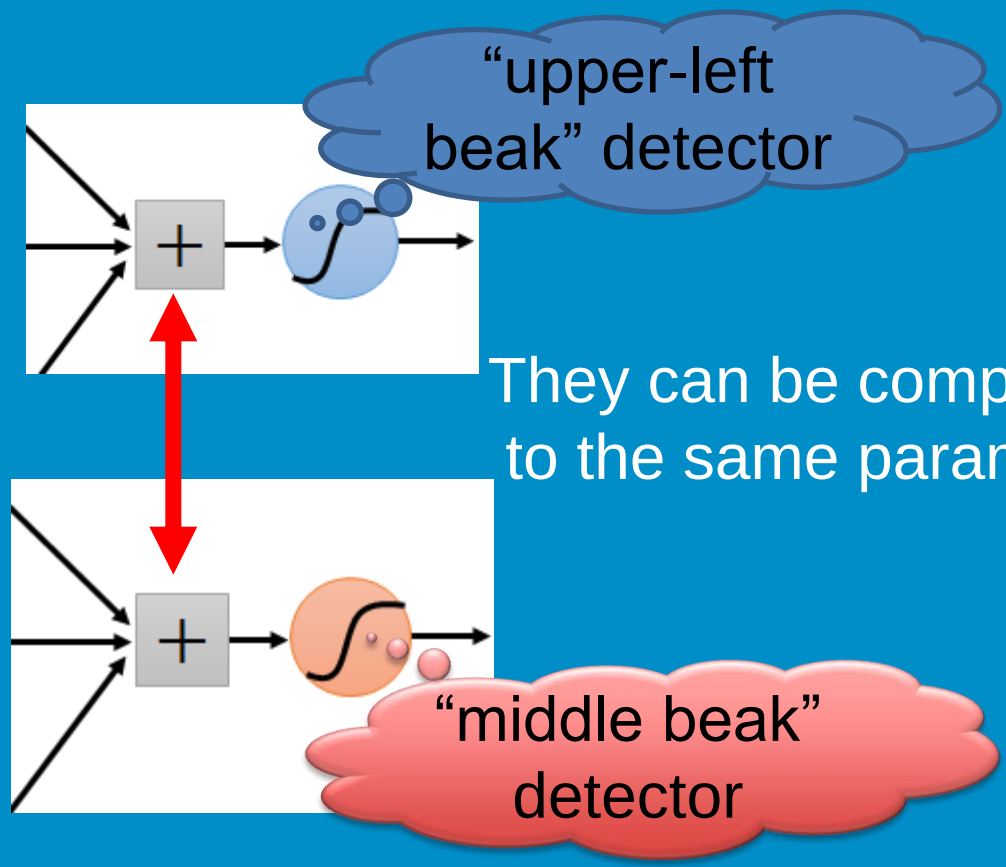
Consider learning an image

- Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters

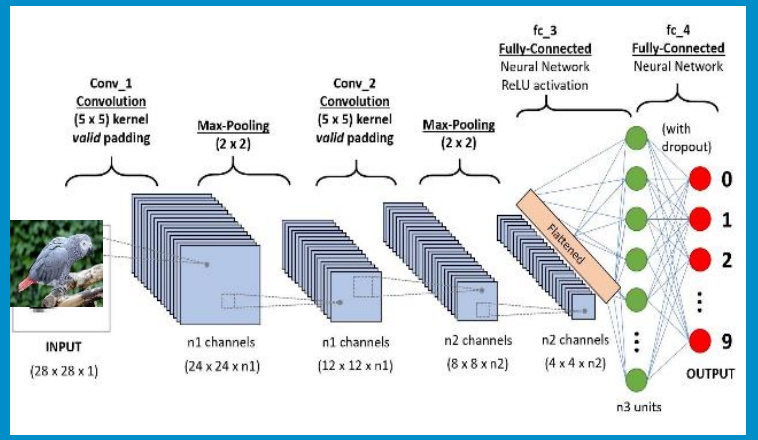
Same pattern appears in different places:
They can be compressed!
What about training a lot of such “small” detectors
and each detector must “move around”.





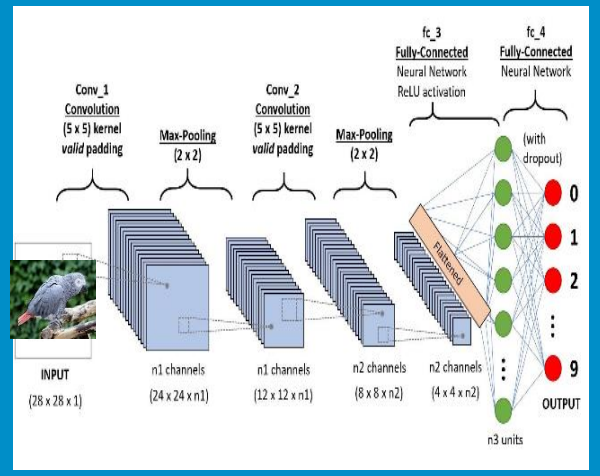
Motivation behind CNN

- Sparse interaction
  - Reducing number of connections (making kernel/filter smaller than the input).
- Parameter sharing
  - Shared weights on the edges (same weights are the same for all hidden neurons in a given layer)
- Pooling further reduces the complexity (It reduces the dimensions of the feature maps, thus reduce training time)



