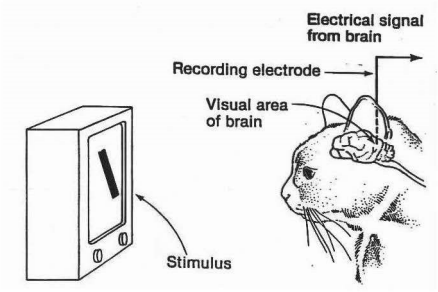
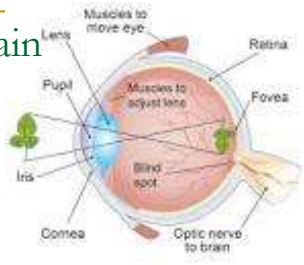
Convolutional Neural Networks

Visual Information Processed in Brain

The Visual Cortex

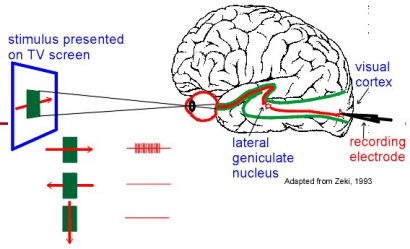




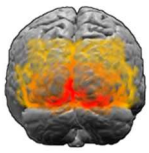
1) Spatial Invariance

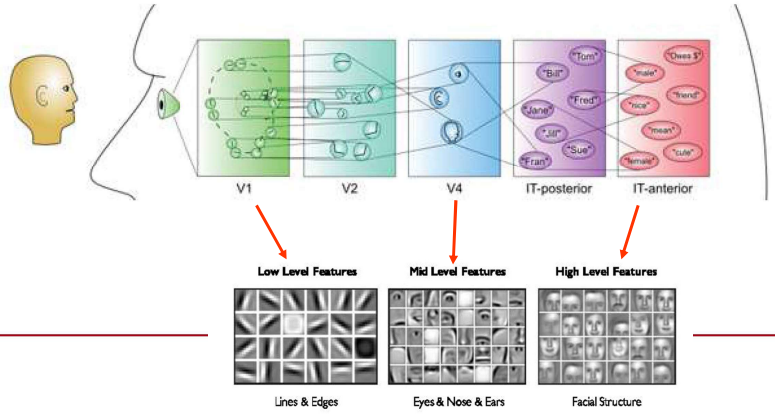
2) Receptive Field

3) Hierarchy



The Visual Cortex

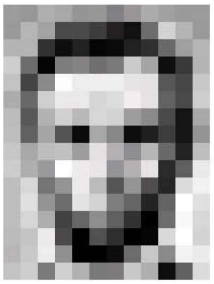


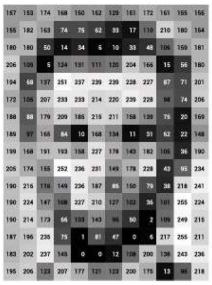


3

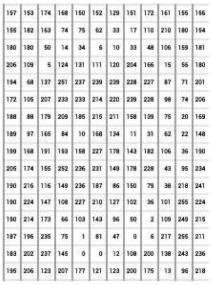
Images are Numbers in Computer

Images are Numbers





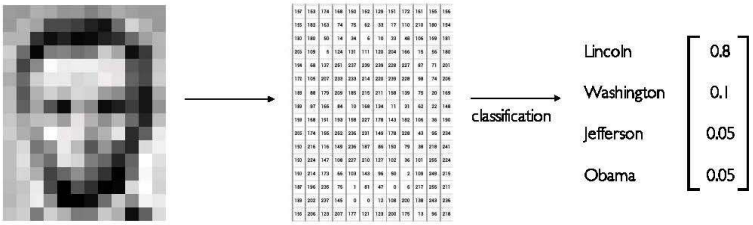
What the computer sees



An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

4

Tasks in Computer Vision



- **Regression**: output variable takes continuous value
- **Classification**: output variable takes class label. Can produce probability of belonging to a particular class

