Università degli Studi di Messina

Data Science

Big Data Acquisition

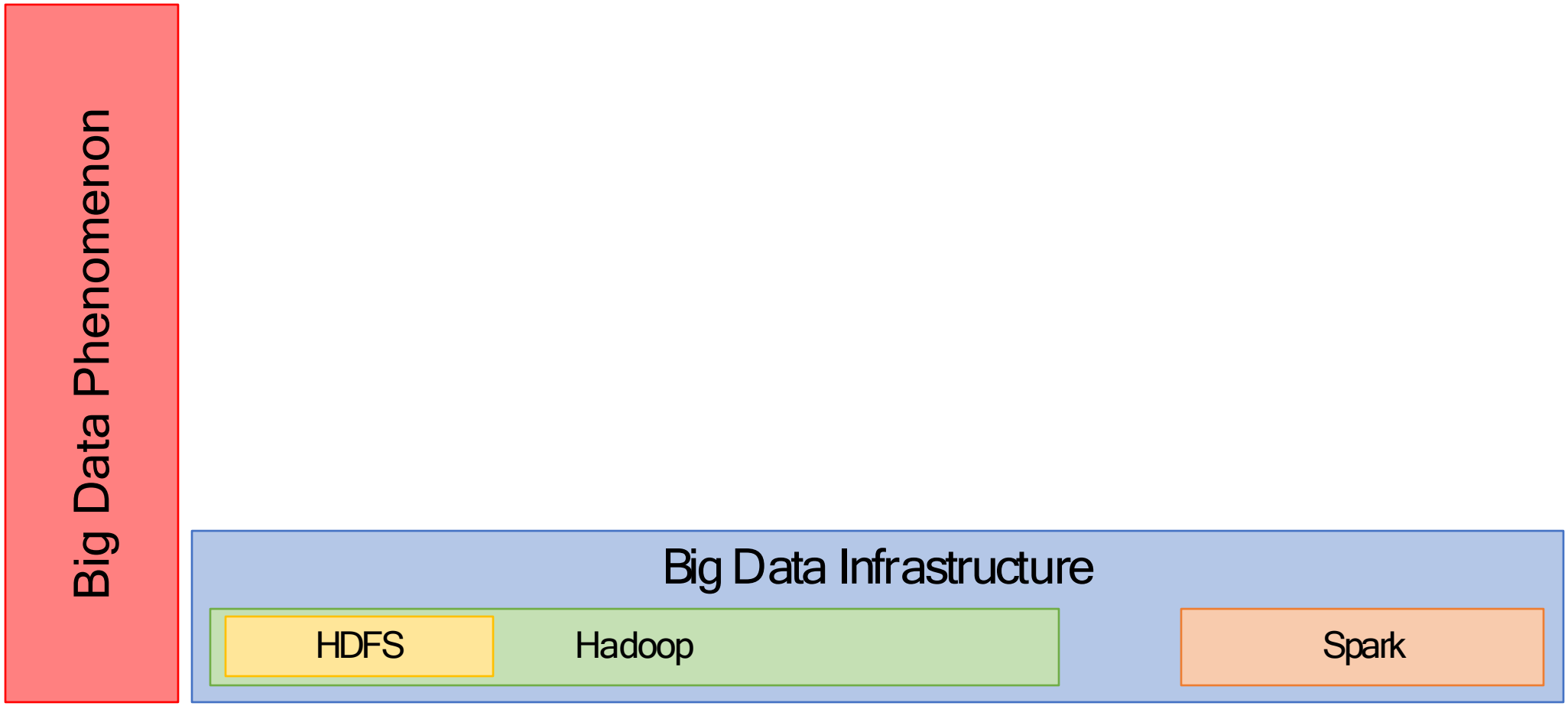
Prof. Daniele Ravi

Honorary Associate Professor
University College London (UCL) – UK
Department of Computer Science
d.ravi@ucl.ac.uk

dravi@unime.it

Outline of the Course

Outline of the Course



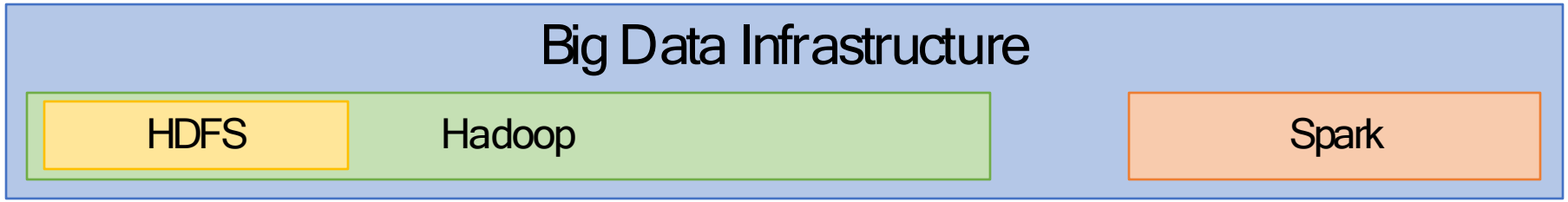
Outline of the Course

Big Data Phenomenon

Hadoop is an open-source framework that allows for the distributed **processing** of large datasets across **clusters** of computers using simple programming models.
It is designed to scale from a single server to **thousands of machines**.

•**Components of Hadoop:**

- **MapReduce**: A processing technique that divides tasks into small chunks and processes them in parallel across many machines.
- **HDFS (Hadoop Distributed File System)**: The storage system designed for Hadoop, which allows data to be distributed and stored across multiple nodes.
- **YARN (Yet Another Resource Negotiator)**: **The resource management** layer that allocates system resources to various applications running in Hadoop.



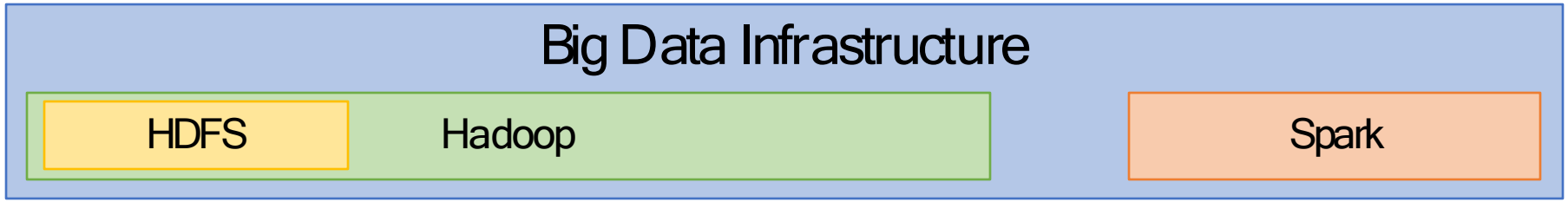
Outline of the Course

Big Data Phenomenon

HDFS is a part of Hadoop and is responsible for storing data across multiple machines in a reliable manner.
It ensures fault tolerance by replicating data across different machines.

•**Key Features of HDFS:**

- **Distributed Storage**: Data is split into blocks and distributed across many nodes in the cluster.
- **Replication**: Data is replicated (usually 3 times) to ensure that if one machine fails, the data is still accessible from another.
- **Large-File Optimized**: HDFS is designed for storing large files and is optimized for high-throughput access to big datasets.



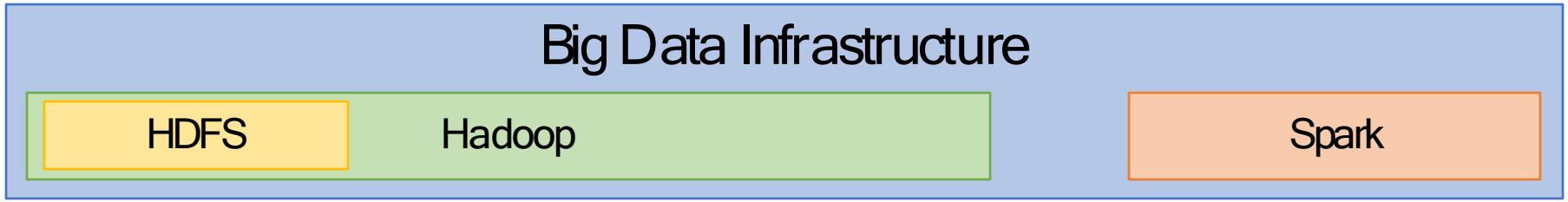
Outline of the Course

Big Data Phenomenon

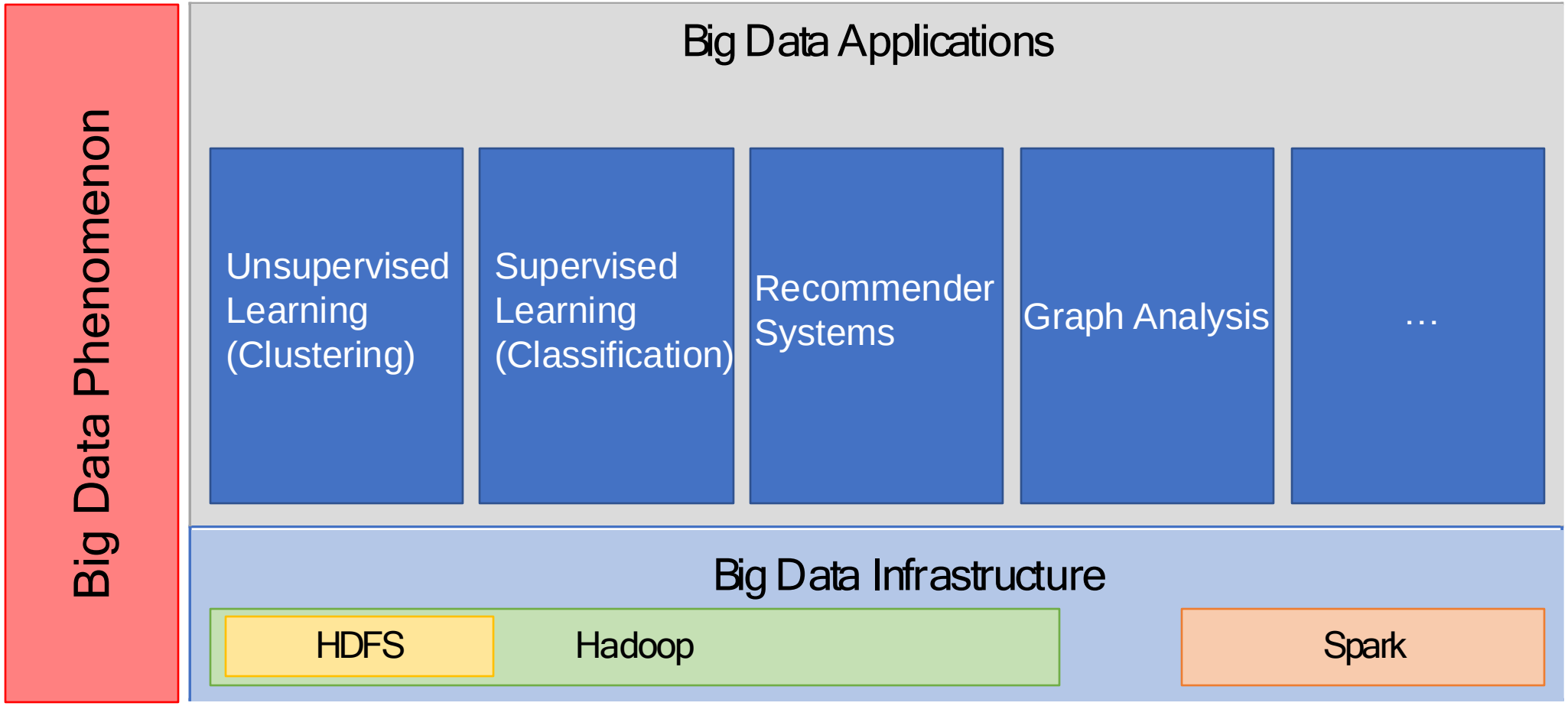
Apache Spark is an open-source, distributed **data processing engine** that can work with large datasets. Unlike Hadoop's MapReduce, which writes data to disk after every map and reduce operation, Spark keeps data in memory, which makes it significantly faster for certain workloads.

