# Introductory seminar on Computer Vision

Sara Concas, Cagliari Digital Lab 2024 - Day 1

sara.concas90c@unica.it

# !!IMPORTANT!!

Friday's class has been moved from the morning to the afternoon

Room:

3:00 p.m.-6:00 p.m.

# About Us

Pattern Recognition
and Applications Lab

Emanuele Ledda

May 2024

Introductory Seminar on Artificial Intelligence and Machine Learning

Daniele Angioni

July 2024

Introductory Seminar on PyTorch for Deep Learning

Sara Concas

September 2024

**Introductory Seminar on Computer Vision**

# About me

- B.Sc. Degree in Electrical and Electronic Engineering from University of Cagliari in November 2018
- M.S. degree in Computer Engineering, Cybersecurity and Artificial Intelligence from the University of Cagliari on February 2021
- PhD student in Electronic and Computer Engineering (DRIEI) at the University of Cagliari since 2021
- Cat lover (As you will understand from the pictures chosen for the examples!)

# About me

Email: sara.concas90c@unica.it
(Also on Teams)

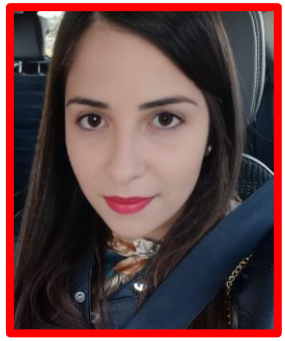
https://sites.unica.it/pralab/people/sara-concas/

PhD students classroom, building M (Next to classroom B0)
(I'm not always there, please send me an email to arrange an appointment!)

# My Research

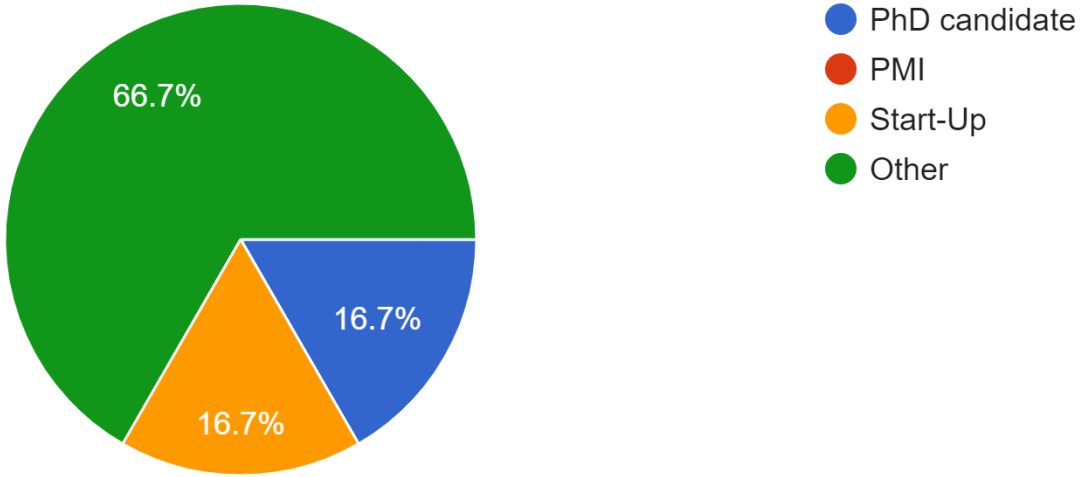
## Deepfake Detection




Publications List

# About You



# Prerequisite

- Basic knowledge **of Python**
- Understanding the basic concepts of **Linear Algebra**
- Basic theoretical knowledge of image processing, algebra (vectors, matrices, etc.), and machine learning techniques.
- Bring a **PC** 

# Seminar Objectives

- **Introduction to Computer Vision**: Understand the basics of computer vision, its applications, and its significance in the field of AI and ML.
- **Digital Image Processing**: Gain knowledge on digital images and how to process them using tools like Numpy and OpenCV.
- **Classification Techniques**: Learn about different classification methods, including supervised and unsupervised learning, binary and multi-label classification for images and videos.
- **Semantic Segmentation**: Explore semantic segmentation techniques using various models such as SVMs, Random Forests, and Deep Learning methods.
- **Object Detection**: Understand the difference between object detection and semantic segmentation, and learn practical techniques like the Viola-Jones algorithm, histogram of oriented gradients, and CNNs.
- **Alignment and Tracking**: Study methods for object and face alignment, as well as single and multi-object tracking in images and videos, and the associated challenges.

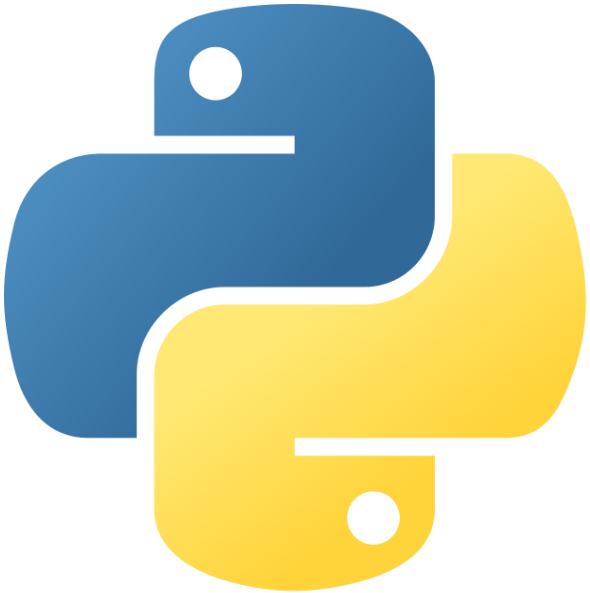
# Technical Python Aspects

- Learning **numpy** essentials for managing tensor manipulation, essential for every ML task
- Basic data management skills with **pandas**
- General knowledge on using **SciKit-Learn** classifiers


# Course language: Python

## Why?

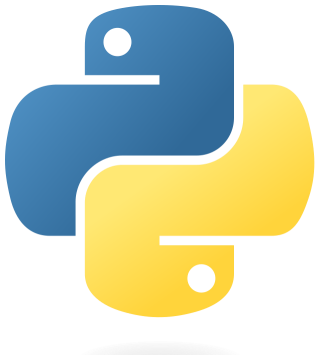
- Major code language for AI and ML
- Great choice of libraries
- Easy to learn & to read
- Flexibility
- Platform independence



https://www.w3schools.com/python/default.asp

# Platform: Google Colaboratory

- Hosted Jupyter Notebook service
- No setup required
- free access to computing resources (GPUs, TPUs)

... You just need a Google account!

https://colab.research.google.com/?hl=en-GB

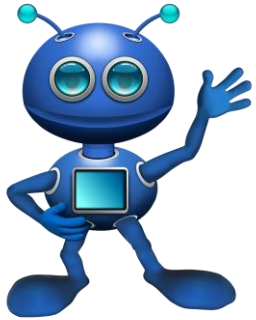
# Python Main Data Analysis Libraries

# Image Processing in Python

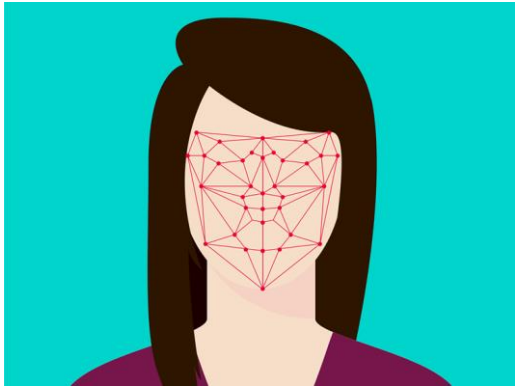
## OpenCV:

- World's biggest Computer Vision library
- Cross-platform library
- Image processing, video capture and analysis including features like face detection and object detection

# Artificial Intelligence (AI)

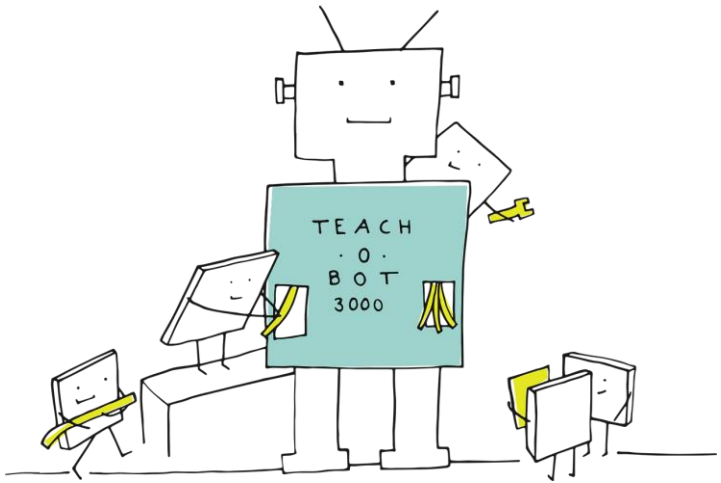


Concept and development of computer systems that can carry out tasks traditionally requiring human intelligence, such as speech recognition, decision-making, and pattern recognition.

