On-device AI / AIoT

EEL 4745C: Microprocessor Applications II

Fall 2022

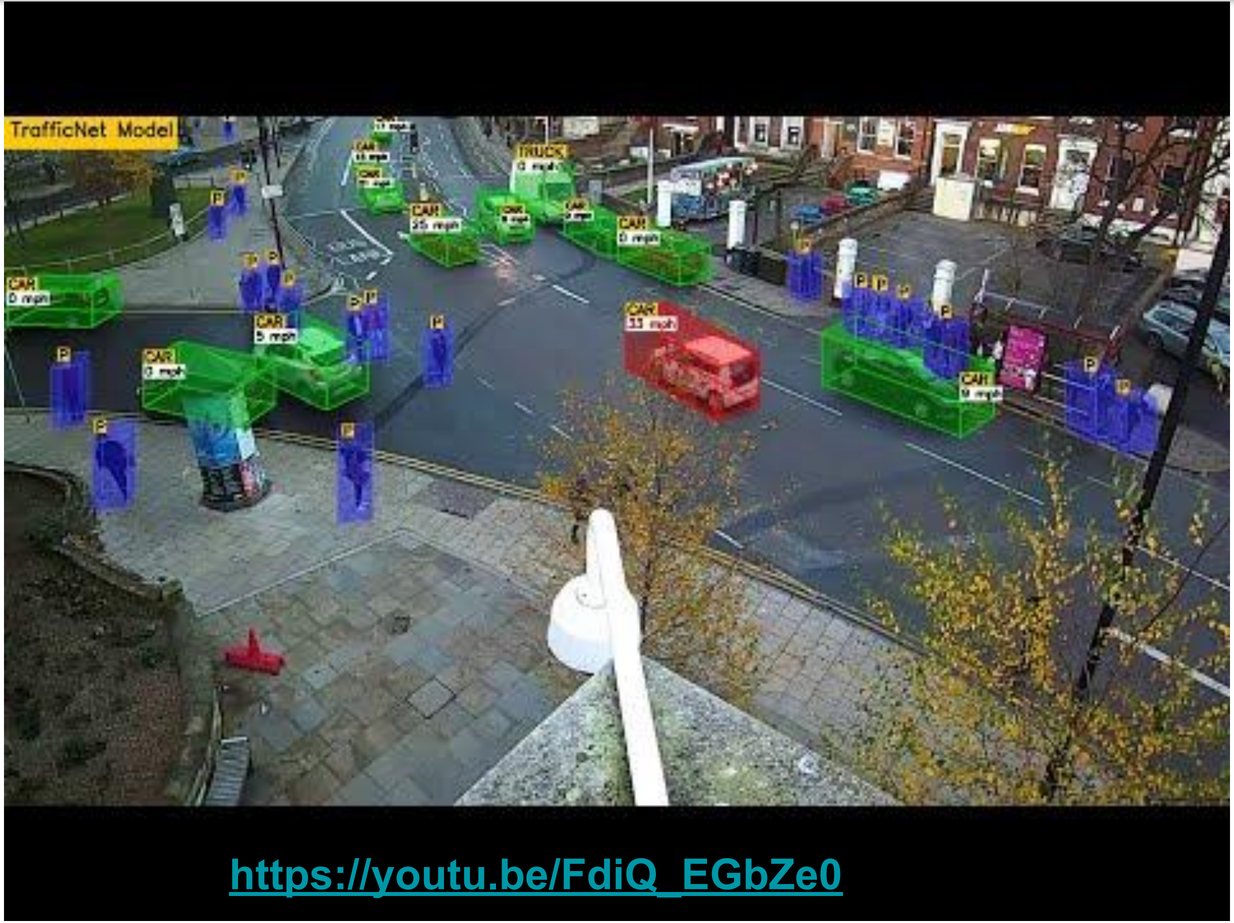
Md Jahidul Islam

Lecture 8

ECE | Florida
Electrical & Computer Engineering

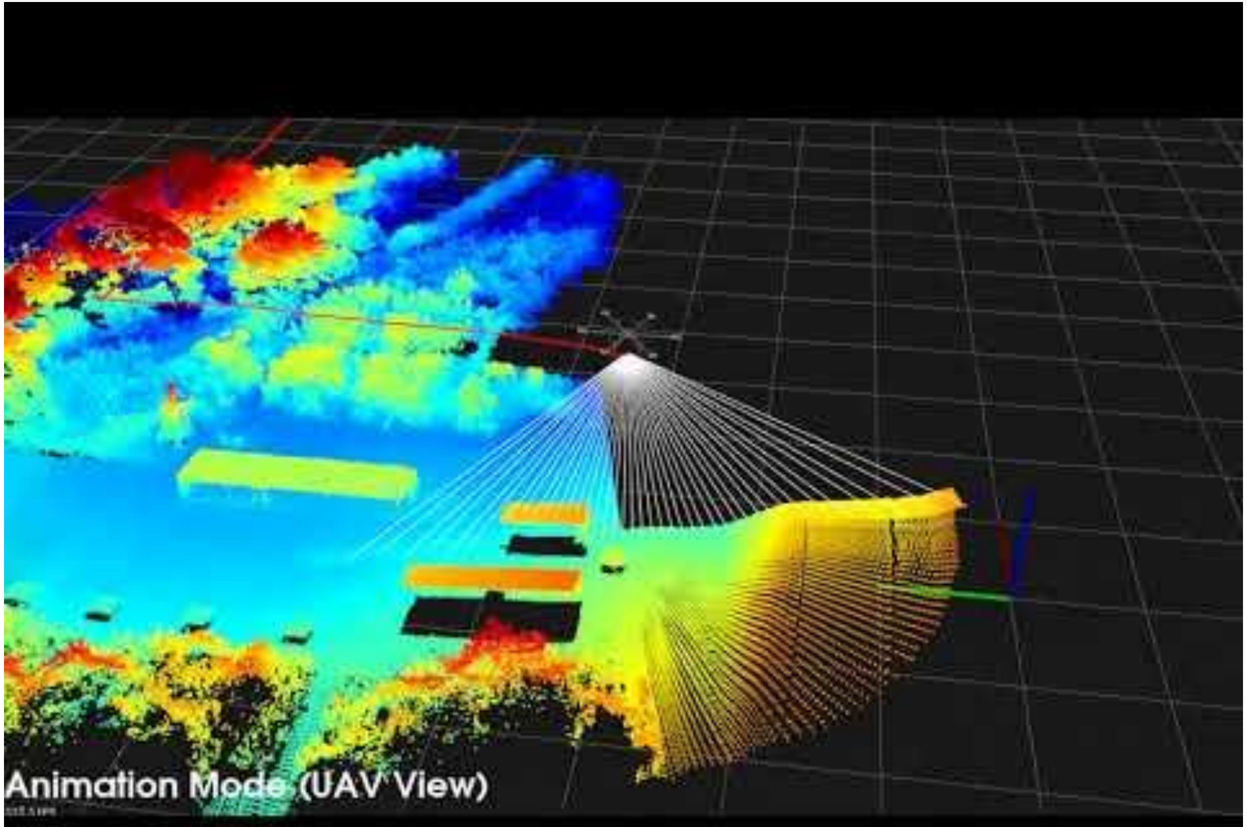
UF | UNIVERSITY of FLORIDA

On-Device AI: Traffic Monitoring



https://youtu.be/FdiQ_EGbZe0

On-Device AI: Aerial Mapping



https://youtu.be/cbczfgH1x0s

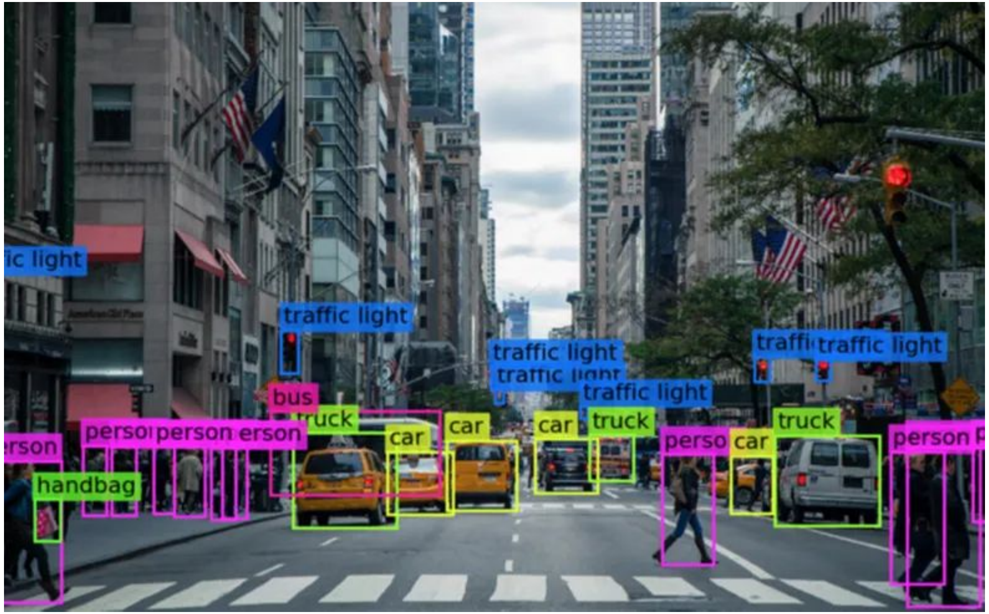
On-Device AI: Underwater Robot Convoying

https://youtu.be/Em7V-vBApHc

Use Case: Robot Vision

- **Robot vision / machine vision requirements**
  - Real-time sensing
  - Low-power hardware (RTOS)
  - Fast on-device operation
    - Object detection, tracking
    - Often actions based on perception
    - Online learning and feedback

- Examples:
  - Autonomous cars/drones/robots
  - Security cameras
  - Cellphones!
  - Smart AIoT devices



Typical WorkFlow

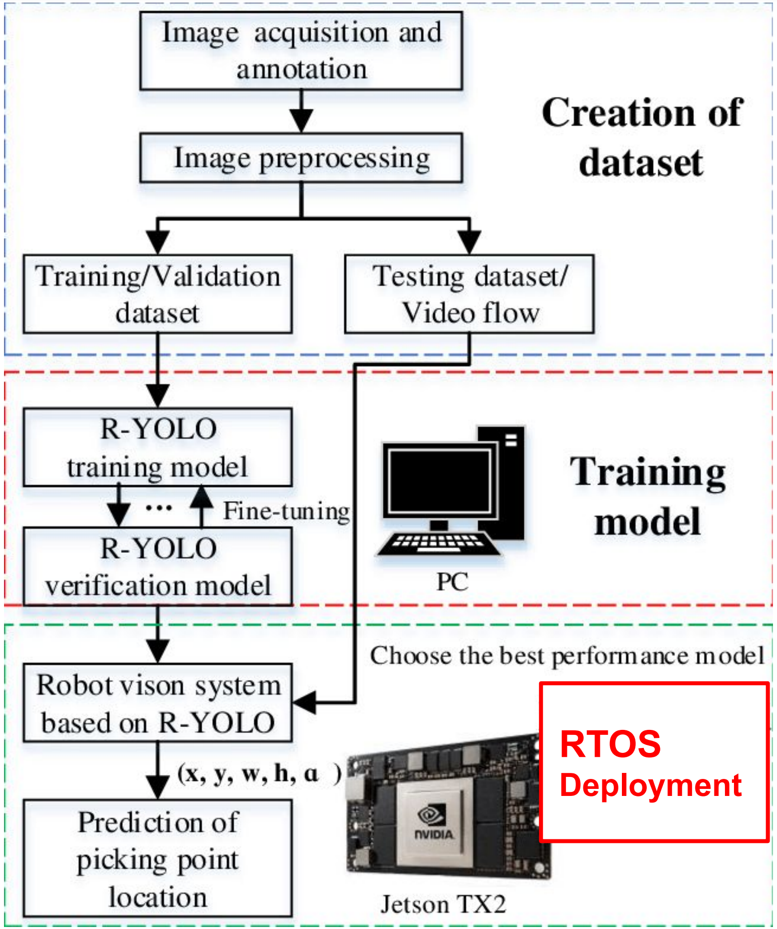


Image acquisition and annotation

Image preprocessing

Training/Validation dataset

Testing dataset/ Video flow

Creation of dataset

R-YOLO training model

… Fine-tuning

R-YOLO verification model

PC

Training model

Choose the best performance model

Robot vison system based on R-YOLO

(x, y, w, h, α)

Prediction of picking point location

Jetson TX2

RTOS Deployment

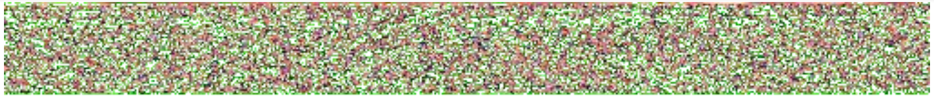
Our RTOS Experiments

- **Low-power real-time face detection on video data**
  - Input: video feed
  - Output: bounding box drawn on the face(s)
  - Hardware:
    - Beaglebone Black
    - See Lab-5 manual and code templates
  - Face detection model:
    - Haar-cascade detection model
    - See this tutorial

- Why use *cascade detection* model?
  - Very fast operation on embedded systems
  - Good choice for low number of object categories

Face Detection: Viola-Jones

- **The famous Viola-Jones Algorithm**
  - Works with frontal face images with visible
    - Eyes and eyebrows, nose, and lips.
    - Symmetry and standard positioning of facial features.
  - Uses Haar features (see this)
  - Calculates features on every pixel with different window sizes
  - Then it finds the best features using Adaptive Boosting (Adaboost) an ML algorithm. See this for more information.
  - Then uses a cascade of classifiers to identify the presence of each features.
  - The accumulated scores gives the final result.