•**Key Features of Spark:**

- **In-Memory Processing**: Spark processes data in memory, making it faster for iterative algorithms and repeated operations on the same data.
- **Ease of Use**: Spark provides high-level APIs in Java, Scala, Python, and R, and supports complex operations like SQL queries, machine learning, and graph processing.
- **Distributed Processing**: Like Hadoop, Spark works in a distributed environment and can scale across clusters of machines.
- **Spark vs. Hadoop (MapReduce)**: Spark can be up to 100 times faster than Hadoop MapReduce

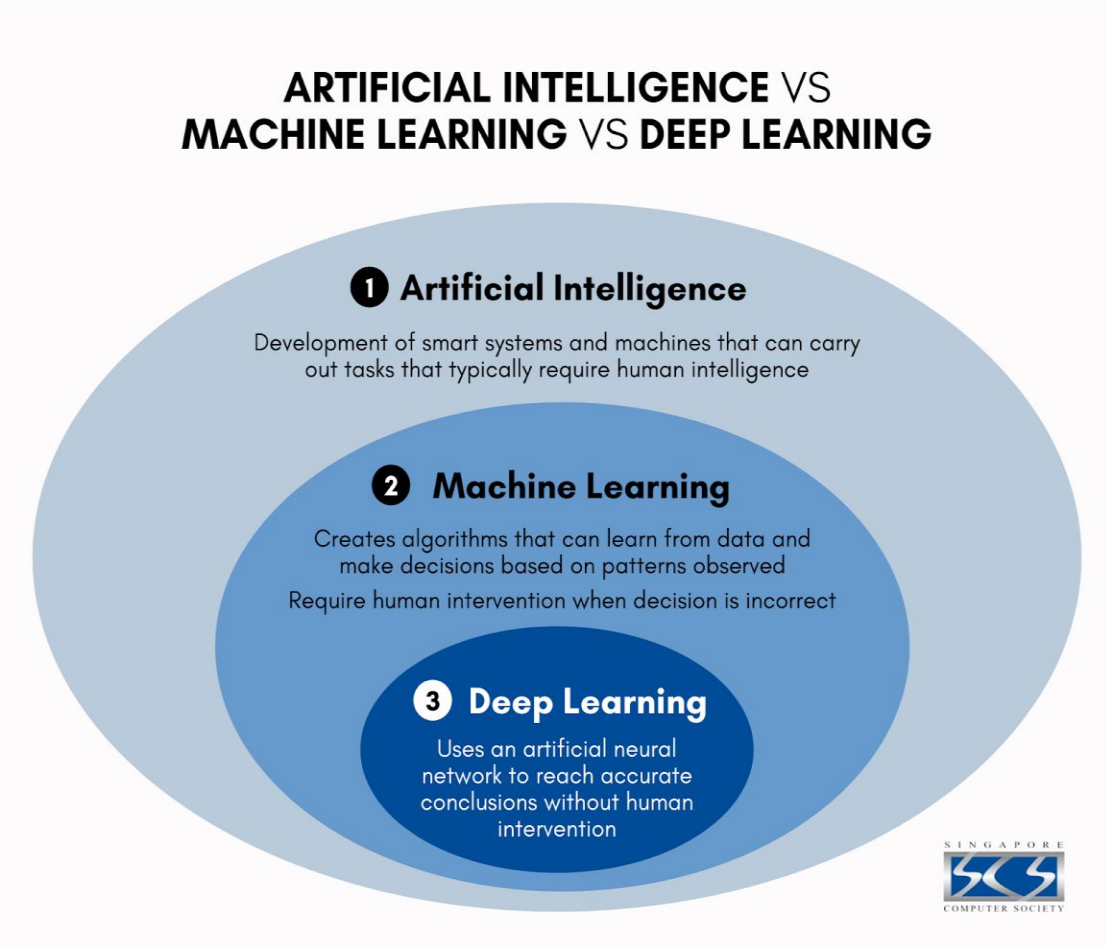


Outline of the Course

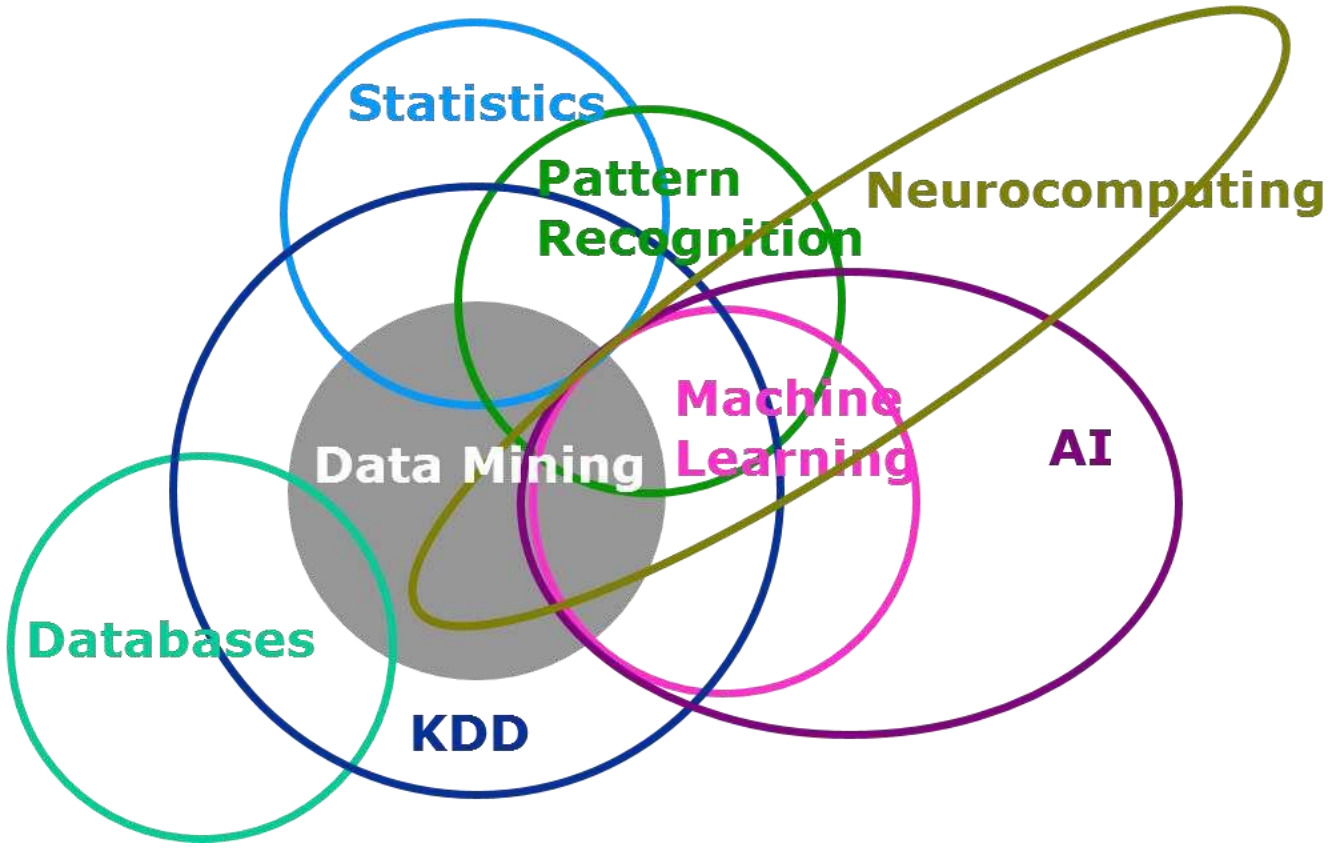


Why Big Data?

- **Machine learning and AI are based on Big Data :**
  - Develop algorithms that allow **computers** to change behavior (**learn**) based on **data**
  - Learn to recognize **complex patterns** and make **intelligent decisions**
  - We need knowledge of:
    - statistics
    - probability theory
    - data mining
    - pattern recognition
    - artificial intelligence



Big Data Analysis: Landscape



Why Big Data?

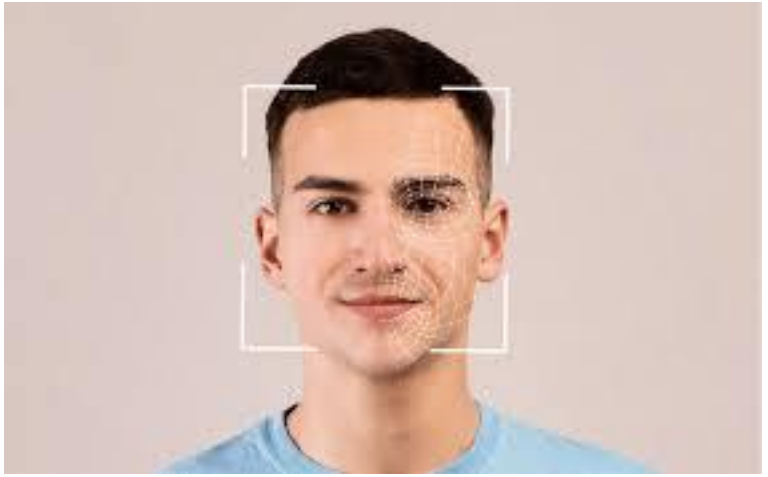
- The main idea is to automatize some process

BUT

- **In real worlds problems**
  - We may not have the expertise
  - We cannot explain how (speech recognition)
  - Problem changes over time
  - Need customized solutions (spam filtering)

- Therefore, we can learn it from past experience (our DATA)

What can we do with Big Data?



- Face, speech, handwriting recognition
  - Pattern recognition
- Spam filtering, terrain navigability (rovers)
  - Classification
- Credit risk assessment, weather forecasting, stock market prediction
  - Regression
- Future: Self-driving cars? Translating phones?

What can we do with Big Data?

- **Aim:**
  - Extract knowledge (information) from past experience (training data)
  - Use this knowledge to analyze new experiences (new samples)

- **Problem:**
  - Designing rules to deal with new data by hand can be difficult
    - How to write a program to detect a cat in an image?

- **Solution:**
  - Collecting data can be easier
    - Find images with cats, and without
  - Use machine learning to automatically find such rules

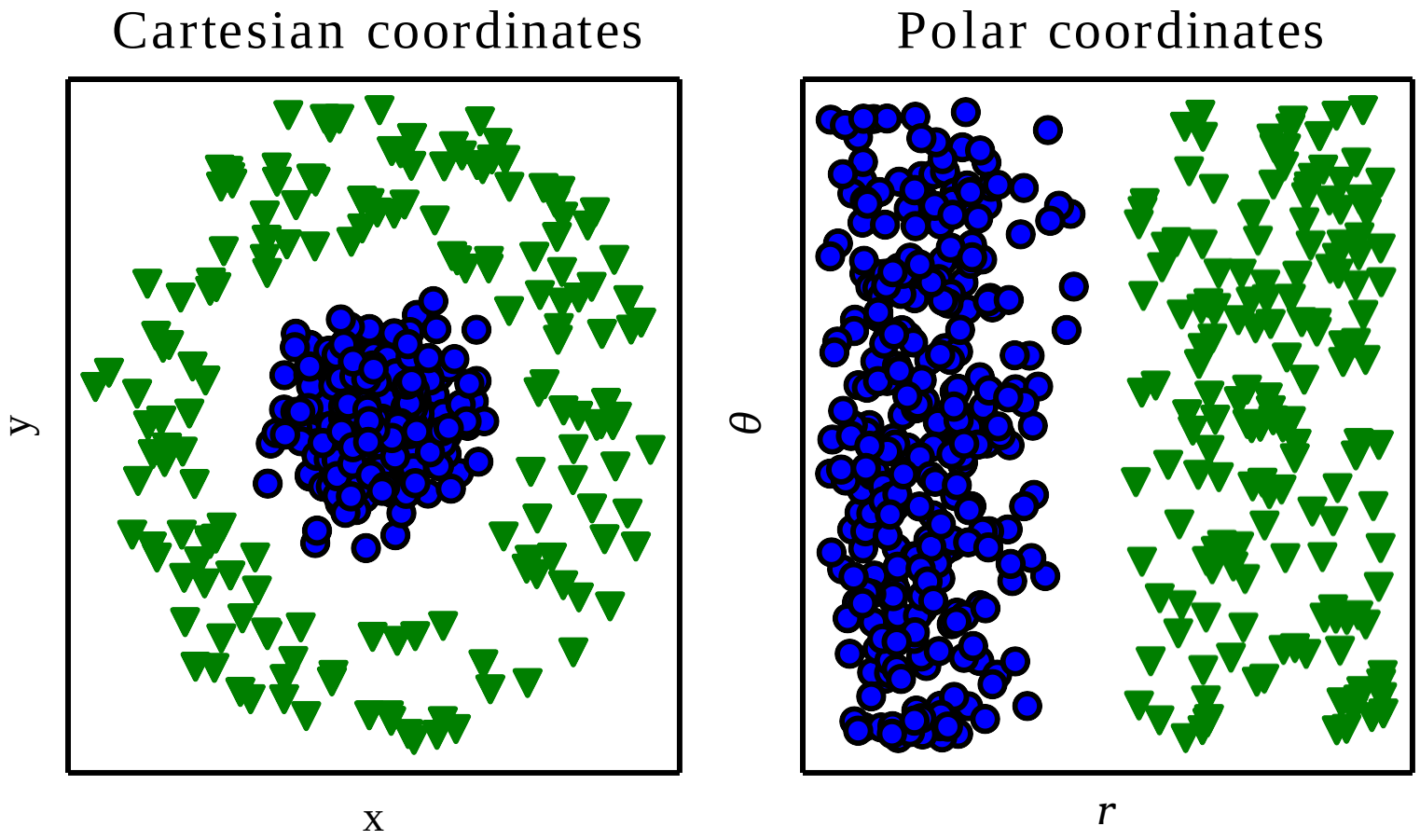
Steps to do with big data

1. **Data collection:** “training data”, optionally with “labels” provided by an “expert”
2. **Representation:** how the data are encoded into “features” when presented to learning algorithm
3. **Modeling:** choose the class of models that the learning algorithm will choose from
4. **Estimation:** find the model that best explains the data: *simple and fits well*
5. **Validation:** evaluate the learned model and compare to solution found using other model classes
6. **Test:** apply learned model to new “test” data

Data Representation

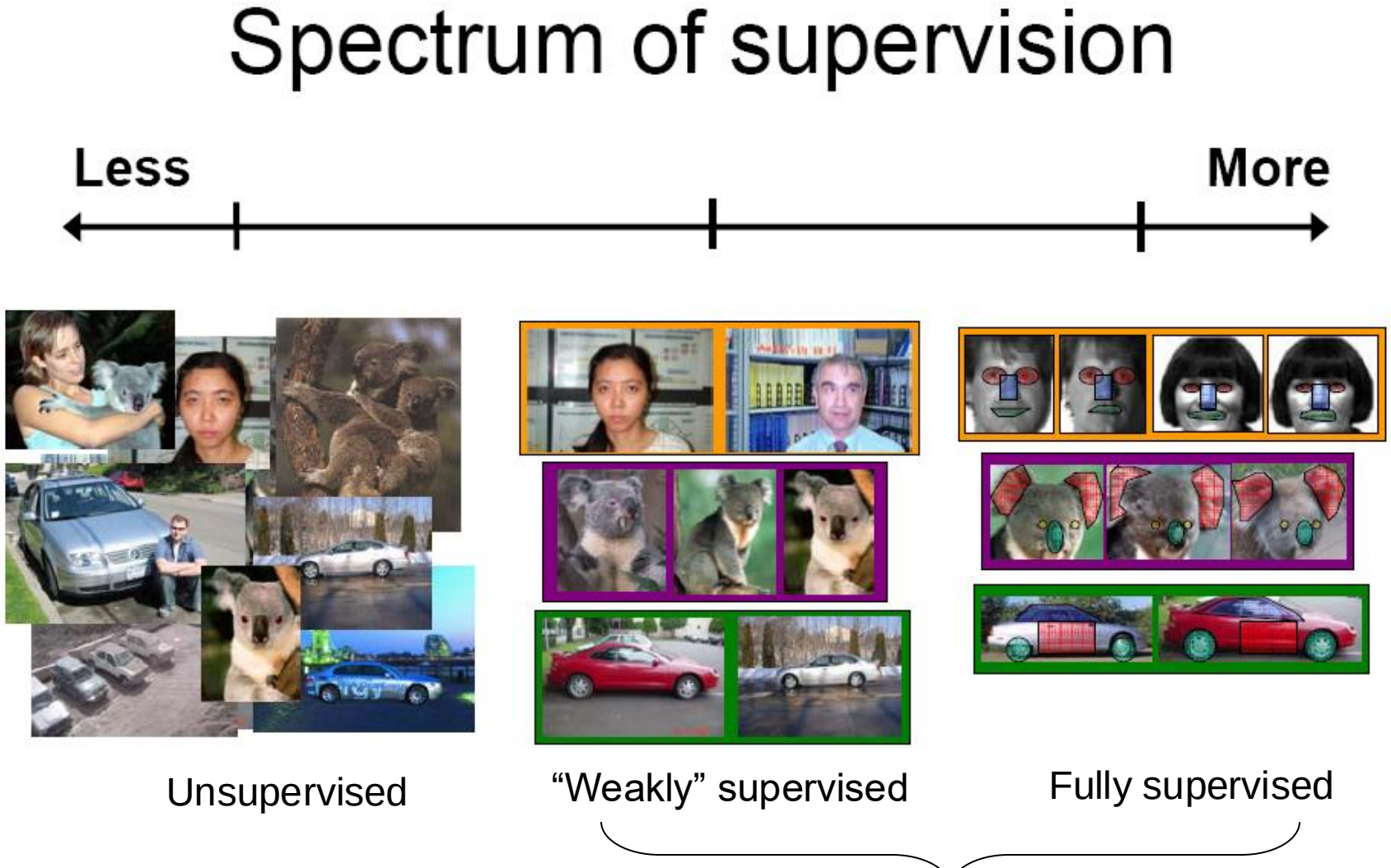
- Many type of representations:
  - Numbers, vectors, graphs, images, text
  - **Homogeneous** or **heterogeneous**
- Choice of representation may impact the choice of learning algorithm
- **Domain knowledge** can help to design or select good features
  - The ultimate feature would solve the learning problem
- We can use automatic methods known as "**feature selection**" approaches