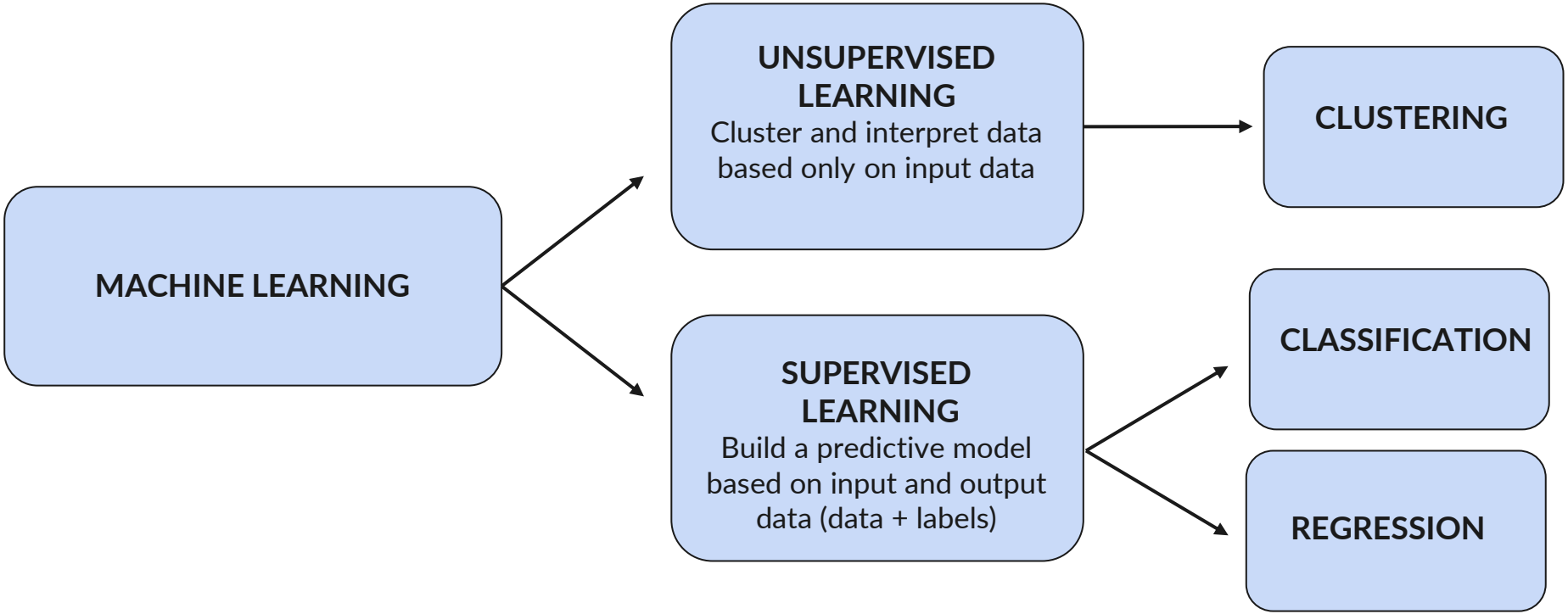



# Machine Learning

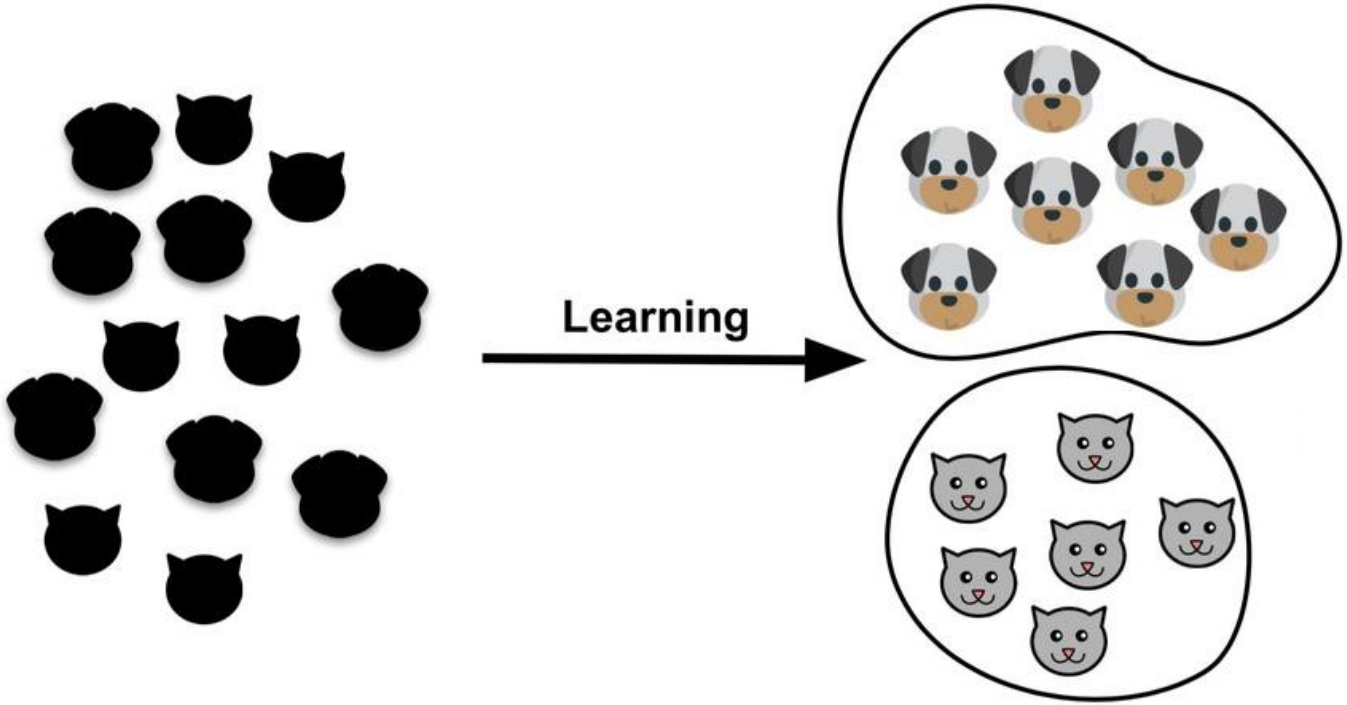
Teaches computers to ***learn from experience***
Learn information from data and adaptively improve the performance as the available samples increase