https://medium.datadriveninvestor.com/how-the-facial-detection-algorithms-work-viola-jones-algorithm-and-opencv-bd694936512f

Face Detection in Images

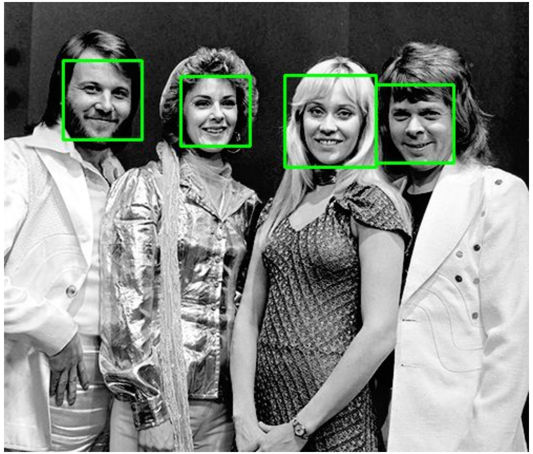
- Read an image (grayscale mode) file given the path

  -
    ```
    image = cv2.imread(img_path, 0) #grayscale-mode
    ```
- Load the cascade classifier model

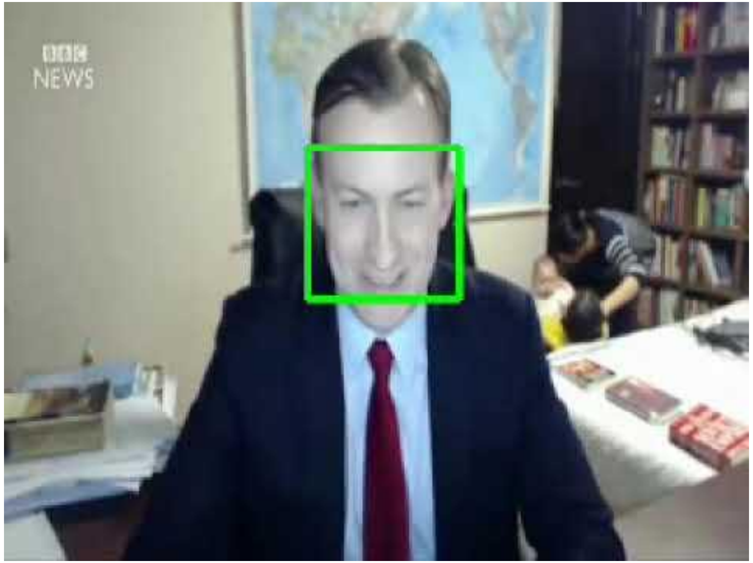
  -
    ```
    faceCascade = cv2.CascadeClassifier(cascade_path)
    ```
- Detect faces

  -
    ```
    faces = faceCascade.detectMultiScale(image,
                                         scaleFactor=1.1,
                                         minNeighbors=5,
                                         minSize=(30, 30)
                                         )
    ```
- Detect bounding boxes on the image

  -
    ```
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h),
                      color = (0, 255, 0),
                      thickness = 2
                      )
    ```



Face Detection in Video



https://youtu.be/prSROEIjuaY

https://youtu.be/IC_y8wD7F3Y

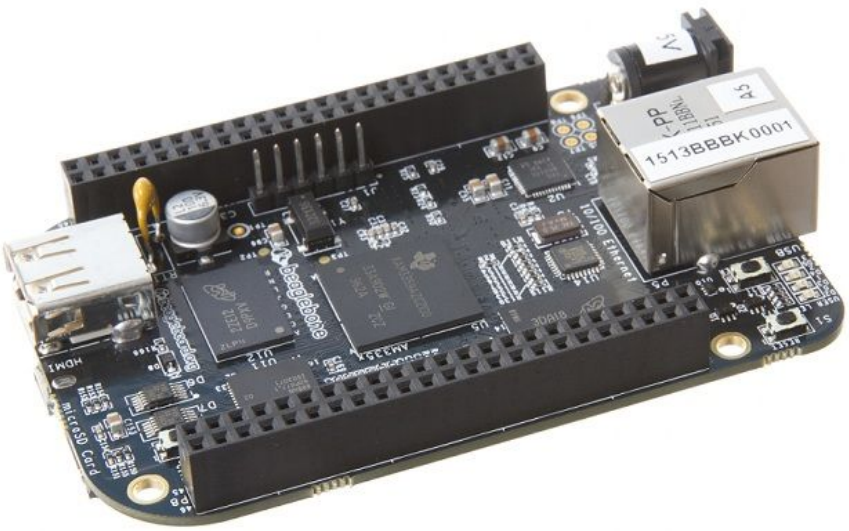
Lab 5: AI/AIoT with RTOS

Getting started with Beagleboards

- Part A: Interfacing and communication with a BeagleBoard
- Part B: Setting up Python-OpenCV pipeline
- Part C: Face detection in RTOS

Part A: Getting Started With a Beagle

Open your Beagle box



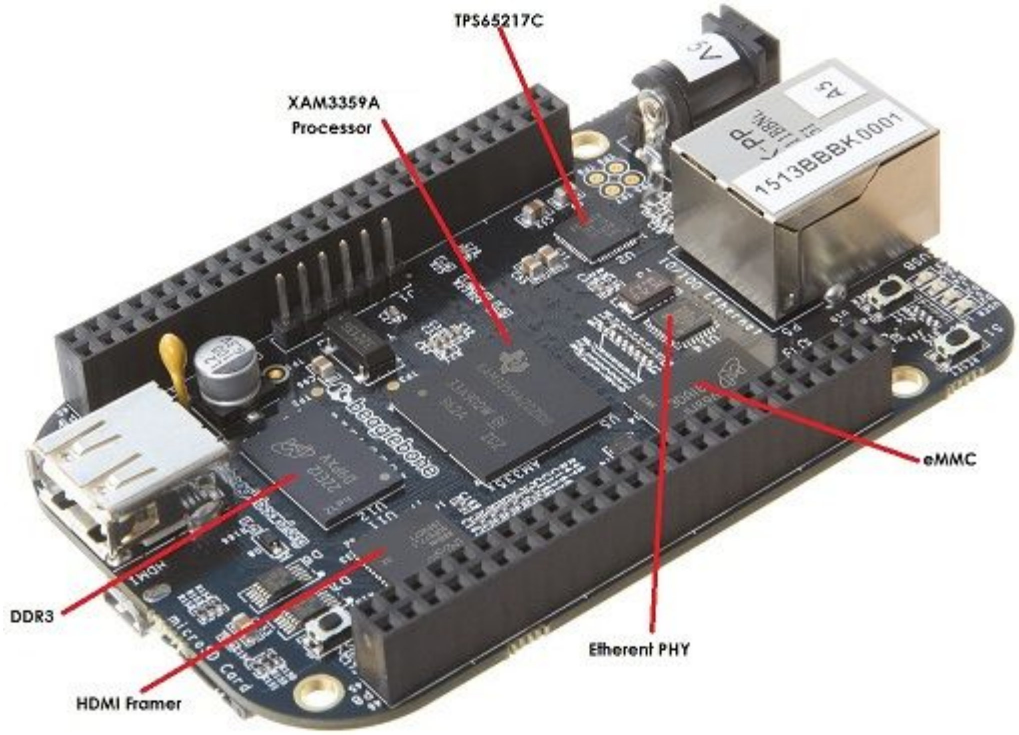
Beaglebone Black board

USB mini cable

Open the Beagle box and you will see a Beaglebone Black board and a USB mini cable.

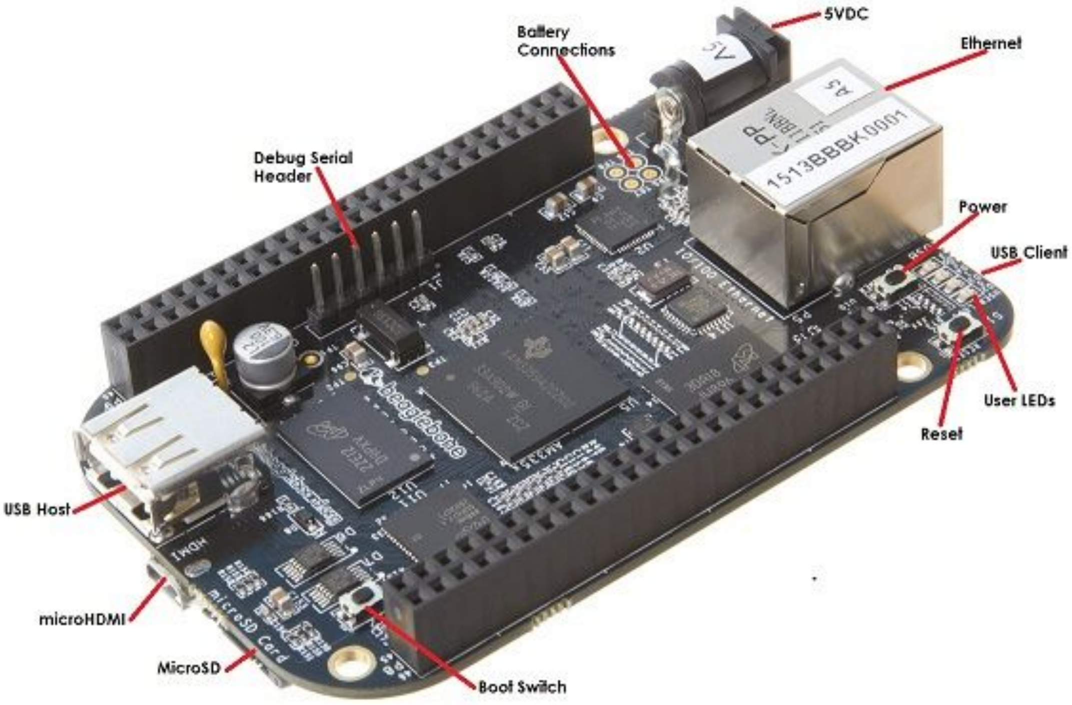
USB mini cable provide a convenient way to provide both power to your Beagle and connectivity to your PC.

Part A: Getting Started With a Beagle



BeagleBone Black Key Component Locations

Part A: Getting Started With a Beagle



BeagleBone Black Connector and Switch Locations

Part A: Getting Started With a Beagle

Update USB CDC Ethernet gadget to use NCM *(MacOS Only)*

The USB CDC Ethernet gadget of Beagle is currently configured to use the ECM protocol. Unfortunately this protocol is no longer supported by MacOS hosts. However, the newer NCM protocol is supported.

So if you are using MacOS operating system, you also need to follow the Step 1 to Step 6 to update the USB CDC Ethernet gadget configuration scripts to use the NCM protocol instead of the ECM protocol.

Note: You need to do Step 1 to Step 5 on a windows operating system.

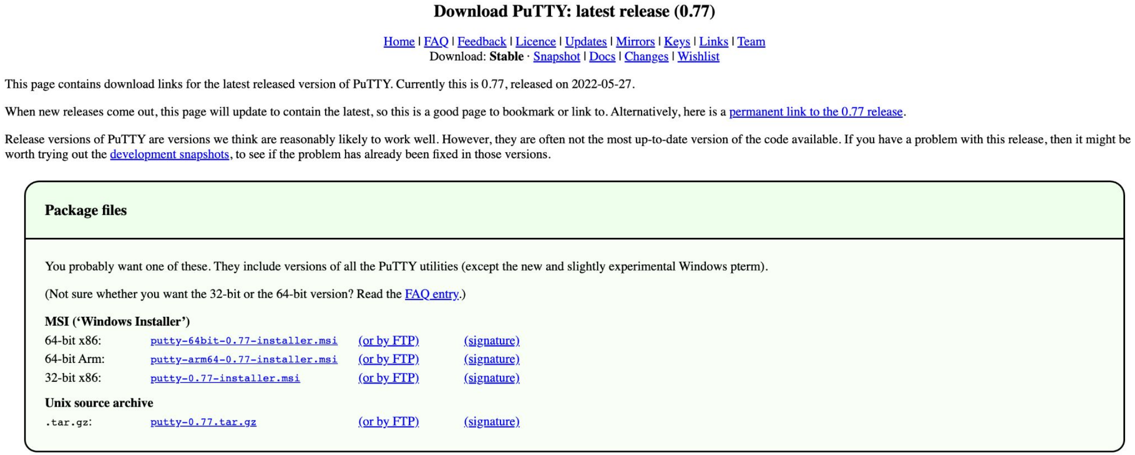
Part A: Getting Started With a Beagle

1. Download Putty

Step 1: Go to https://www.putty.org/

Step 2: Click Download PuTTY

Step 3: Choose the package file that matches your operating system and click to download



Download PuTTY: latest release (0.77)

Home | FAQ | Feedback | Licence | Updates | Mirrors | Keys | Links | Team
Download: **Stable** · Snapshot | Docs | Changes | Wishlist

This page contains download links for the latest released version of PuTTY. Currently this is 0.77, released on 2022-05-27.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a permanent link to the 0.77 release.