Representation of Data Matter



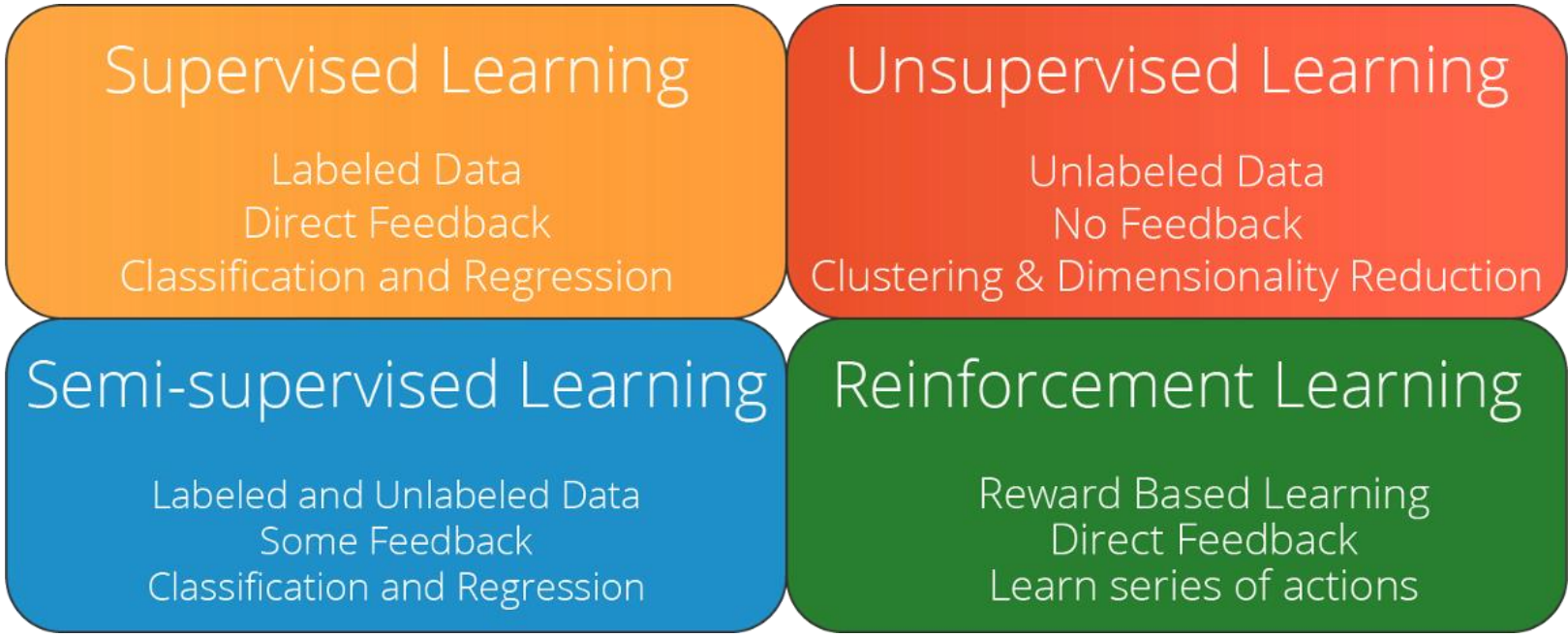
Chapter 1 of the book *Deep Learning (*www.deeplearningbook.org)

Machine Learning and AI are a very rich family!

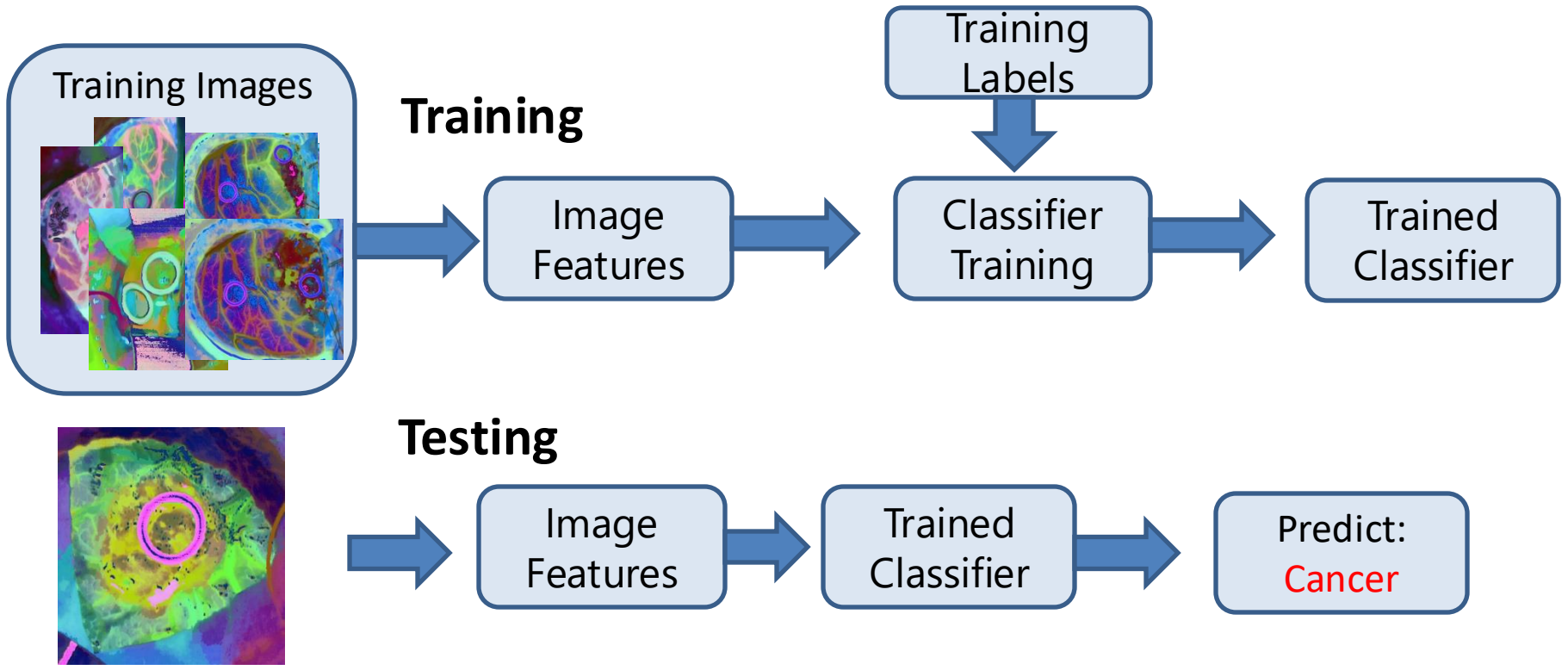


Machine Learning is a very rich family!

- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimension reduction
  - Density estimation
- Semi-supervised
  - Combine labeled data with
  - unlabeled data
- Active learning
  - Determine the most useful data to label next



Supervised Learning Example



Typical features for images are:

SIFT, HOG, LBP, FAST, etc.

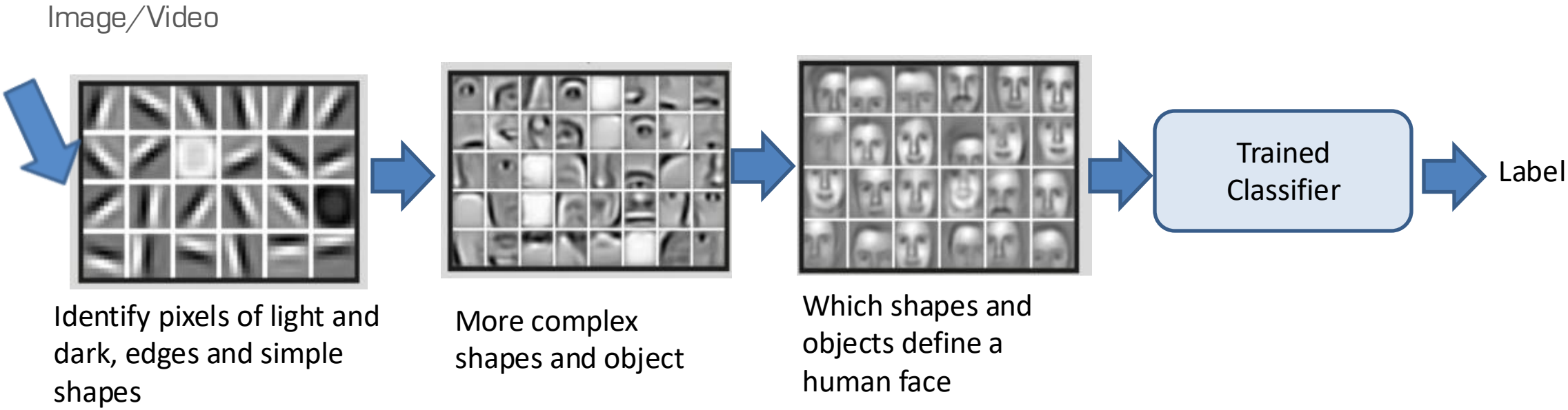
- These features are not learned (**hand-crafted**)

Machine learning VS Deep Learning

Learning the features automatically:

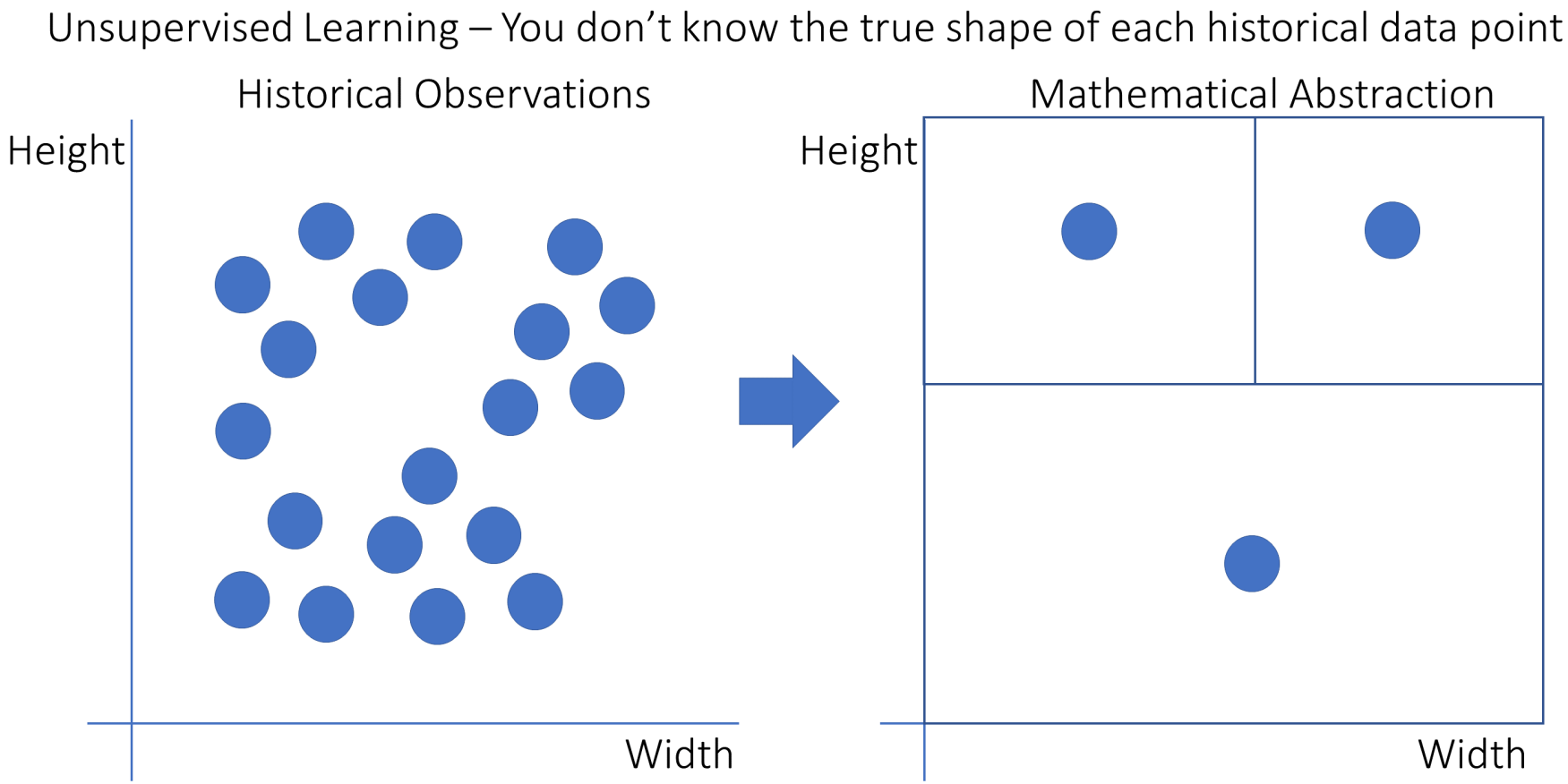
I.e., in a Convolutional Neural Network (CNN), each layer of the hierarchy extracts features from the output of the previous layer.