# Machine Learning Techniques

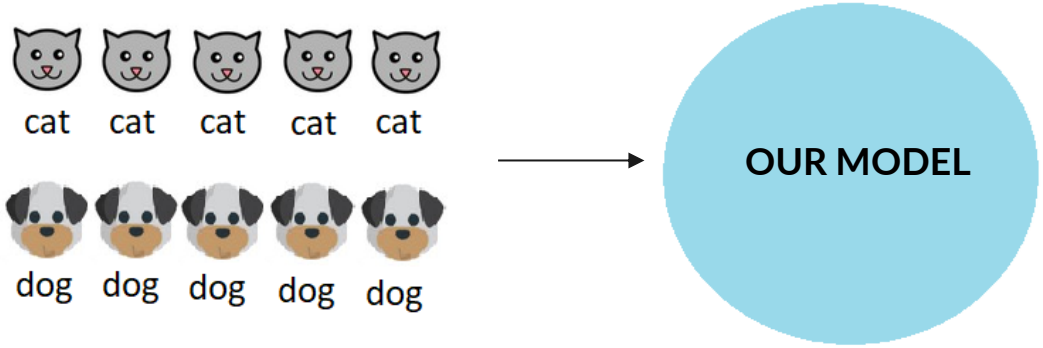


# Unsupervised learning

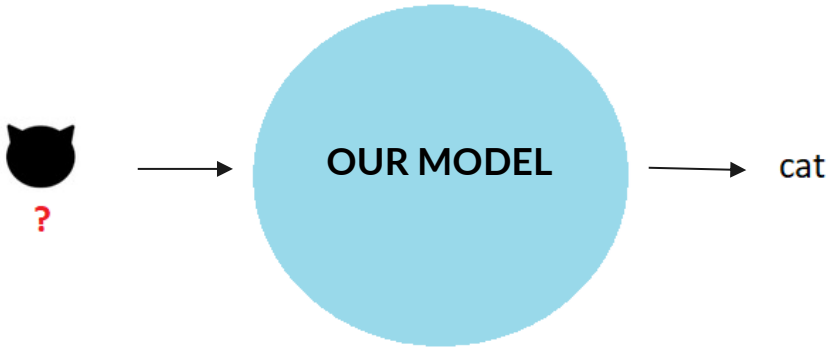


# Supervised learning

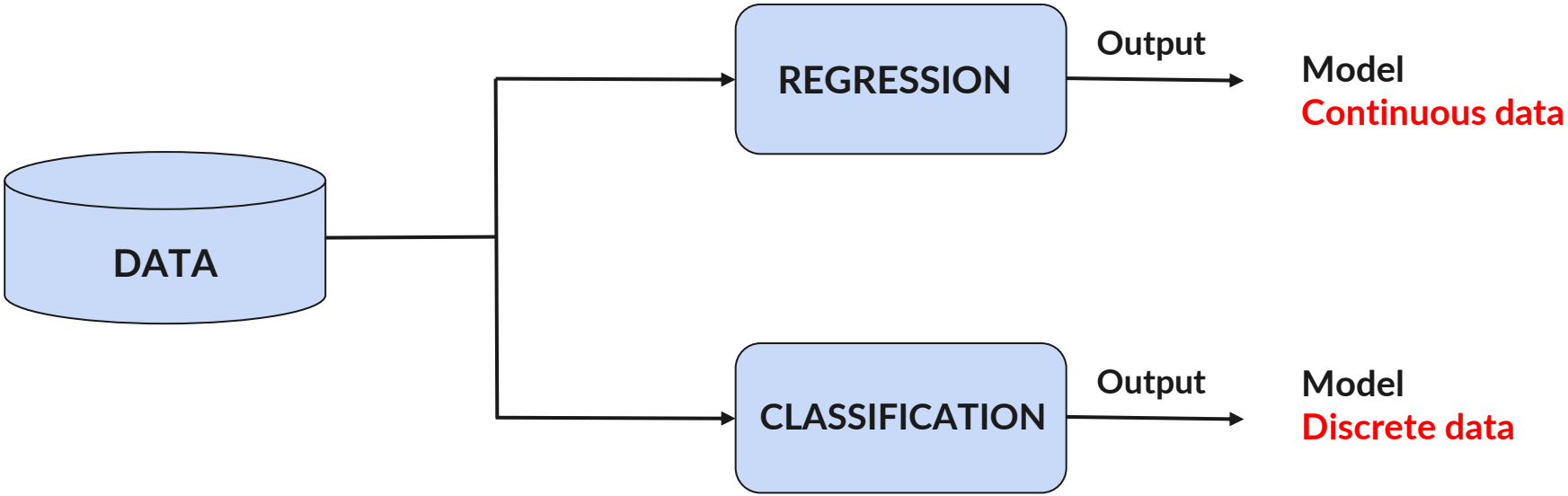
**Training**




**Testing**



# Supervised learning



enjoy algorithms

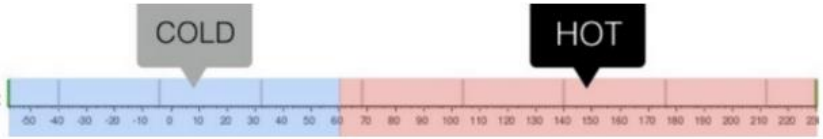
## Regression

What will be the temperature tomorrow?


Fahrenheit

## Classification

Will it be hot or cold tomorrow?




Fahrenheit

https://www.enjoyalgorithms.com/blogs/classification-and-regression-in-machine-learning

# Machine Learning stages

1 - Problem Definition

2 - Data Collection

3 – Data Annotation (Supervised)

4 - Data Preprocessing

5 – Data Splitting

6 - Model Selection

7 - Model Training and Parameter Tuning

8 - Model Evaluation

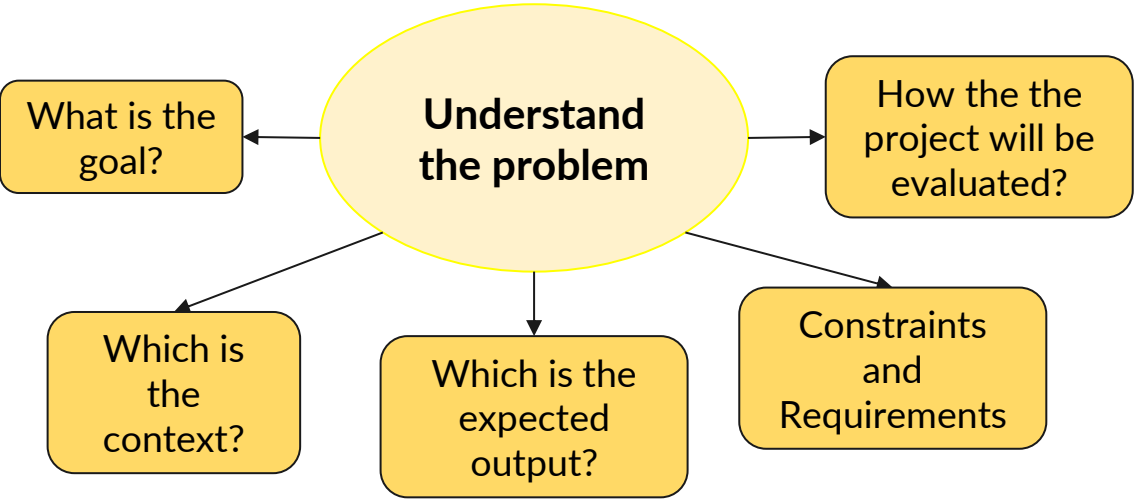
9 - Model Deployment, Monitoring and Maintenance

10 - Documentation and Reporting

# Machine Learning stages - Problem Definition

Initial and crucial stage where the goals and scope of a project are clearly defined.

This step influences all subsequent stages in the machine learning workflow




**Is ML the appropriate tool to solve it?**

# Machine Learning stages - Data Collection

Gathering the relevant data needed to train and evaluate a model

The quality and quantity of data significantly impact the performance of the model

- Identifying Data Sources
- Data Types and Formats
- Data Requirements
- Ensuring Data Quality
- Handling Privacy and Security Concerns
- Sampling
- Data Storage and Management
- Documentation

# Machine Learning stages - Data Annotation

Process of labeling or tagging data to make it understandable and usable for training supervised algorithms.
The annotations provide the ground truth, which the model uses to understand patterns, make predictions, and generalize to new, unseen data.


**Tools and Techniques:**

- Manual Annotation: Human annotators manually label data.
- Automated Annotation: Leveraging pre-existing models or scripts to automatically label data
- Crowdsourcing: Platforms like Amazon Mechanical Turk or Labelbox allow large numbers of annotators to work on datasets simultaneously

# Machine Learning stages - Data Preprocessing

- **Handling Null Values**: deleting or substituting the missing values

| | | | | |
|---|---|---|---|---|
| 1.00 | 2.00 | 3.00 | 14.00 | 5.00 |
| 6.00 | 7.00 | 8.00 | 9.00 | 10.00 |
| 11.00 | nan | 13.00 | 14.00 | 15.00 |
| 16.00 | 17.00 | 18.00 | 19.00 | nan |
| 11.00 | 22.00 | 23.00 | 24.00 | 25.00 |
| 26.00 | 27.00 | nan | 29.00 | 30.00 |
| 31.00 | 32.00 | 33.00 | 34.00 | 35.00 |



| | | | | |
|---|---|---|---|---|
| 1.00 | 2.00 | 3.00 | 14.00 | 5.00 |
| 6.00 | 7.00 | 8.00 | 9.00 | 10.00 |
| 11.00 | 22.00 | 23.00 | 24.00 | 25.00 |
| 31.00 | 32.00 | 33.00 | 34.00 | 35.00 |

Remove the rows containing the null values

| | | | | |
|---|---|---|---|---|
| 1.00 | 2.00 | 3.00 | 14.00 | 5.00 |
| 6.00 | 7.00 | 8.00 | 9.00 | 10.00 |
| 11.00 | 0.00 | 13.00 | 14.00 | 15.00 |
| 16.00 | 17.00 | 18.00 | 19.00 | 0.00 |
| 11.00 | 22.00 | 23.00 | 24.00 | 25.00 |
| 26.00 | 27.00 | 0.00 | 29.00 | 30.00 |
| 31.00 | 32.00 | 33.00 | 34.00 | 35.00 |

Substitute the null values

# Machine Learning stages - Data Preprocessing

- **Normalization/Standardization**: Normalization scales data to a range of 0 to 1, while standardization adjusts data to have a mean of 0 and a standard deviation of 1.

| | | | | |
|---|---|---|---|---|
| 1.00 | 2.00 | 3.00 | 0.00 | 5.00 |
| 6.00 | 7.00 | 8.00 | 9.00 | 10.00 |
| 11.00 | 0.00 | 13.00 | 14.00 | 15.00 |
| 16.00 | 17.00 | 18.00 | 19.00 | 0.00 |
| 0.00 | 22.00 | 23.00 | 24.00 | 25.00 |
| 26.00 | 27.00 | 0.00 | 29.00 | 30.00 |
| 31.00 | 32.00 | 33.00 | 34.00 | 35.00 |

| | | | | |
|---|---|---|---|---|
| 1.00 | 2.00 | 3.00 | 21.50 | 5.00 |
| 6.00 | 7.00 | 8.00 | 9.00 | 10.00 |
| 11.00 | 17.83 | 13.00 | 14.00 | 15.00 |
| 16.00 | 17.00 | 18.00 | 19.00 | 20.00 |
| 15.17 | 22.00 | 23.00 | 24.00 | 25.00 |
| 26.00 | 27.00 | 16.33 | 29.00 | 30.00 |
| 31.00 | 32.00 | 33.00 | 34.00 | 35.00 |

Filling 0 values with column mean

| | | | | |
|---|---|---|---|---|
| 0.00 | 0.00 | 0.00 | 0.50 | 0.00 |
| 0.17 | 0.17 | 0.17 | 0.00 | 0.17 |
| 0.33 | 0.53 | 0.33 | 0.20 | 0.33 |
| 0.50 | 0.50 | 0.50 | 0.40 | 0.50 |
| 0.47 | 0.67 | 0.67 | 0.60 | 0.67 |
| 0.83 | 0.83 | 0.44 | 0.80 | 0.83 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Normalization

| | | | | |
|---|---|---|---|---|
| -1.45 | -1.62 | -1.46 | 0.00 | -1.50 |
| -0.94 | -1.11 | -0.91 | -1.58 | -1.00 |
| -0.43 | 0.00 | -0.37 | -0.95 | -0.50 |
| 0.09 | -0.09 | 0.18 | -0.32 | 0.00 |
| 0.00 | 0.43 | 0.73 | 0.32 | 0.50 |
| 1.11 | 0.94 | 0.00 | 0.95 | 1.00 |
| 1.62 | 1.45 | 1.83 | 1.58 | 1.50 |

Standardization

# Machine Learning stages - Data Preprocessing

- **Encoding Categorical Variables**: non-numeric data needs to be converted into numerical form.