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the development snapshots, to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

MSI ('Windows Installer')

64-bit x86:	`putty-64bit-0.77-installer.msi`	(or by FTP)	(signature)
64-bit Arm:	`putty-arm64-0.77-installer.msi`	(or by FTP)	(signature)
32-bit x86:	`putty-0.77-installer.msi`	(or by FTP)	(signature)

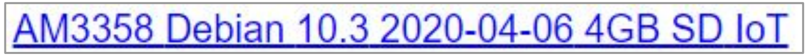
Unix source archive

`.tar.gz`:	`putty-0.77.tar.gz`	(or by FTP)	(signature)

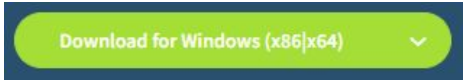
Part A: Getting Started With a Beagle

2. Download the software image to SD card

Step 1: Go to https://beagleboard.org/latest-images

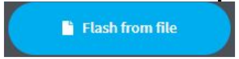
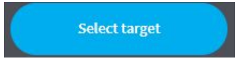
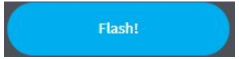
Step 2: Click  AM3358 Debian 10.3 2020-04-06 4GB SD IoT to download the latest Debian image

Step 3: Go to https://www.balena.io/etcher/

Step 4: Click  to download and install the SD card programming utility

Step 5: Use your computer's SD slot or a USB adapter to connect the SD card to your computer

Step 6: Write the software image to your SD card

(1) Open the SD card programming utility (balenaEtcher) you just installed
(2) Click  and select the image file with .img.xz extension you just downloaded
(3) Click  and select your SD card
(4) Click  to start writing the image to your SD card
(5) Eject the newly programmed SD card

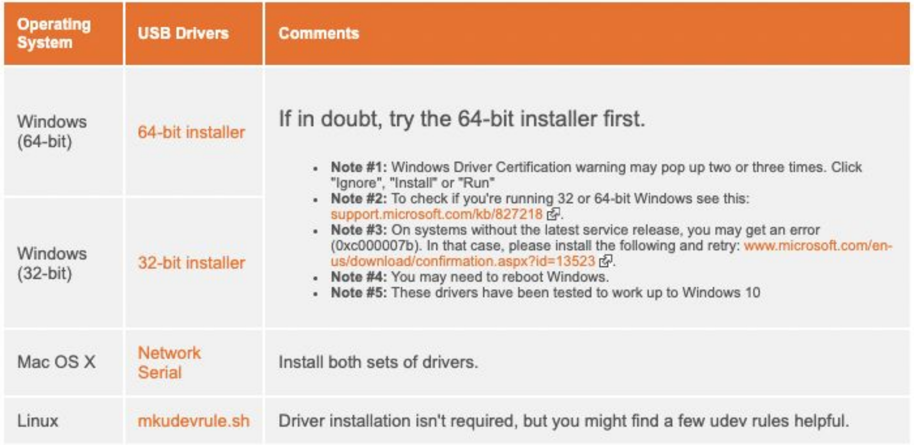
Part A: Getting Started With a Beagle

3. Download USB Driver

Step 1: Go to https://beagleboard.org/getting-started

Step 2: Choose the USB Driver file that matches your operating system and click to download

Step 3: Install the USB Driver



Operating System	USB Drivers	Comments
Windows (64-bit)	64-bit installer	If in doubt, try the 64-bit installer first. • **Note #1:** Windows Driver Certification warning may pop up two or three times. Click "Ignore", "Install" or "Run" • **Note #2:** To check if you're running 32 or 64-bit Windows see this: support.microsoft.com/kb/827218. • **Note #3:** On systems without the latest service release, you may get an error (0xc000007b). In that case, please install the following and retry: www.microsoft.com/en-us/download/confirmation.aspx?id=13523. • **Note #4:** You may need to reboot Windows. • **Note #5:** These drivers have been tested to work up to Windows 10
Windows (32-bit)	32-bit installer	
Mac OS X	Network Serial	Install both sets of drivers.
Linux	mkudevrule.sh	Driver installation isn't required, but you might find a few udev rules helpful.

If the installation fails, follow this tutorial to reinstall:
https://support.viewsonic.com/en/support/solutions/articles/33000252416-how-to-install-unsigned-drivers-in-windows-10

Part A: Getting Started With a Beagle

4. Power On and Boot the Beagle off of the SD card

Step 1: Insert SD card into your board *(Make sure SD card is inserted ahead of providing power!)*

Step 2: Connect the USB cable to your Windows PC

Step 3: Hold down the Boot Switch button and connect the other end of the USB mini cable to your beagle

After step 3, you'll see the power (PWR or ON) LED lit steadily. Within a minute or so, you should see the other LEDs blinking in their default configurations. Below is the information for LEDs:

- USR0 is typically configured at boot to blink in a heartbeat pattern
- USR1 is typically configured at boot to light during SD (microSD) card accesses
- USR2 is typically configured at boot to light during CPU activity
- USR3 is typically configured at boot to light during eMMC accesses
- USR4/WIFI is typically configured at boot to light with WiFi (client) network association (*BeagleBone Blue and BeagleBone AI only*)

Part A: Getting Started With a Beagle

5. SSH Into the Beagle

Step 1: Start Putty

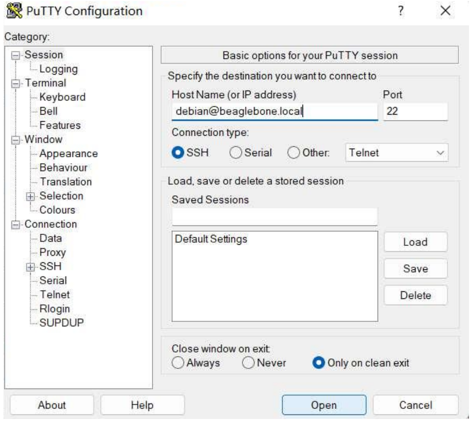
Step 2: In the Host Name (or IP address) text box
type **`debian@beaglebone.local`**

Step 3: In the Port textbox, type **22**

Step 4: Click **Open**

Step 5: A PuTTY security alert about the server's host key appears the first time you connect
Click **Yes**

Step 6: Type initial password **`temppwd`** when prompted
Press **Enter**



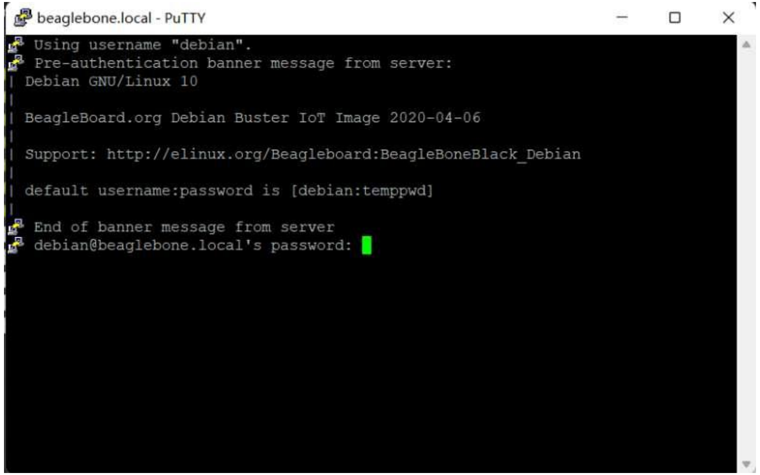
Specify the destination you want to connect to

Host Name (or IP address): debian@beaglebone.local

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Close window on exit: ○ Always ○ Never ● Only on clean exit



```
Using username "debian".
Pre-authentication banner message from server:
| Debian GNU/Linux 10
|
| BeagleBoard.org Debian Buster IoT Image 2020-04-06
|
| Support: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
|
| default username:password is [debian:temppwd]
|
End of banner message from server
debian@beaglebone.local's password:
```

Part A: Getting Started With a Beagle

6. Update USB CDC Ethernet gadget to use NCM *(MacOS Only)*

Follow this tutorial to update USB CDC Ethernet gadget:
https://github.com/RobertCNelson/boot-scripts/commit/b61125c1485bee929340cacc06c85c6fcfd678bc

Step 1: Enter the following command in the console of the putty window:

```
$ cd ../../opt/scripts/boot
$ vim am335x_evm.sh
```

Step 2: Press **i** to switch to Insert mode

Step 3: Press ⬇️ to go line 484, change “ecm” in lines 484, 485, 486, 488 to “ncm”

Step 4: Press **Esc** to leave Insert mode

Step 5: Type **:wq** and press **Enter** to save file

Step 6: Now you have updated USB gadget in “am335x_evm.sh”, then update “bbai.sh”, “bbai_usb_gadget.sh”, “boot/beagle_x15.sh” in the same way according to the tutorial.

Part A: Getting Started With a Beagle

7. Connect your Beagle with Mac PC *(MacOS Only)*

Step 1: Now you have updated USB CDC Ethernet gadget. Eject and unplug the Beagle from Windows

Step 2: Connect the USB cable to your Mac PC via an adapter

Step 3: Hold down the Boot Switch button and connect the other end of the USB mini cable to your beagle

Step 4: Open the Terminal

Step 5: Enter the following command in the Terminal console to ssh your Beagle:

```
$ ssh debian@beaglebone.local
```

Step 6: Type initial password **temppwd** when prompted, and then press **Enter**

8. Connect Beagle to the Internet

There are two ways to share the Internet with Beagle.

You can do this by plugging an ethernet cable into the beagle's Ethernet port.

You can also connect the beagle to WiFi by following the next three pages.

Note: You cannot share the Internet by connecting to eduroam or other WiFi that needs authentication

Part A: Getting Started With a Beagle

8. Connect Beagle to WiFi *(For MacOS)*

Step 1: Enter **`ifconfig`** command in the Terminal console, check the interface of the beagle

Step 2: Enter the following command in the Terminal console to share network with your Beagle*(You should do step 2 every time you SSH the beagle and want to connect to the internet*):

```
$ sudo dhclient usb1
```

Note: Replace usb1 with the interface used by the beagle on your PC

Step 3: Type initial password **temppwd** when prompted, and then press **Enter**

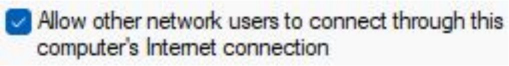
8. Connect Beagle to WiFi *(For Windows)*

Step 1: Enter the following command in the Terminal console:

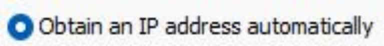
```
$ sudo /sbin/route add default gw 192.168.7.1
```

Step 2: Open the Control Panel and select **Network and Internet**

- For Windows 11 PC
  - Select **Network and Sharing Center**
  - Click on the PC’s Internet connection icon and choose the **Properties** option
  - Click on **Sharing** tab at the top, enable ☑ Allow other network users to connect through this computer's Internet connection option

    
  - Select Beagle’s connection in the drop down text box of Home networking connection
  - Click **OK**, go back to **Network and Sharing Center**
  - Click on the Beagle’s connection icon and select the **Properties** option
  - Enable ☑ Internet Protocol Version 4 (TCP/IPv4) option and select the **Properties** option

  - Enable ● Obtain an IP address automatically option , and then click **OK**

    
- For Windows 10 PC and more details
  - Visit https://ofitselfso.com/BeagleNotes/HowToConnectPocketBeagleToTheInternetViaUSB.php

Part A: Getting Started With a Beagle

8. Connect Beagle to WiFi *(For Windows)*

Step 3: Restart Putty and SSH into the beagle

Step 4: Enter the following command in the Terminal console *(You should do step 2 and 3 every time you SSH the beagle and want to connect to the internet)* :

```
$ sudo /sbin/route add default gw 192.168.7.1
```

Step 5: Configure a nameserver

(1) Enter the following command in the Terminal console:

```
$ sudo nano /etc/resolv.conf
```

(2) Type the following into the editor :

```
domain localdomain
search localdomain
nameserver 8.8.8.8
nameserver 8.8.4.4
```

(3) Press **Ctrl+o** followed with Enter to save changes, and press **Ctrl+x** to exit nano

Part B: Setup Python-OpenCV Pipeline

1. Install Python

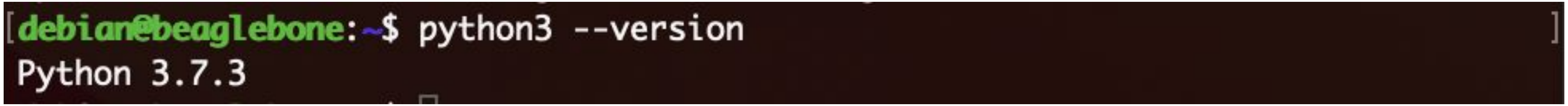
Step 1: Enter the following commands in the Terminal console:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install python3
```

Step 2: Enter the following command in the Terminal console to verify the installation:

```
$ python3 --version
```

You should get the version like **Python 3.7.3** if Python 3 is installed successfully



```
debian@beaglebone:~$ python3 --version
Python 3.7.3
```

Part B: Setup Python-OpenCV Pipeline

2. Install Python-OpenCV

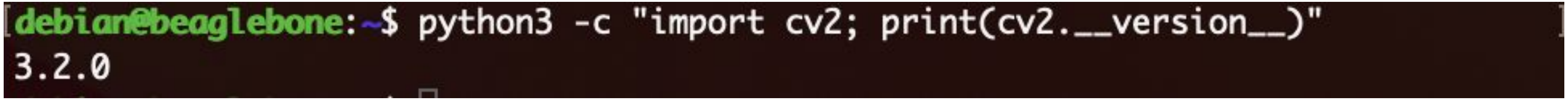
Step 1: Enter the following command in the Terminal console:

```
$ sudo apt install python3-opencv
```

Step 2: Enter the following command in the Terminal console to verify the installation:

```
$ python3 -c "import cv2; print(cv2.__version__)"
```

You should get the version like **3.2.0** if Python-Opencv is installed successfully