Applications of CNN

- Image classification
- Speech recognition programs
- Facial Recognition
- Analyzing Documents
- Collecting Historic and Environmental Elements
- Understanding Climate
- Data-driven personalized advertising



Architecture of Convolutional Neural Network

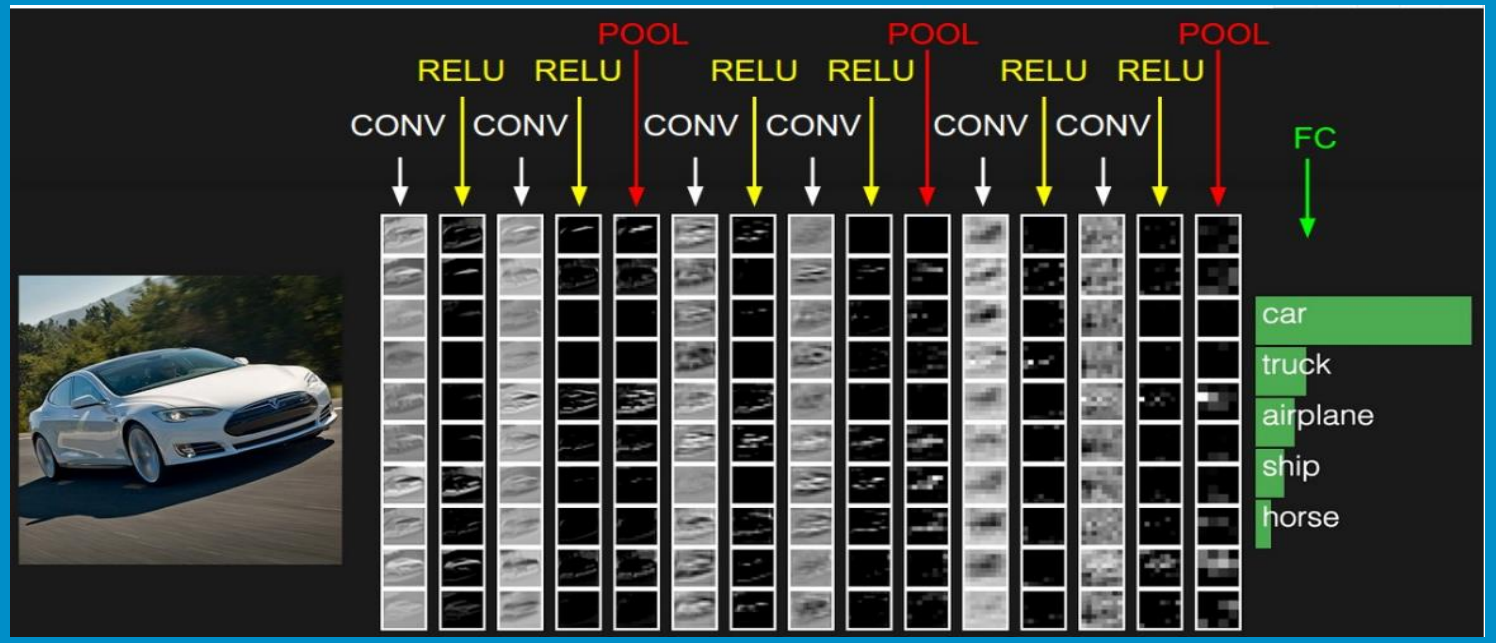
[3D visualization of a ANN and CNN]

HUMBER

WE ARE HUMBER

A simple CNN structure

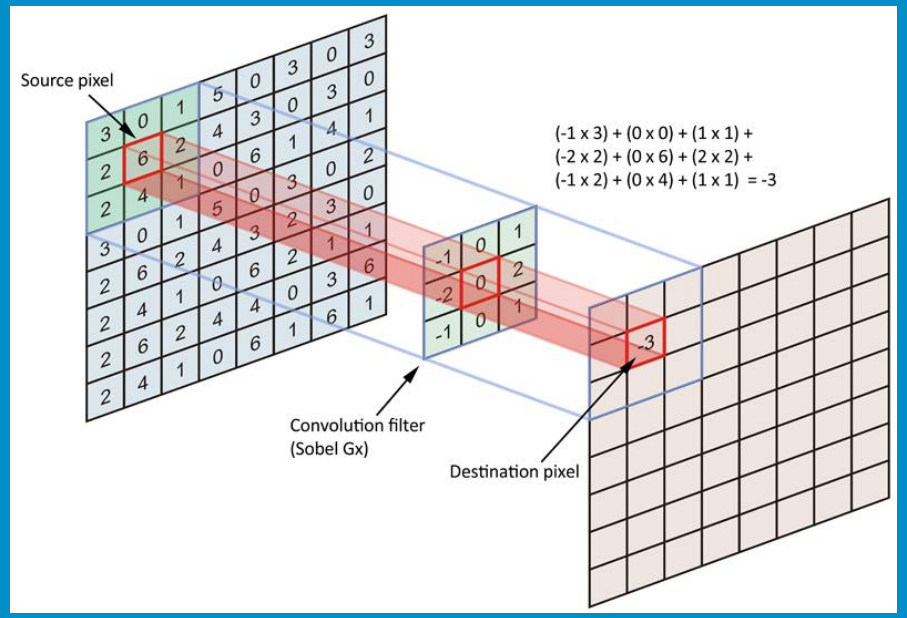
- A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

