

# Neural networks and image classification

Andrey Stotskiy



3 October 2023

# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Outline

## I. Image classification task

- I.1. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Computer vision (recap)



**Vision task:** understand, what is depicted on an image

**Computer vision:** developing a computer model for vision. CV is a part of Artificial Intelligence (AI)

**Turing test for CV:** answer any question about the image that can be answered by a human

# Outline

## I. Image classification task

### I.I. Classification and related tasks

- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Binary classification



- Does this image contain a pedestrian?
- Binary answer  $y \in \begin{cases} 0, & \text{no} \\ 1, & \text{yes} \end{cases}$
- Alternatively, the estimated probability of the positive answer  $p_{\text{yes}} \in [0; 1]$

# Multiclass classification



- Which object is shown on this image?
- The set of *allowed* object classes is determined in advance!
- Integer answer  $y \in \left\{ \underset{\text{cat}}{1}, \underset{\text{car}}{2}, \dots, \underset{\text{rat}}{S} \right\}$
- Alternatively, a list of estimated probabilities

$$p_i \in [0; 1] \quad i \in 1, \dots, S \quad \sum_{i=1}^S p_i = 1$$

# Attribute recognition



Male  
Asian  
Bearded  
Smiling

- *Attributes* are properties or characteristics that are commonly expressed by some object
- Human attributes may include race, sex, age, color of hair, current emotional state or the presence of wearable accessories such as masks, glasses and hats
- Attribute recognition can often be reduced to one or more classification tasks, for example:
  - *sex* → binary
  - *race* → multiclass
  - *age* → multiclass (over discrete age groups)

# Verification



Are these two images of the same person?

# Identification



- Given a database of images and a single "query" image, determine if the person in the "query" image is present in the database
  - "white" or allow list (face authentication for staff)
  - "black" or deny list (find criminals from surveillance)
- Identification can be reduced to  $N$  pairwise Verification tasks

# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.I. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.I. Neural networks
- 3.2. End-to-end training

# Metrics: Goodhart's law



When a measure becomes a target,  
it ceases to be a good measure.

# Metrics: Accuracy

- Accuracy - percentage of correctly classified samples

Dataset	CNN	Original	BP[23]	CBP[11]	KP	Others
CUB [43]	VGG-16 [38]	73.1*	84.1	84.3	<b>86.2</b>	82.0 84.1
	ResNet-50 [15]	78.4	N/A	81.6	84.7	[18] [16]
Stanford Car [19]	VGG-16	79.8*	91.3	91.2	<b>92.4</b>	<b>92.6</b> 82.7
	ResNet-50	84.7	N/A	88.6	91.1	[18] [14]
Aircraft [27]	VGG-16	74.1*	84.1	84.1	<b>86.9</b>	80.7
	ResNet-50	79.2	N/A	81.6	85.7	[14]
Food-101 [4]	VGG-16	81.2	82.4	82.4	84.2	50.76
	ResNet-50	82.1	N/A	83.2	<b>85.5</b>	[4]

Table 2. Performance comparisons among all baselines, where KP is the proposed kernel pooling method with learned coefficients. Following the standard experimental setup, we use the input size of  $448 \times 448$  for CUB, Stanford Car and Aircraft datasets except the original VGG-16 (marked by an asterisk \*), which requires a fixed input size of  $224 \times 224$ . For Food-101, we use the input size of  $224 \times 224$  for all the baselines.

- Top-K Accuracy (aka Rank) - percentage of sample for which the correct class is within K most likely predicted classes (often K=5)

# Data domains and modalities



Every computer vision algorithm is designed to operate on images sampled from some *statistical population*. This population is described by an empirical distribution over the set of all "valid" (for that algorithm) images.

$$img \sim P(\mathbb{I}) \quad \mathbb{I} \subseteq \mathbb{R}^{H \times W \times C}$$

These algorithms work by exploiting the inherent properties and invariants of the *statistical population* they support.

# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

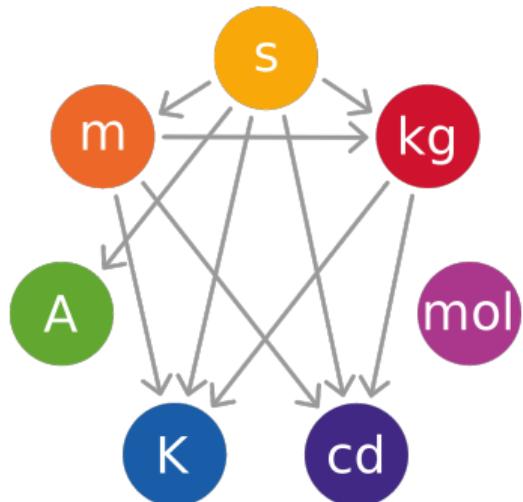
## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Datasets

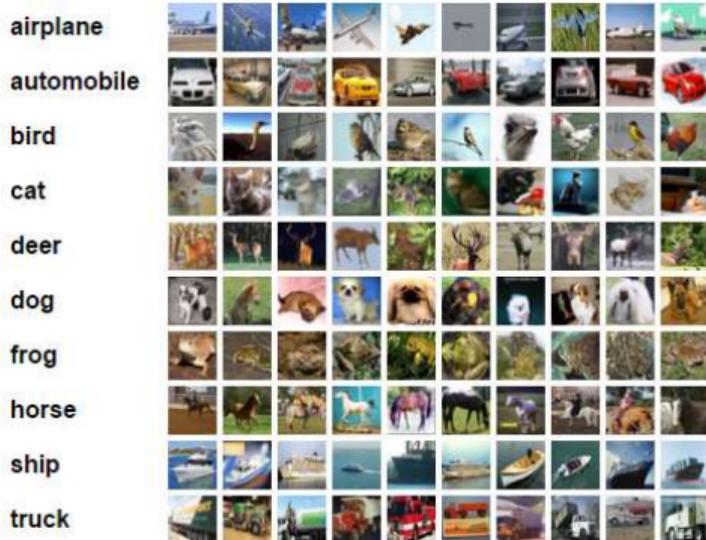


These leaderboards are used to track progress in Classification

Trend	Dataset	Best Model
	InDL	ConvNext
	N-ImageNet	Event Spike Tensor
	N-ImageNet (mini)	Event Image
	ImageNet C-OOD (class-out-of-distribution)	ViT-L/32-384 with Max-logit
	Autoimmune Dataset	Swin Transformer Base (Patch 4 Window 12)
⋮		

[paperswithcode.com/task/classification-1](https://paperswithcode.com/task/classification-1)

# CIFAR-10 and CIFAR-100



Subset of the TinyImages collection

- 60000 images total
- CIFAR-10: 10 classes
  - 5000 training images per class
  - 1000 testing images per class
- CIFAR-100: 100 classes
  - 500 training images per class
  - 100 testing images per class

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

[www.cs.toronto.edu/~kriz/cifar.html](http://www.cs.toronto.edu/~kriz/cifar.html)

# ImageNet

Goal: create a dataset with at least 1000 images for each of the original 117 000 synsets/classes

~14 000 000 images

(~1 000 000 images with bounding box annotations)

~22 000 non-empty classes (~10 000 classes with at least 1000 examples)

primer housing animal weight  
offspring computer drop headquarters egg white  
teacher album down flowers television  
register gallery court key structure light date spread  
king fireplace church press market lighter  
sustaining consumer cup connect side site door pack  
sport screen tree file tower tall camp fish, coffee  
sky plant wall means fan ball lamp  
bread table top man car study bird  
weakening cover cloud leashes net menu ball fish glass  
spring range fruit shop sign  
bed shop to to goal  
kitchen main camera box center step  
engine to memory/sleep cell kid  
dinner stone child case student stand  
apple girl flat  
flag bank home room office club  
radio support level line street golf  
beach library stage video food building  
tool material player machine security call clock  
football hospital much equipment cell phone mountain crowd  
short circuit bridge telephone  
gas pedal microphone recording

The screenshot shows the ImageNet homepage. At the top, there's a search bar with the placeholder "14,197,122 images, 21,611 synsets indexed" and a green "SEARCH" button. To the right are links for "Home", "About", "Explore", and "Download". Below the search bar, it says "Not logged in | Login | Signup". The main content area has a heading "Start exploring here" and three sections of images and definitions:

- Synset: people** has bounding box  
Definition: (plural) any group of human beings (men or women or children) collectively; "old people"; "there were at least 200 people in the audience".
- Synset: homo, man, human being, human** has bounding box  
Definition: any living or extinct member of the family Hominidae characterized by superior intelligence, articulate speech, and erect carriage.
- Synset: child, kid** has bounding box  
Definition: a human offspring (son or daughter) of any age; "they had three children"; "they were able to send their kids to college".