```
debian@beaglebone:~$ python3 -c "import cv2; print(cv2.__version__)"
3.2.0
```

Part C: Face Detection in Beagle

1. Copy file to Beagle and run python scripts

Step 1: Enter the following commands in the Terminal console to create folder:

```
$ cd ~
$ mkdir code/
```

Step 2: Download the face_detection folder to your PC’s Downloads folder

Step 3: Enter the following command in your local Terminal console to copy folder from your PC to beagle:

```
$ scp -r ~/Downloads/Lab5 debian@beaglebone.local:~/code/
```

Step 4: Enter the following commands in the Beagle’s Terminal console to run the python scripts:

```
$ cd code/Lab5/
$ python3 face_detect_img.py
```

Image / Video Credits:

- *test_image.png: https://en.wikipedia.org/wiki/ABBA#/media/File:ABBA_-_TopPop_1974_5.png*
- *test_video.mp4: https://www.youtube.com/watch?v=Mh4f9AYRCZY*

Part C: Face Detection in Beagle

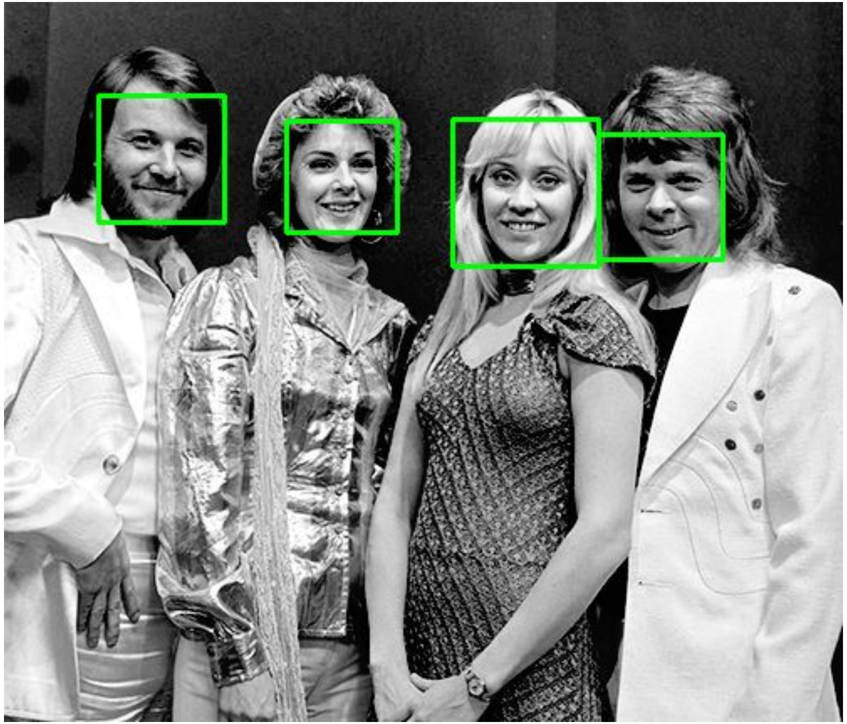
2. Your Tasks

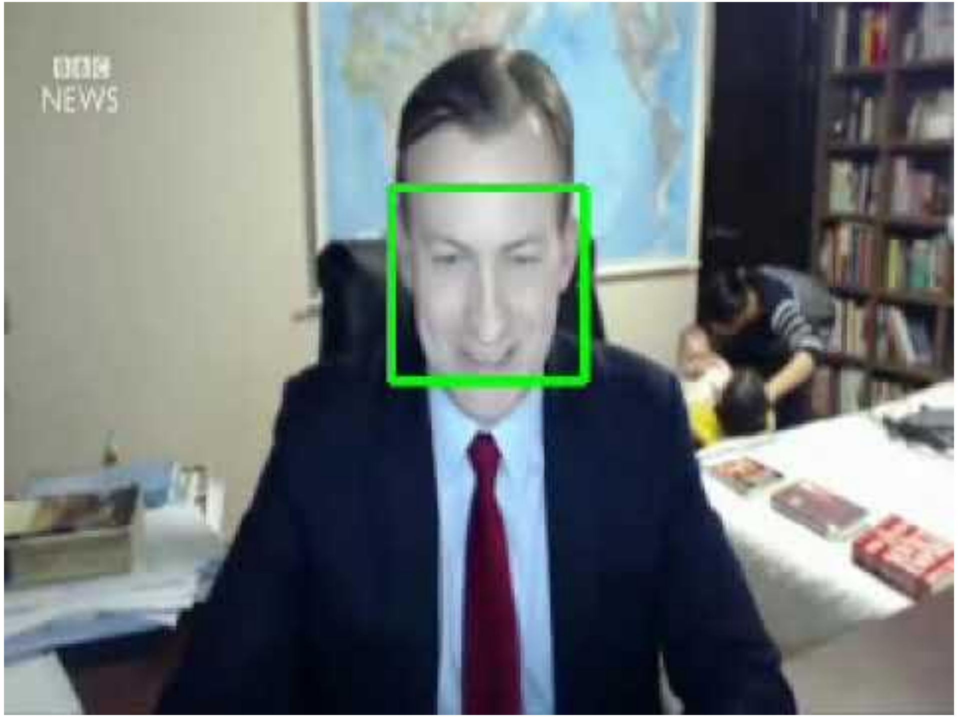
- Check and verify the output of **`face_detect_img.py`**
- Complete the **`face_detect_video.py`** file so that
  - It takes input a path for a video file
    - Reads the video file frame by frame
    - Detects faces in each frame, and draws bounding boxes if any faces are found
    - The final output should be a video with all the face boxes
    - Check: https://youtu.be/prSROEIjuaY

- Note that the **`face_detect_webcam.py`** file is given for your reference
  - Try it on your laptop (not on beagle because there are no cameras connected to it!)
  - Move your face and check the performance; maybe try it with multiple people.

- *Bonus points (1-2):*
  - *Implement some tracking features (by Kalman filtering or any other filtering approach)*