CAT DOG

| Animal ID | Animal Type | Breed |
|---|---|---|
| 1 | cat | Siamese |
| 2 | dog | Labrador |
| 3 | cat | Persian |
| 4 | dog | Beagle |
| 5 | cat | Maine Coon |

| Animal ID | Animal Type | Breed_Siamese | Breed_Labrador | Breed_Persian | Breed_Beagle | Breed_Maine Coon |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 |

# Machine Learning stages - Data Preprocessing

- **Scaling**: adjusts the range of the data features

| House ID | Size (sq ft) | Bedrooms | Price (USD) |
|---|---|---|---|
| 1 | 1200 | 3 | 300,000 |
| 2 | 1500 | 4 | 350,000 |
| 3 | 800 | 2 | 200,000 |
| 4 | 2000 | 5 | 500,000 |
| 5 | 950 | 2 | 225,000 |

| House ID | Size (scaled) | Bedrooms | Price (USD) (scaled) |
|---|---|---|---|
| 1 | 0.33 | 3 | 0.33 |
| 2 | 0.58 | 4 | 0.5 |
| 3 | 0.0 | 2 | 0.0 |
| 4 | 1.0 | 5 | 1.0 |
| 5 | 0.13 | 2 | 0.083 |

# Machine Learning stages - Data Preprocessing

- **Feature Selection and Extraction**: selecting the most relevant features (attributes) for the model or create them from existing ones to better capture the underlying patterns in the data

Features

| Animal ID | Animal Type | Breed | Age (Years) | Weight (kg) | Color | Fur Length | Special Diet (Target) |
|---|---|---|---|---|---|---|---|
| 1 | Cat | Siamese | 2 | 4 | White | Short | No |
| 2 | Dog | Labrador | 5 | 30 | Black | Short | Yes |
| 3 | Cat | Persian | 3 | 5 | Grey | Long | No |
| 4 | Dog | Beagle | 4 | 20 | Brown | Short | Yes |
| 5 | Cat | Maine Coon | 1 | 6 | Orange | Long | No |

Prediction we want from our model

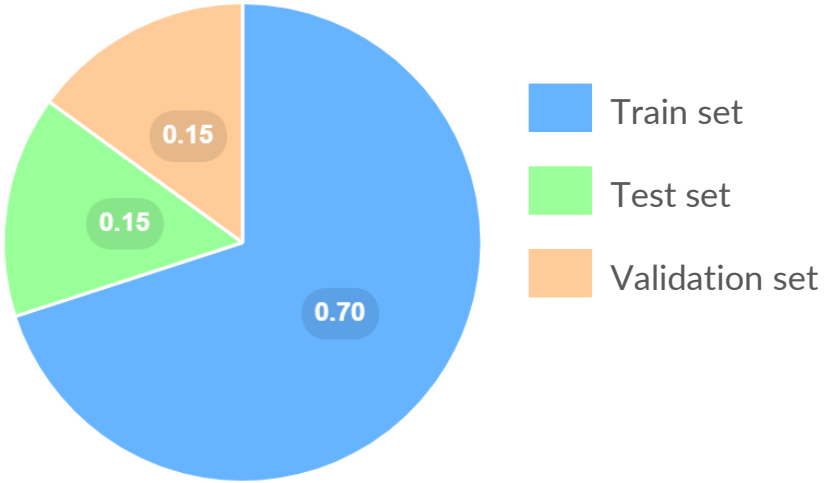
Remove Irrelevant Features

Select Relevant Features

# Machine Learning stages - Data Splitting

- **Data Splitting**: The dataset is typically split into training and testing sets. A common split is 70-80% for training and 20-30% for testing. Additionally, a validation set may be used to tune hyperparameters.

**REMEMBER**: NEVER TEST THE SYSTEM ON THE SAME DATA WITH WHICH IT WAS TRAINED

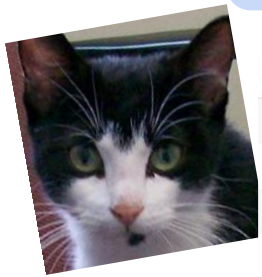


# Machine Learning stages - Model Selection

Find the most appropriate model for the problem at hand

**You can choose a model based on**

Type of data



| EmployeeID | Name | Age | Department | Salary |
|---|---|---|---|---|
| 101 | Alice | 28 | HR | 60000 |
| 102 | Bob | 34 | Finance | 70000 |
| 103 | Charlie | 29 | IT | 65000 |
| 104 | David | 42 | Marketing | 80000 |

Hello, my name is Sara!

Task

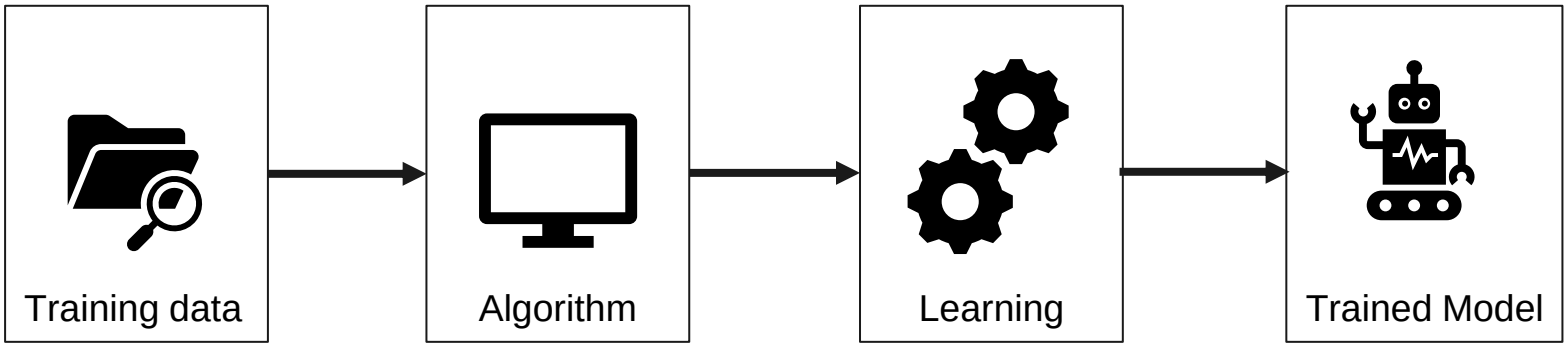
Classification

Regression

Clustering

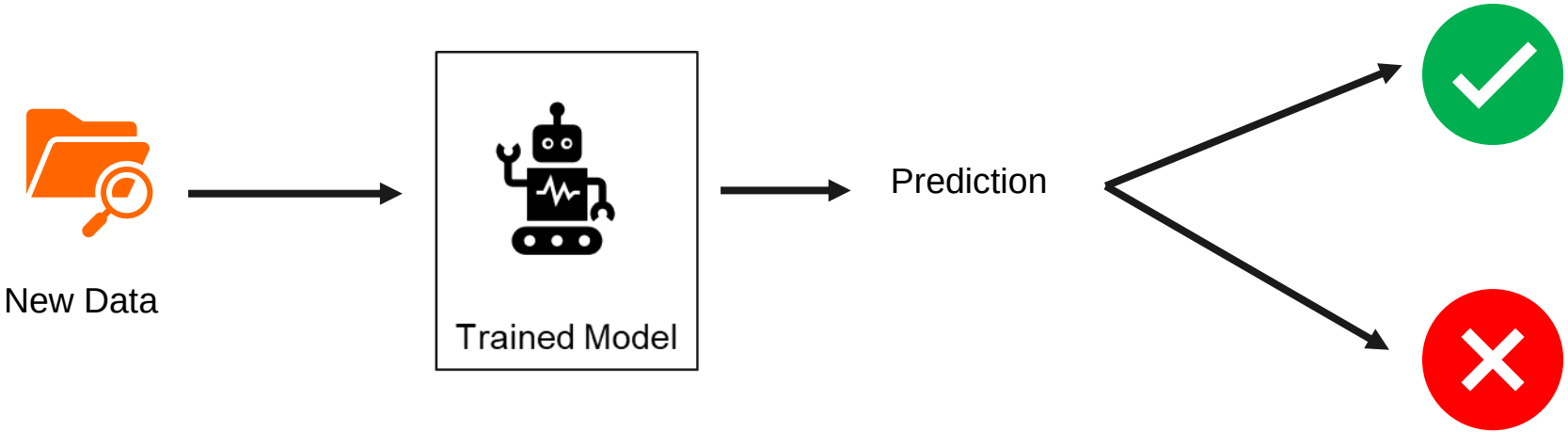
# Machine Learning stages - Model Training

The algorithm uses the data to adjust its internal parameters to minimize the difference between the predicted and actual outcomes.