Increasingly complex rules are used to categorize complicated shapes, such as faces.



Unsupervised Learning Example

- Unsupervised learning tries to group data by evaluating their similarity
- You can never be certain that grouped data points belong to the same cluster



Why big data is important?

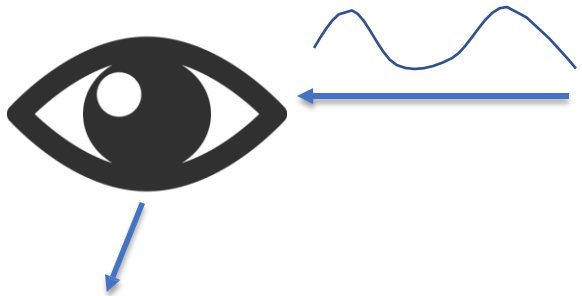
Application Domains

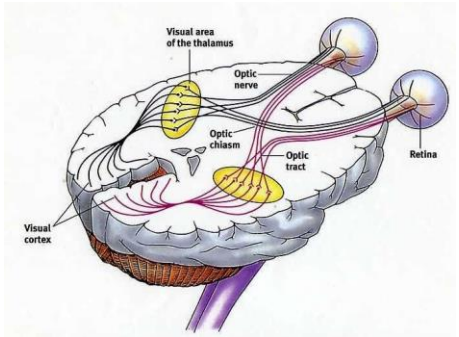
- Computer Vision
- Robotics
- Gaming
- Natural Language Processing and Speech
- Art
- Healthcare

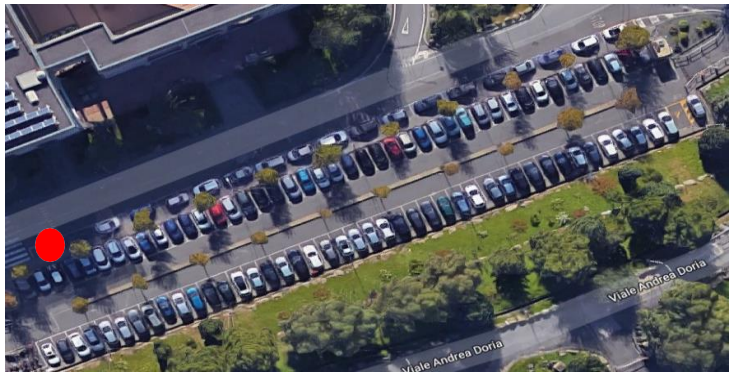
Computer Vision

"learning to answer questions about images and videos"

Human Visual System







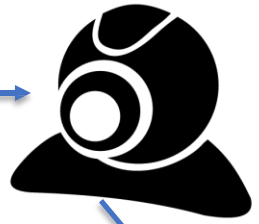
TASK

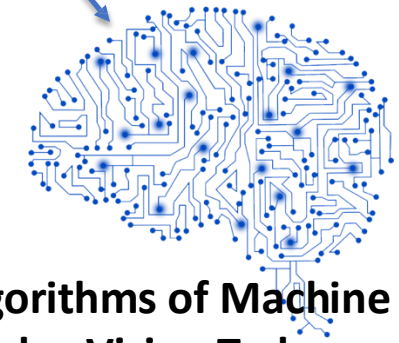
How many cars are in the parking area?

103

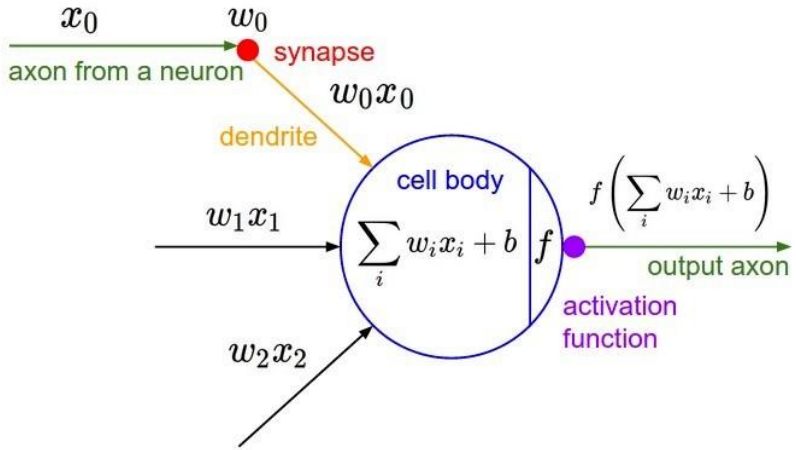
Visual Artificial System

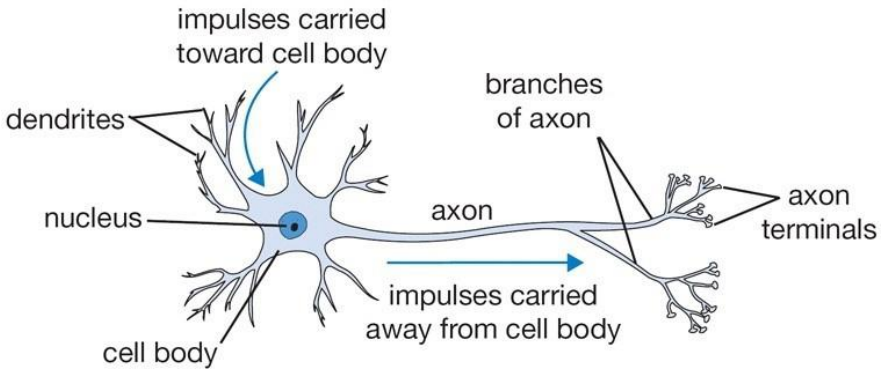
101011001





Advanced Algorithms of Machine Learning to Solve Vision Tasks





Computer Vision

- Vision is challenging!
  - For a small patch of resolution 256x256 with 256 pixel values
  - A total of $2^{8x256x256} = 2^{524288}$ of possible images
  - Similar to the stars in the universe

Can You Spot the Baby?

Computer Vision

- Object Recognition and Segmentation
- Image Captioning
  - Is the process of generating textual description of an image
- Human pose Estimation
- Summarization and Indexing
- Localization

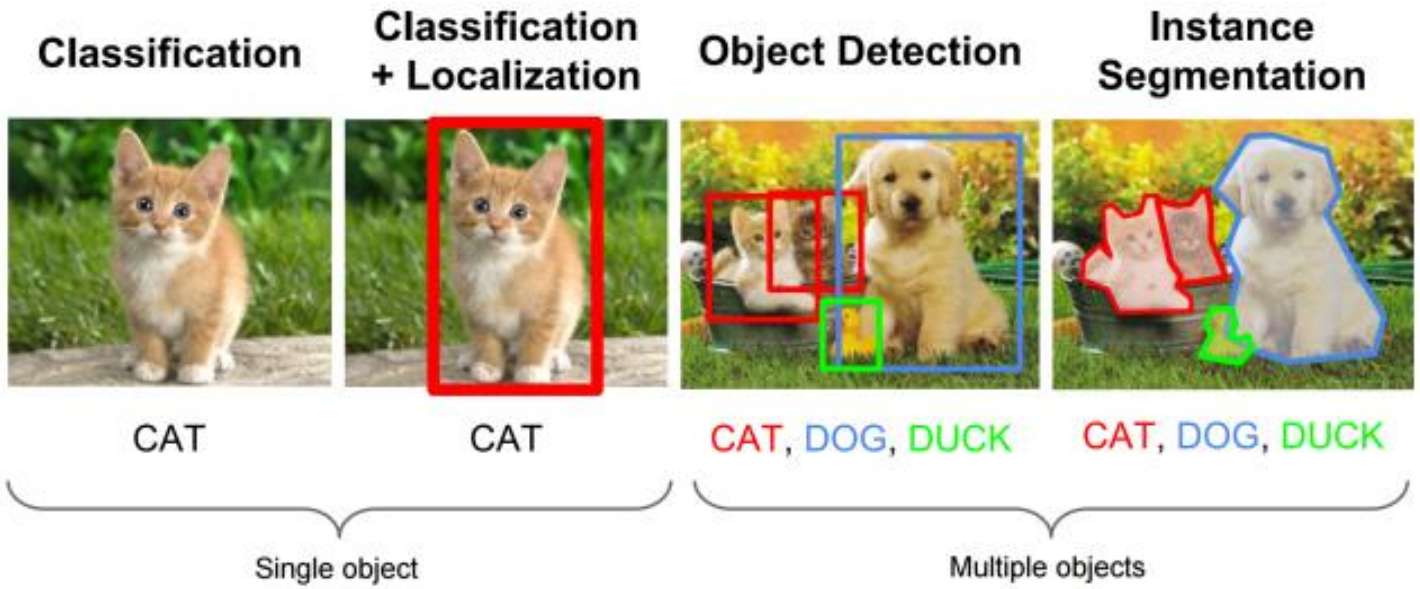
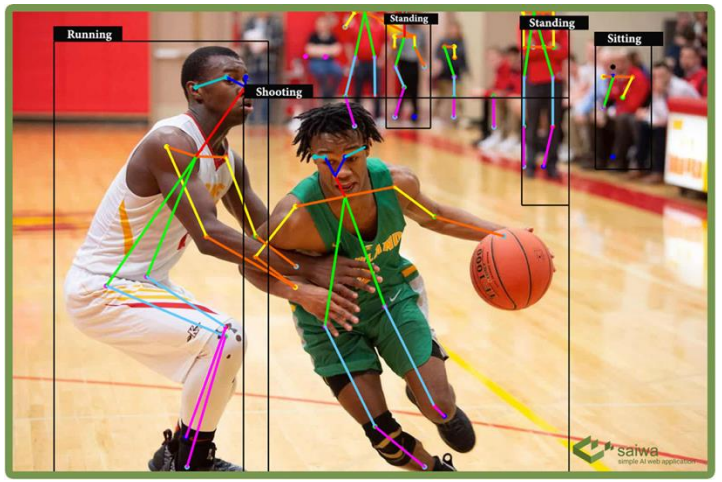


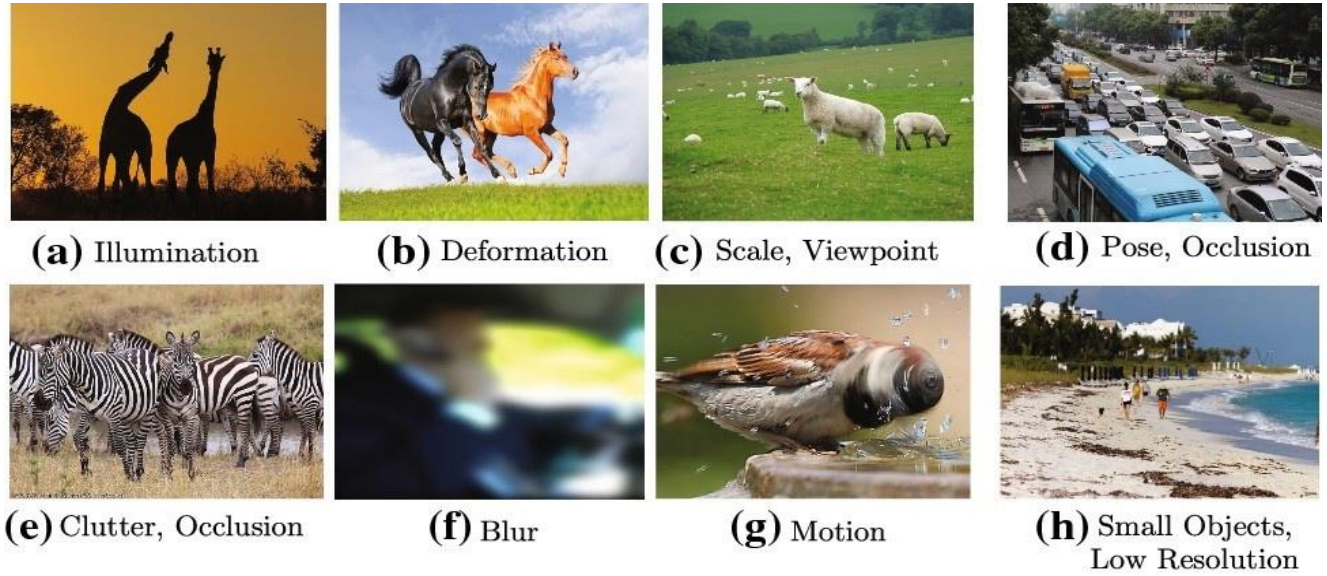
Image Captioning

Human pose Estimation



Computer Vision

- Image variability:
  - viewpoint, scales, deformations, occlusions, illuminations, pose, resolutions