Lab5 Output: Face Detection in Image/Video

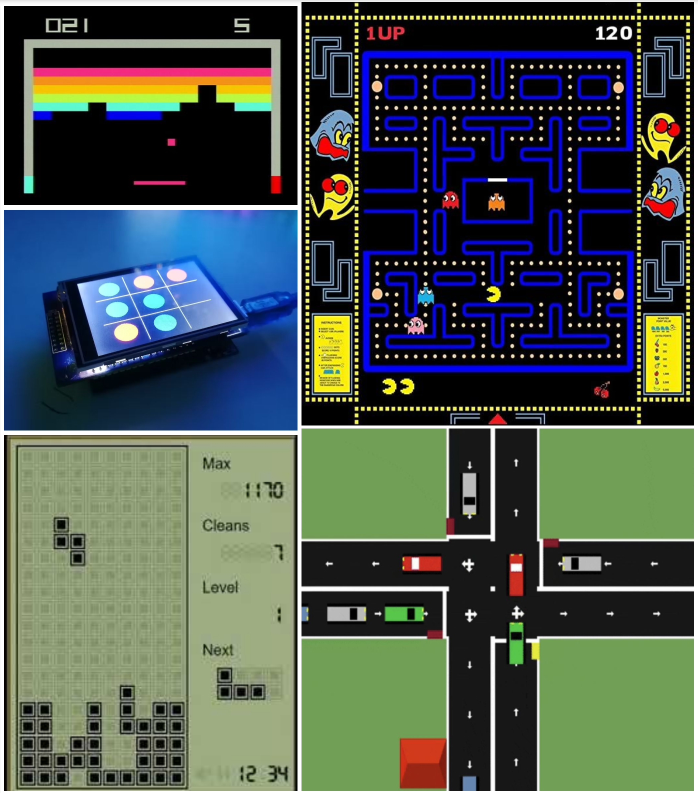




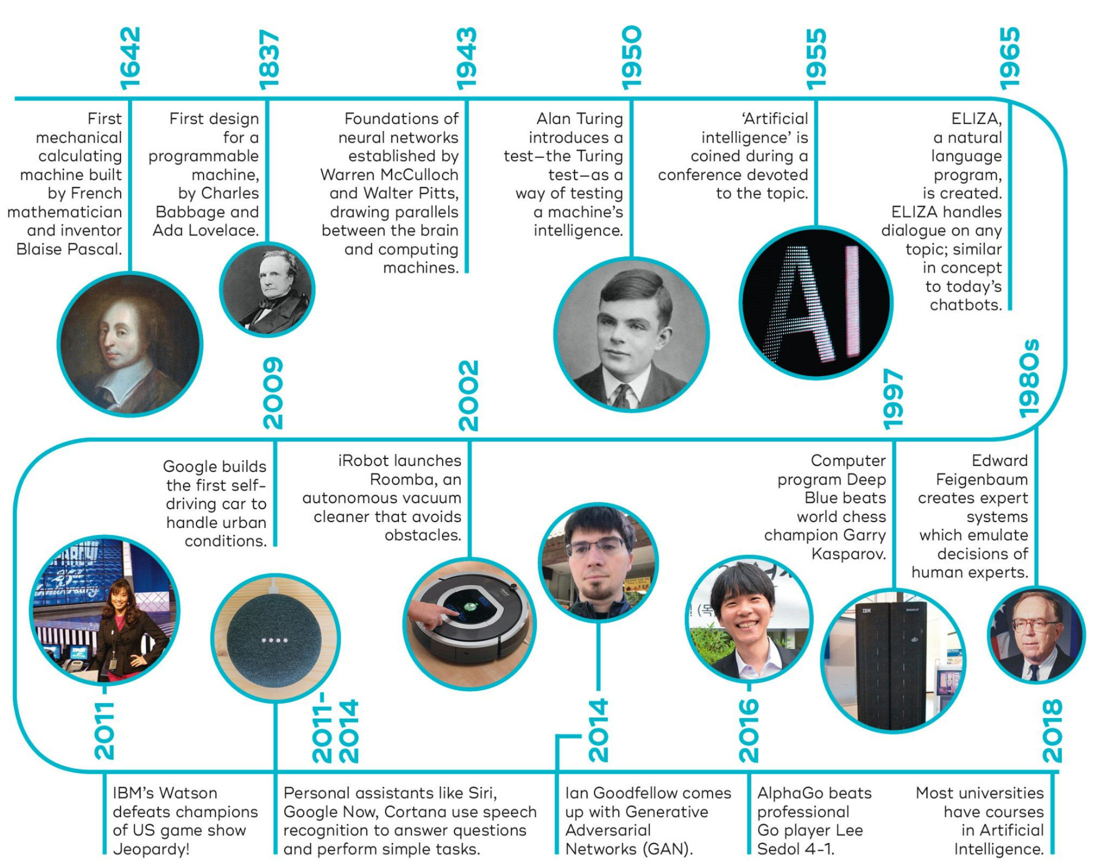
Logistics: Lab 5 and Project

- **Lab-5 days: 11/7 - 11/18**
  - Very easy lab, extended for over two weeks so that you find time to formulate and have the initial sketch of your project.
  - **Demo due by 11/18:** must be during your own lab or any office hours

- **Project idea formulation by 11/15**
  - By **November 15th**, decide on your project and prepare an 1-slide or 1-page document stating the following
    - Project/game title: what it is and how it will operate (a brief description unless it is a popular well known game)
    - **Your RTOS design sketch**
      - Threads, semaphores, memory managements
      - How the game will be played and what are the steps from start to finish (similar to Lecture 6 slide 23-25)
  - Show this document to your TA or me and get feedback
  - We will discuss with you and set three milestones
    - Milestone #1: RTOS skeleton with some operational threads
    - Milestone #2: Shared memory access and IPC done
    - Milestone #3: final operational demo of the whole game/ project



AI: History



1642 First mechanical calculating machine built by French mathematician and inventor Blaise Pascal.

1837 First design for a programmable machine, by Charles Babbage and Ada Lovelace.

1943 Foundations of neural networks established by Warren McCulloch and Walter Pitts, drawing parallels between the brain and computing machines.

1950 Alan Turing introduces a test—the Turing test—as a way of testing a machine's intelligence.

1955 'Artificial intelligence' is coined during a conference devoted to the topic.