# ImageNet: Annotation problems



mite

container ship

motor scooter

leopard

mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat



grille

agaric

cherry

Madagascar cat

convertible	mushroom	dalmatian	squirrel monkey
grille	mushroom	grape	spider monkey
pickup	jelly fungus	elderberry	titi
beach wagon	gill fungus	ffordshire bulterrier	indri
fire engine	dead-man's-fingers	currant	howler monkey

# OpenImages



Goal: create the largest **open** dataset of real-life photographs with diverse annotations

- ~9 000 000 images  
**licensed under CC BY 2.0**
- ~60 000 000 annotations for  
~20 000 categories
- Various supplementary annotations are also available  
(for example, localized text descriptions)

[storage.googleapis.com/openimages/web/index.html](https://storage.googleapis.com/openimages/web/index.html)

# Fine-grained classification

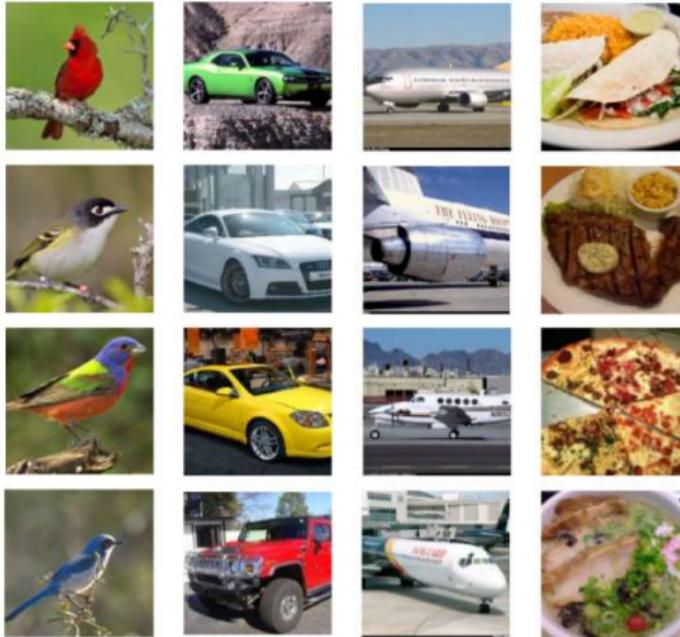


Figure 7. Images we used for visual recognition. From left to right, each column contains examples from CUB Bird [43], Stanford Car [19], Aircraft [27] and Food-101 [4].

# Outline

## I. Image classification task

- I.1. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

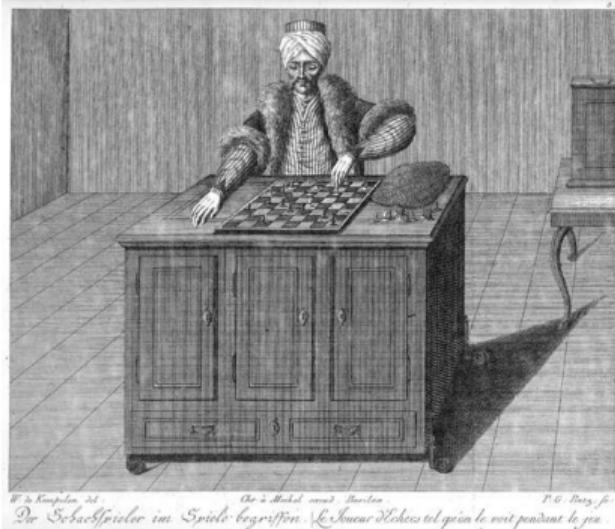
## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Mechanical Turk



"Mechanical Turk, Automaton Chess Player" was a robot created **in 1770** that could play chess (and even beat competent players). In 1820 it was revealed that the robot couldn't actually play chess by itself and that it was instead **controlled by a human sitting in a hidden compartment**.

# Galaxy Zoo



GALAXY ZOO [galaxyzoo.org](http://galaxyzoo.org)

- Classification of galaxy images
- The first large scale project of this kind
- More than 150 000 volunteers created over 60 000 000 annotations in a single year **for free (!)**

# XKCD: 1897

TO COMPLETE YOUR REGISTRATION, PLEASE TELL US  
WHETHER OR NOT THIS IMAGE CONTAINS A STOP SIGN:



NO

YES

ANSWER QUICKLY—OUR SELF-DRIVING  
CAR IS ALMOST AT THE INTERSECTION.

SO MUCH OF "AI" IS JUST FIGURING OUT WAYS  
TO OFFLOAD WORK ONTO RANDOM STRANGERS.

# Annotation as a service

**amazonmechanical turk** Artificial Artificial Intelligence

Your Account    HITs    Qualifications

Xiaodan Zhou | Account Settings | Sign Out | Help

Introduction | Dashboard | Status | Account Settings

**Mechanical Turk is a marketplace for work.**  
We give businesses and developers access to an on-demand, scalable workforce.  
Workers select from thousands of tasks and work whenever it's convenient.

**264,053 HITs** available. [View them now.](#)

### Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. [Find HITs now.](#)

**As a Mechanical Turk Worker you:**

- Can work from home
- Choose your own work hours
- Get paid for doing good work

[Find HITs Now](#)

or learn more about being a [Worker](#)

### Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. [Register Now](#)

**As a Mechanical Turk Requester you:**

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

[Get Started](#)

FAQ | Contact Us | Careers at Amazon | Developers | Press | Policies | Blog  
©2005-2012 Amazon.com, Inc. or its Affiliates

An company

# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

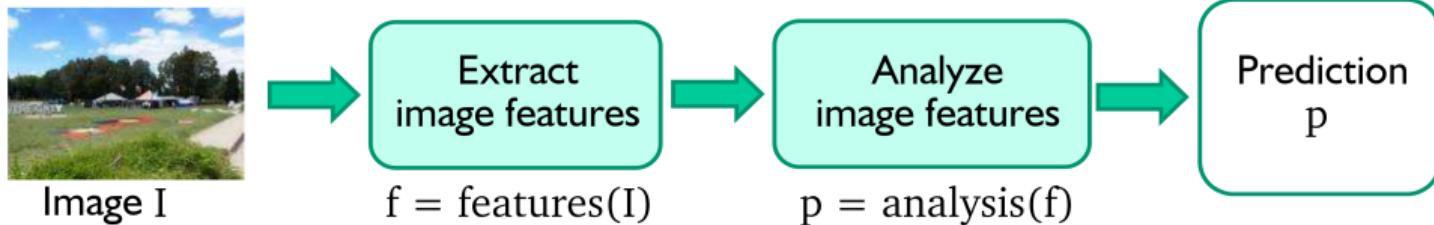
## 2. Classical computer vision methods (overview)