5

High Level Feature Detection

High Level Feature Detection

Let's identify key features in each image category

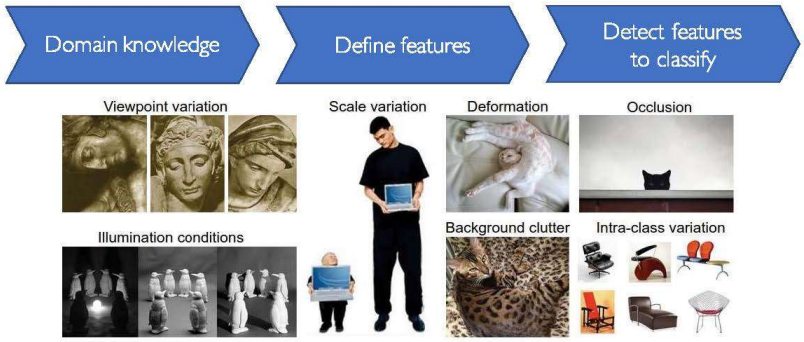
Nose,
Eyes,
Mouth

Wheels,
License Plate,
Headlights

Door,
Windows,
Steps

6

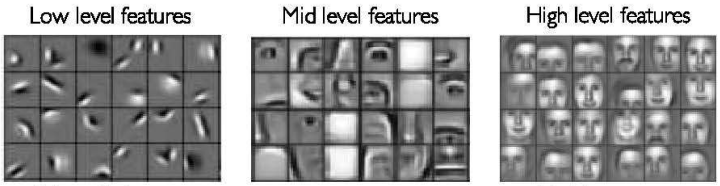
Manual Feature Extraction



7

Learning Feature Representations

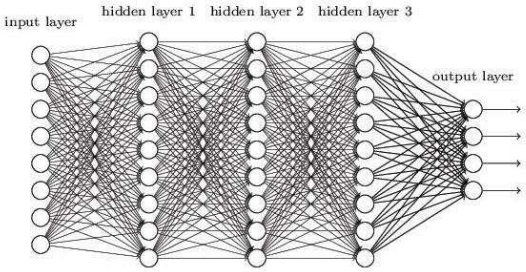
Can we **learn hierarchy of features** directly from the data instead of hand engineering?



8

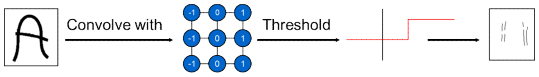
Learning Visual Features

Fully Connected Neural Network

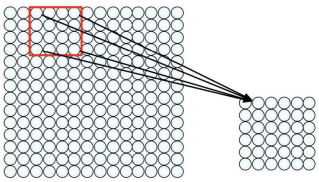


- Connecting neuron in hidden layer to all neurons in input layer
- No spatial information
- Too many parameters

9



Feature Extraction with Convolution



- Filter of size 4x4 : 16 different weights
- Apply this same filter to 4x4 patches in input
- Shift by 2 pixels for next patch

This "patchy" operation is **convolution**

1) Apply a set of weights – a filter – to extract **local features**

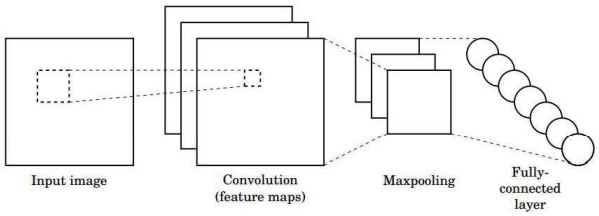
2) Use **multiple filters** to extract different features

3) **Spatially share** parameters of each filter

10

Convolutional Neural Network

CNNs for Classification



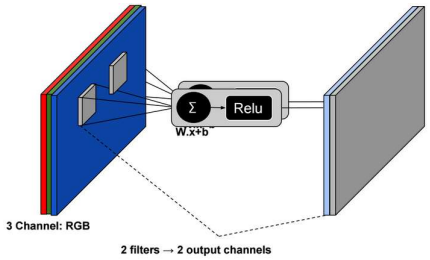
1. **Convolution**: Apply filters with learned weights to generate feature maps.
2. **Non-linearity**: Often ReLU.
3. **Pooling**: Downsampling operation on each feature map.

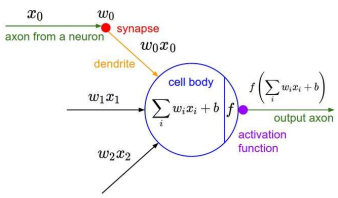
Train model with image data.
Learn weights of filters in convolutional layers.

11

Convolution

A convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as a filter.