1965 ELIZA, a natural language program, is created. ELIZA handles dialogue on any topic; similar in concept to today's chatbots.

1980s Edward Feigenbaum creates expert systems which emulate decisions of human experts.

1997 Computer program Deep Blue beats world chess champion Garry Kasparov.

2002 iRobot launches Roomba, an autonomous vacuum cleaner that avoids obstacles.

2009 Google builds the first self-driving car to handle urban conditions.

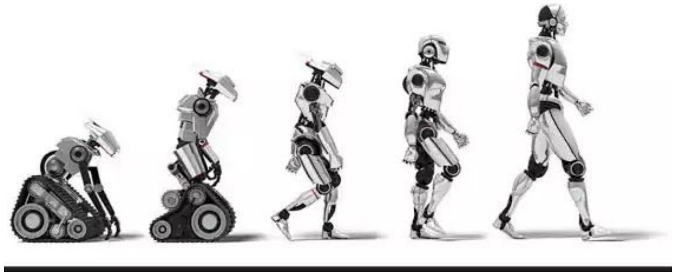
2011 IBM's Watson defeats champions of US game show Jeopardy!

2011-2014 Personal assistants like Siri, Google Now, Cortana use speech recognition to answer questions and perform simple tasks.

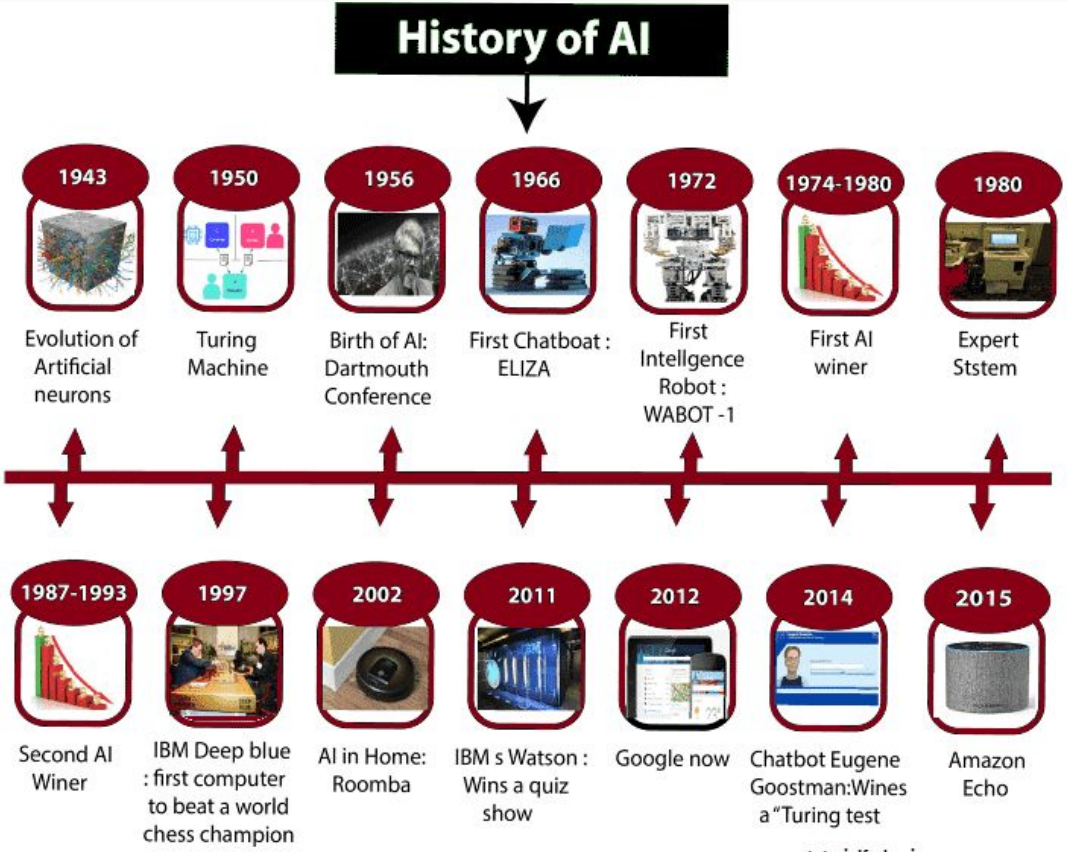
2014 Ian Goodfellow comes up with Generative Adversarial Networks (GAN).

2016 AlphaGo beats professional Go player Lee Sedol 4-1.

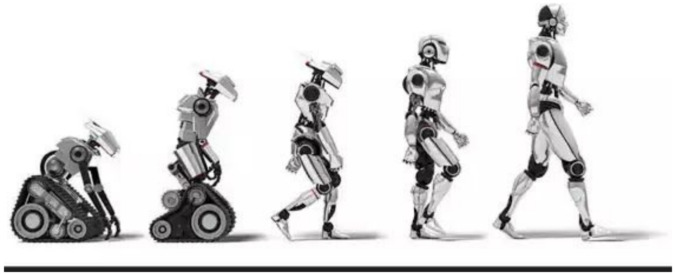
2018 Most universities have courses in Artificial Intelligence.



AI: History

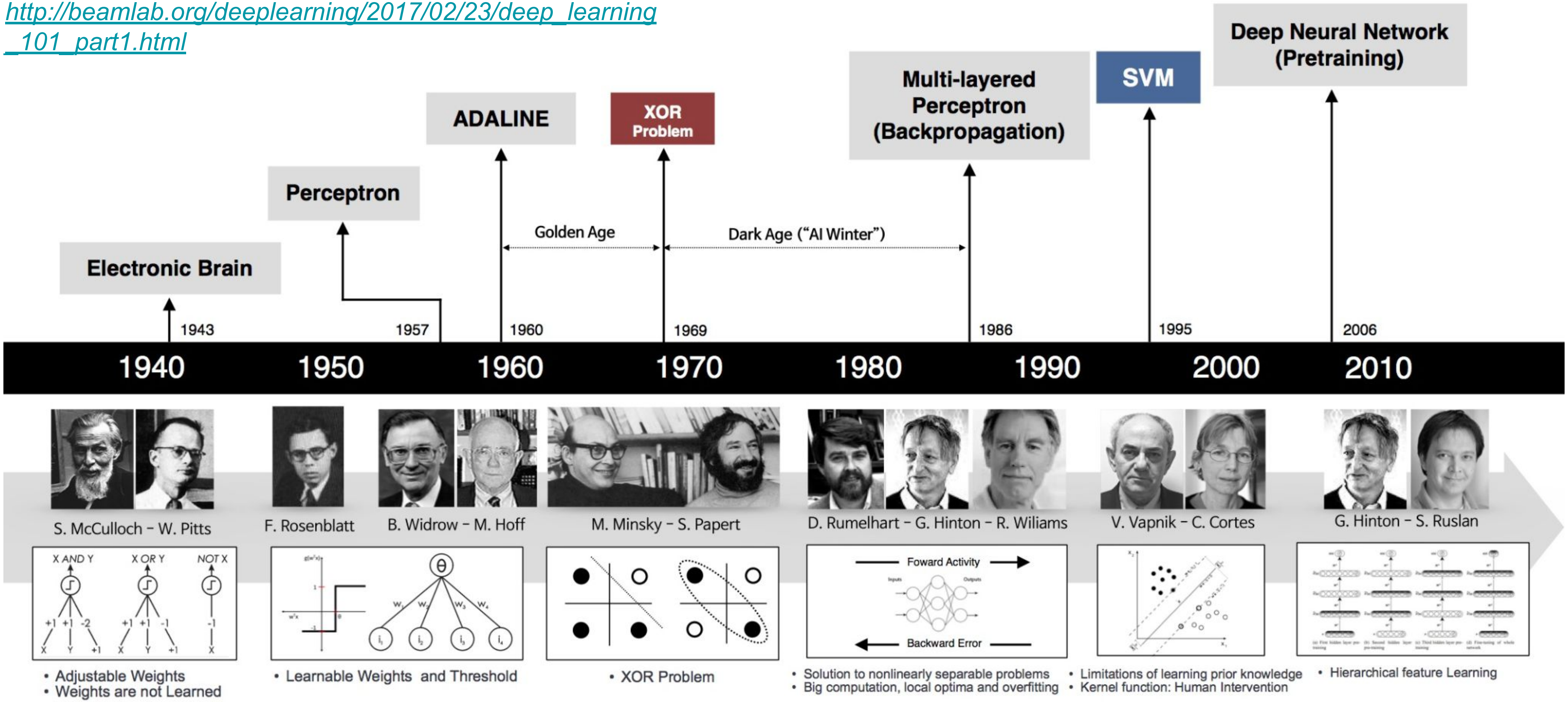

History of AI
1943
Evolution of Artificial neurons
1950
Turing Machine
1956
Birth of AI: Dartmouth Conference
1966
First Chatboat : ELIZA
1972
First Intellgence Robot : WABOT -1
1974-1980
First AI winer
1980
Expert Ststem
1987-1993
Second AI Winer
1997
IBM Deep blue : first computer to beat a world chess champion
2002
AI in Home: Roomba
2011
IBM s Watson : Wins a quiz show
2012
Google now
2014
Chatbot Eugene Goostman:Wines a "Turing test
2015
Amazon Echo

tutorialforbeginner.com



Deep Learning: History

http://beamlab.org/deeplearning/2017/02/23/deep_learning_101_part1.html



Electronic Brain — 1943

Perceptron — 1957

ADALINE — 1960

XOR Problem — 1969

Golden Age

Dark Age ("AI Winter")

Multi-layered Perceptron (Backpropagation) — 1986

SVM — 1995