# Machine Learning stages - Model Evaluation

Assessing the performance of a trained model using a set of criteria to determine how well it generalizes to new, unseen data



# Machine Learning stages - Model Deployment, Monitoring and Maintenance

- **Model Deployment**: process of integrating a trained machine learning model into a production environment where it can provide predictions or insights on new data in real-time

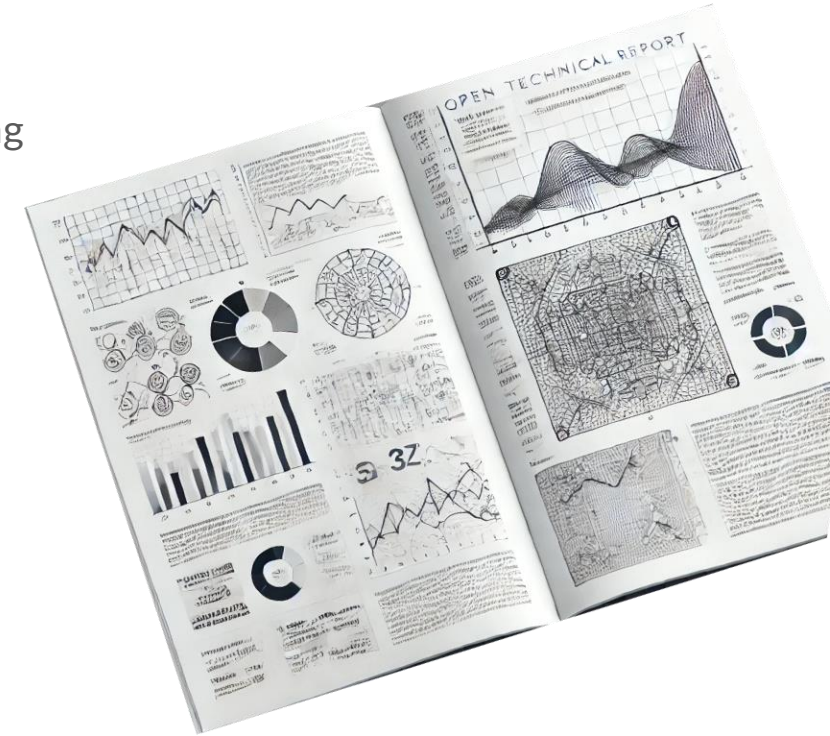
- **Model Monitoring**: continuously tracking the performance and behavior of a deployed model to ensure it remains accurate, reliable, and effective over time

- **Model Maintenance**: ongoing process of managing, updating, and optimizing machine learning models after they have been deployed in a production environment

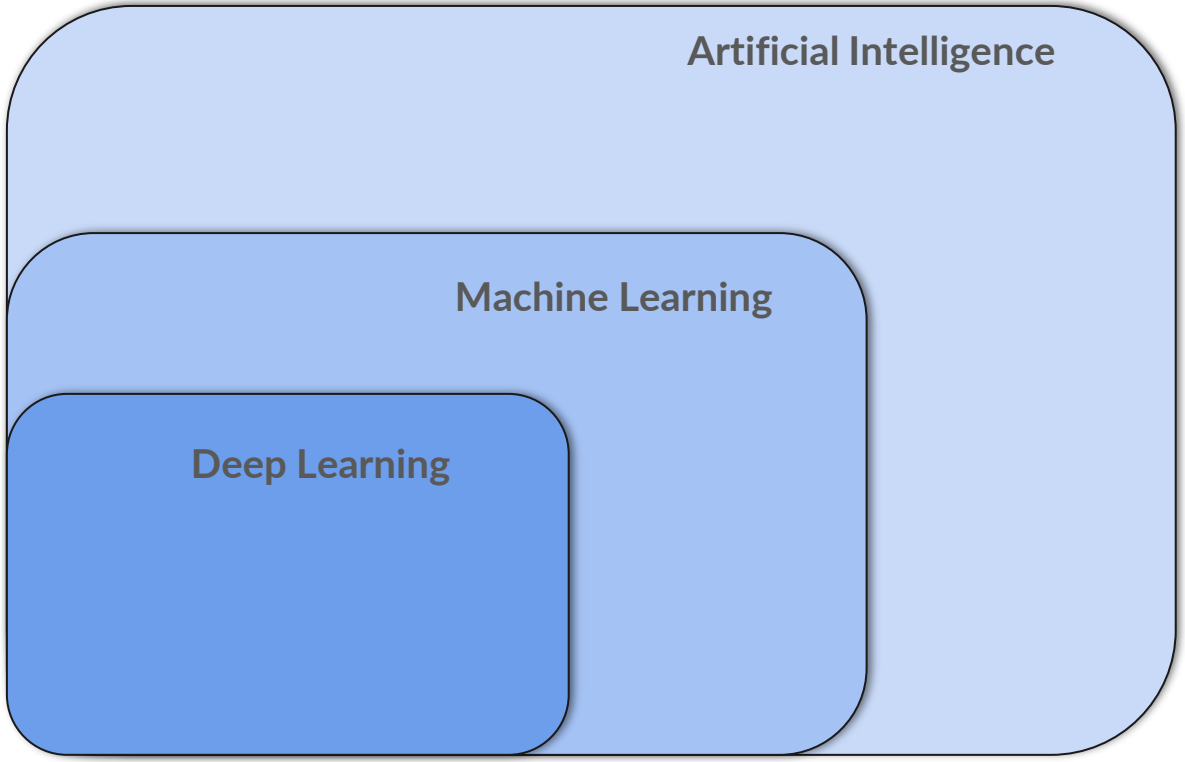
# Machine Learning stages - Documentation and Reporting

The documentation should include:

- **Executive Summary**: an overview of the project, including objectives, approach, and key findings (without technical details)
- **Technical Summary**:  technical aspects of the project, including the models used, their performance, and any challenges encountered
- **Data Analysis Report**
- **Model Performance Report**
- **Error Analysis**
- **Reproducibility Report**
- **Others**: Bias and Fairness Analysis, Privacy and Security, Ethical Considerations