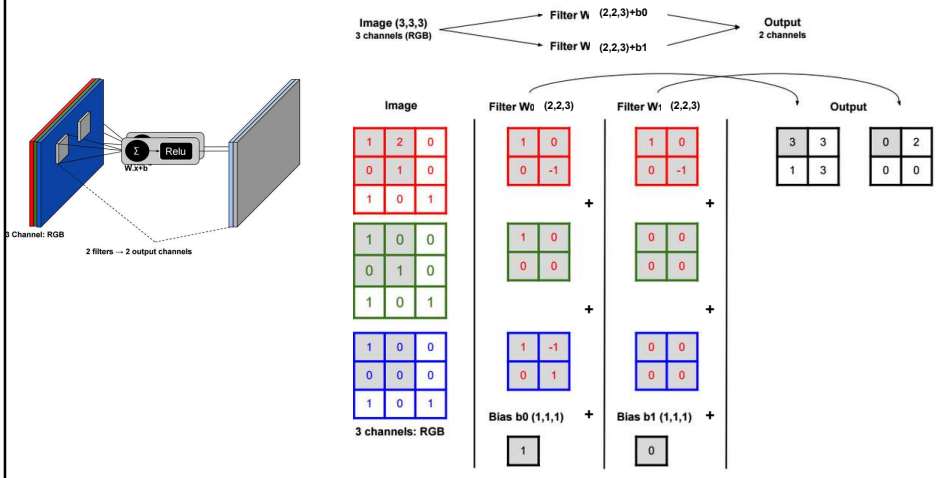




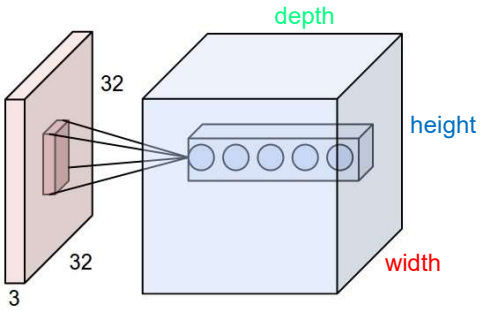
Actually, each filter corresponds to one perceptron. The perceptron is moved around the whole image to extract local features.

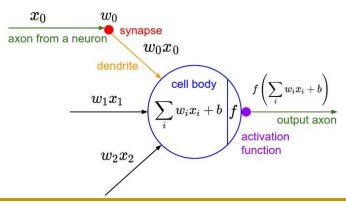
12

Convolution



Spatial Arrangement of Output Volume





Layer Dimensions:

$$h \times w \times d$$

where *h* and *w* are spatial dimensions
d = number of filters

Stride:

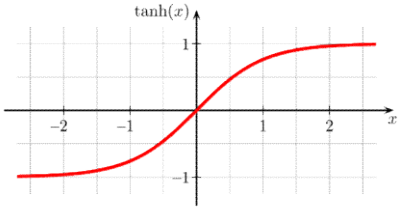
Filter step size

Receptive Field:

Locations in input image that a node is path connected to. Typically, it is equal to filter size e.g. (3x3).

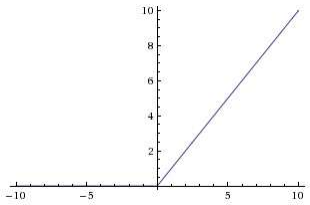
Nonlinearity

Tanh(x)



$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

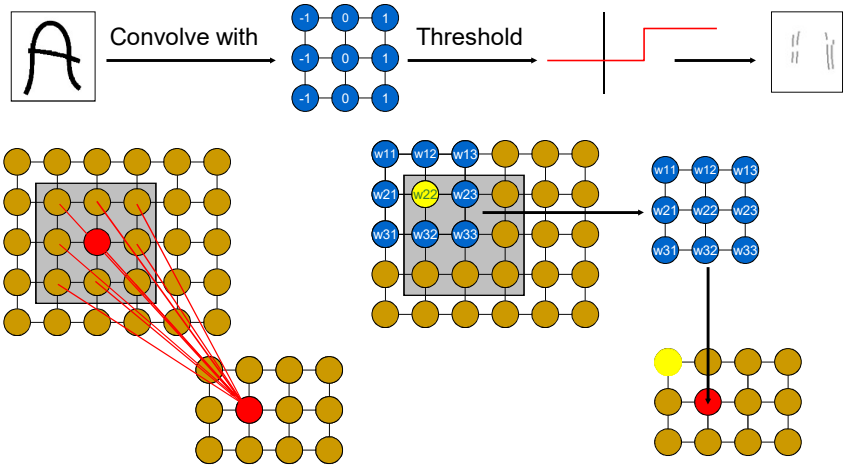
ReLU



$$f(x) = \max(0, x)$$

15

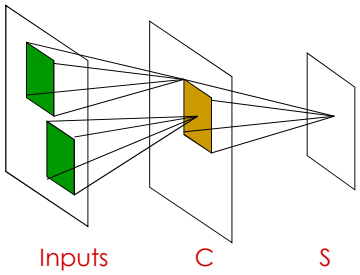
Nonlinearity (Feature extraction)



16

Nonlinearity (feature extraction)

- Shared weights: all neurons in a feature share the same weights (but not the biases).
- In this way all neurons detect the same feature at different positions in the input image.
- Reduce the number of free parameters.