- 2.I. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.I. Neural networks
- 3.2. End-to-end training

# Classical method pipeline



# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

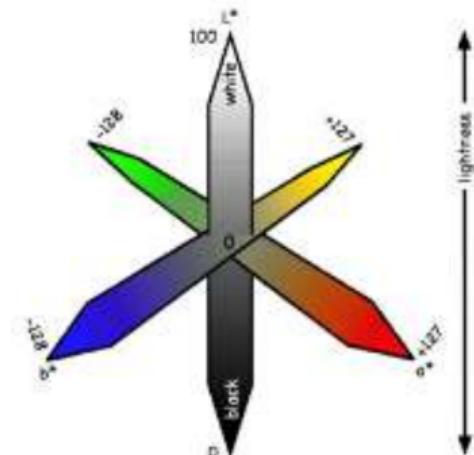
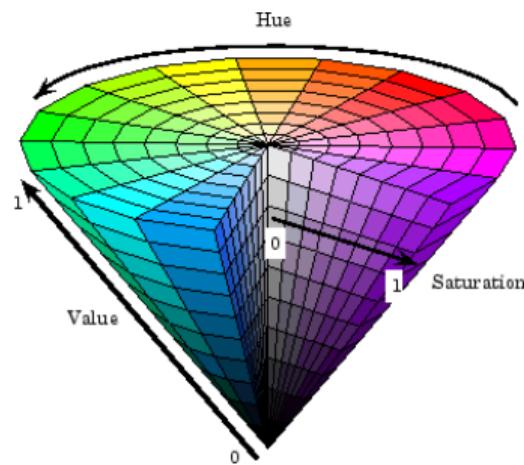
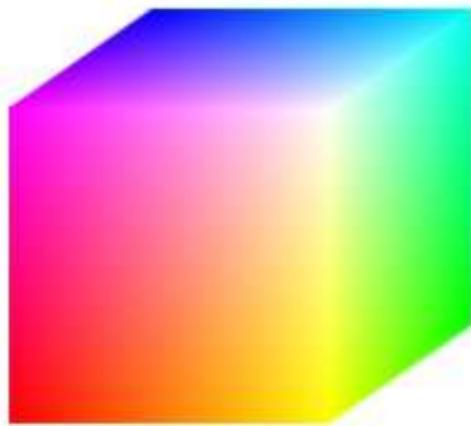
## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

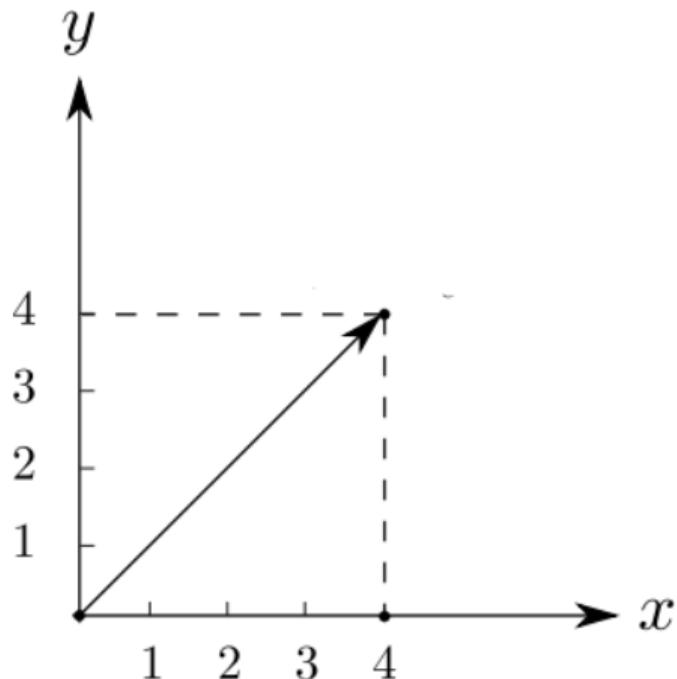
## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

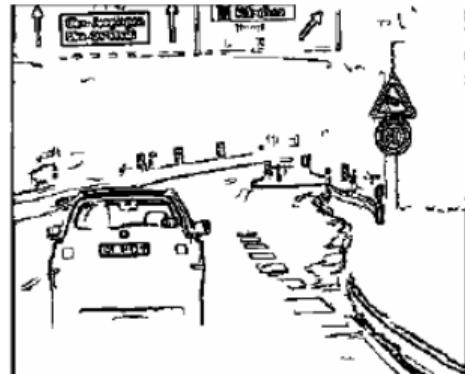
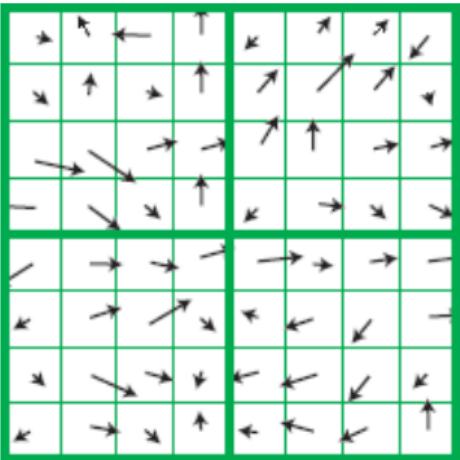
# Features: color



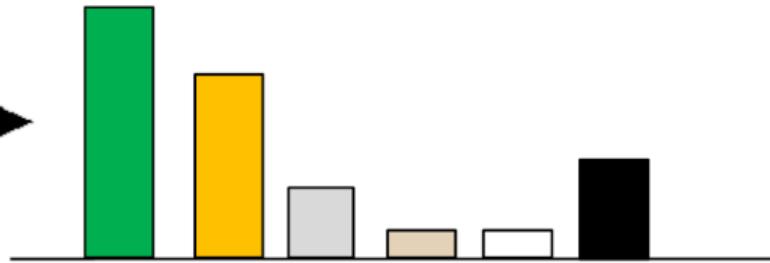
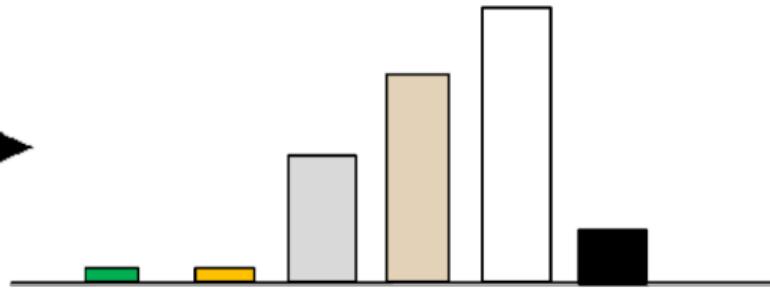
# Features: position