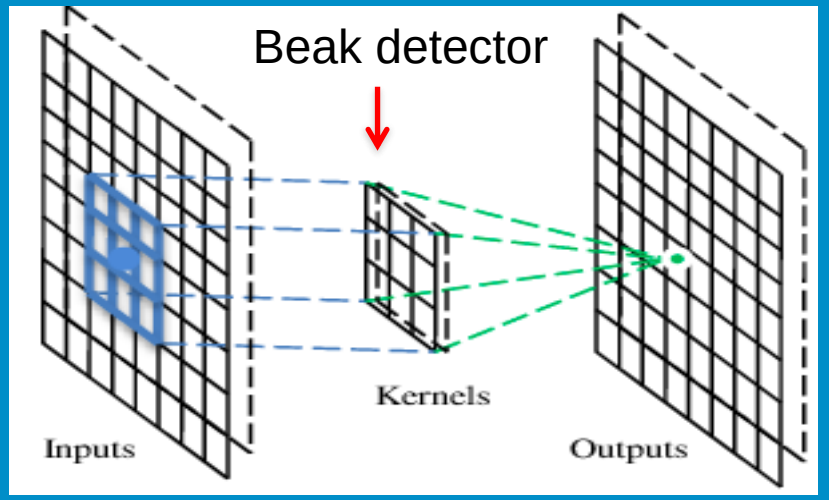


- **CONV: Convolutional kernel layer**
- **POOL: Dimension reduction layer**
- **FC: Fully connection layer**
- **RELU: Activation function**

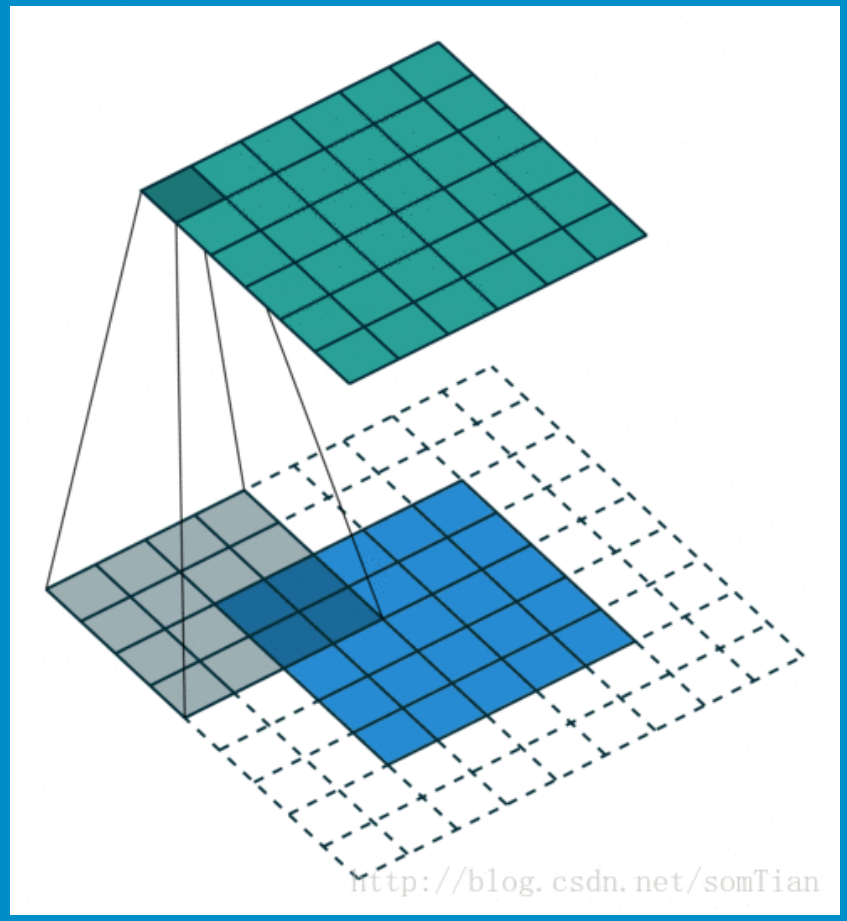
A convolutional layer

- A CNN is a neural network with some convolutional layers (and some other layers).
- The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.
- A convolutional layer has a number of filters that does convolutional operation.