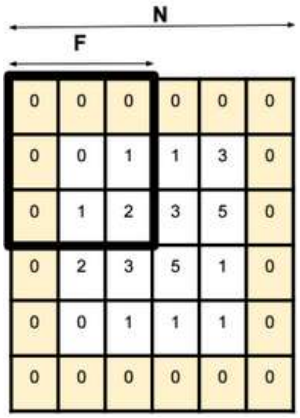


17

Padding

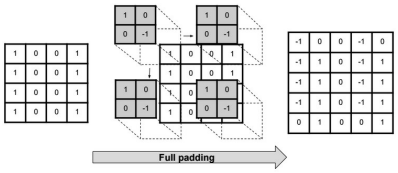
Padding is basically adding rows or columns of zeros to the borders of an image input. It helps control the output size of the convolution layer. The formula to calculate the output size is: **(N – F) / stride + 1**.

For a 32x32x3 image and using 10 5 × 5 filters with stride 1 and pad 2, we get an output with size 32x32x10.

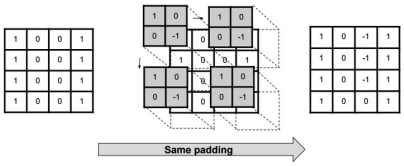


18

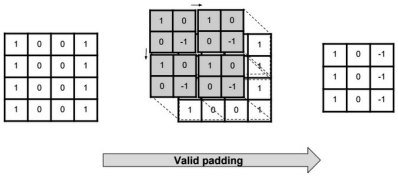
- Full padding



- Same padding (half padding)



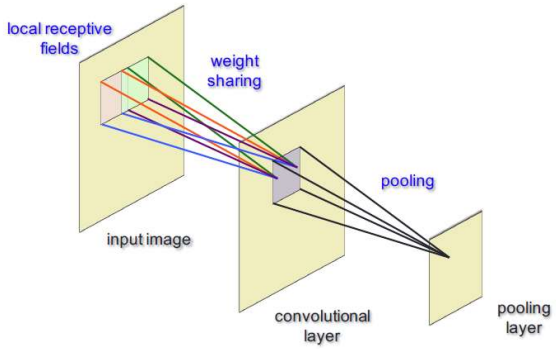
- Valid padding (no padding)



19

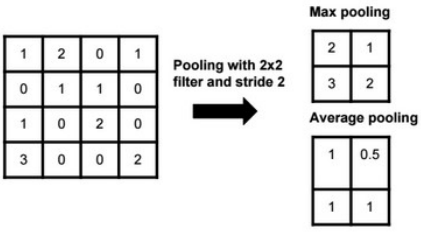
Pooling

- Common pooling operations:
  - Max pooling: reports the maximum output within a rectangular neighborhood.
  - Average pooling: reports the average output of a rectangular neighborhood (possibly weighted by the distance from the central pixel).



20

An Example

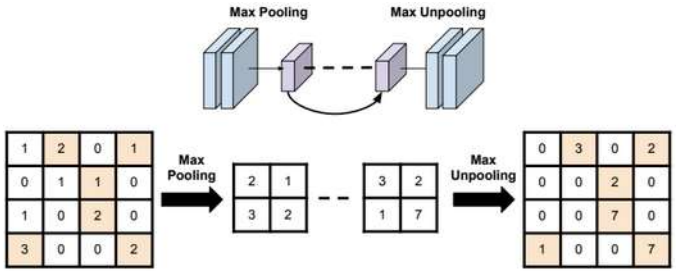


With pooling we reduce the size of the data without changing the depth. Max pooling preserves edge, resulting in spatial invariance.

The output size of a polling operation on a 8x8x10 representation using a 2×2 filter and with stride 2 is 4x4x10 (We can use the same formula: **(N – F) / stride + 1**).