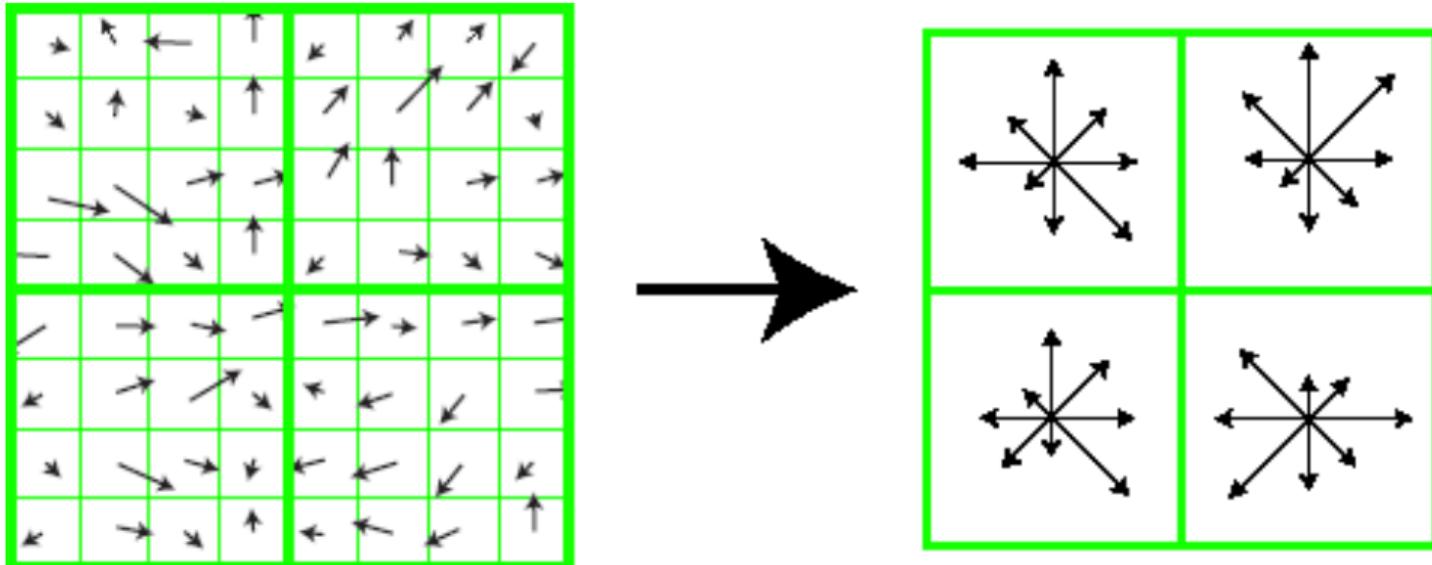
# Features: gradients and edges



# Features: statistics and histograms



# Features: statistics and histograms

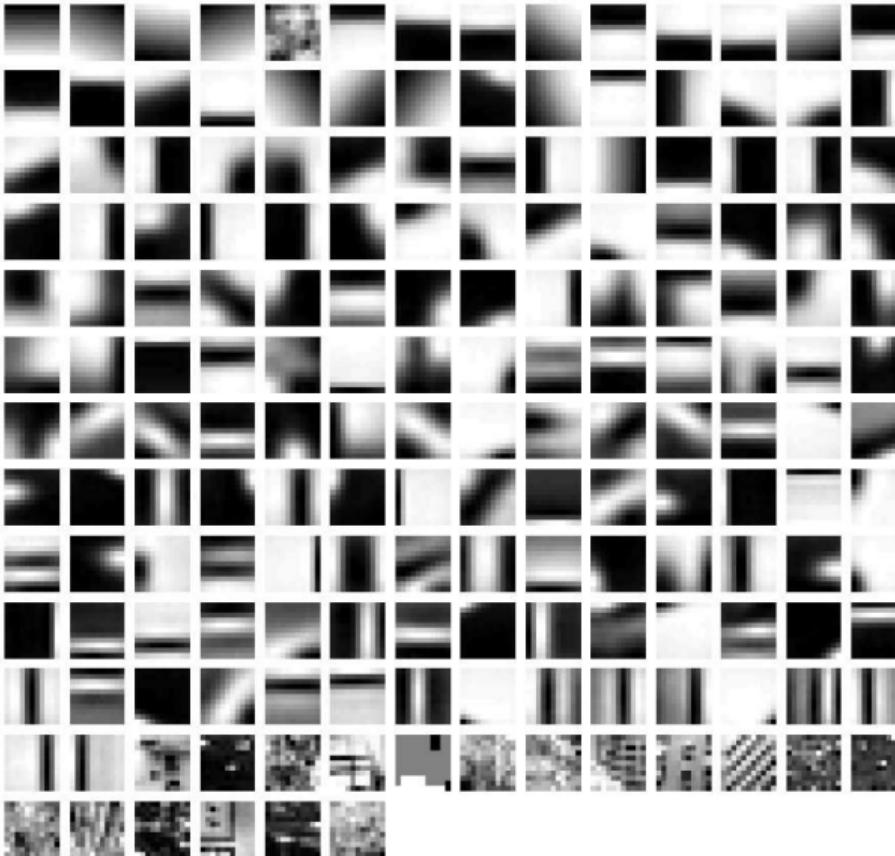


**HoG:** Histogram of oriented Gradients

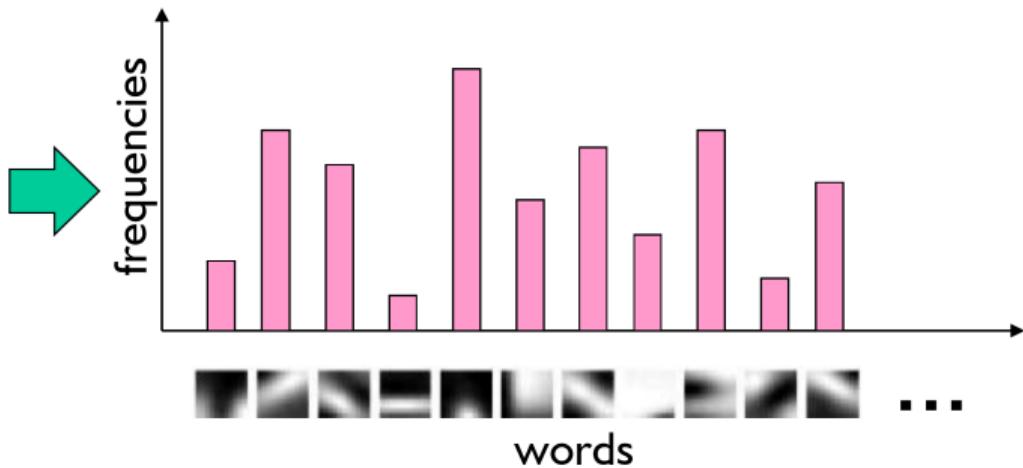
# Features: patterns and bags of words



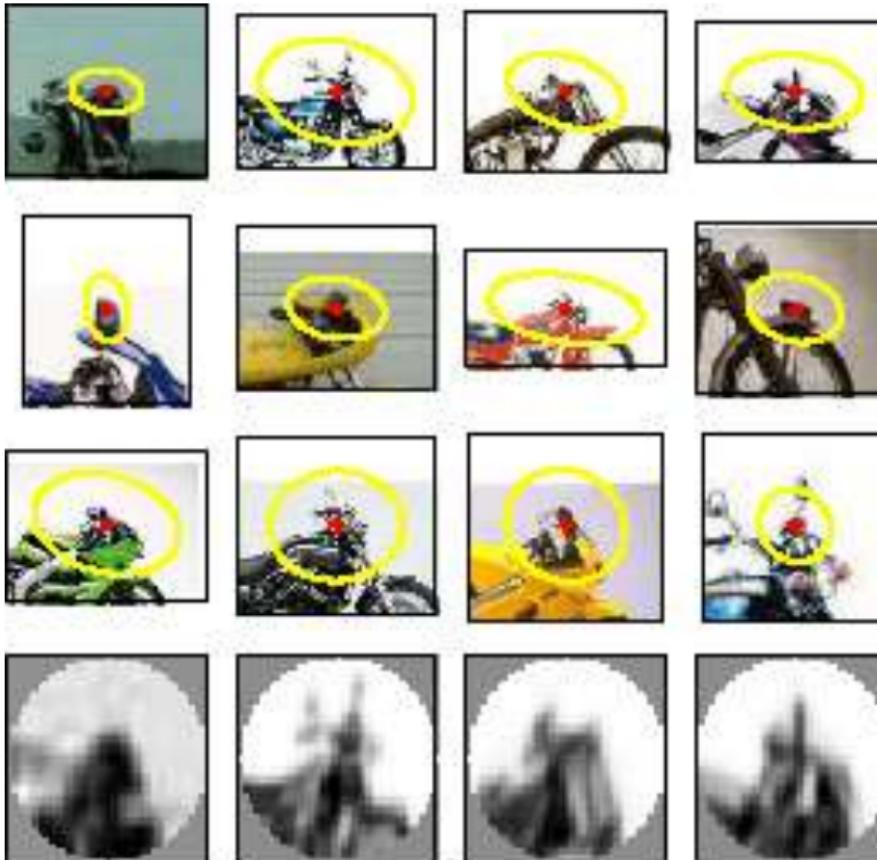
# Features: patterns and bags of words



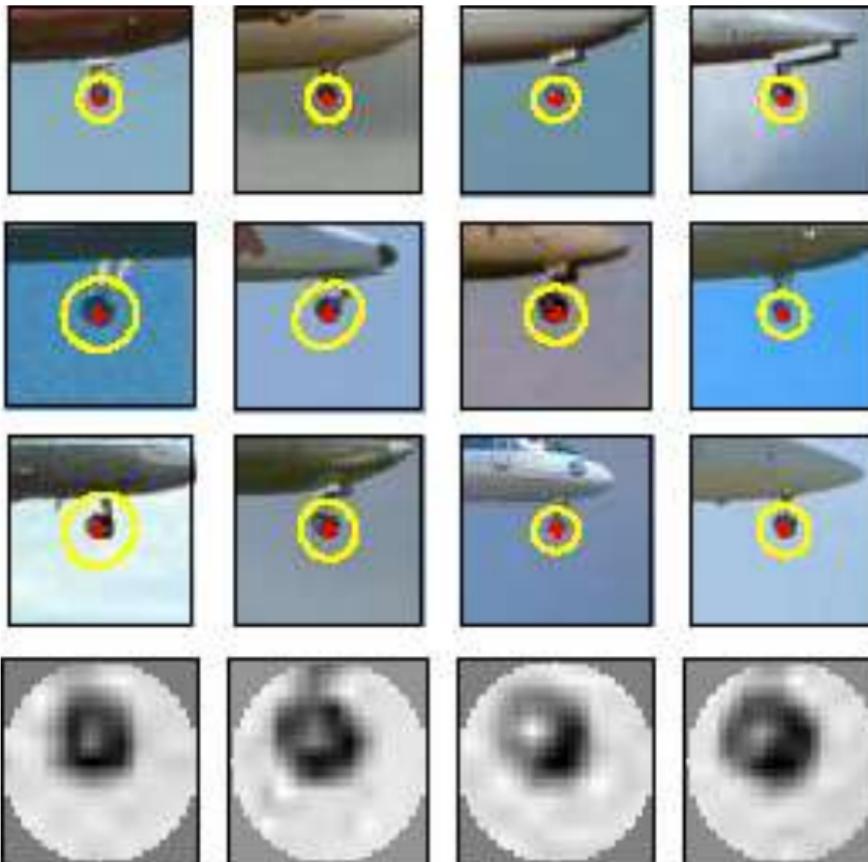