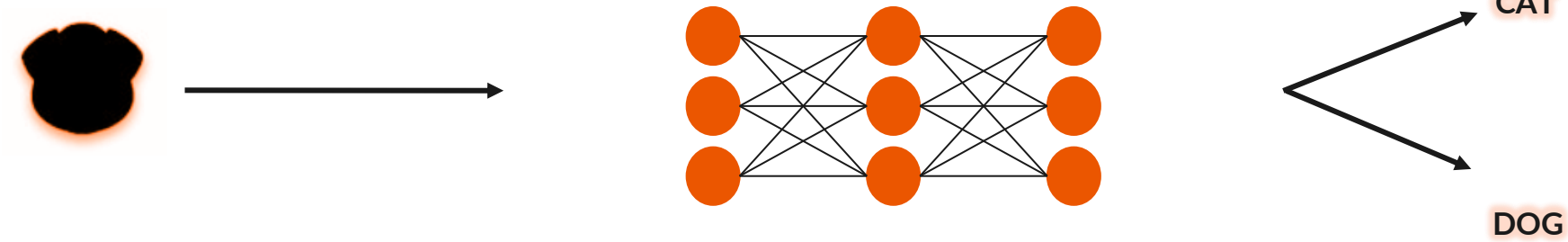
# Machine Learning vs Deep Learning



## Machine Learning


Input | Feature extraction | Classification | Output

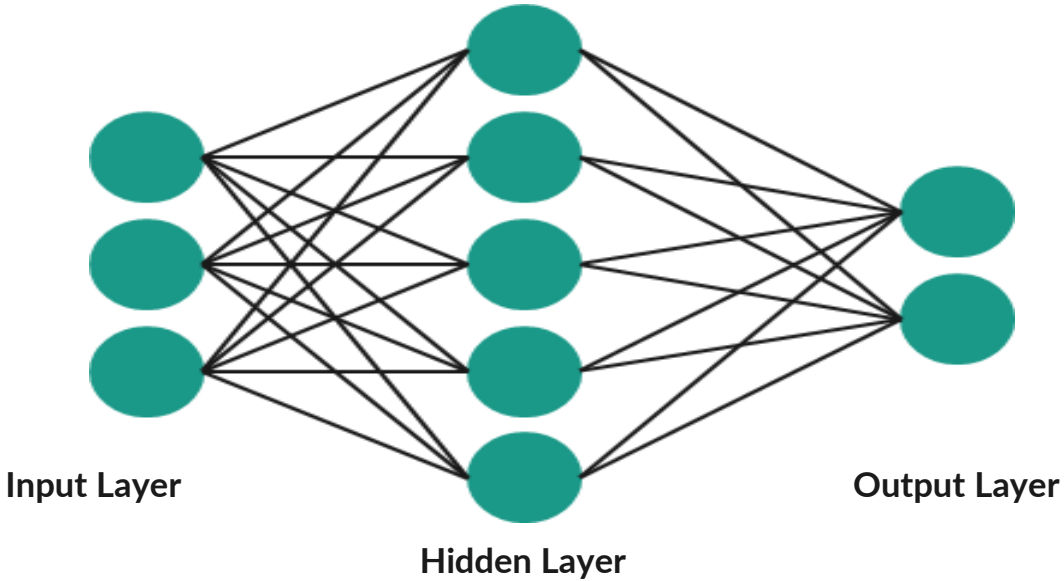
## Deep Learning




Input | Feature extraction + Classification | Output

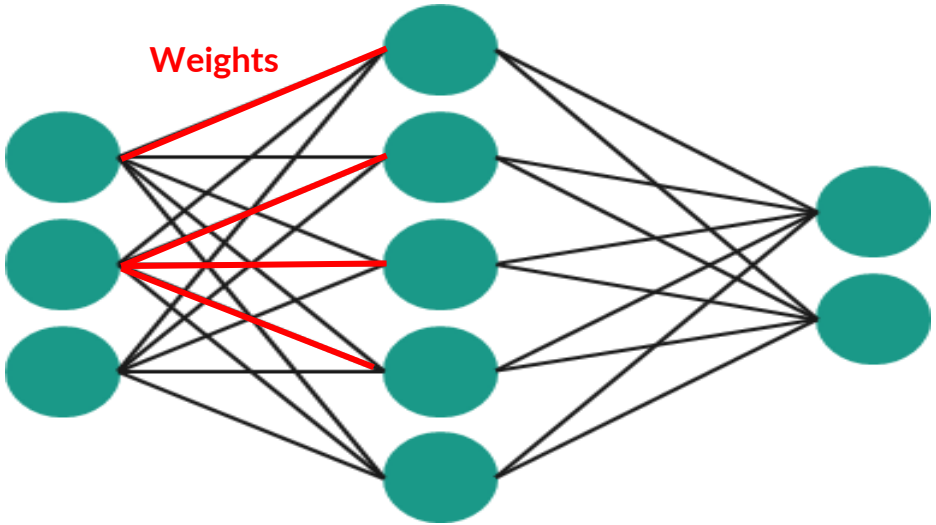
# Neural Networks

Type of machine learning model inspired by the structure and functioning of the human brain. They consist of interconnected layers of nodes, or "neurons," that process data in a way that allows the network to learn and make decisions.



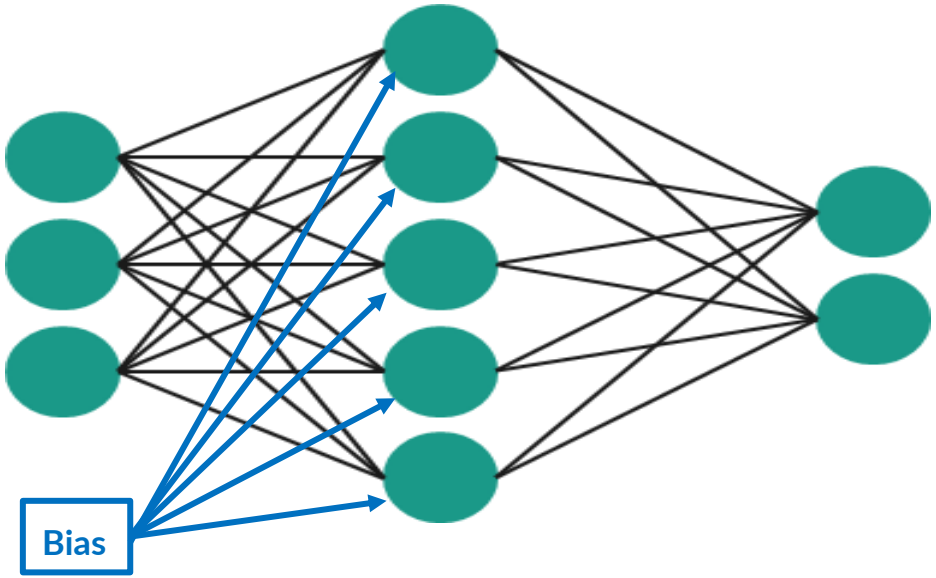
# Neural Networks

Weights are the parameters that determine the strength of the connection between neurons.



# Neural Networks

Biases are additional parameters in the model that help the network fit the data better.



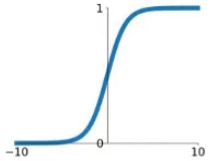
# Neural Networks

An activation function is applied to introduce non-linearity into the model.

## Activation Functions

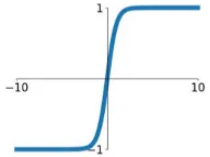
**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



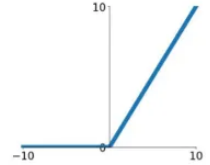

**tanh**

$$\tanh(x)$$



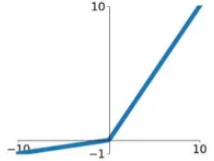

**ReLU**

$$\max(0, x)$$




**Leaky ReLU**

$$\max(0.1x, x)$$

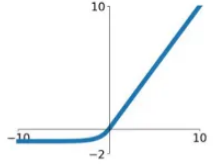



**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

**ELU**

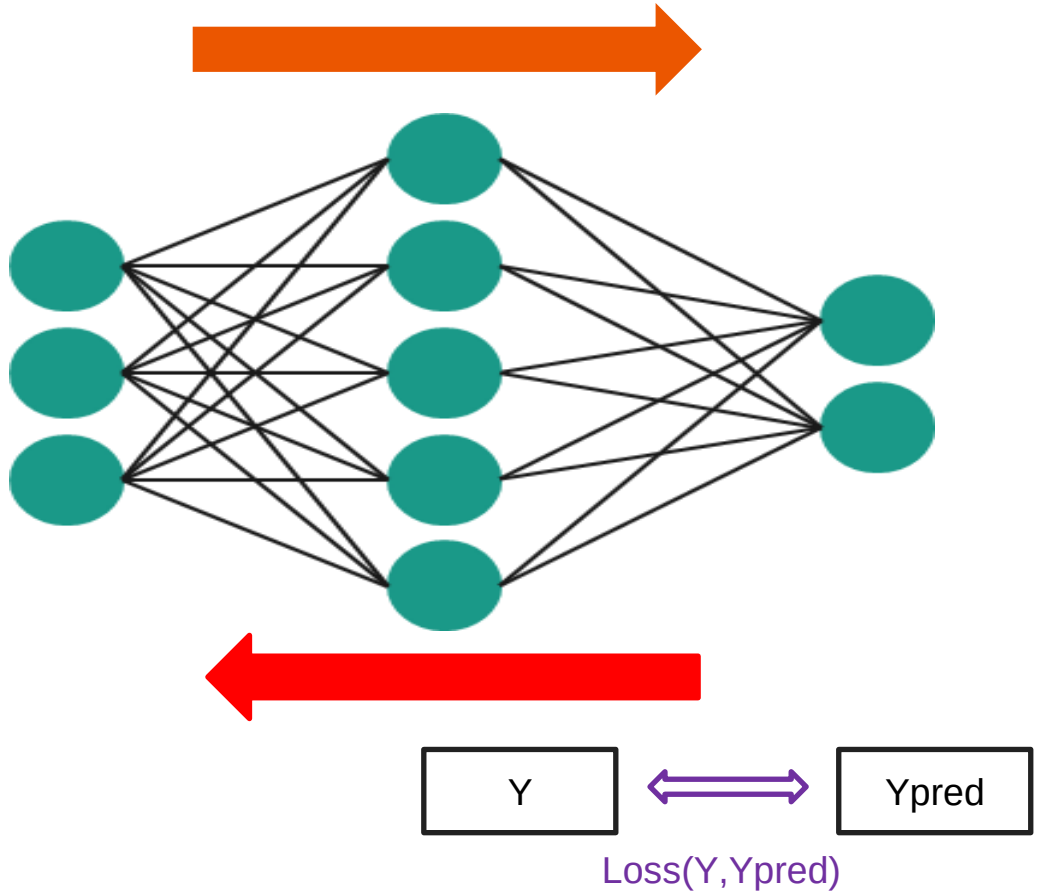
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$




https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092

# Neural Networks

- **Forward Propagation**: Data moves from the input layer, through the hidden layers, to the output layer.
- **Loss Function**: The difference between the network's prediction and the actual result is calculated using a loss function.
- **Backpropagation**: The network uses the loss to adjust the weights and biases by propagating the error backward through the network, updating the parameters to reduce the loss.



# Digital Images

A real scene is transformed into a digital image through a digitization process.