21

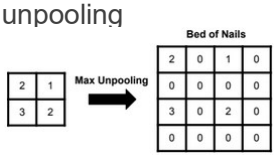
Unpooling



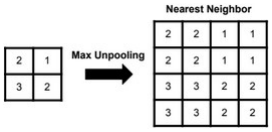
22

Unpooling

Bed of nails unpooling



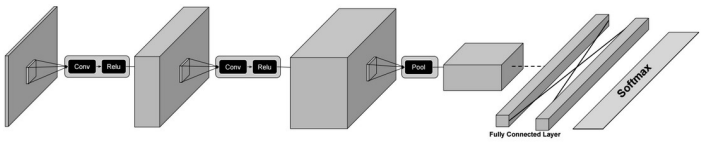
Nearest neighbor unpooling



23

Architecture

In general the architecture of a convolutional neural network is as below:



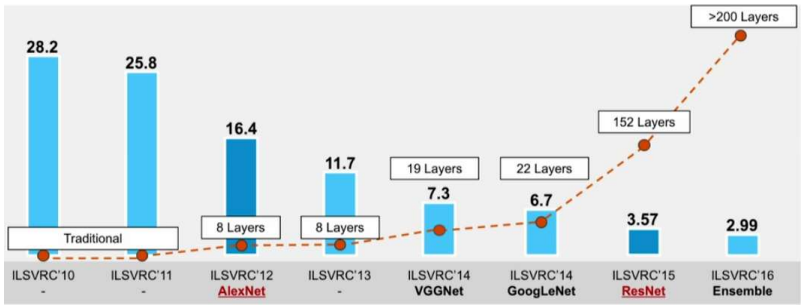
Conv→ Relu → Conv → Relu→ Pool → … → Conv → Fully Connected Layer → Softmax

Some well known CNN architectures are: AlexNet (8 layers), VGG (16-19 layers), GoogLeNet (22 layers) and ResNet (152 layers).

24

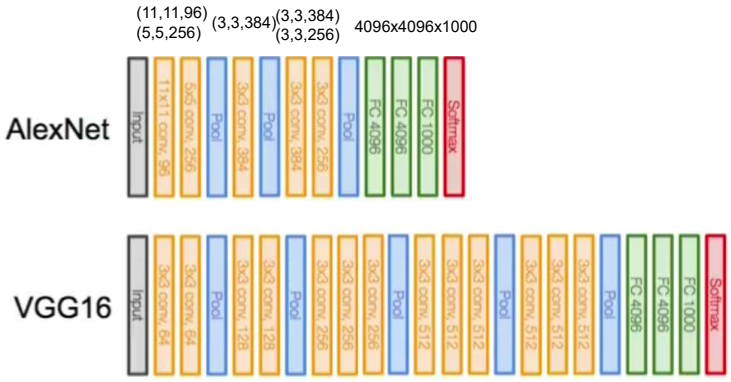
ILSVRC

(ImageNet large scale visual recognition challenge)