# Features: patterns and bags of words



# Features: patterns and bags of words



# Features: patterns and bags of words



# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

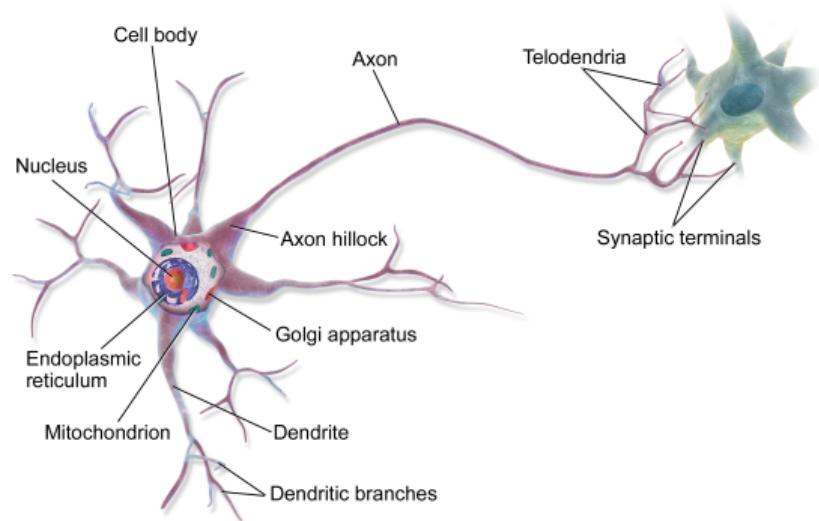
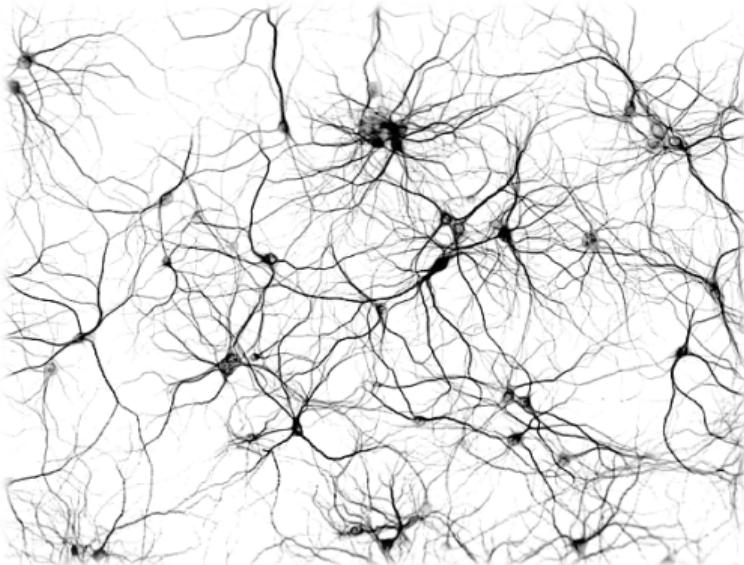
## 2. Classical computer vision methods (overview)

- 2.I. Hand-crafted features
- 2.2. Classical machine learning models

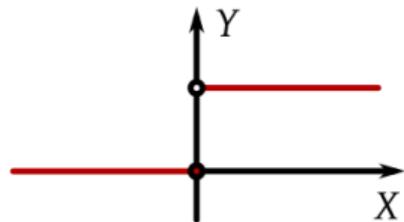
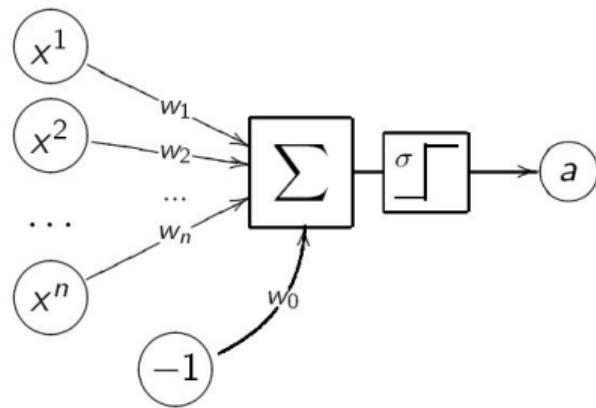
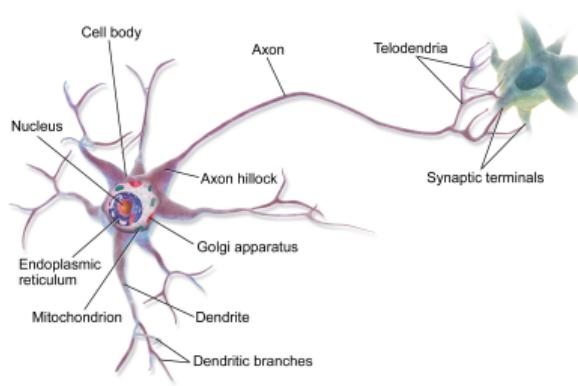
## 3. Modern computer vision methods (overview)