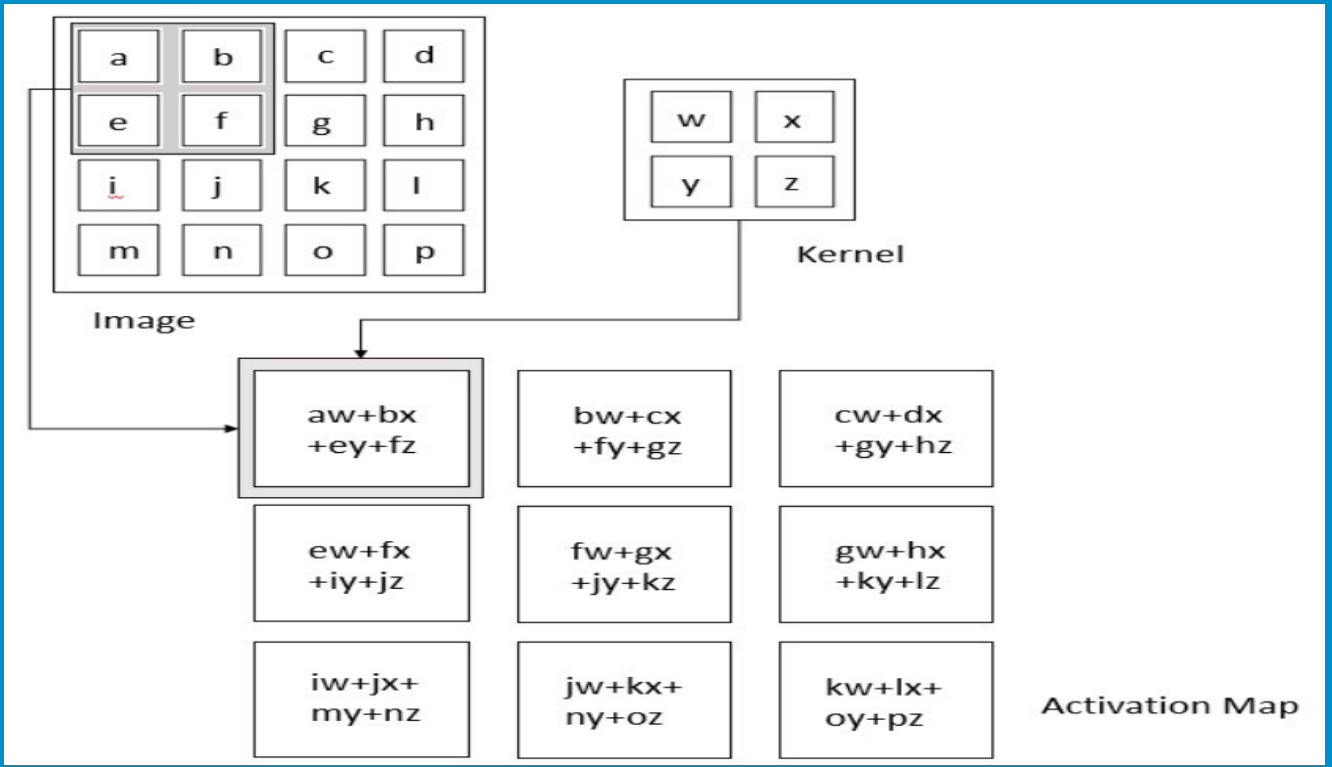
Convolutional kernel



Convolutional Layer

- Forward pass:
  - Kernel slides across the height and width of the image-producing the image representation of that receptive region.
  - This produces a two-dimensional representation of the image known as an **activation map** that gives the response of the kernel at each spatial position of the image.
  - The sliding size of the kernel is called a **stride**.

Convolutional Layer (Cont'd)



Convolution Operation (Source: Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville)