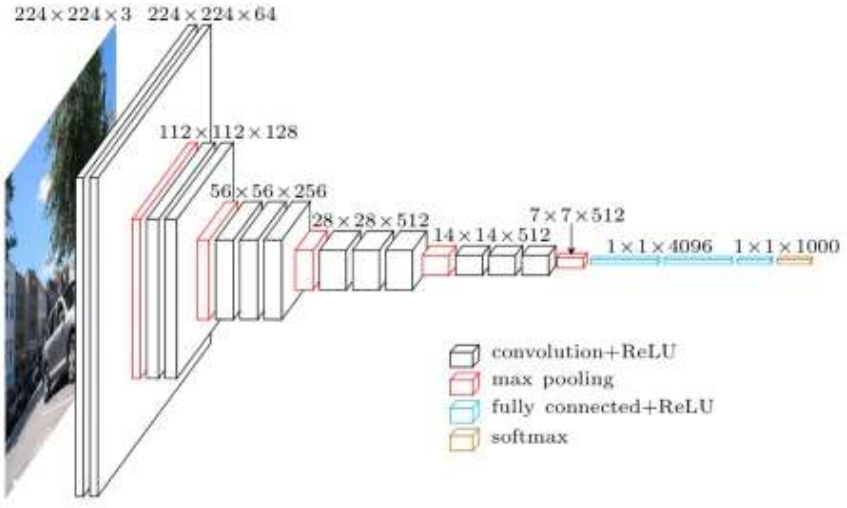
25

Architecture



26

VGG16 architecture

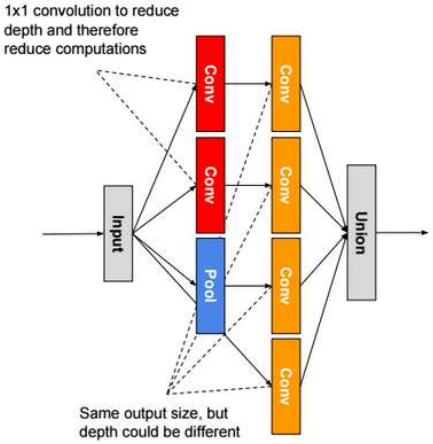


27

Architecture

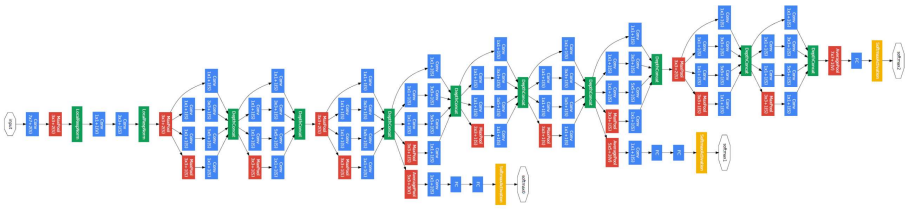
GoogLeNet

For GoogleNet, the architecture is slightly different. It uses Inception modules which combine multiple parallel convolutions.



28

GoogLeNet architecture

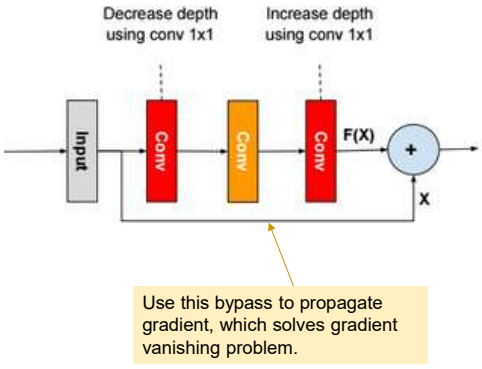


29

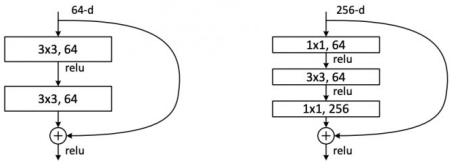
Architecture

ResNet

For ResNet, we use residual blocks. The output of a residual block is the sum of the input X + the output of last convolution layer F(X). If weights are zeros, then the output of a residual block is the input X.

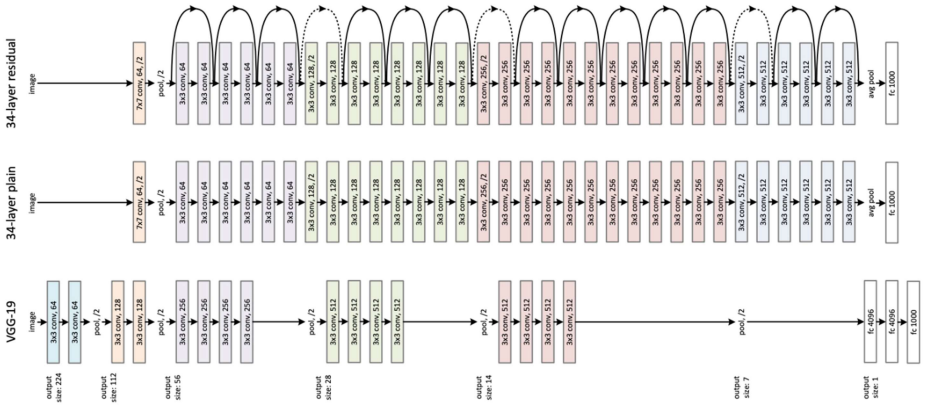


There are other architectures like Network in Network, FractalNet, Densely Connected CNN, SqueezeNet.