By formulating the brightness distribution of an image as a mathematical representation:

- from f(x,y) in the continuous space (real image)
- to f(m,n) in the discrete space

with M rows and N columns

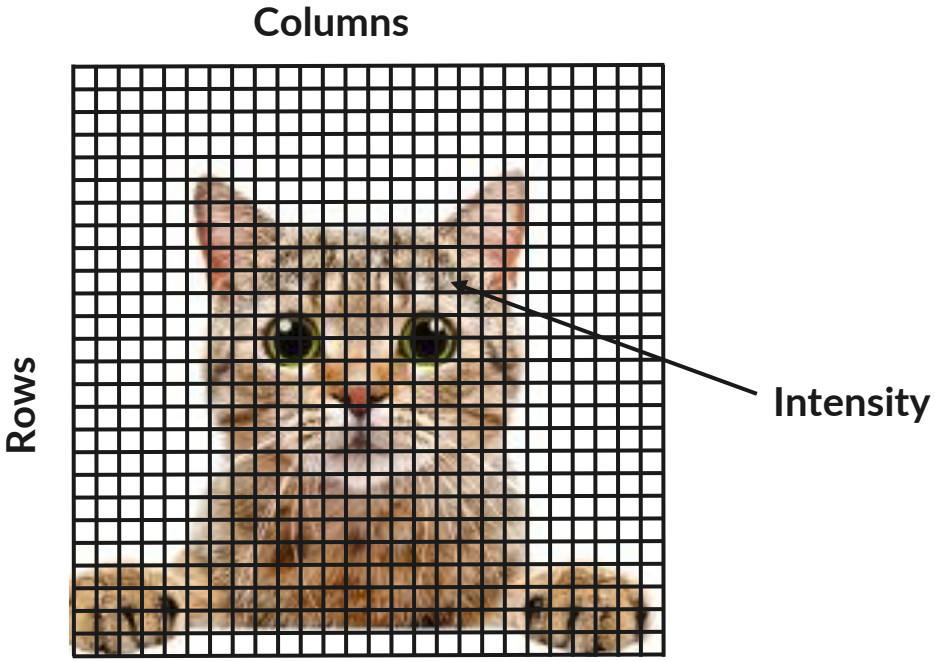
- {m=1,2,3,….,M}
- {n=1,2,3,….,N}

# Digital Images

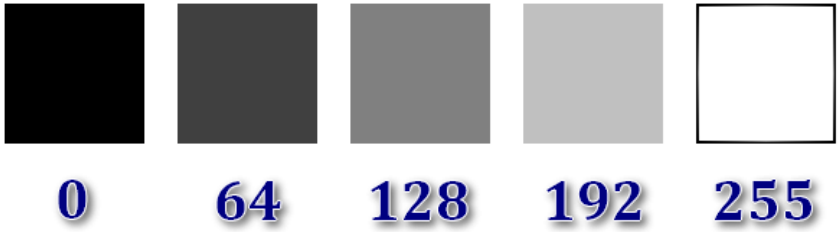
with M rows and N columns

- {m=1,2,3,….,M}
- {n=1,2,3,….,N}

- f(m,n) indicates the intensity, i.e. the value of the image in those specific coordinates

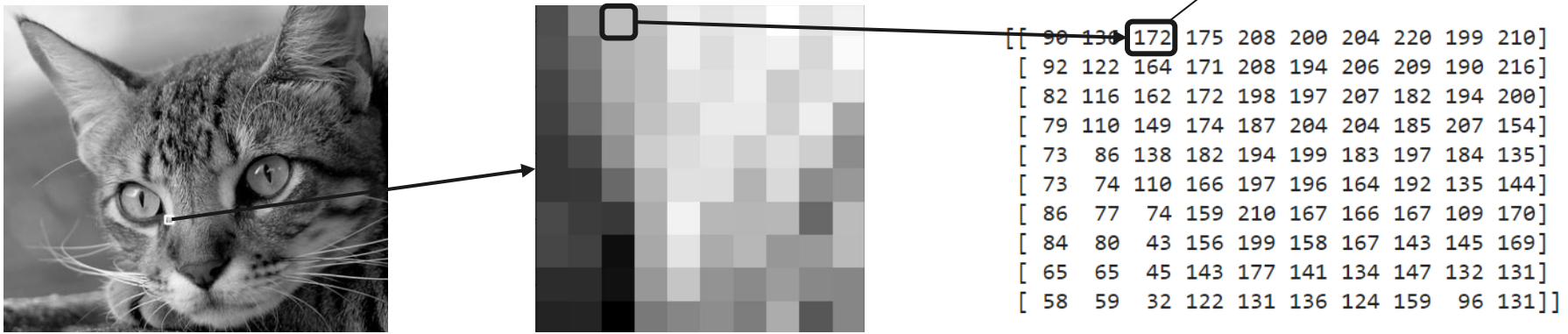


# Digital Images




In grayscale images, the intensity levels can vary depending on the type of image

- typically they are 256 with values between 0 and 255



# Digital images

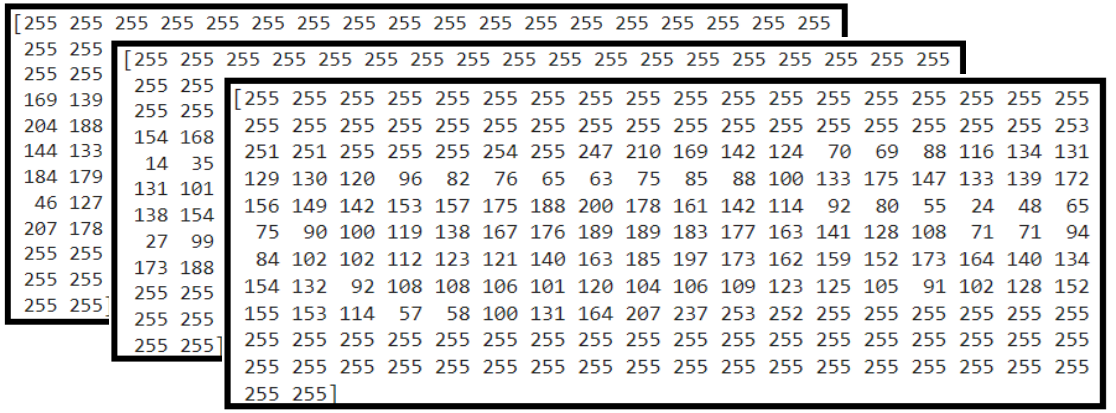
- Colors are represented as a sum of 3 **primary colors**: **Red**, **Green**, and **Blue** (RGB)
- Monitors and televisions work this way
- **256 levels** (**8 bits**) for each channel (i.e. primary color), thus **3 bytes** for each pixel

$2^{3*8}$ colors = $2^{24}$ colors = **16.777.216** colors

| Color | R | G | B |
|---|---|---|---|
| Black | 0 | 0 | 0 |
| White | 255 | 255 | 255 |
| Red | 255 | 0 | 0 |
| Yellow | 255 | 255 | 0 |
| Gray | 127 | 127 | 127 |

# Digital Images

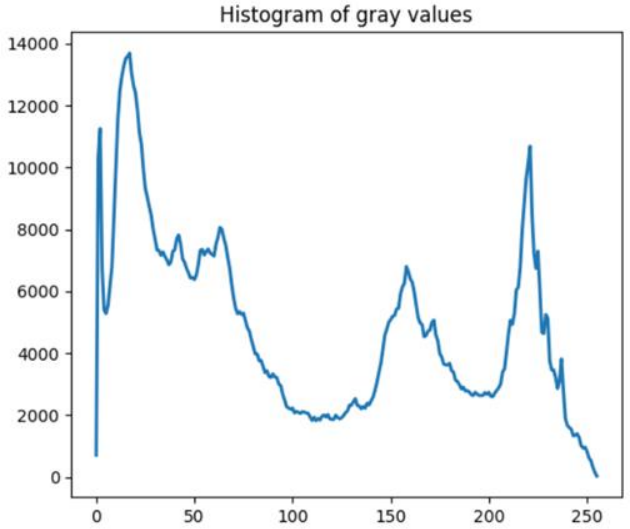
BLUE GREEN RED



# Histogram of an image

- The histogram of a grayscale digital image provides
  - frequency $n_k$ of a certain intensity value k within the image
  - therefore indicates the number of pixels for each level of gray