(a) Illumination (b) Deformation (c) Scale, Viewpoint (d) Pose, Occlusion

(e) Clutter, Occlusion (f) Blur (g) Motion (h) Small Objects, Low Resolution

- Semantic Variation:
  - Intra-class Variation

(i) Different instances of the "chair" category

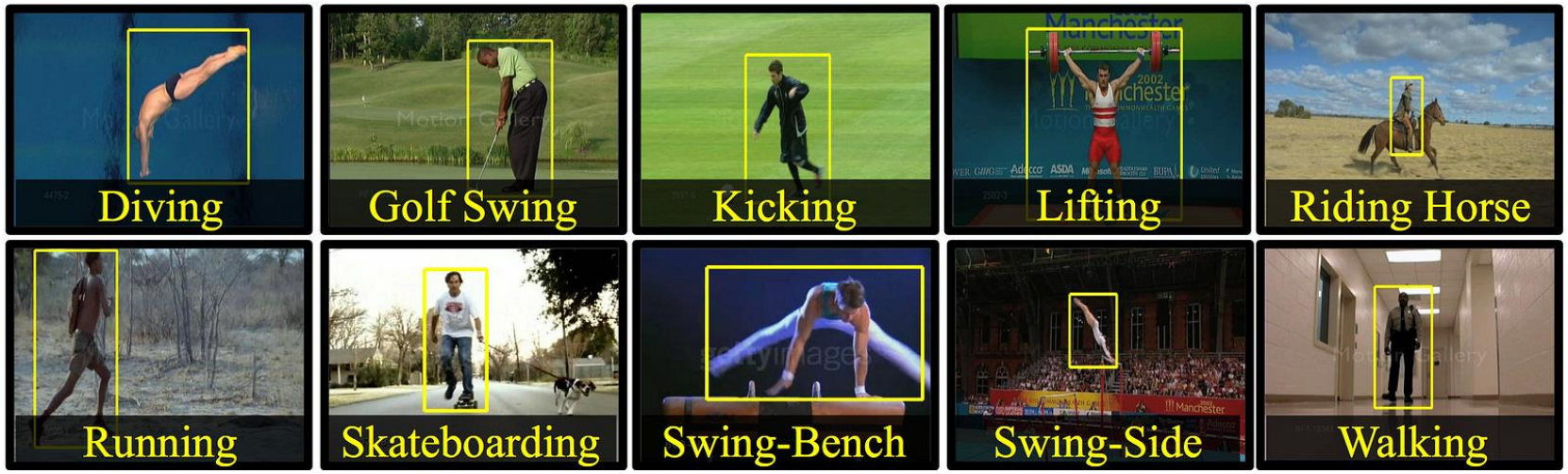
  - Inter-class similarity

(j) Small Interclass Variations: four different categories

Robots / Artificial Agents

Tracking and Location Estimation

Activity Recognition



Factory Robots

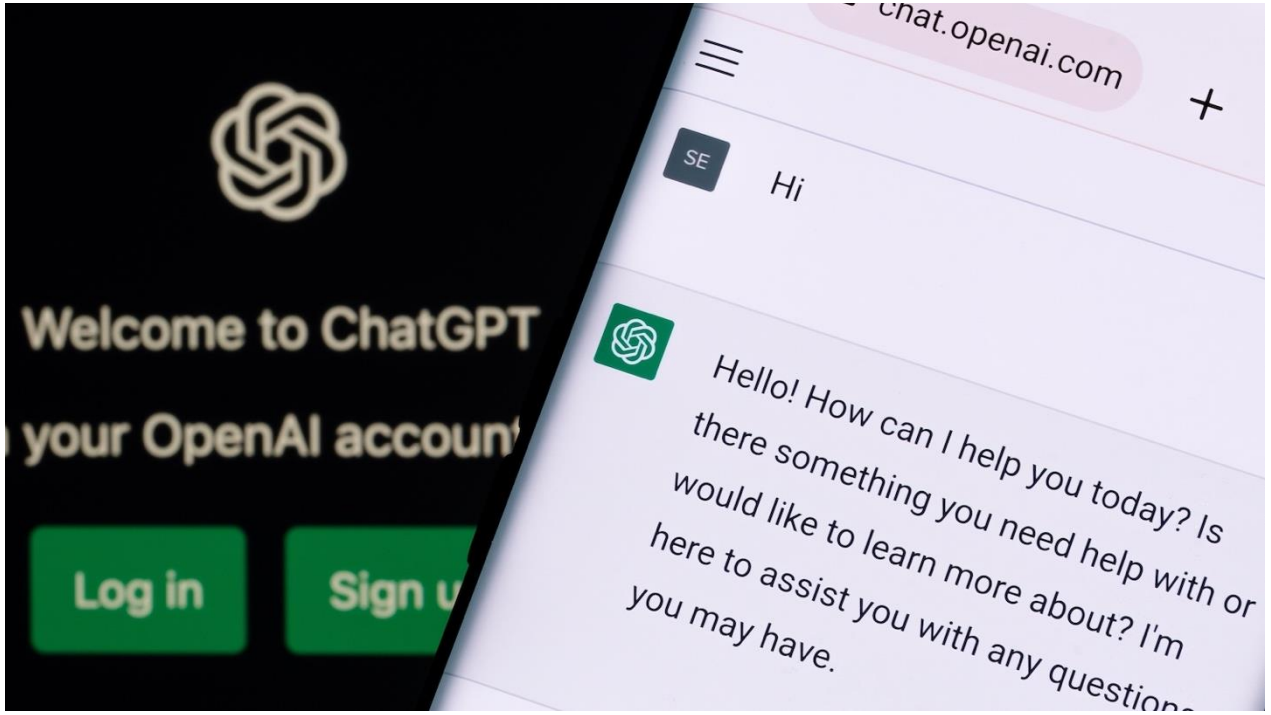
Robots / Artificial Agents

Self-Driving Cars

Natural Language Processing, Speech Recognition, Audio Recognition

Audio recognition

Language Understanding



NLP is a complex task

- Synonymy
- Ambiguity

- NLP is very high dimensional
  - I.e. English dictionary contains over 170,000 words

Big Data for Art

- **Image to Image Translation**
  - https://affinelayer.com/pixsrv/

- **Generating Music**
  - https://openai.com/research/musenet
  - https://openai.com/research/jukebox

- **Generating Realistic Images or Imitating Famous Painters**
  - https://labs.openai.com/
  - https://www.midjourney.com/showcase/recent/

Big Datafor Art

- Generating Avatars
- Generating Video

https://synthesys.io/

Big Data for Art

- Music, painting, etc. are tasks that are **uniquely human**
  - Difficult to model
  - Even more difficult to evaluate (if not impossible)

- If machines can generate novel pieces resembling art, they must have understood something about “beauty”, “harmony”, etc.

- Have they really learned to generate new art?
  - Or do they just fool us by memorizing data?

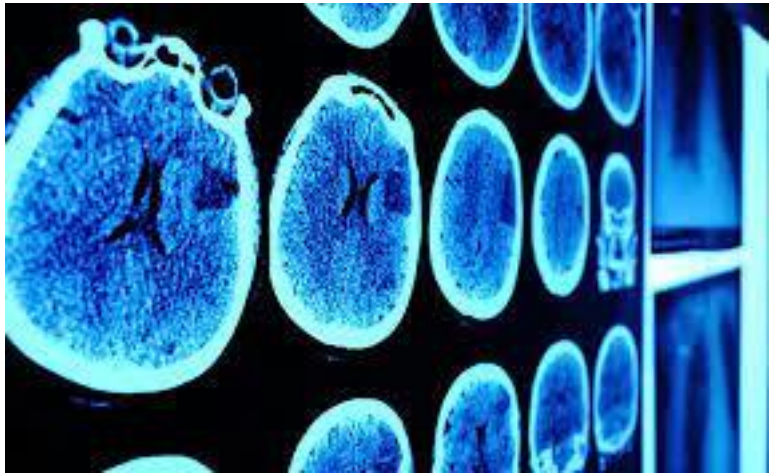
Big Data in healthcare

Diagnose with unprecedent accuracy

AI can improve medical services:

1. Decrease the costs
2. Make them more efficient
3. Less prone to mistakes

Augment doctors



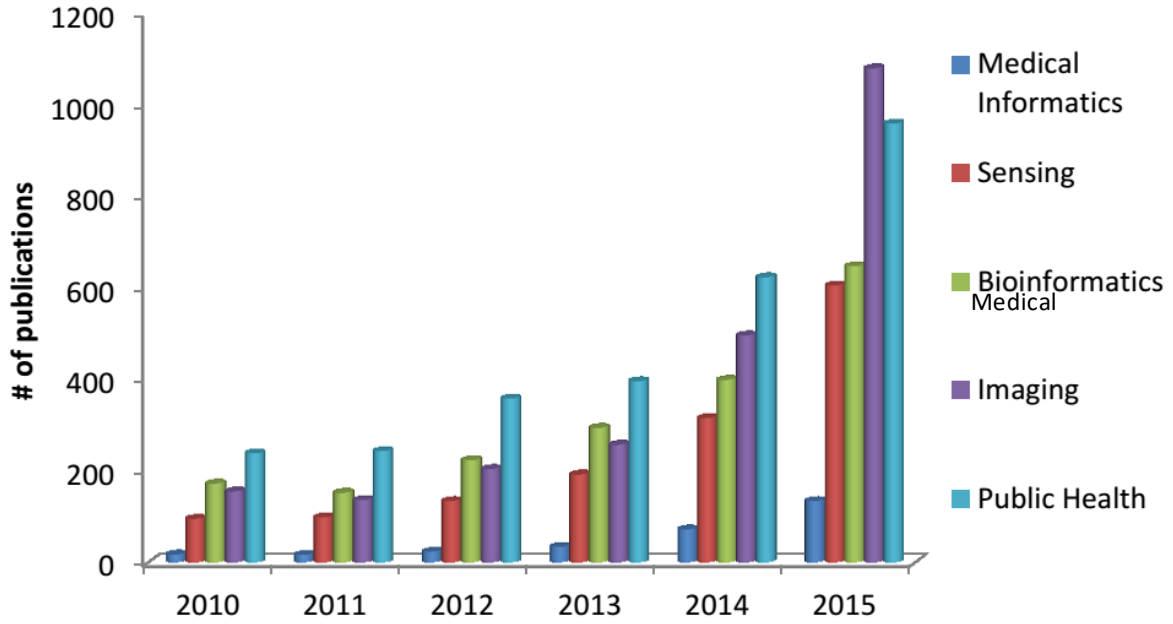
- Early Diagnosis
- Improving patient care
  - (i.e. increase expectation life)

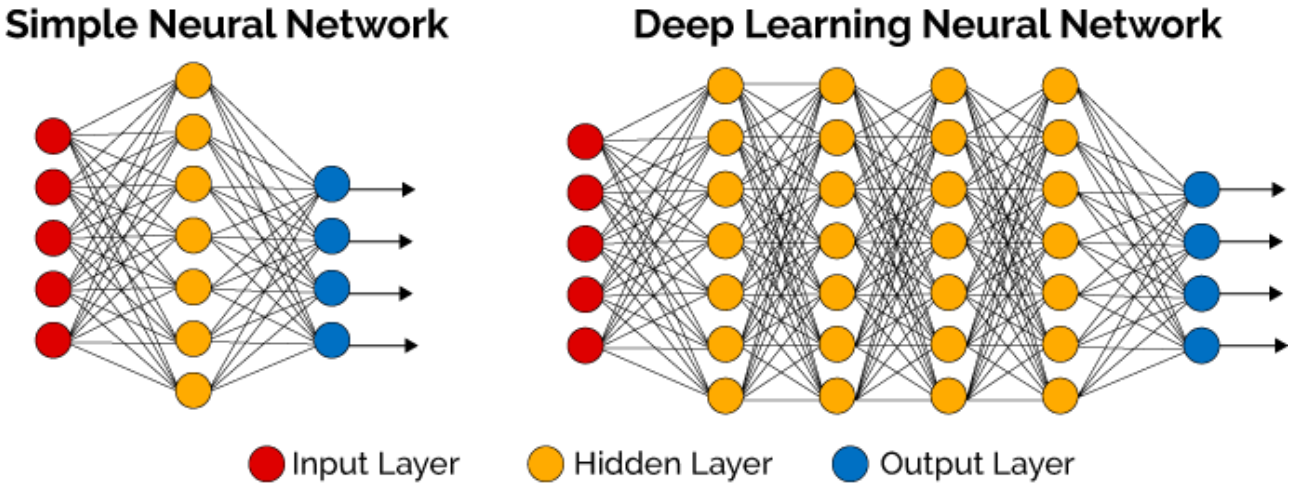
Big Data in healthcare

Rapid uptake in healthcare due to:

1. Many medical centers that collect and organize large sets of patient data

2. Computational hardware improvements
   High performance computing
   Cloud computing
   GPUs
   Fast data storage

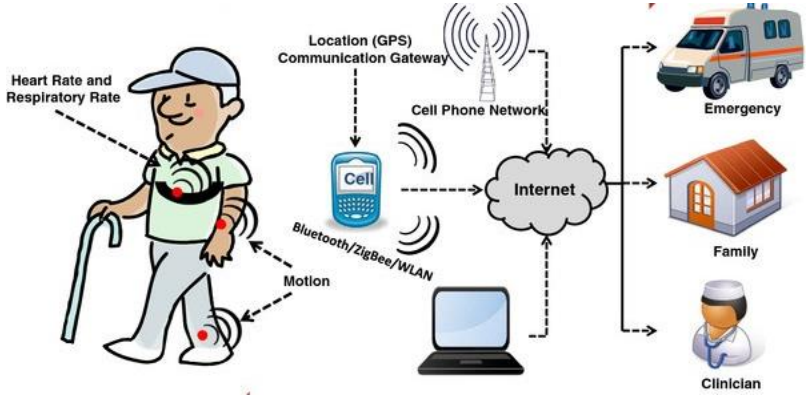




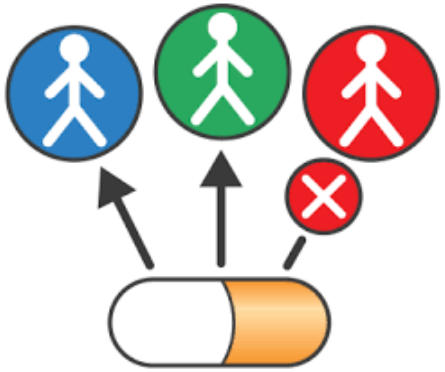
Ravì, Daniele, et al. "Deep learning for health informatics." *IEEE journal of biomedical and health informatics* 21.1 (2016): 4-21