Convolution

These are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Dot product →

3 -1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Convolution

If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

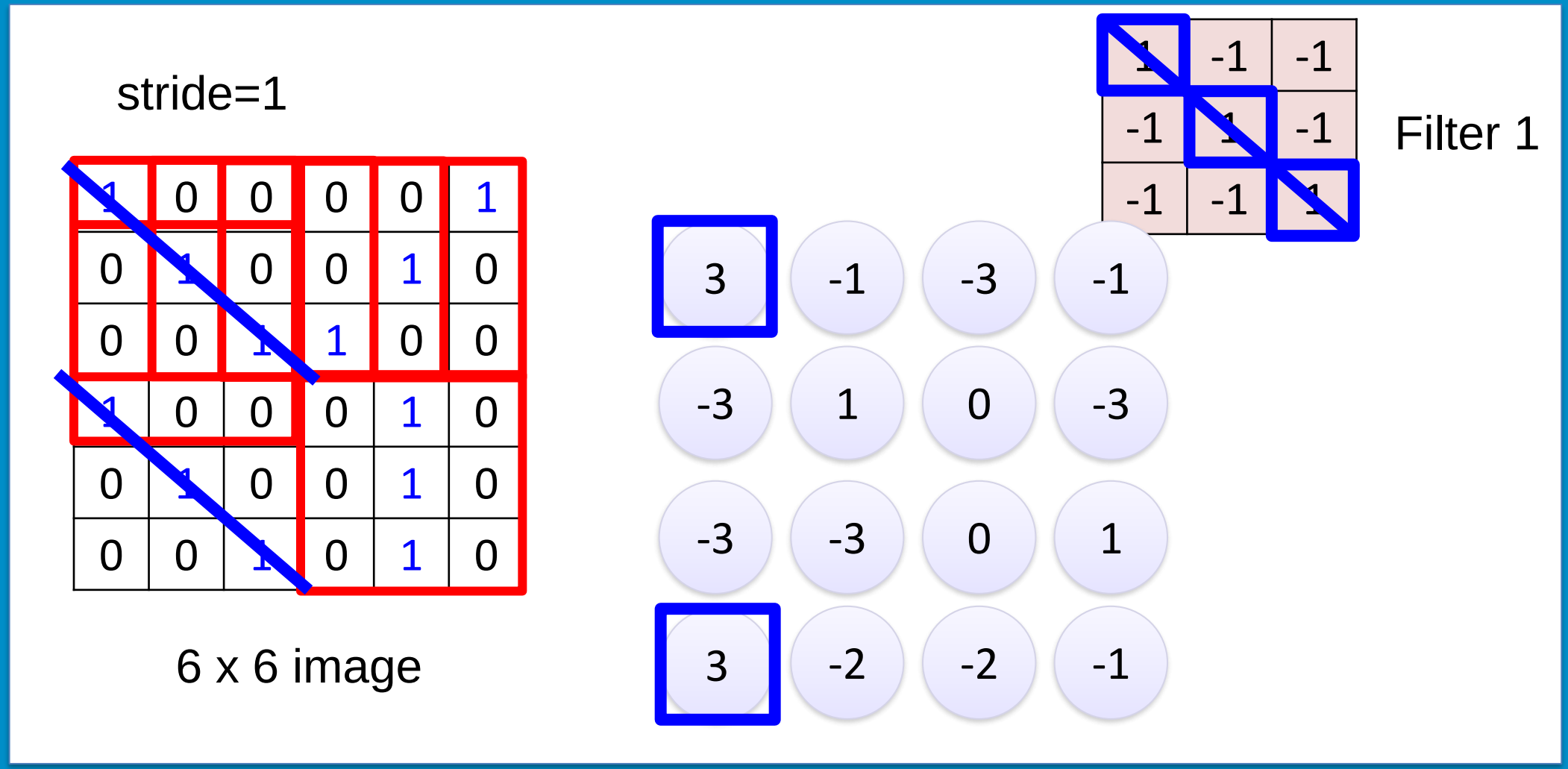
6 x 6 image

3 -3

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Convolution



Convolution

stride=1

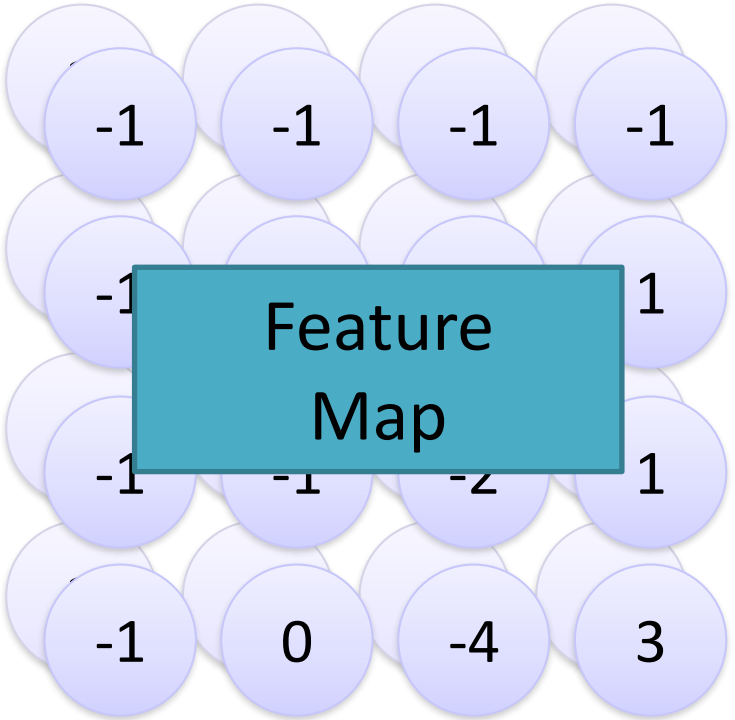
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Repeat this for each filter

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



59

Padding

- In a convolutional layer, we observe that the pixels located on the corners and the edges are used much less than those in the middle
- Padding, adds rows and columns of zeros to the input image.
- If we apply padding P in an input image of size W X H, the output image has dimensions (W+2P) X (H+2P).
- By using padding in a convolutional layer, we increase the contribution of pixels at the corners and the edges to the learning procedure.