- 3.I. Neural networks
- 3.2. End-to-end training

# Biological neurons



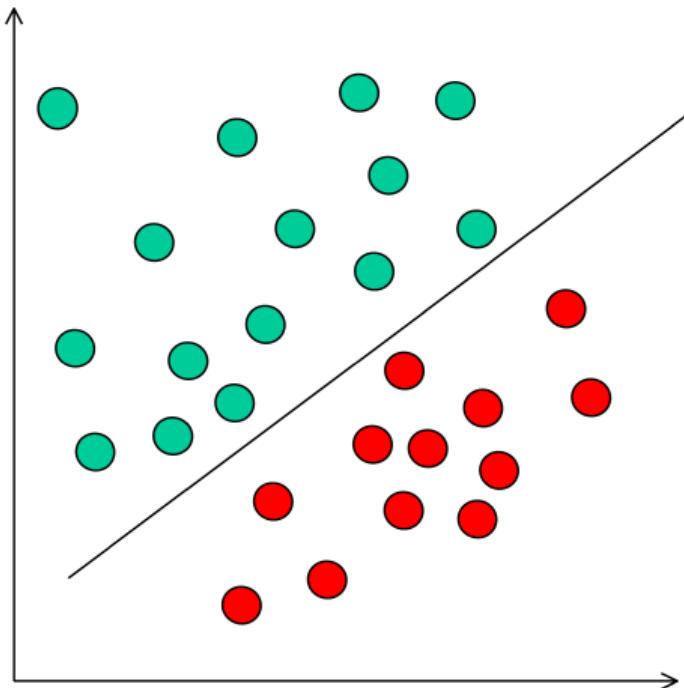
# McCulloch-Pitts neuron model



$$a(x, w) = \sigma \left( \sum_{i=1}^n w_i x_i - w_0 \right)$$

Warren S. McCulloch; Walter Pitts (1943)

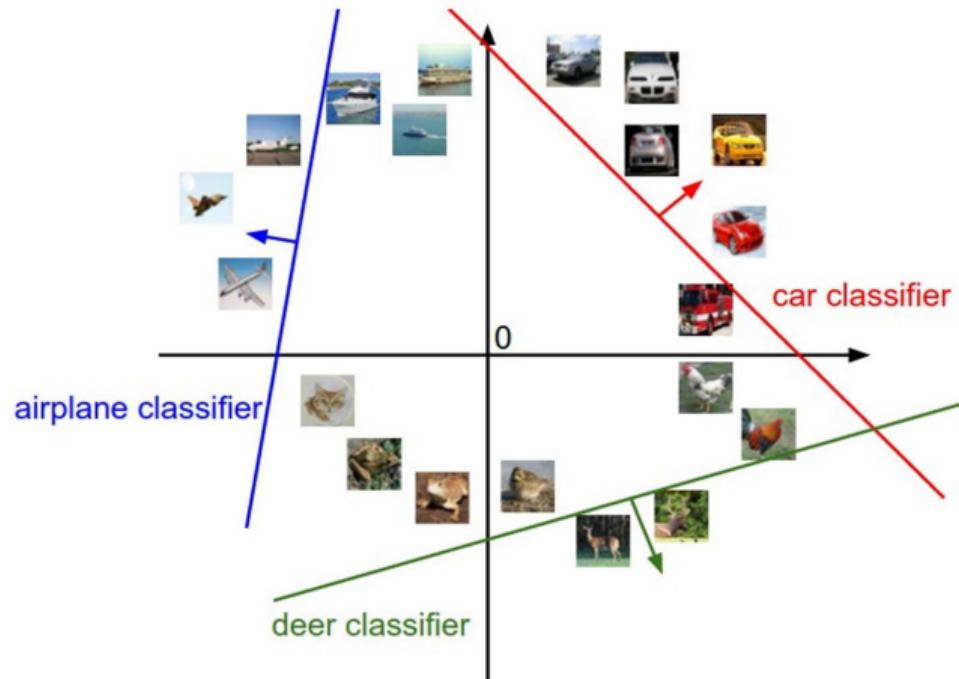
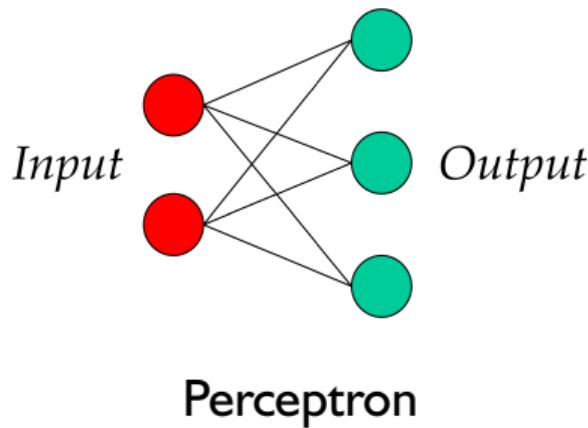
# Neurons as linear classifiers



$$\sigma \left( \sum_{i=1}^n w_i x_i - w_0 \right)$$

Optimal parameters  $w_i$  can be found using classical iterative methods.

# Multiple classes



# Outline

## I. Image classification task

- I.1. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# Outline

## I. Image classification task

- I.1. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

## 2. Classical computer vision methods (overview)

- 2.1. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.1. Neural networks
- 3.2. End-to-end training

# XKCD: 2173

OH, HEY, YOU ORGANIZED  
OUR PHOTO ARCHIVE!

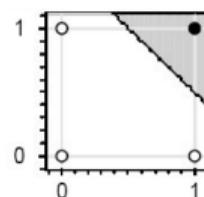
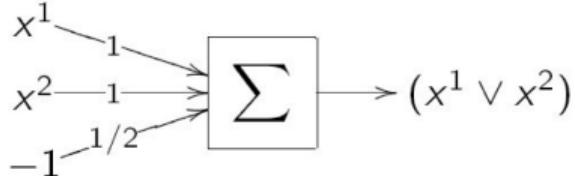
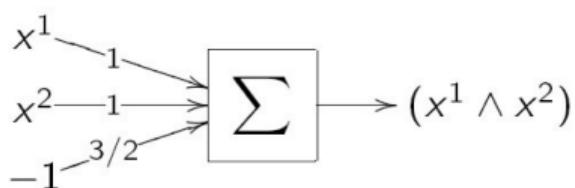
YEAH, I TRAINED A NEURAL  
NET TO SORT THE UNLABELED  
PHOTOS INTO CATEGORIES.

WHOA! NICE WORK!

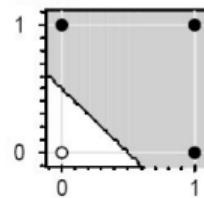


ENGINEERING TIP:  
WHEN YOU DO A TASK BY HAND,  
YOU CAN TECHNICALLY SAY YOU  
TRAINED A NEURAL NET TO DO IT.