Big Data in healthcare

Monitor patients in a free-living environment

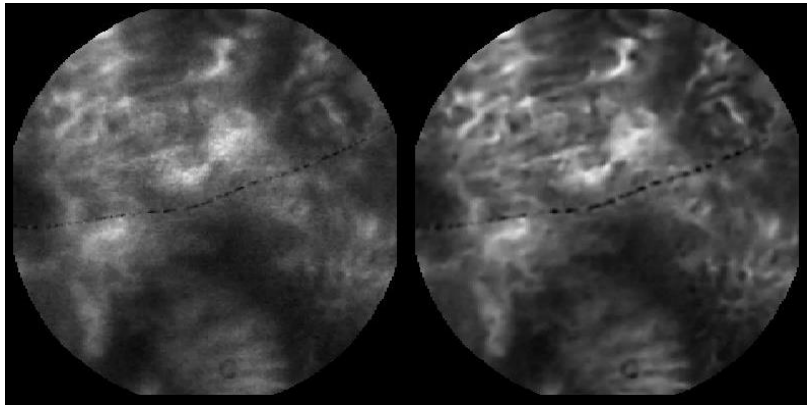


Predict patient outcomes to provide personalized treatments



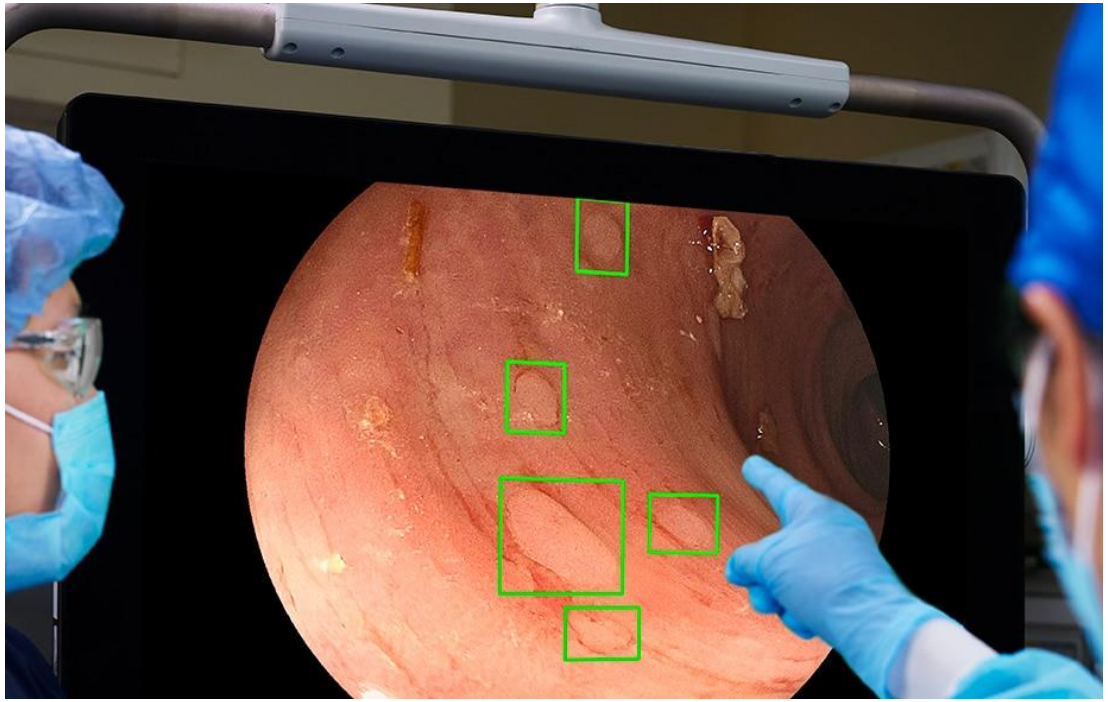
Assist surgeons during surgical procedure

Improve diagnosis during optical biopsy



Big Data in healthcare: endoscopy

- The first-to-market, computer-aided polyp detection system powered by AI
  - Approved by FDA
- Detect colorectal polyps through enhanced visualization during colonoscopy



GI Genius™ has been shown to increase adenoma detection rates by up to 14.4%.

Big Data in healthcare: Smart Sensing

Applications

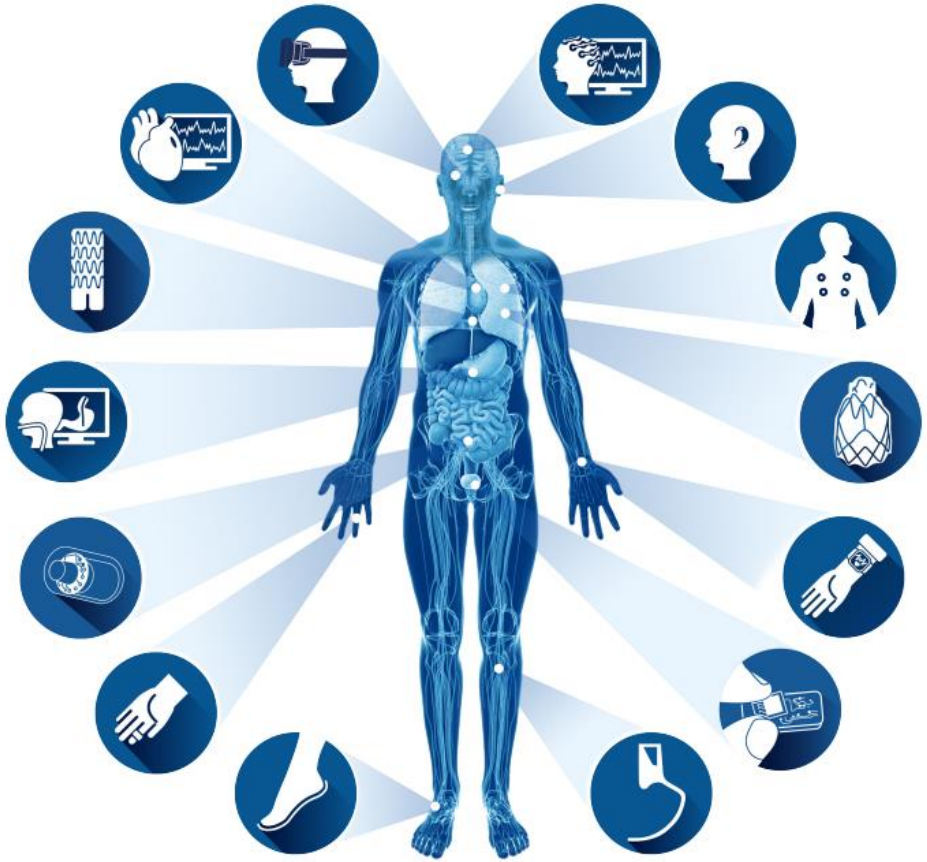
Patient monitoring

Wellbeing

Diseases status change

Sports

Wearable and implantable sensors



Challenges:

- **Dance data (50Hz)**
- **Continues data (entire day)**

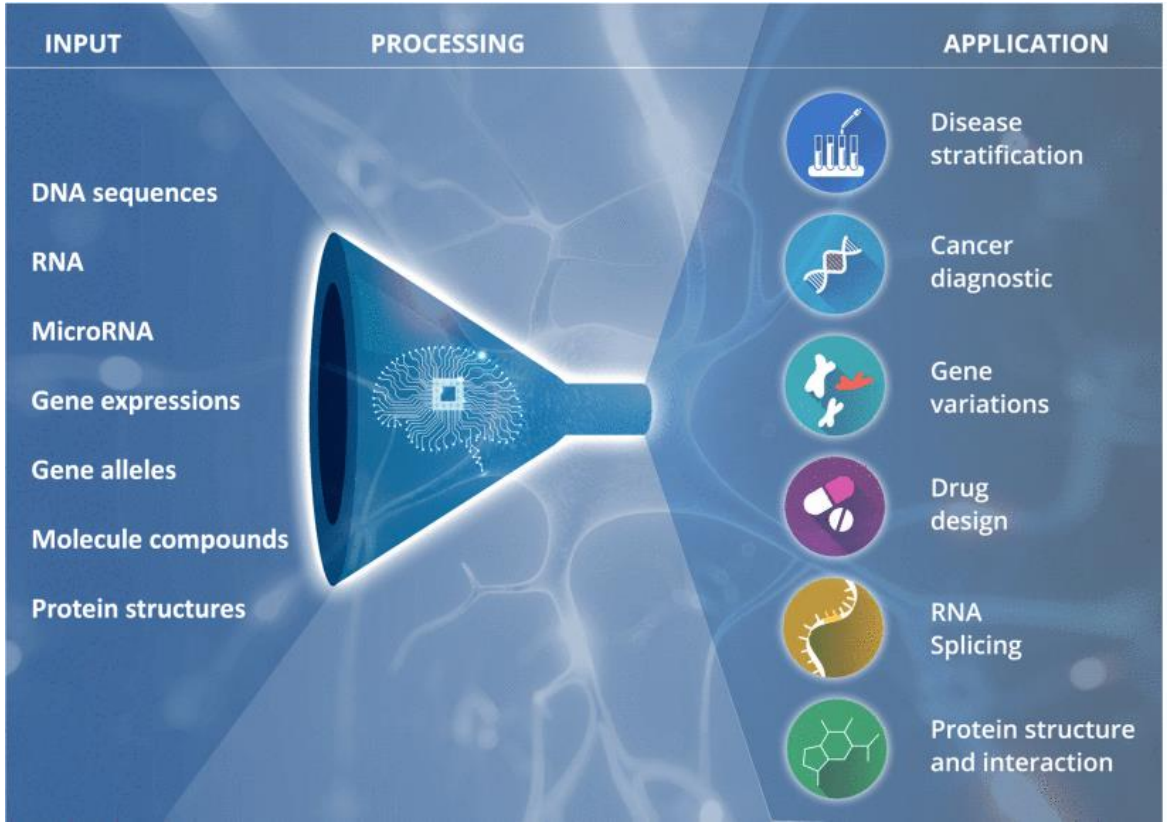
Big Data in healthcare: Smart Sensing

- **Food intake monitoring**
- Motivations:
  - Traditional methods based on **questionnaires are unreliable**
- Solution:
  - **Automatically** food intake detection using **egocentric cameras**

D. Ravì, B. Lo, G-Z Yang, “Real-time Food Intake Classification and Energy Expenditure Estimation on a Mobile Device”, IEEE Body Sensor Networks (BSN) 2015

Big Data in healthcare

	Applications	Input Data
Bioinformatics	Cancer diagnosis Gene selection/classification Gene variants	Gene expression MicroRNA Microarray data
	Drug design	Molecule compounds
	Compound-Protein interaction RNA binding protein DNA methylation	Protein structures Molecule compounds Genes/RNA/DNA sequences
Medical Imaging	3D brain reconstruction Neural cells classification Brain tissues classification Alzheimer/MCI diagnosis	MRI/fMRI Fundus images PET scans
	Tissue classification Organ segmentation Cell clustering Hemorrhage detection Tumour detection	MRI/CT Images Endoscopy images Microscopy Fundus Images X-ray images Hyperspectral images



Big Data in healthcare

	Applications	Input Data
Pervasive Sensing	Anomaly detection Biological parameters monitoring	EEG ECG Implantable device
	Human activity recognition	Video Wearable device
	Hand gesture recognition Obstacle detection Sign language recognition	Depth camera RGB-D camera Real-Sense camera
	Food intake Energy expenditure	Wearable device RGB Image Mobile device
Medical Informatics	Prediction of disease Human behaviour monitoring Data mining	Electronic health records Big medical dataset Blood/Lab tests
Public Health	Predicting demographic info Lifestyle diseases Infectious disease epidemics Air pollutant prediction	Social media data Mobile phone metadata Geo-tagged images Text messages

We can continue giving examples on:

security

industry

etc....

but you have got the point!

Nowadays, the use of Big Data is basically everywhere

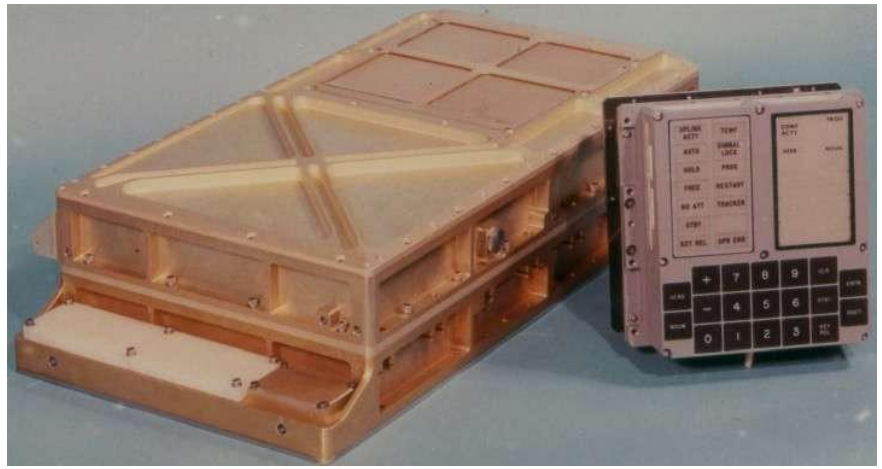
Let's Get Started!

What the He…ck isThat?

source: Wikipedia

The Apollo Guidance Computer (AGC)

The computer installed on each command and lunar module of all the Apollo program's missions



A few numbers:

- ~2 MHz CPU clock frequency
- 16-bit architecture
- 3,840 bytes of main memory (RAM)
- 69,120 bytes of non-volatile read-only memory (ROM)

All the running software was written in AGC assembly language, now also available on GitHub

Almost 55 Years Have Passed…

… And The World Has Changed

AGC vs. Our Smartphone

- Most recent smartphones have
  - > 3 GHz CPU clock frequency
  - 4 ÷ 16 GB of RAM
  - 64 ÷ 1000 GB of internal storage (**don't** call it ROM!)

AGC vs. Our Smartphone

- Most recent smartphones have
  - >3 GHz CPU clock frequency
  - 4÷16 GB of RAM
  - 64÷1000 GB of internal storage (**don't** call it ROM!)