30

ResNet34 architecture

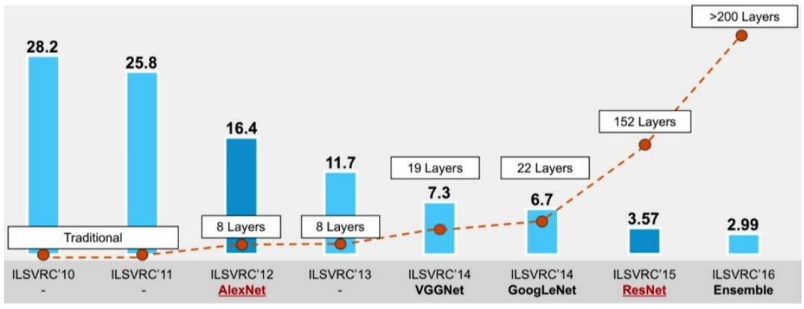


31

Computer Vision Datasets

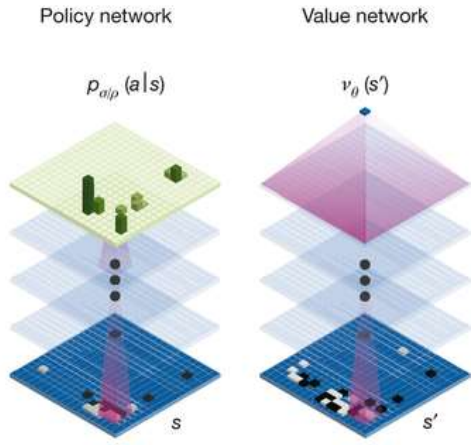
Famous computer Vision datasets are: **MNIST**, **ImageNet**, **CIFAR-10(0)**, **Places**.
Below the error rate for deep neural networks trained on the ImageNet dataset.

ILSVRC (ImageNet large scale visual recognition challenge)



32

Deep CNN in AlphaGO



(Silver et al, 2016)

Policy network:

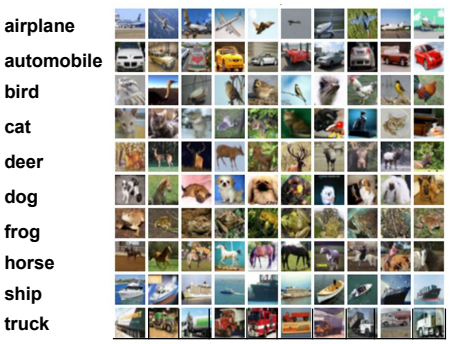
- Input: 19x19, 48 input channels
- Layer 1: 5x5 kernel, 192 filters
- Layer 2 to 12: 3x3 kernel, 192 filters
- Layer 13: 1x1 kernel, 1 filter

Value network has similar architecture to policy network

33

Computer Vision Tasks

Image Tagging

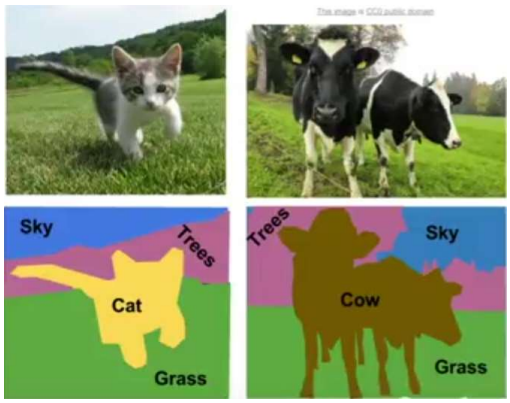


34

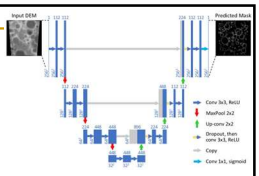
Computer Vision Tasks

Semantic Segmentation

Semantic segmentation is the task of assigning a class-label to each pixel in an image.

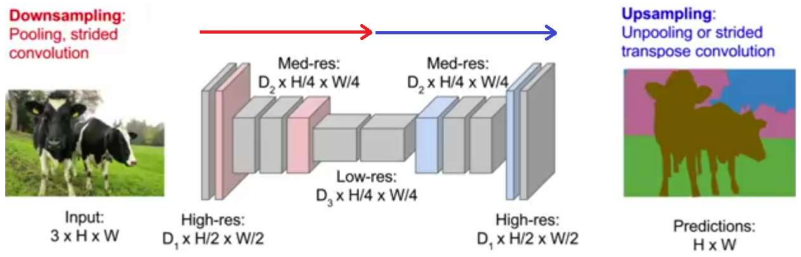


35



Computer Vision Tasks

The general architecture of a CNN for this task is as follow:

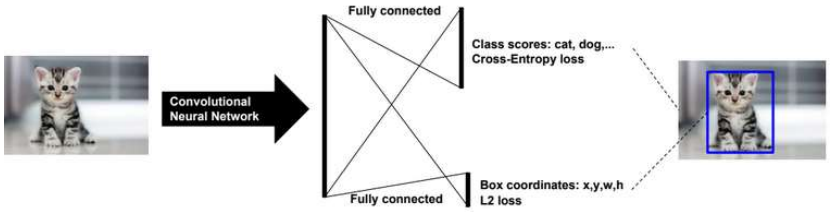


In this task, we minimize the cross-entropy loss over every pixel.