# Perceptron: representable functions



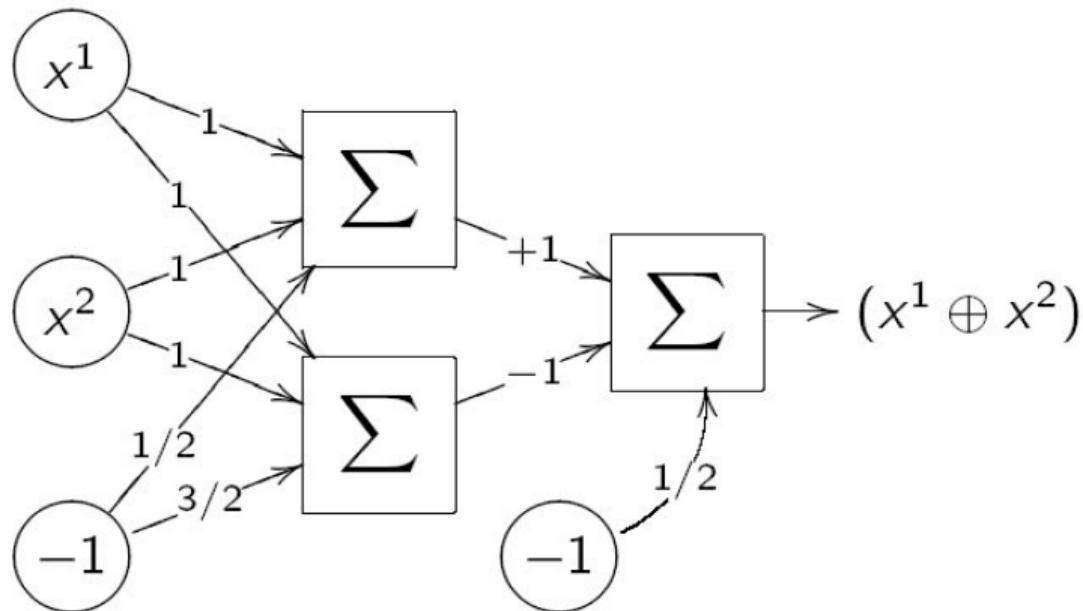
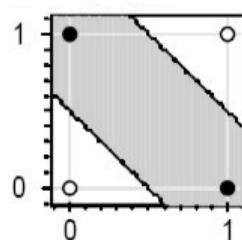
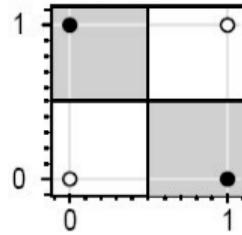
$$x^1 \wedge x^2 = \sigma\left(x^1 + x^2 - \frac{3}{2}\right)$$



$$x^1 \vee x^2 = \sigma\left(x^1 + x^2 - \frac{1}{2}\right)$$

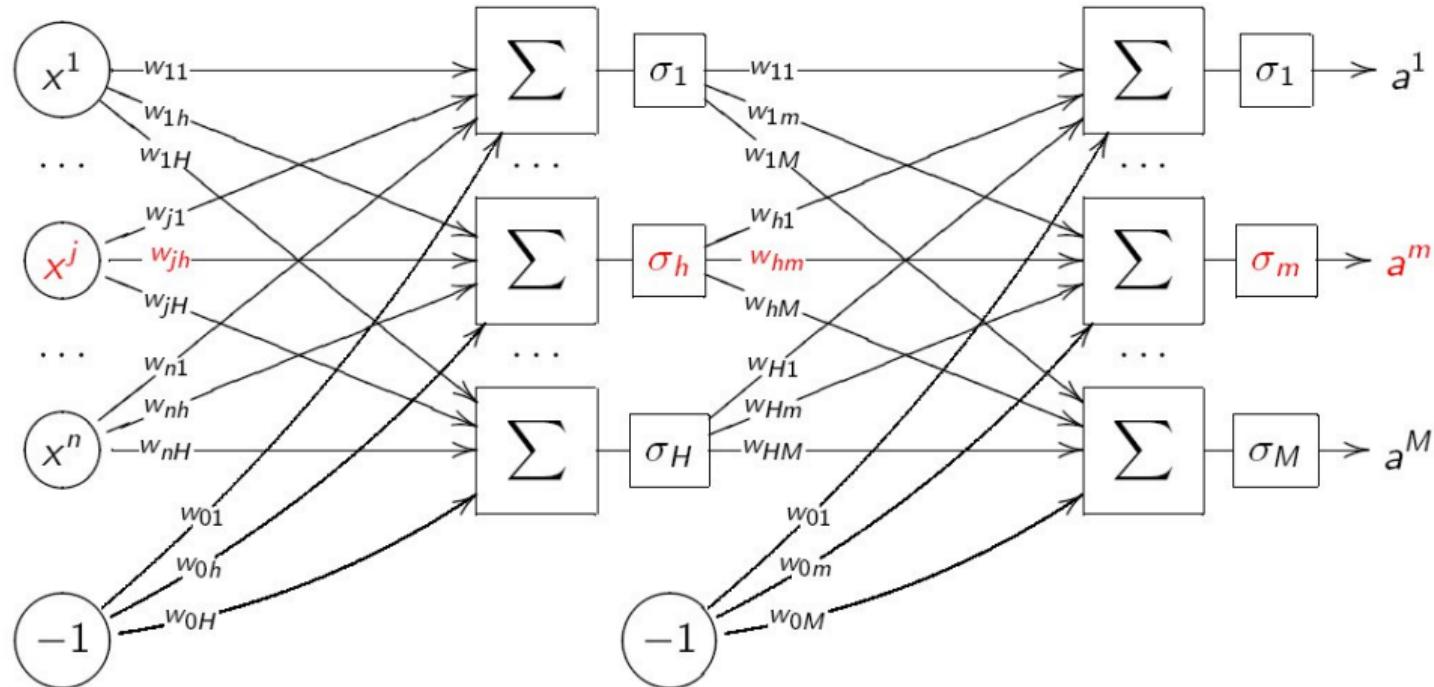
$$\neg x = \sigma\left(-x + \frac{1}{2}\right)$$

# Perceptron: representable functions

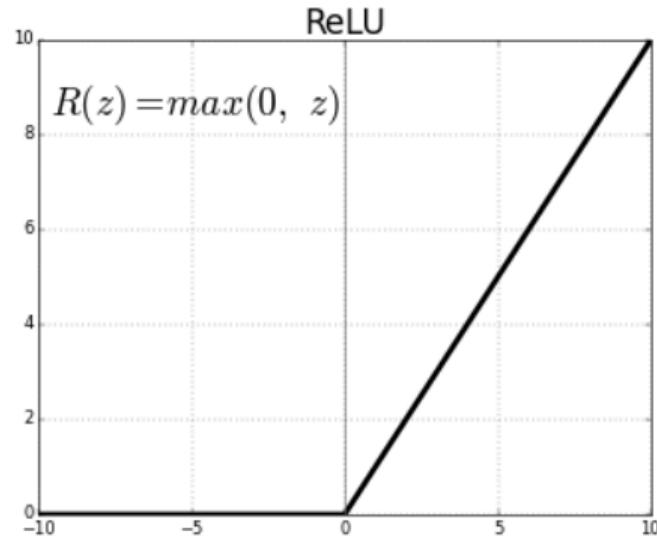
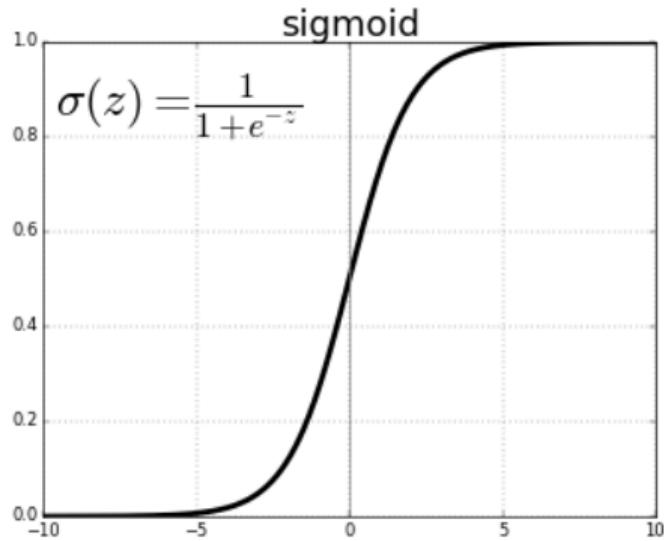


$x^1 \oplus x^2$  is **not** representable by a single perceptron.  
but it **is** representable by two!