~3 orders of magnitude faster (~1,000x)

~6/7 orders of magnitude larger RAM and internal storage (up to 10,000,000x)

A Side Note on Units

Prefixes for multiples of bits (bit) or bytes (B)

Decimal				Binary				
Value		SI		Value		IEC		JEDEC
1000	10^3	k	kilo	1024	2^{10}	Ki	kibi	K kilo
1000^2	10^6	M	mega	1024^2	2^{20}	Mi	mebi	M mega
1000^3	10^9	G	giga	1024^3	2^{30}	Gi	gibi	G giga
1000^4	10^{12}	T	tera	1024^4	2^{40}	Ti	tebi	–
1000^5	10^{15}	P	peta	1024^5	2^{50}	Pi	pebi	–
1000^6	10^{18}	E	exa	1024^6	2^{60}	Ei	exbi	–
1000^7	10^{21}	Z	zetta	1024^7	2^{70}	Zi	zebi	–
1000^8	10^{24}	Y	yotta	1024^8	2^{80}	Yi	yobi	–

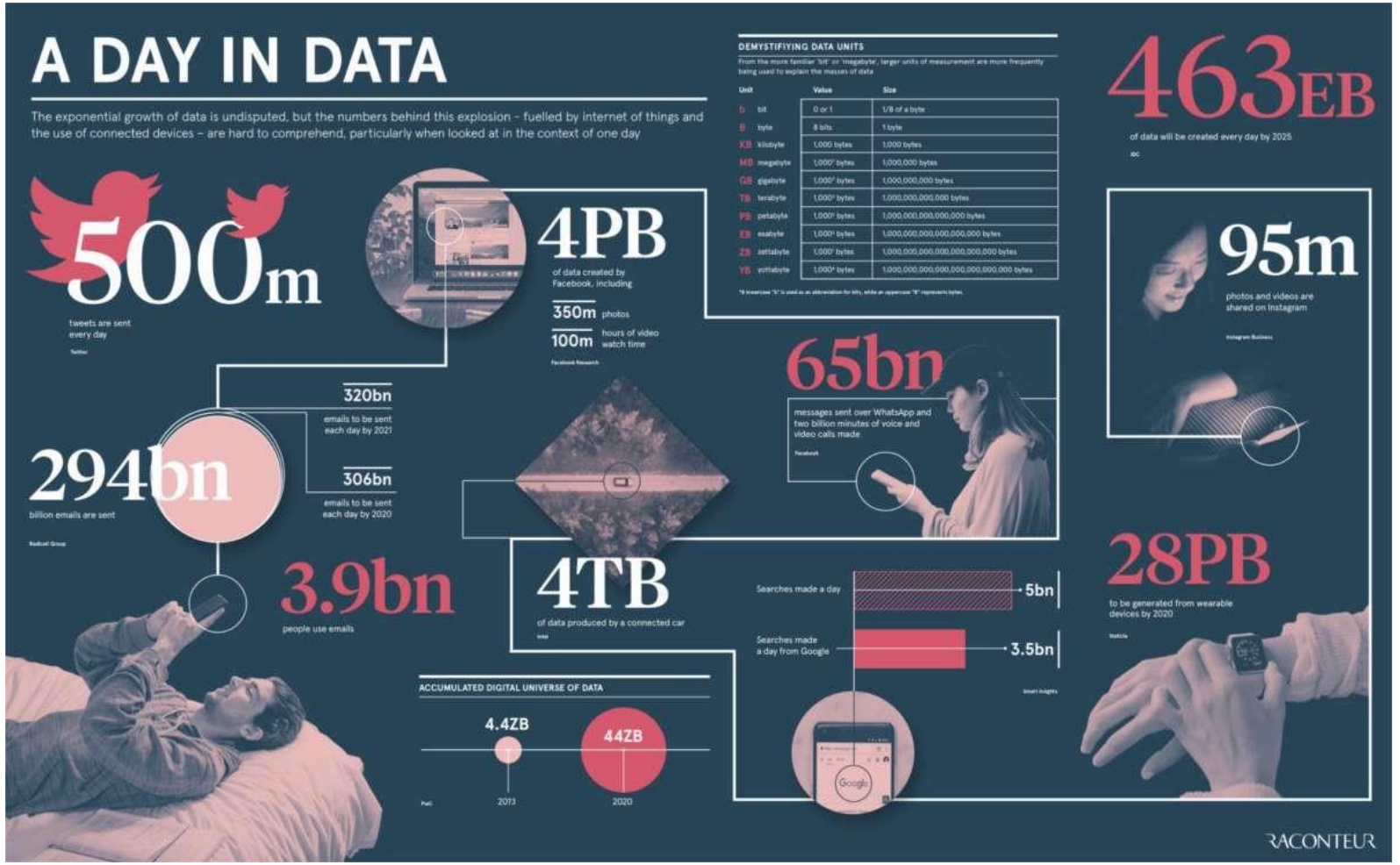
The Information Technology (IT) Revolution

- Started almost 60 years ago and still rocketing
- Driven by:
  - Science/Engineering
  - Business
  - Society

What Happens on the Internet in 1 Minute?

source: LocaliQ

How Much Data is Generated Each Day?

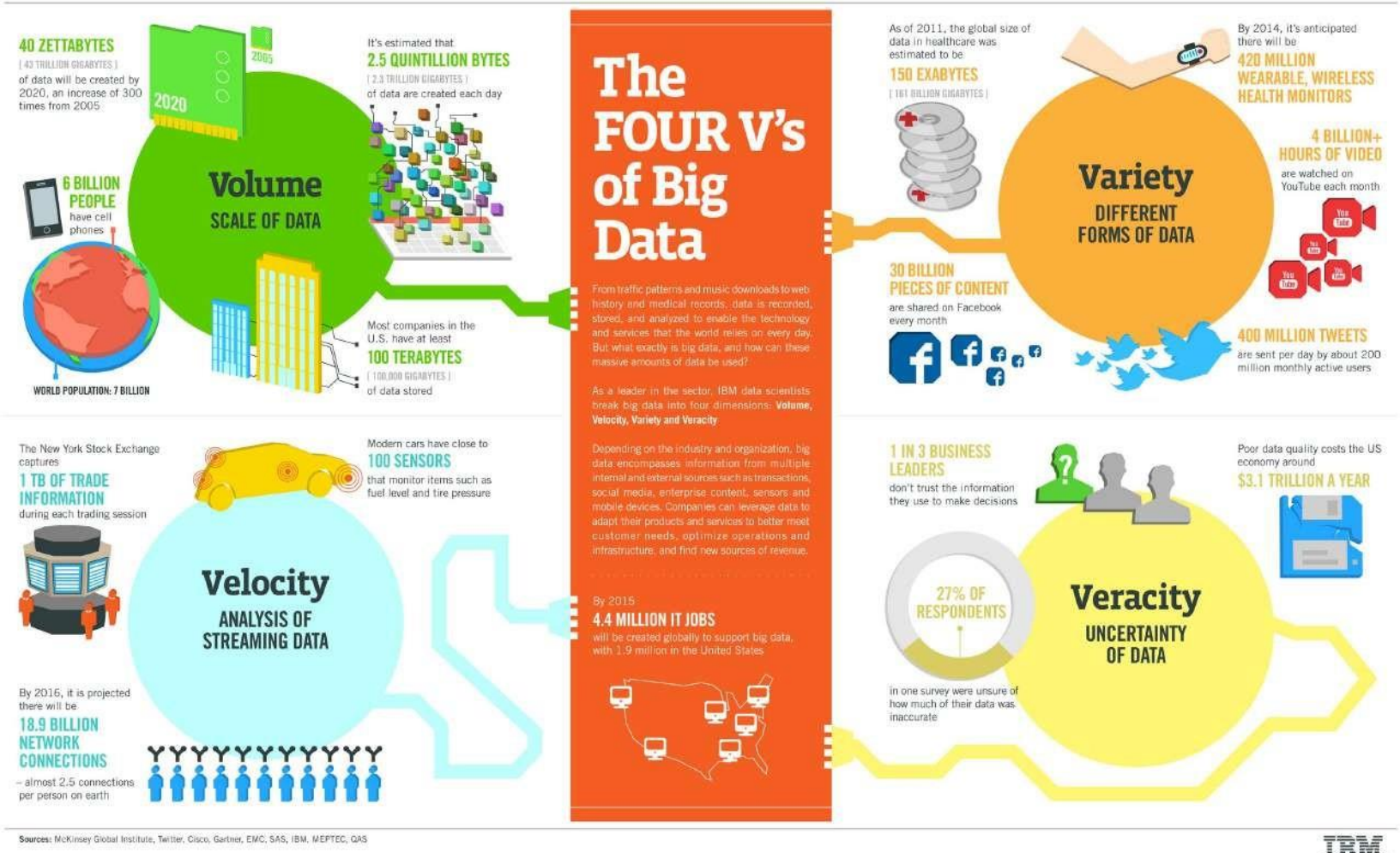


source: VisualCapitalist

What is Big Data?

- Sometimes a buzzword yet describing an actual phenomenon
- **4 V**'s (sometimes, 5, 6 or even 7!)
  - **Volume** very large amount of data (orders of TB or PB)
  - **Variety** different formats of data: structured (relational tables), semi-structured (JSON files), and unstructured (text/audio/video)
  - **Velocity** insane speed at which data is generated (e.g., Twitter stream)
  - **Veracity** reliability of the data used to drive decision processes

The 4 V's of Big Data

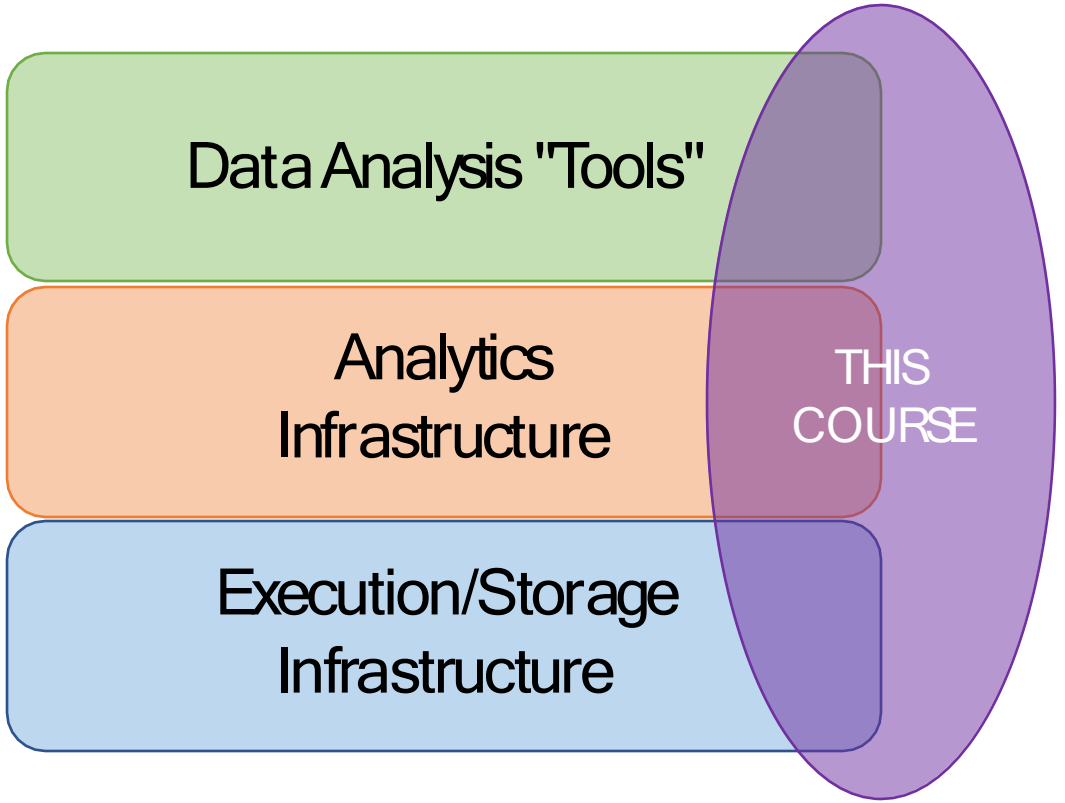


source: IBM

The Value of Big Data

- Extracting knowledge from data is incredibly valuable
  - 5 out of 6 of the biggest companies in the world are "data driven companies"
- To get the most value out of it, data has to be:
  - **Stored**
  - **Managed**
  - **Analyzed**

Big Data Analysis Stack



What Will We Learn?

- To extract knowledge from **different types of data**
  - High-dimensional
  - Unlabeled/Labeled
  - Graph-based
  - Infinite/never-ending streams

What Will We Learn?

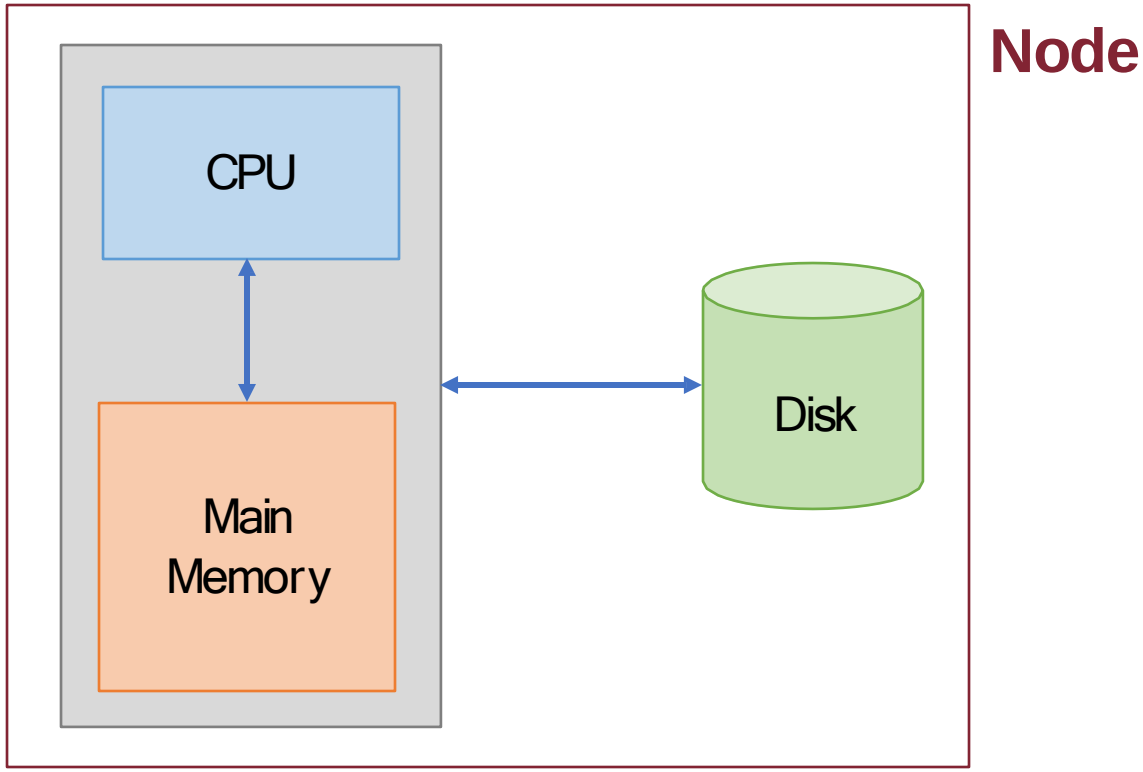
- To use **different models of computation**
  - MapReduce
  - Streams and online algorithms
  - Single machine in-memory

What Will We Learn?