Deep Neural Network (Pretraining) — 2006

1940 1950 1960 1970 1980 1990 2000 2010

S. McCulloch - W. Pitts
- Adjustable Weights
- Weights are not Learned

F. Rosenblatt

B. Widrow - M. Hoff
- Learnable Weights and Threshold

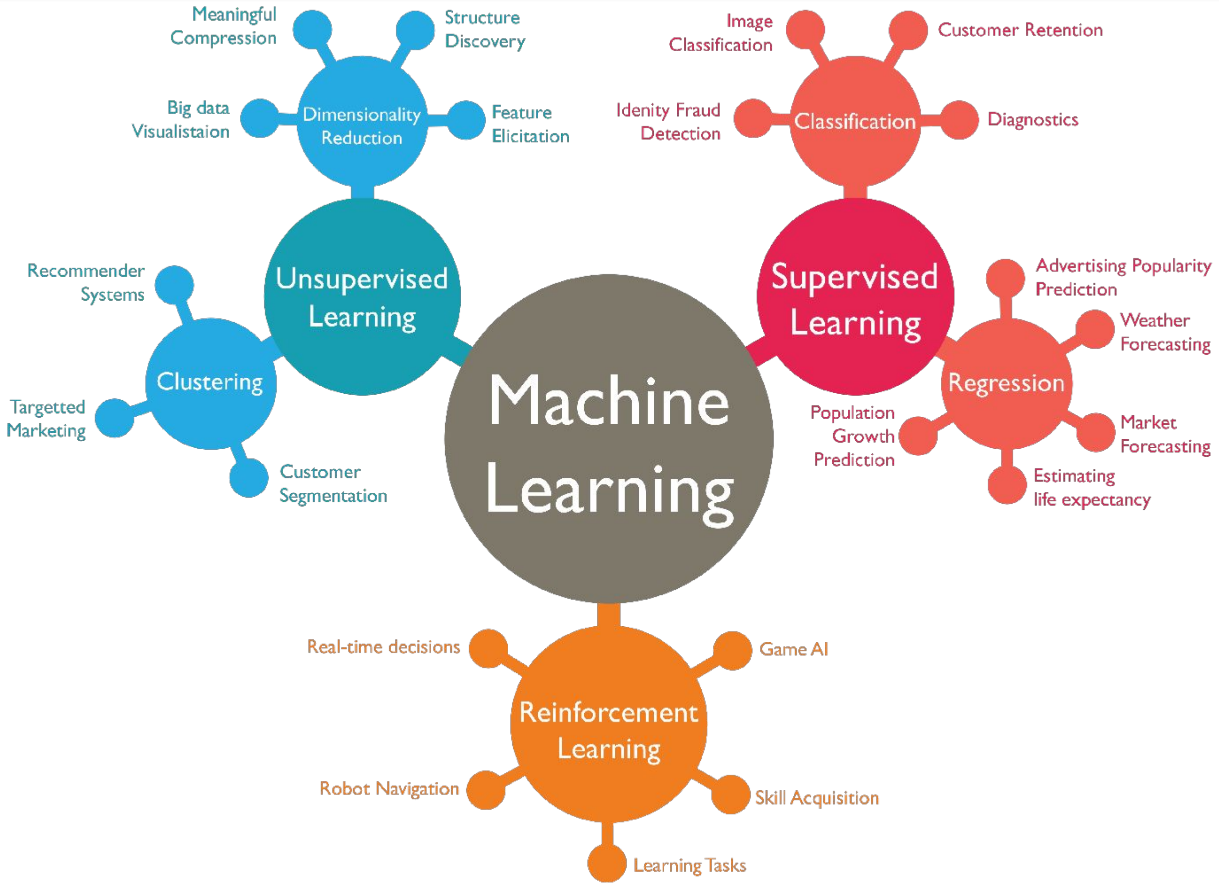
M. Minsky - S. Papert
- XOR Problem

D. Rumelhart - G. Hinton - R. Wiliams
- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting

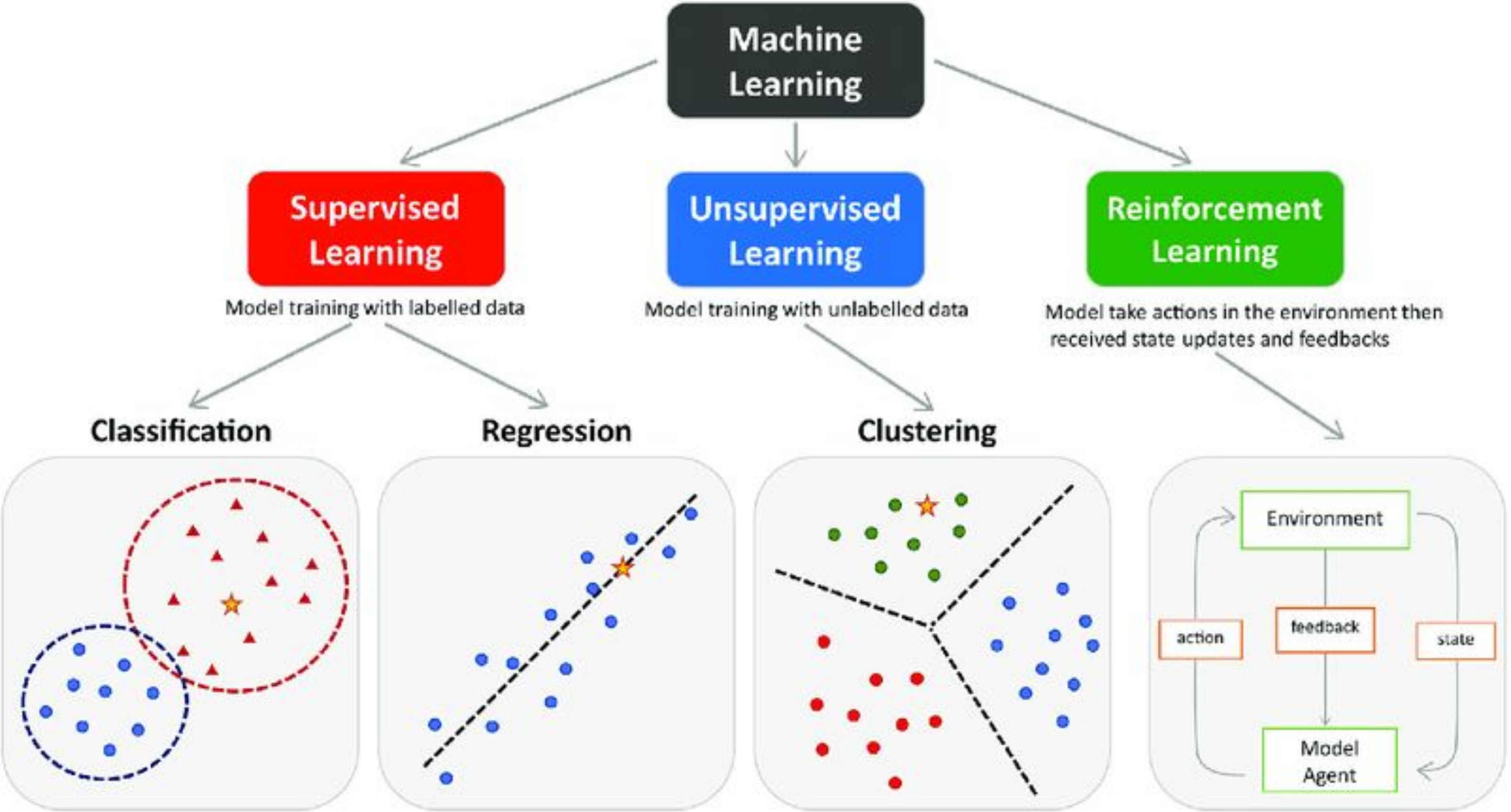
V. Vapnik - C. Cortes
- Limitations of learning prior knowledge
- Kernel function: Human Intervention

G. Hinton - S. Ruslan
- Hierarchical feature Learning

Machine Learning (ML)



Machine Learning (ML)



Machine Learning

Supervised Learning

Model training with labelled data

Unsupervised Learning

Model training with unlabelled data

Reinforcement Learning

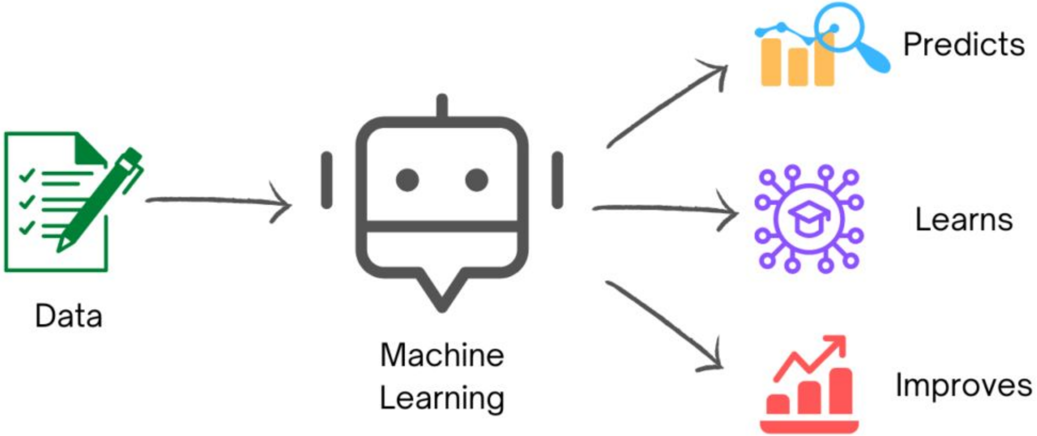
Model take actions in the environment then received state updates and feedbacks

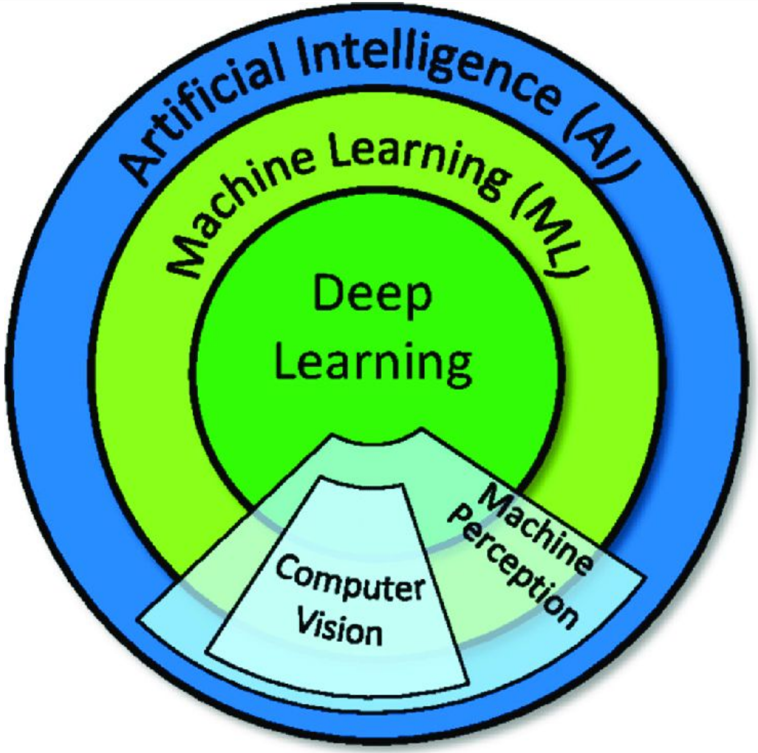
Classification

Regression

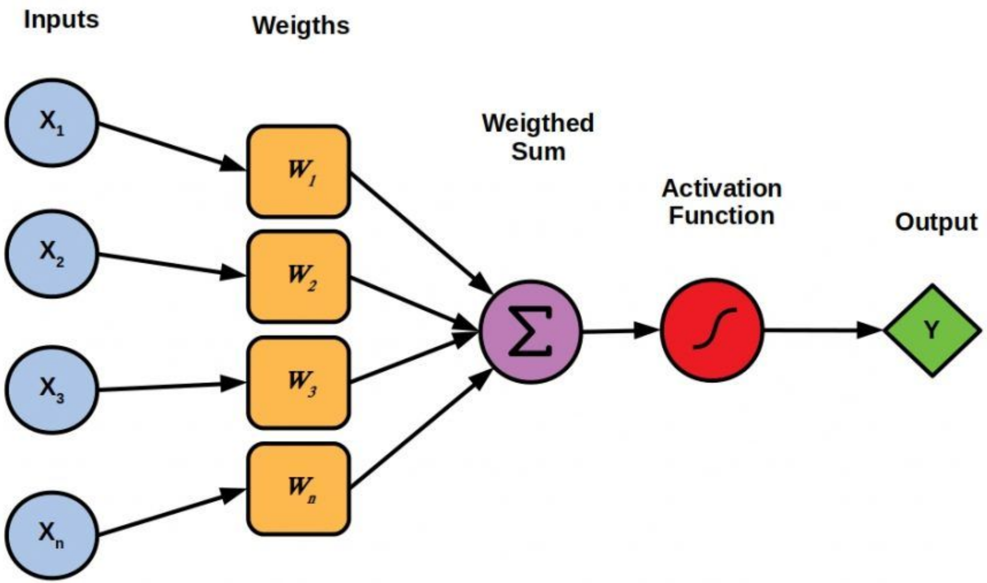
Clustering

AI and ML


Data
Machine Learning
Predicts
Learns
Improves

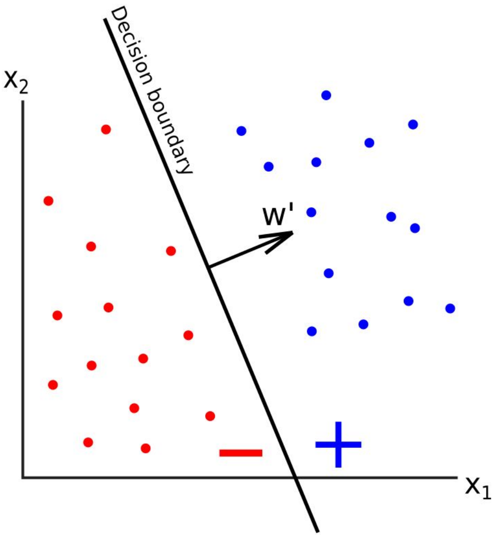

Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning
Computer Vision
Machine Perception

Early Days: The Perceptron

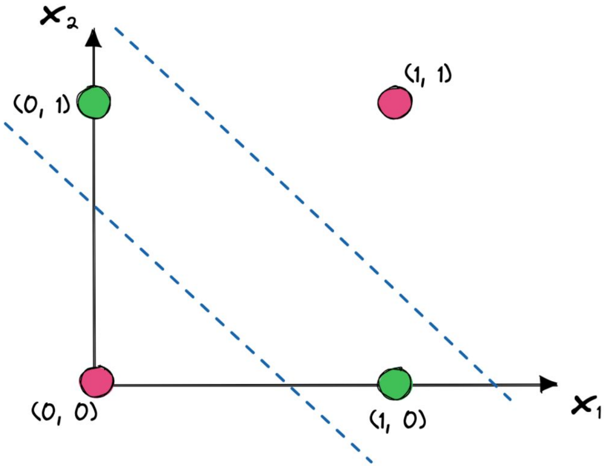


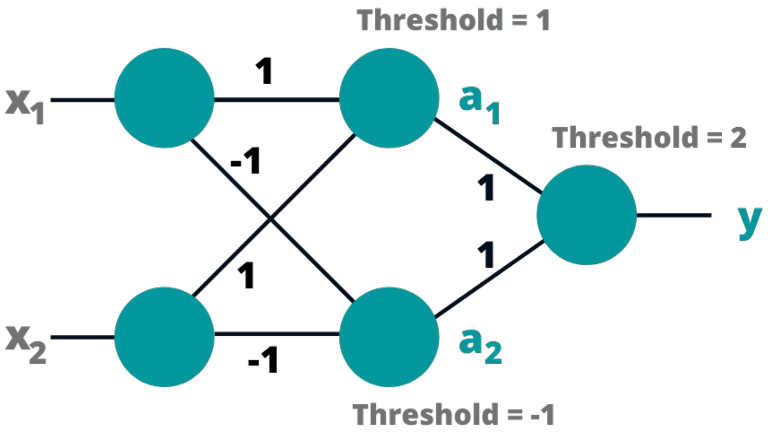
$$z = w_0x_0 + w_1x_1 + \ldots + w_mx_m = \boldsymbol{w}^T\boldsymbol{x}$$

$$\phi(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ -1 \text{ otherwise} \end{cases}$$



Early Days: The XOR Problem

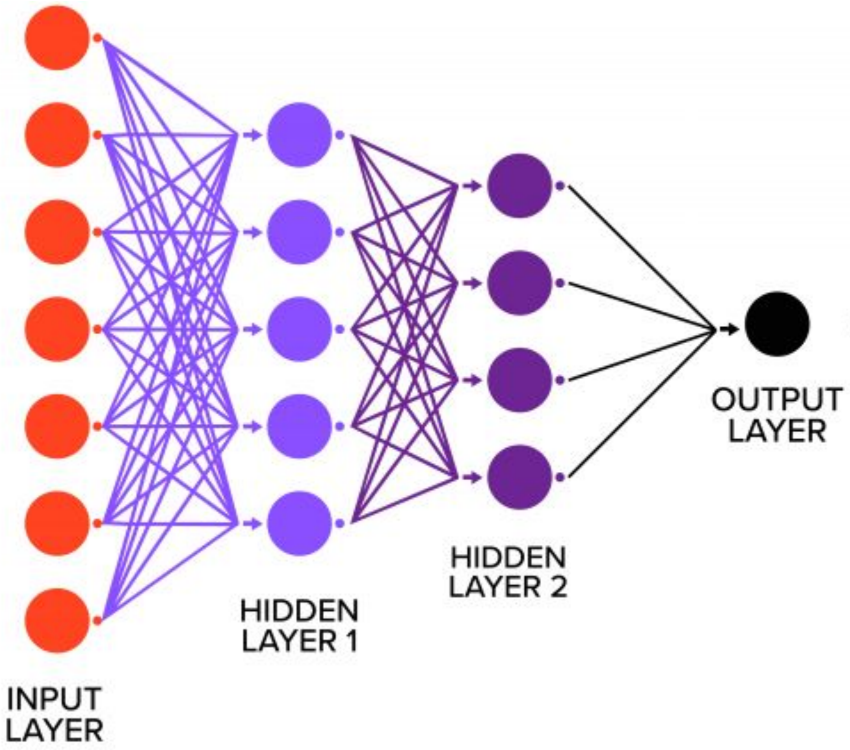




$$a_1 = \begin{cases} 0 & x_1 + x_2 < 1 \\ 1 & x_1 + x_2 \geq 1 \end{cases} \quad a_2 = \begin{cases} 0 & -x_1 - x_2 < -1 \\ 1 & -x_1 - x_2 \geq -1 \end{cases}$$