36

Computer Vision Tasks

Classification & Localisation



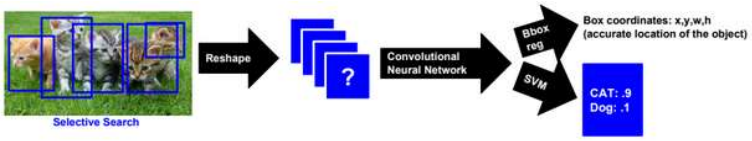
37

Computer Vision Tasks

Object Detection

Sliding window

Region proposals (selective search/R-CNN)



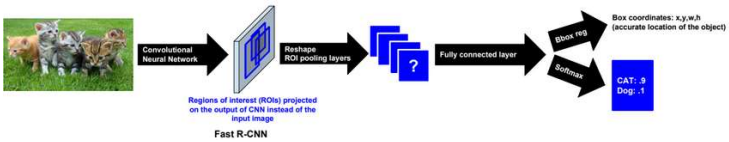
R-CNN

38

Computer Vision Tasks

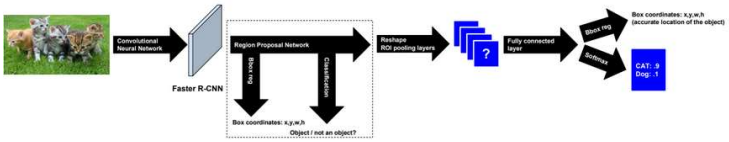
Object Detection

Fast R-CNN



Faster R-CNN

In Faster R-CNN we use and train a Region Proposal network instead of using selective search.

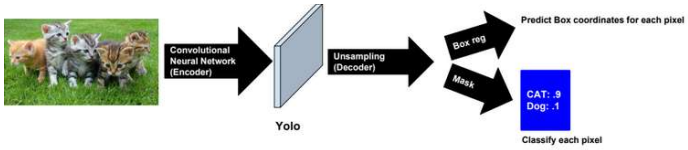


39

Computer Vision Tasks

Object Detection

Yolo



It's recommended to use **Focal loss** function when training the model.
Focal loss is a loss function considered class imbalance defined by

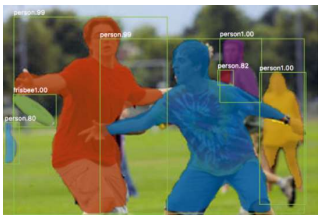
$$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t)$$

40

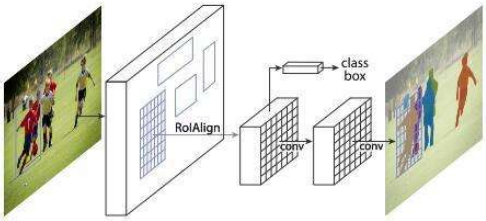
Computer Vision Tasks

Instance Segmentation

Mask R-CNN



Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN.

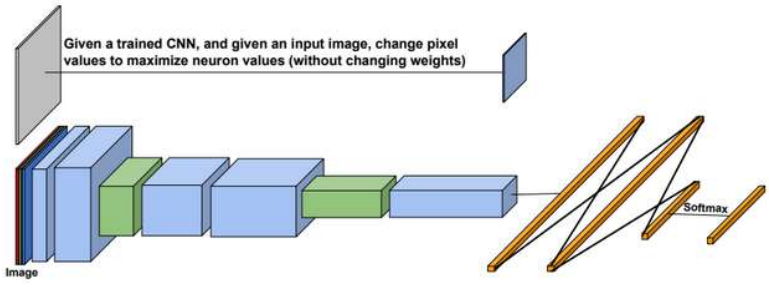


CNN Based Object Detection

41

Inside CNN

To find the pixels in an input image that are important for a classification, we can use the **Gradient Ascent** or **Feature Inversion** algorithms.



42

Adversarial Attacks on Neural Networks

Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(W, x, y)}{\partial x}$$

Fix your weights θ, and true label y

"How does a small change in the input increase our loss"

43