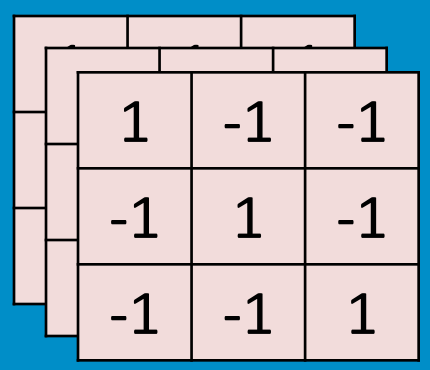
Output size of a convolutional layer

- To compute the output size of a convolutional layer:
  - O = Size (width) of output image.
  - I = Size (width) of input image.
  - K = Size (width) of kernels used in the Conv Layer.
  - N = Number of kernels.
  - S = Stride of the convolution operation.
  - P = Padding

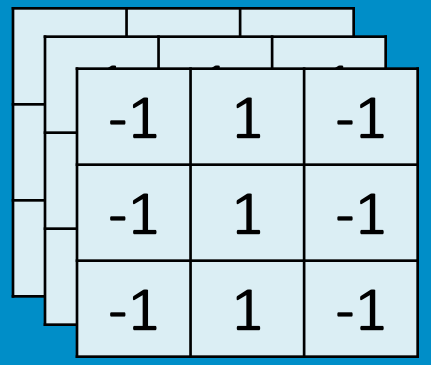
$$O_{width} = (I - K + 2P) / S + 1$$

- Note: to keep the size of the input unchanged after the convolution layer, when s=1:
  - P = (K - 1) / 2