# Multilayer perceptron



# Activation functions



# Universal approximation theorems

- Kolmogorov–Arnold representation theorem (1957)
- George Cybenko - arbitrary width, sigmoid activation function (1989)
- Kurt Hornik - arbitrary width, almost any activation function (1991)
- Gustaf Gripenberg - arbitrary depth, relu activation function (2003)
- Patrick Kidger - arbitrary depth, almost any activation function (2020)

# Outline

## I. Image classification task

- I.I. Classification and related tasks
- I.2. Method comparison and evaluation
- I.3. Datasets and benchmarks
- I.4. Labels and annotations

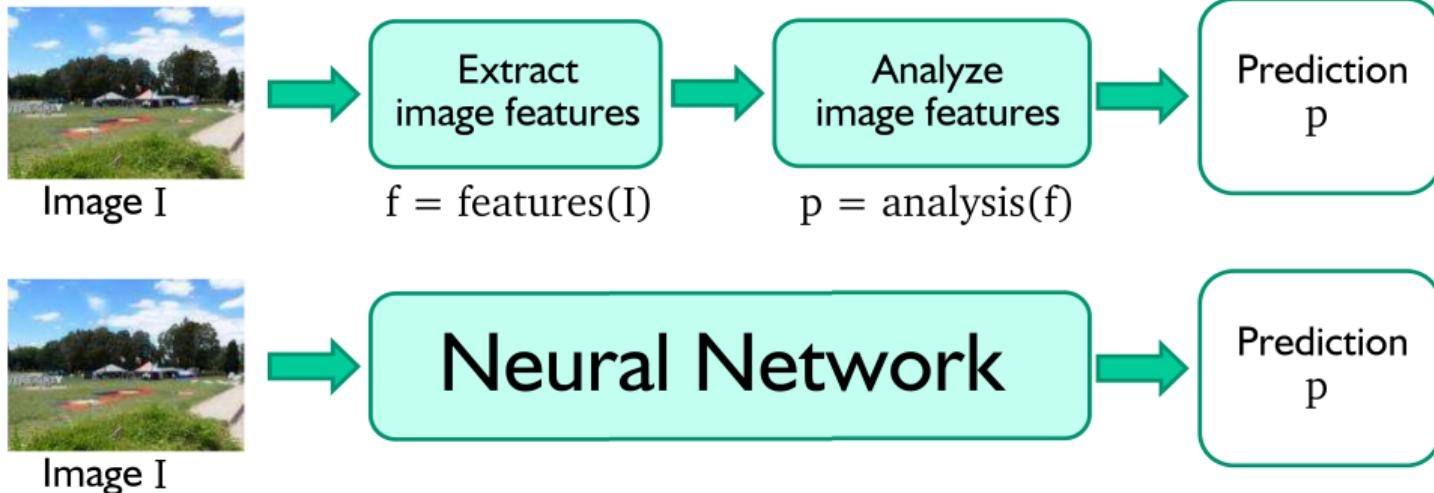
## 2. Classical computer vision methods (overview)

- 2.I. Hand-crafted features
- 2.2. Classical machine learning models

## 3. Modern computer vision methods (overview)

- 3.I. Neural networks
- 3.2. End-to-end training

# Modern computer vision pipeline



# XKCD: 1838



# Loss function

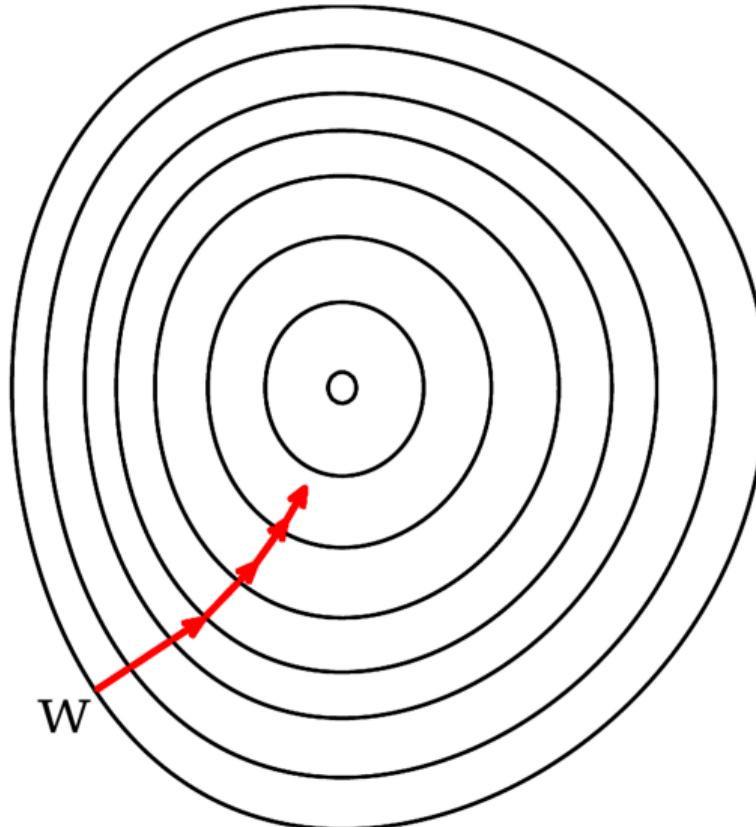
$$p_i^{\text{pr}} = \frac{e^{y_i}}{\sum_{j=1}^N e^{y_j}}$$

Softmax activation function for the last layer

$$L(p^{\text{pr}}, p^{\text{gt}}) = - \sum_{i=1}^N p_i^{\text{gt}} \cdot \log(p_i^{\text{pr}})$$

Categorical cross-entropy

# Gradient descent



# Minibatch stochastic gradient descent

---

## Algorithm 2 Minibatch Stochastic Gradient Descent Training

---

```
1: Input: Function  $f(\mathbf{x}; \theta)$  parameterized with parameters  $\theta$ .  
2: Input: Training set of inputs  $\mathbf{x}_1, \dots, \mathbf{x}_n$  and outputs  $y_1, \dots, y_n$ .  
3: Input: Loss function  $L$ .  
4: while stopping criteria not met do  
5:   Sample a minibatch of  $m$  examples  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$   
6:    $\hat{\mathbf{g}} \leftarrow 0$   
7:   for  $i = 1$  to  $m$  do  
8:     Compute the loss  $L(f(\mathbf{x}_i; \theta), y_i)$   
9:      $\hat{\mathbf{g}} \leftarrow \hat{\mathbf{g}} + \text{gradients of } \frac{1}{m}L(f(\mathbf{x}_i; \theta), y_i) \text{ w.r.t } \theta$   
10:     $\theta \leftarrow \theta + \eta_k \hat{\mathbf{g}}$   
11: return  $\theta$ 
```

---

# Backwards gradient propagation

Will be discussed in more detail during the seminar