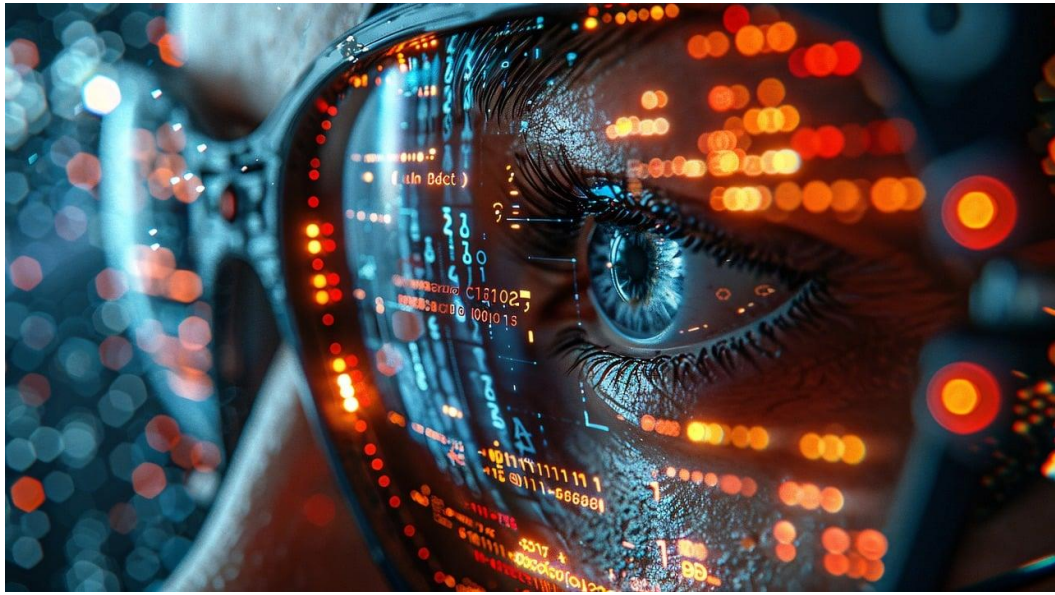
# Tutorial1 - Recalls

https://bit.ly/3VDBRRV

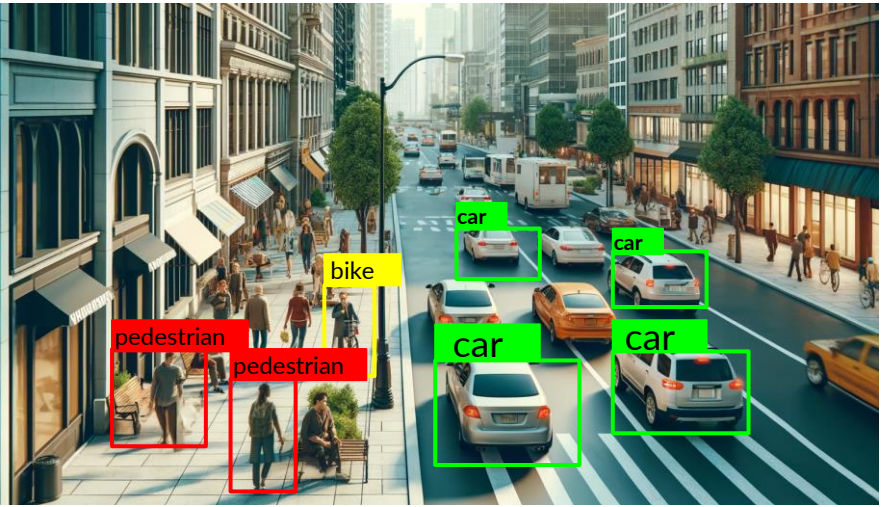
# Computer Vision

Field of artificial intelligence that enables computers to interpret and understand visual information from the world, such as images and videos.

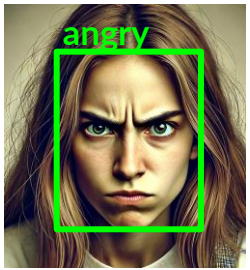
- automate tasks that the human visual system can do



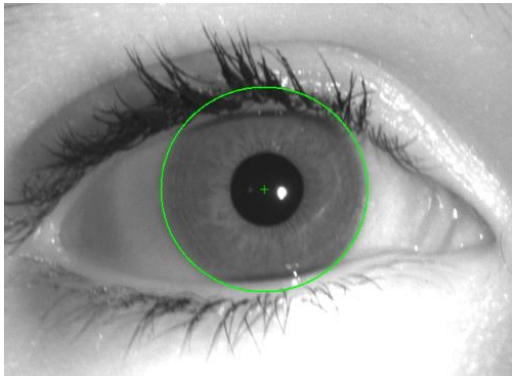
# Computer Vision - Examples



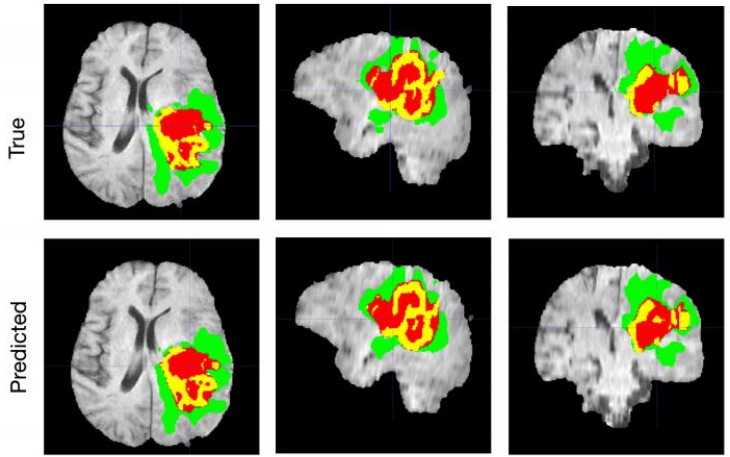

Object recognition




Facial emotion recognition



Biometrics



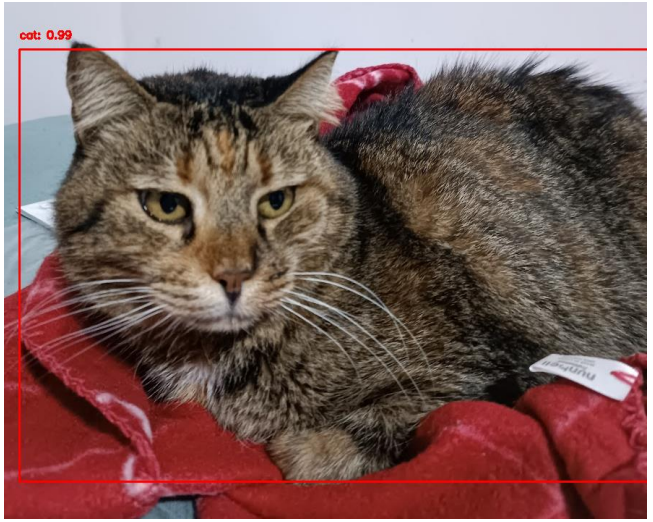

Tumor segmentation.
Source: Nvidia.Developer

# Computer Vision Example "You Only Look Once" (YOLO)

Real-time object detection algorithm that is used to identify and classify objects within an image or a video frame

End-to-end neural network that makes predictions of bounding boxes and class probabilities all at once

YOLO divides the image into a grid and predicts bounding boxes and probabilities for each grid cell.



# Computer Vision Example - "You Only Look Once"（YOLO)

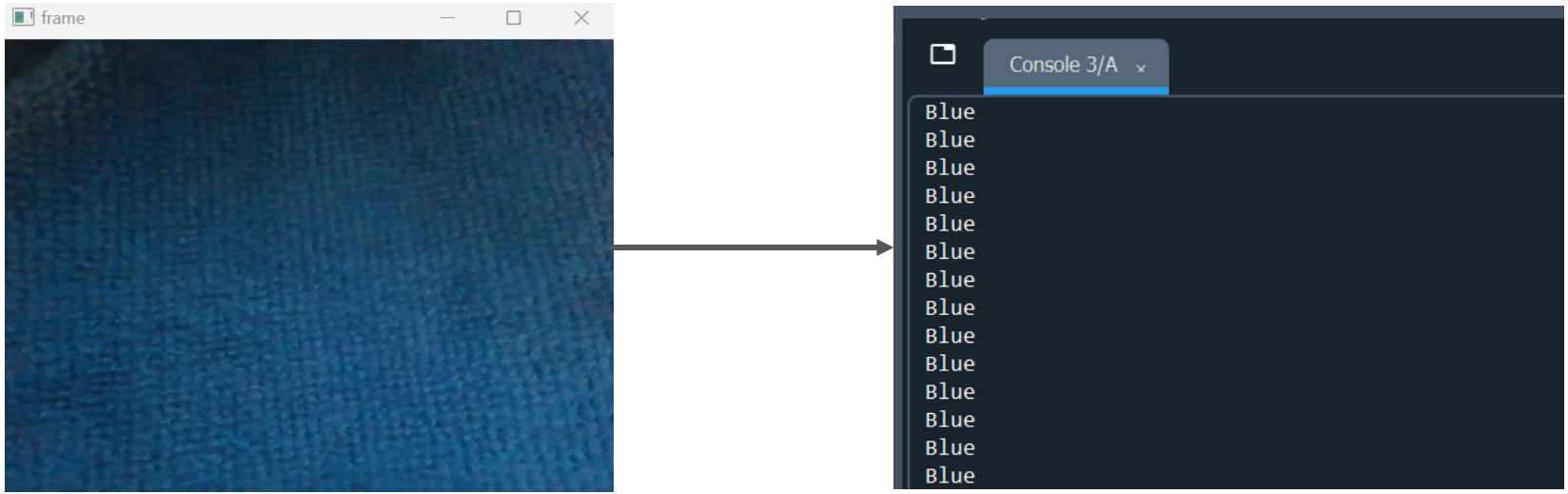
https://bit.ly/3WEBZ41

If you want to know more YOLOv8

# Computer Vision Example - Age Detection


https://www.geeksforgeeks.org/age-detection-using-deep-learning-in-opencv/

# Computer Vision Example -Detect the RGB color from a webcam




https://www.geeksforgeeks.org/detect-the-rgb-color-from-a-webcam-using-python-opencv/