- To apply big data analysis to actually **solve real-world problems**
  - Clustering
  - Predictive Analysis
  - Recommender Systems
  - Graph Analysis
  - Stream Processing
  - ...

The Single-Node Architecture

Everything is ok as long as data fits entirely into main memory
(few accesses to the disk are still tolerated)



Example: Google (Toy) Index

- Google has crawled 50 million web pages (a tiny fraction of the Web!)
- The average size of each web page (HTML only) is ~100 KB
- The total size of the index will be

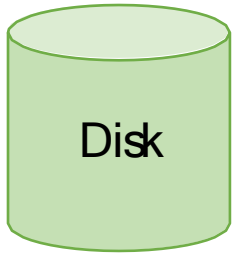
$5x10^7 \times 10^5$ bytes = $5x10^{12}$ bytes = **5 TB**

Main Memory

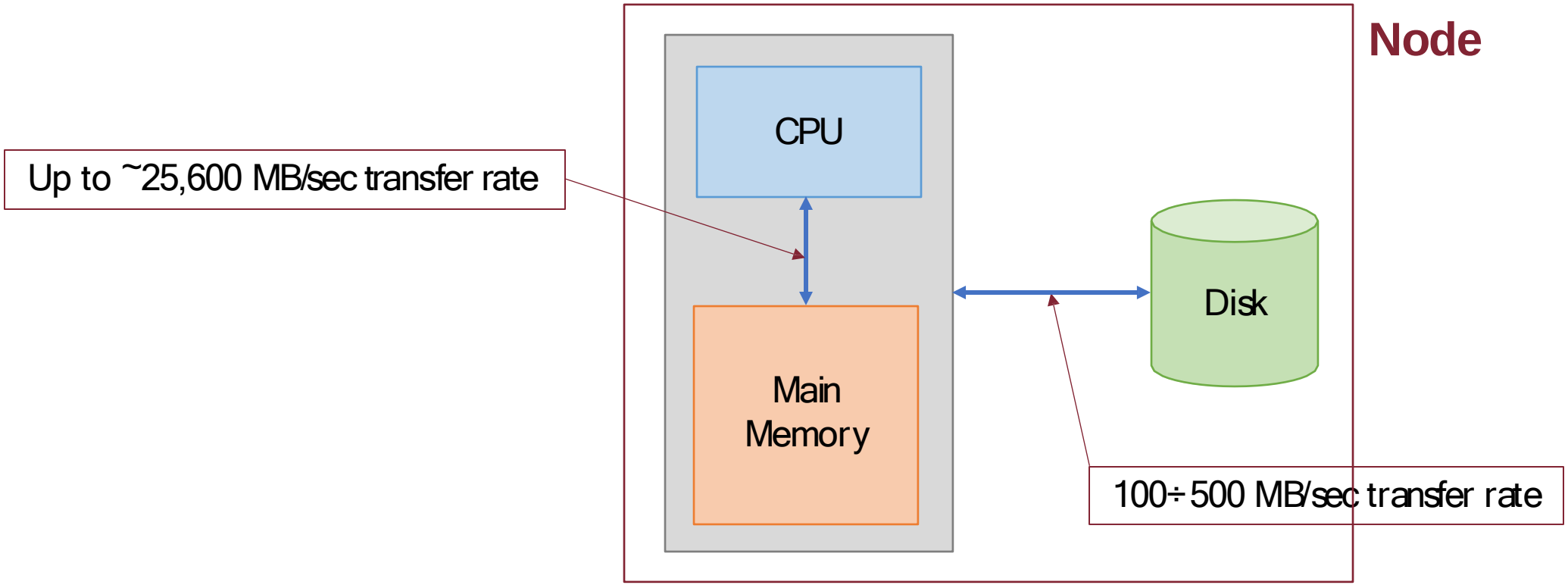
Example: Google (Toy) Index

- Google has crawled 50 million web pages (a tiny fraction of the Web!)
- The average size of each web page (HTML only) is ~100 KB
- The total size of the index will be

$5 \times 10^7 \times 10^5$ bytes = 5×10^{12} bytes = **5 TB**



The Single-Node Architecture



2 orders of magnitude difference between data transfer rate

Example: Google (Toy) Index

- Assuming the disk transfer rate is 100 MB/sec the total time to read the entire index will be:

$5x10^{12}$ bytes/10^{8} bytes/sec = $5x10^{4}$ seconds **~14 hours**

Example: Google (Toy) Index

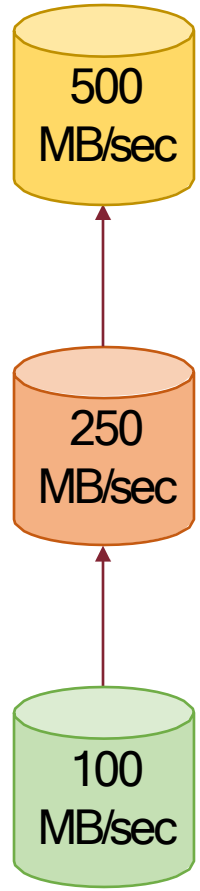
- Assuming the disk transfer rate is 100 MB/sec the total time to read the entire index will be:

$5x10^{12}$ bytes/10^{8} bytes/sec = $5x10^{4}$ seconds **~14 hours**

- More than half a day to just read the index, without even do any computation on it!
- Single-node architecture is clearly not enough here

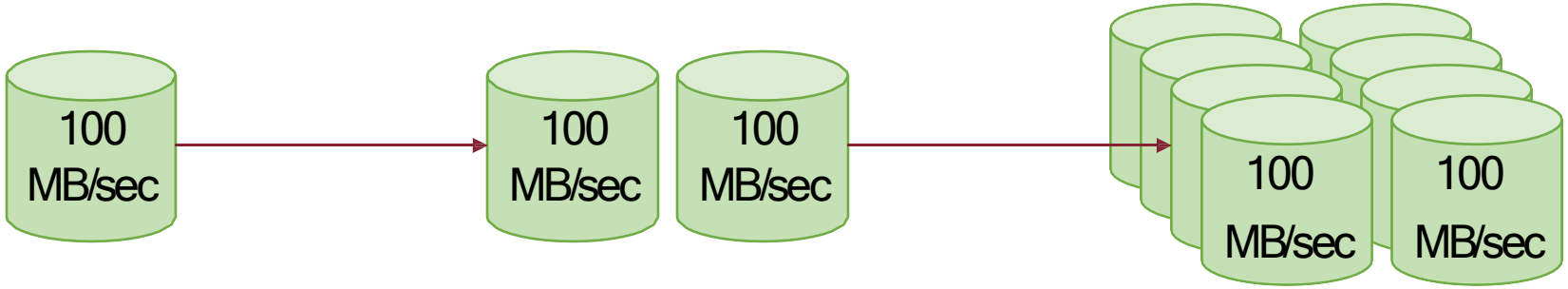
Scaling Up/Vertical Scaling

- Buy a more performing disk (e.g., 250 or 500 MB/sec transfer rate)
- **PRO**
  - Easiest solution
- **CON**
  - Improvement is physically-limited (e.g., 2.5x or 5x)
  - Expensive



Scaling Out/Horizontal Scaling

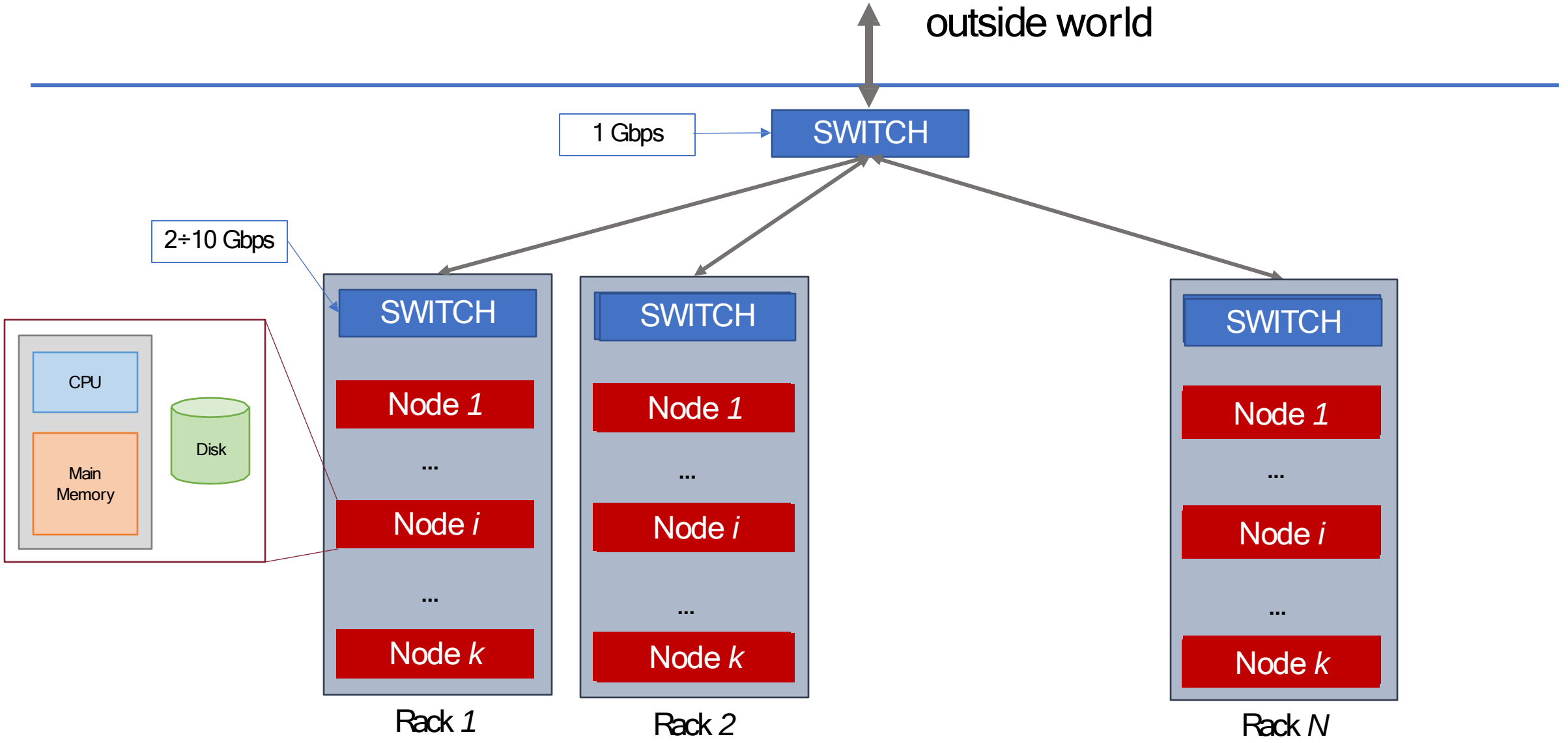
- Buy a set of commodity "cheap" disks and let them work in parallel
- **PRO**
  - Flexibility (improvement is not bound apriori, just add new disks as needed)
- **CON**
  - Extra overhead required to manage parallel work



Cluster Architecture

- Computing architecture based on the **scaling out** principle
- A lot of **commodity nodes** communicating with each other
- Each group of **16÷64 nodes** is arranged in a so-called **rack**
- A **cluster** is made of multiple racks
- Network **switches** enabling node communication
  - 1 Gbps (inter-rack)
  - 2÷10 Gbps (intra-rack)

Cluster Architecture



Cluster Architecture: Challenges

- **3** major **challenges** posed by cluster architecture
  - Ensure **reliability** upon node failure
  - Minimize **network communication** bottleneck
  - Ease **distributed programming** model

Challenge: Node Failure

- Suppose we have a cluster of N nodes
- Each node has a Mean Time To Failure (MTTF) = 3 years ~ 1,000 days

$$p = \text{P}(\text{node}_i \text{ fails}) = 1/1{,}000 = 0.001$$

- Associate with each node a random variable $X_{i,t}$
  - $X_{i,t}$ ~ Bernoulli(p) outputs 1 (failure) with probability $p = 0.001$ and 0 (working) with probability $(1\text{-}p) = 0.999$
  - Assume for semplicity p is the **same** for all nodes and **independent** from each other

Challenge: Node Failure

- A single-node failure on a day may be a quite a rare event (0.1% chance)
- Things are not so infrequent when we deal with several nodes:
  - 1 (expected) failure per day with $N = 1{,}000$ nodes
  - 1,000 (expected) failures per day with $N = 1{,}000{,}000$ nodes

Q1: How to make data and computation resilient to node failures?

Challenge: Network Bottleneck

- Moving data across nodes both intra- and inter racks may be costly
- For example, if we have to transfer 10 TB of data at 1 Gbps

8×10^{13} bits / 1×10^{9} bit/sec = 8×10^{4} secs ~ 1 day

Q2: How to minimize data tranfers so as to reduce network communications?

Challenge: Distributed Programming

- Distributed programming can be really complex
- Programmers should focus on the (distributed) task rather than dealing with the complexities of the cluster architecture

Q3: How to implement algorithms which take advantage of the distributed infrastructure without worrying about its complexities?

Summary

- **Hadoop** provides a framework for distributed data storage and processing.
- **HDFS** is the storage layer within Hadoop.
- **Spark** is a powerful data processing engine that can work with data stored in HDFS but processes it much faster than Hadoop MapReduce due to its in-memory capabilities.

Take-Home Message of Today

- Data is generated at an unprecedented rate -> Big Data
- Extracting knowledge from such big data is incredibly valuable
- Traditional algorithms/techniques often don't scale very well
- There is the need for new "tools" which allow storing, managing, and analyzing big data painlessly