Color image: RGB 3 channels

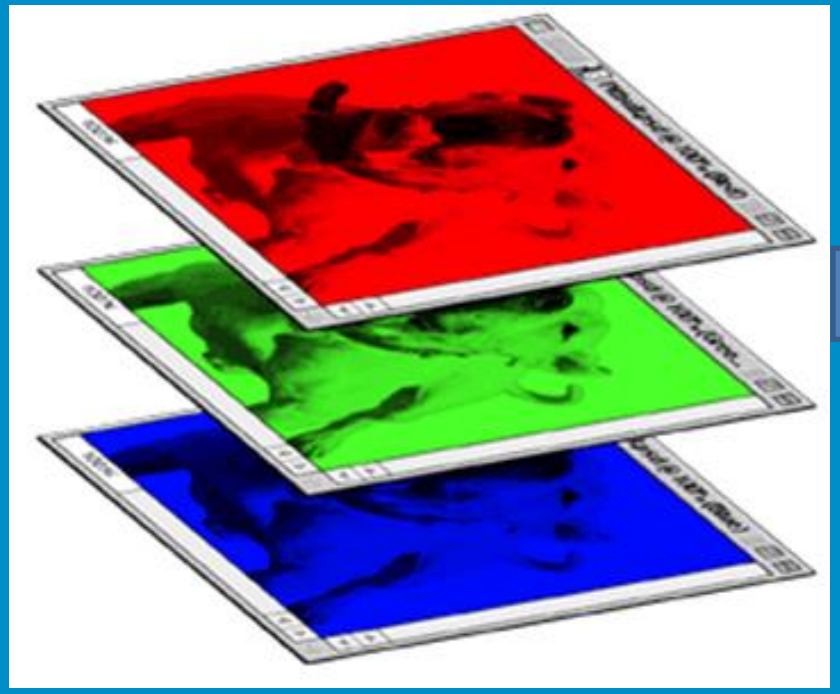


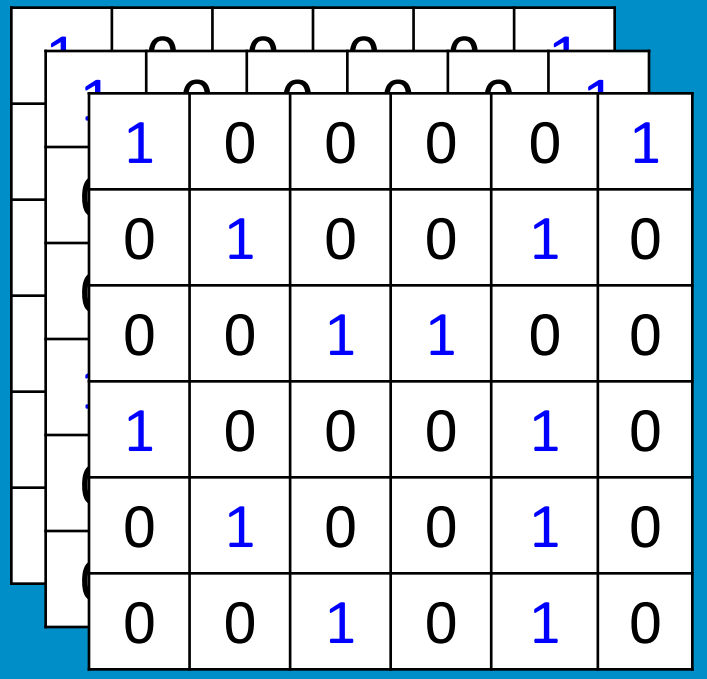
Filter 1



Filter 2

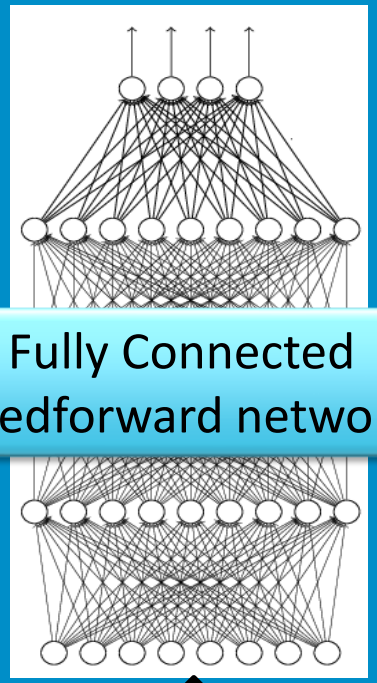
Color image





The whole CNN

Bird African-Grey ……



Fully Connected Feedforward network

Convolution

Pooling

Convolution

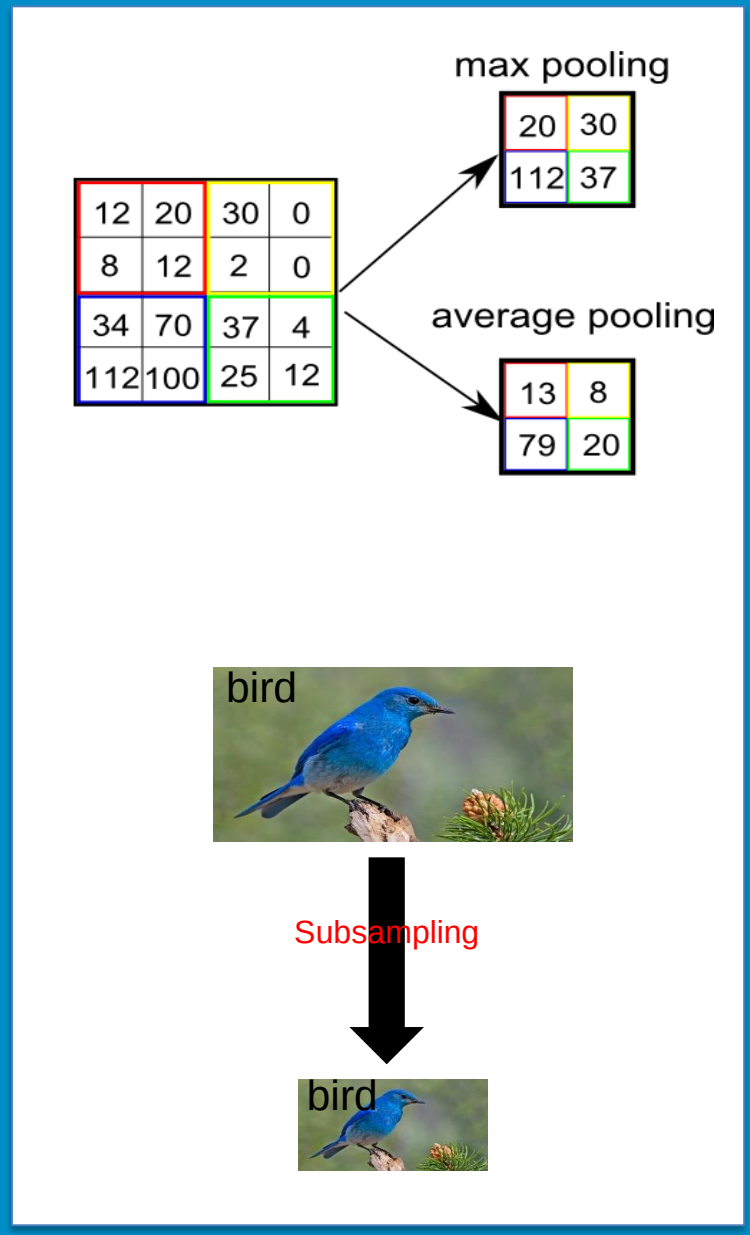
Pooling

Can repeat many times

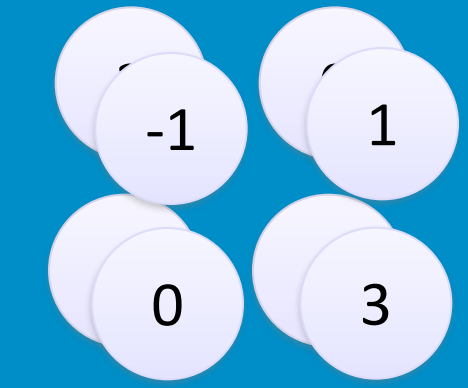
Flattened

Pooling layer

- The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs.
  - Reduce spatial size of the representation,
  - Decrease required amount of computation and weights.
- Pooling operation is processed on every slice individually.
- Pooling functions:
  - Average of the rectangular neighborhood
  - L2 norm of the rectangular neighborhood
  - Weighted average based on the distance from the central pixel.
  - Max pooling: most popular, reports the maximum output from the neighborhood.



The whole CNN



A new image

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Smaller than the original image

- Pooling filter size is almost always 2×2 pixels applied with a stride of 2 pixels
- Reduce the size of each feature map by a factor of 2

The number of channels is the number of filters

Pooling Layers in PyTorch

- A pooling layer can be created like this:
  - Usually, the kernel size and the stride length will be equal.
  - No trainable parameters:

The whole CNN

Bird African-Grey ……

Convolution

Max Pooling

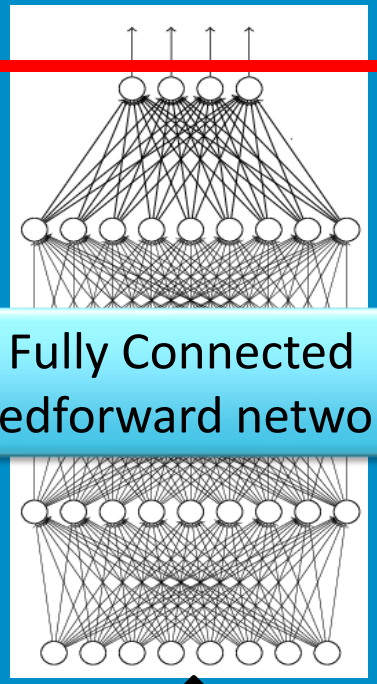
A new image

Convolution

Max Pooling

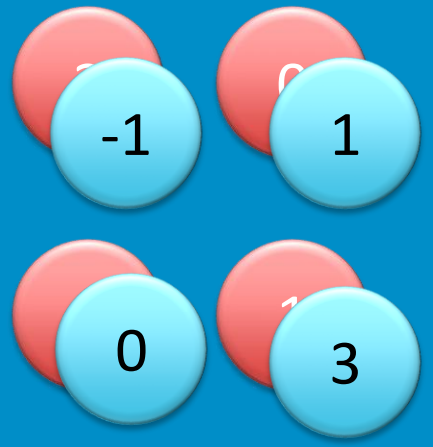
A new image

Flattened

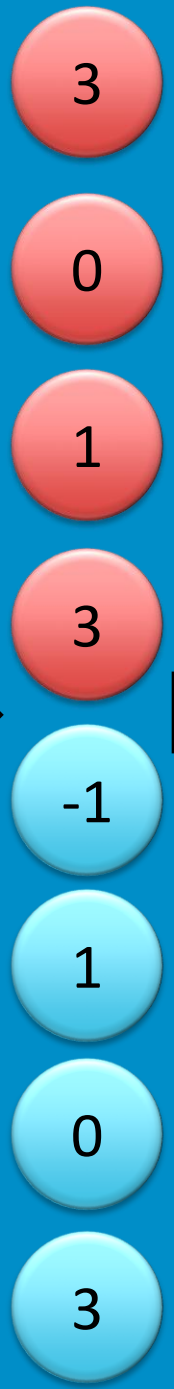


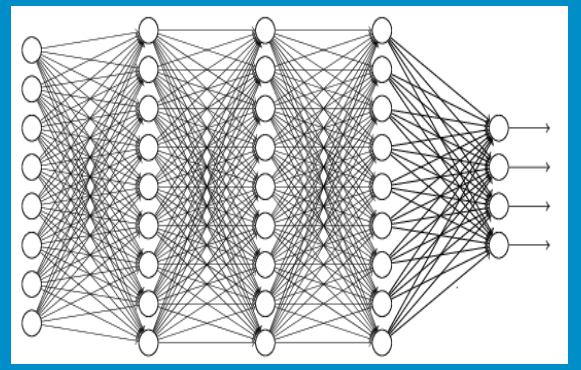
Fully Connected Feedforward network

Flattening



Flattened





Fully Connected
Feedforward network

Fully Connected Layer

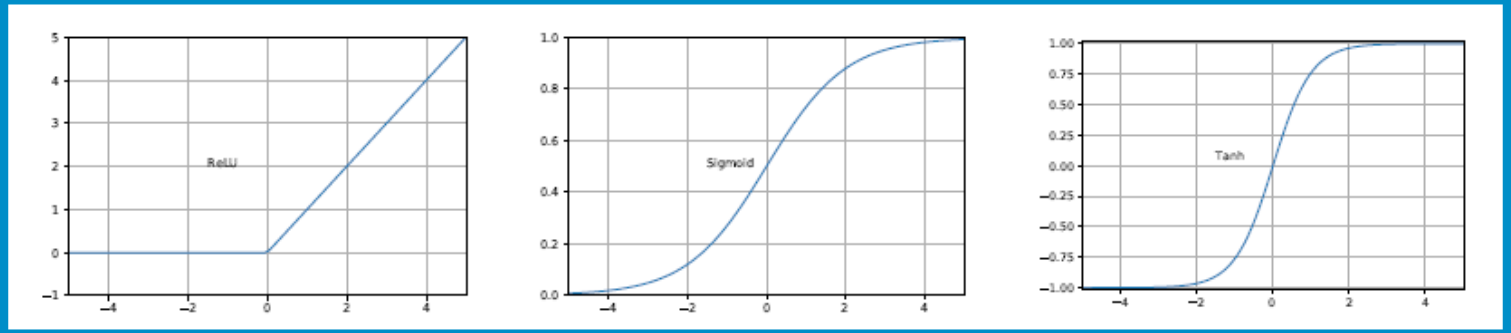
- Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect.

- The FC layer helps to map the representation between the input and the output.

Activation Functions

- Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map.
- There are several types of non-linear operations, the popular ones being:

1. **Sigmoid**
2. **Tanh**
3. **ReLU**



ReLu **Sigmoid** **Tanh**

Summary

- CNN is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image.
- Convolution leverages important ideas that motivated computer vision researchers: sparse interaction, and parameter sharing. Pooling further reduces the complexity
- A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer
- Non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map. Popular ones being sigmoid, Tanh, ReLu.