$$y = \begin{cases} 0 & a_1 + a_2 < 2 \\ 1 & a_1 + a_2 \geq 2 \end{cases}$$

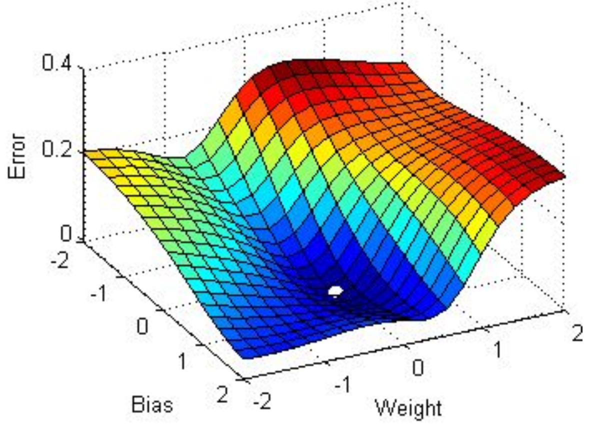
Early Days: Neural Networks (NN)



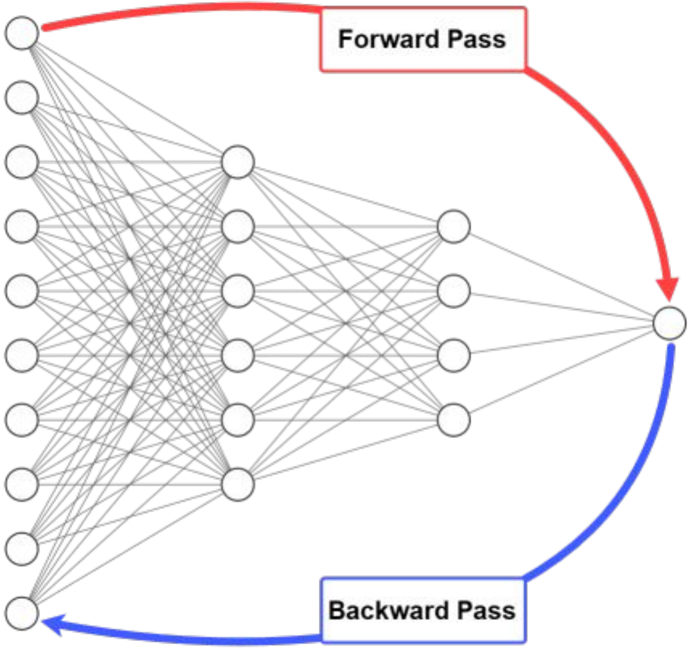
$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

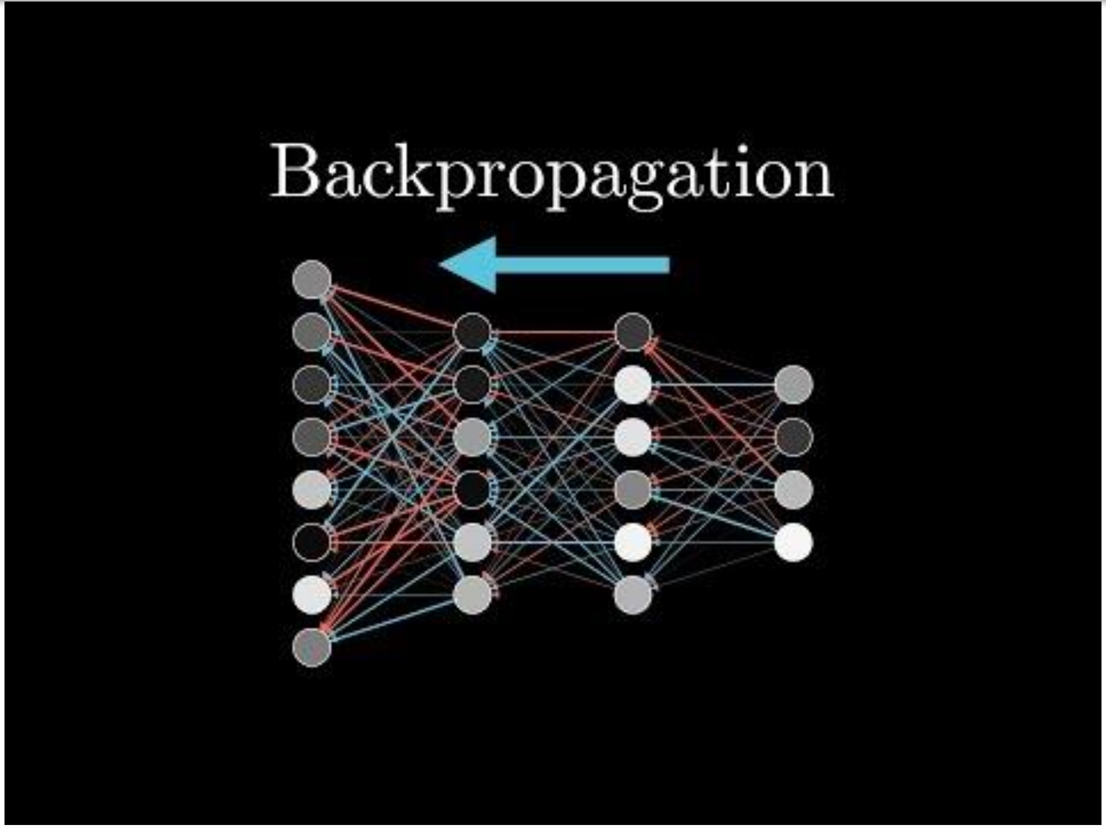
$$\text{output} = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$

$$\textit{Cost Function} = \textit{MSE} = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$



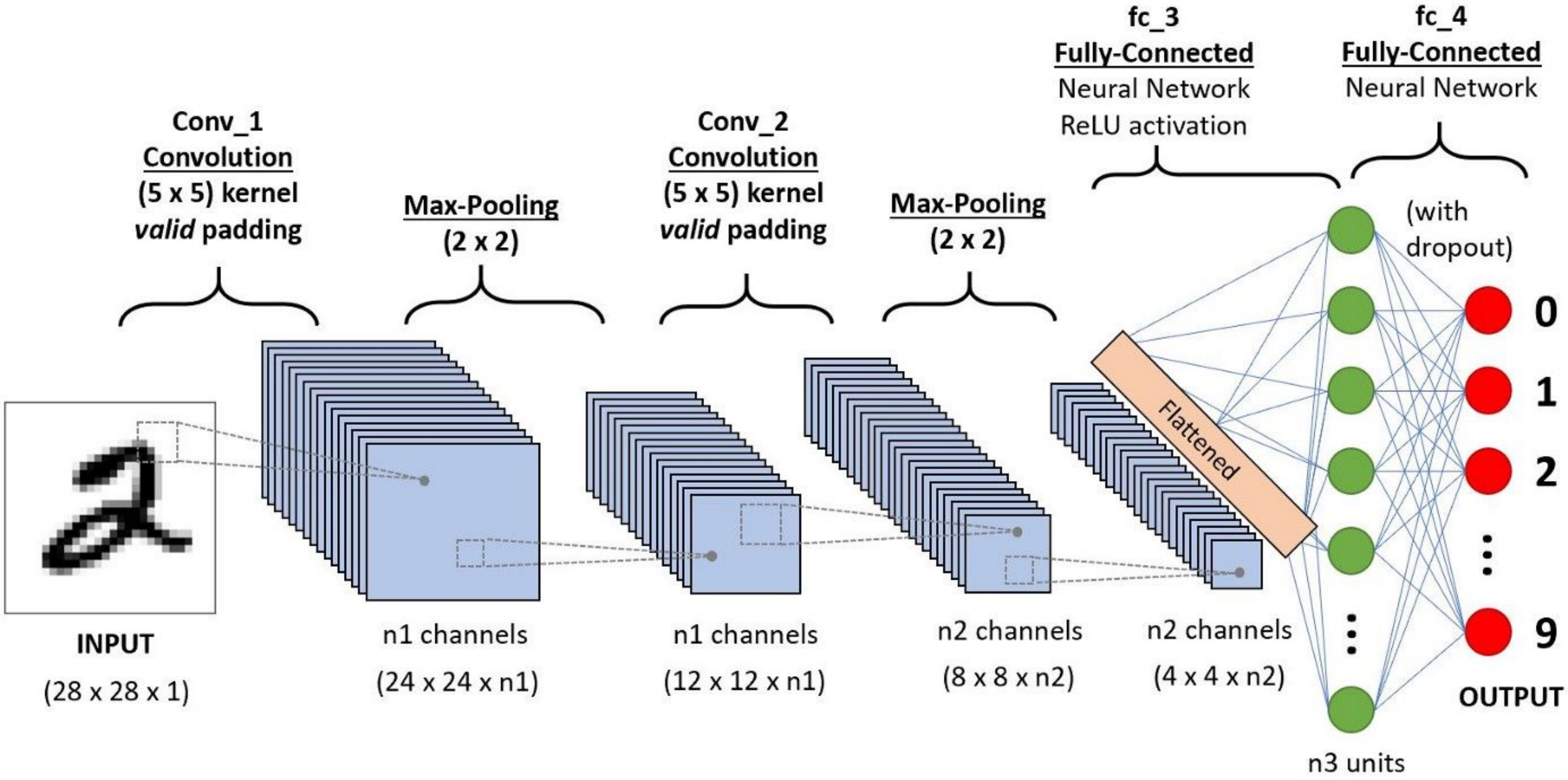
DNNs: Deep Neural Networks



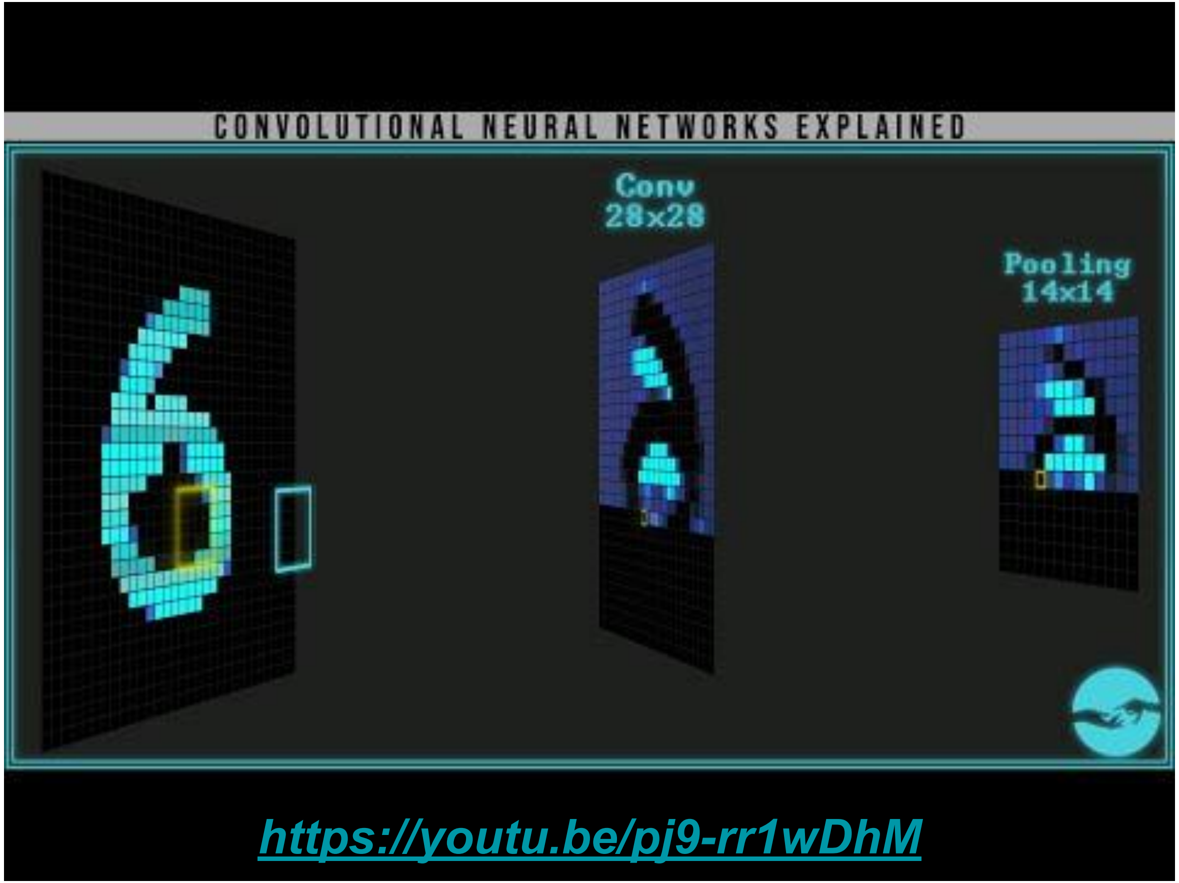


https://youtu.be/Ilg3gGewQ5U

CNNs: Convolutional Neural Networks



CNNs: Convolutional Neural Networks



https://youtu.be/pj9-rr1wDhM

ML/DL: SW Frameworks

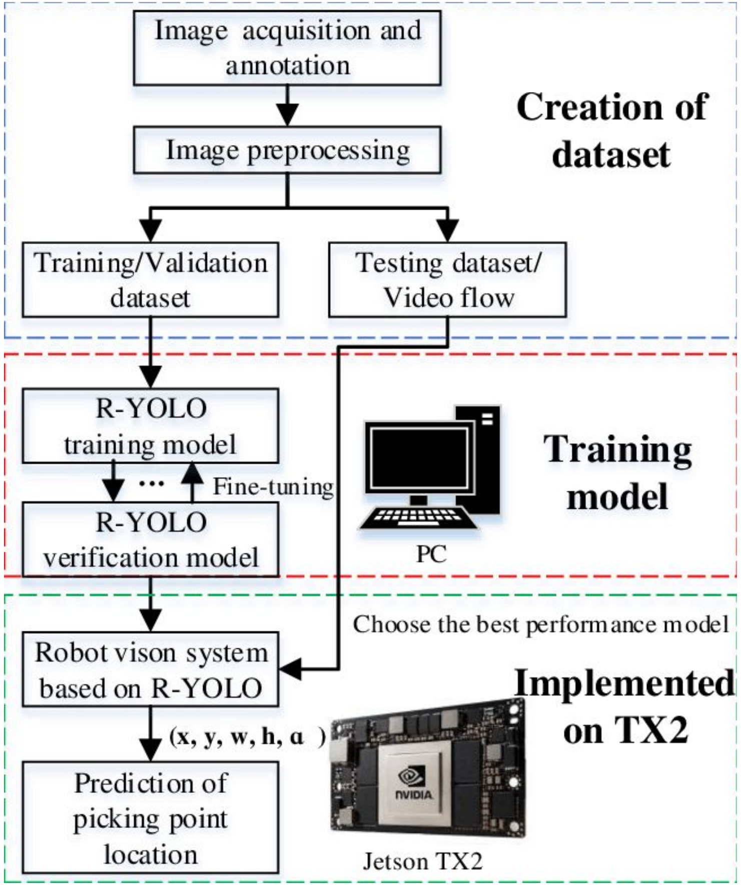


Image acquisition and annotation
Image preprocessing
Creation of dataset
Training/Validation dataset
Testing dataset/ Video flow
R-YOLO training model
...
Fine-tuning
R-YOLO verification model
PC
Training model
Choose the best performance model
Robot vison system based on R-YOLO
(x, y, w, h, α)
Prediction of picking point location
Jetson TX2
Implemented on TX2

PyTorch

TensorFlow

ML/DL: Edge HW

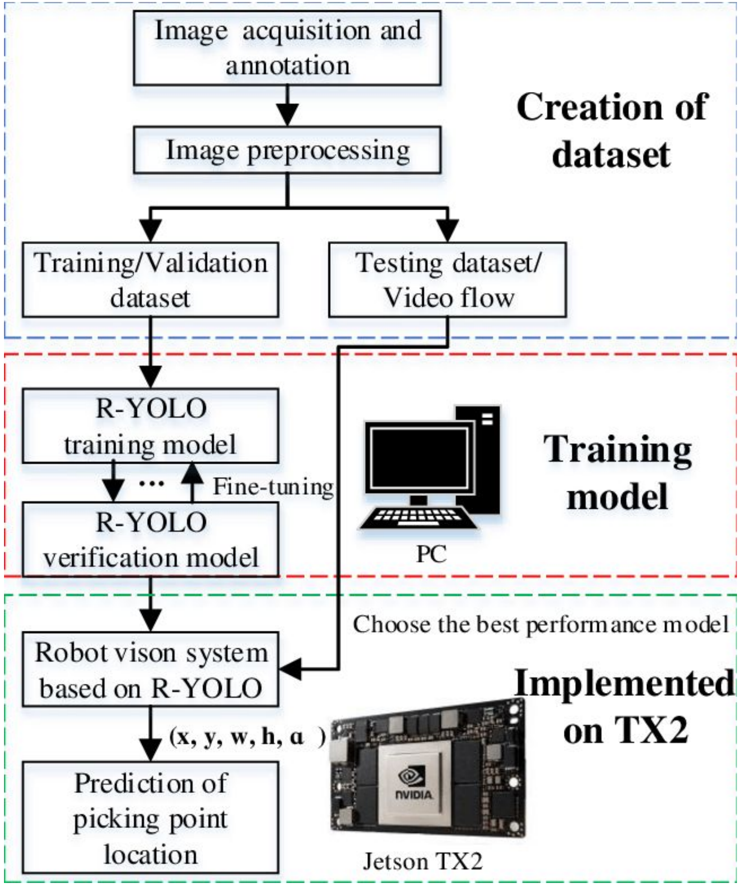


Image acquisition and annotation

Image preprocessing

Training/Validation dataset

Testing dataset/ Video flow

Creation of dataset

R-YOLO training model

…

Fine-tuning

R-YOLO verification model

PC

Training model

Choose the best performance model

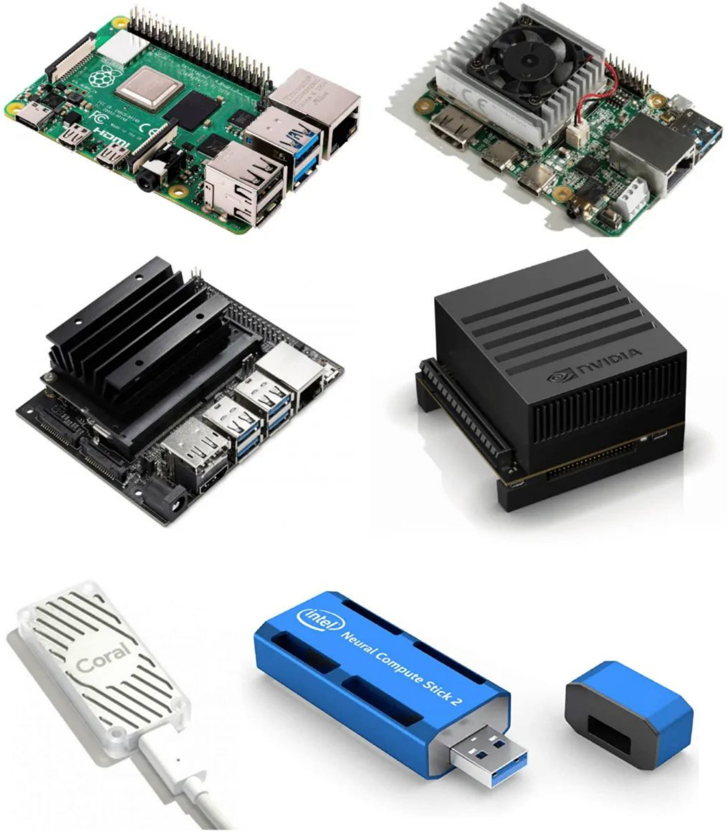
Robot vison system based on R-YOLO

(x, y, w, h, α)

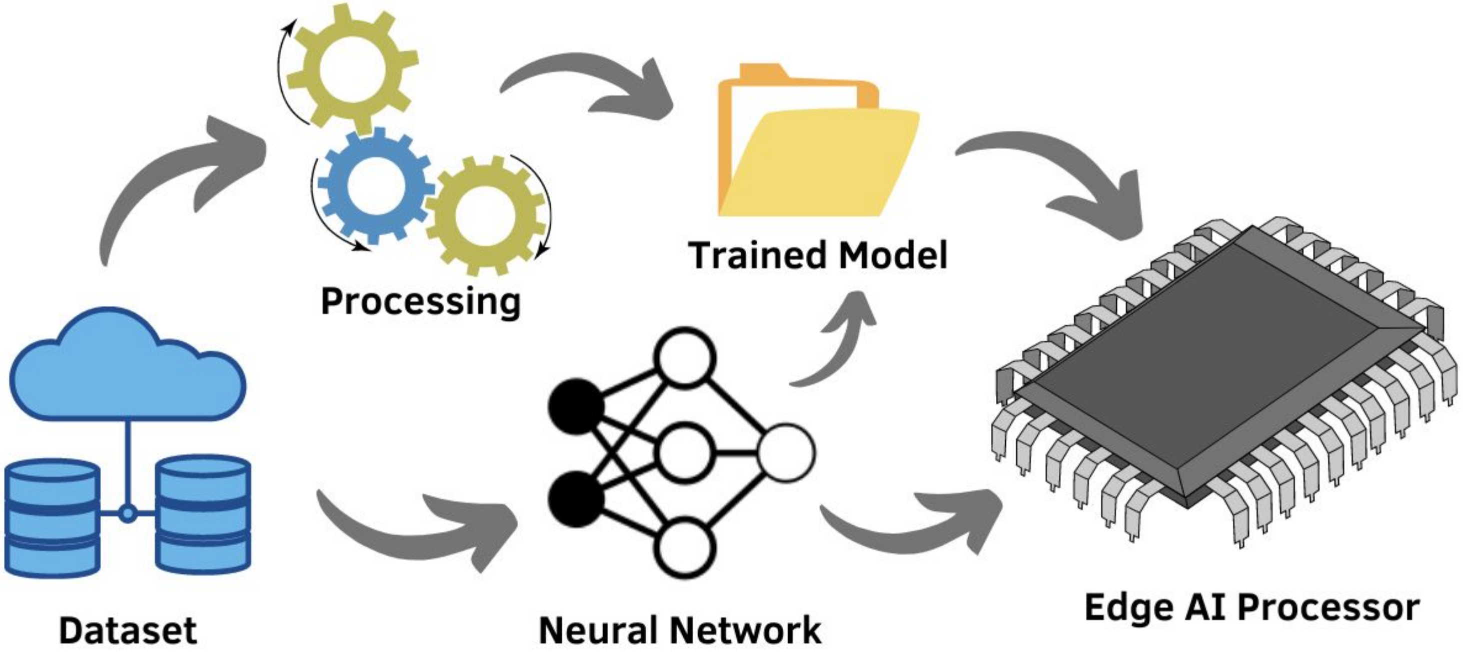
Prediction of picking point location

Jetson TX2

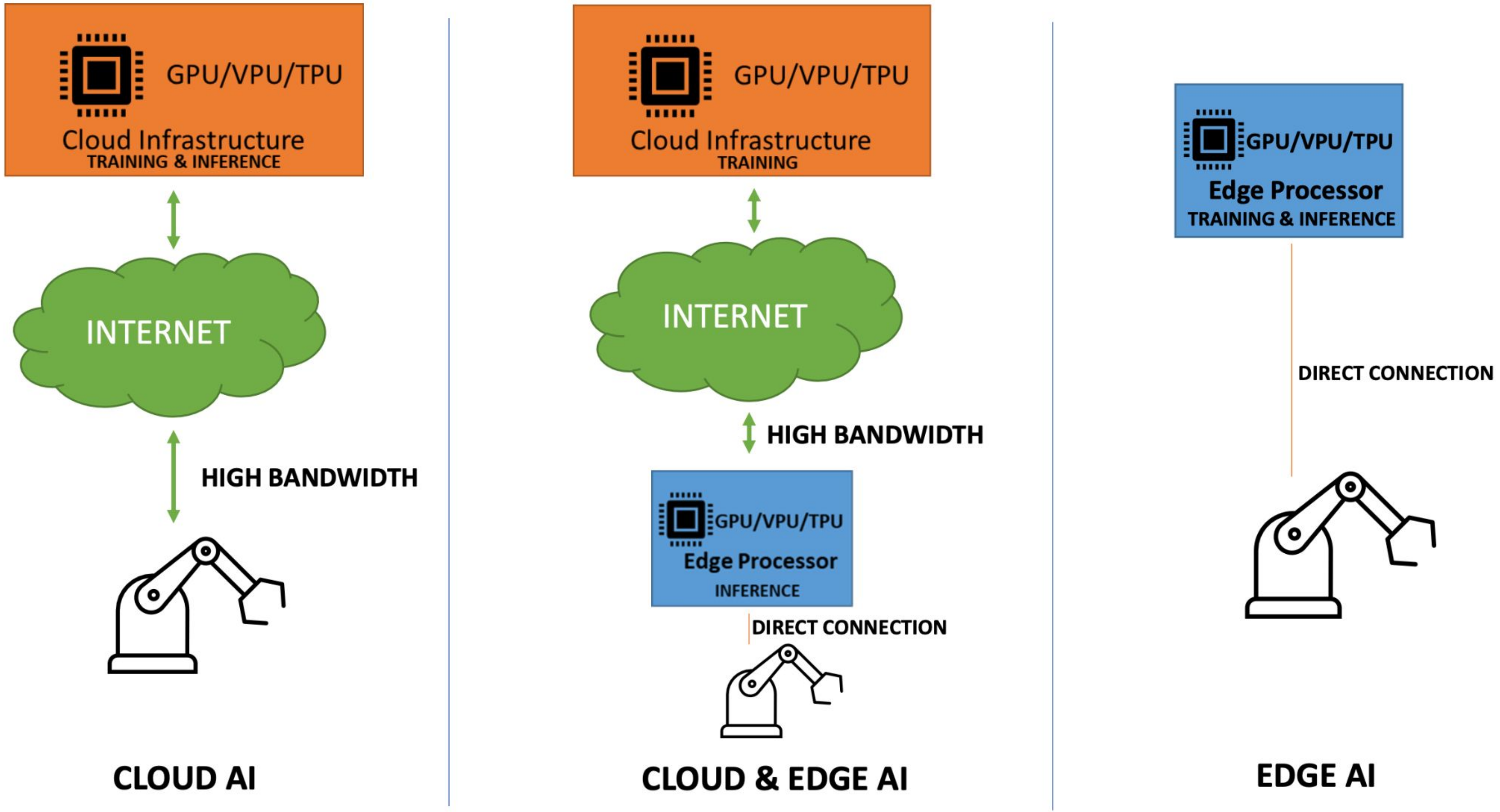
Implemented on TX2



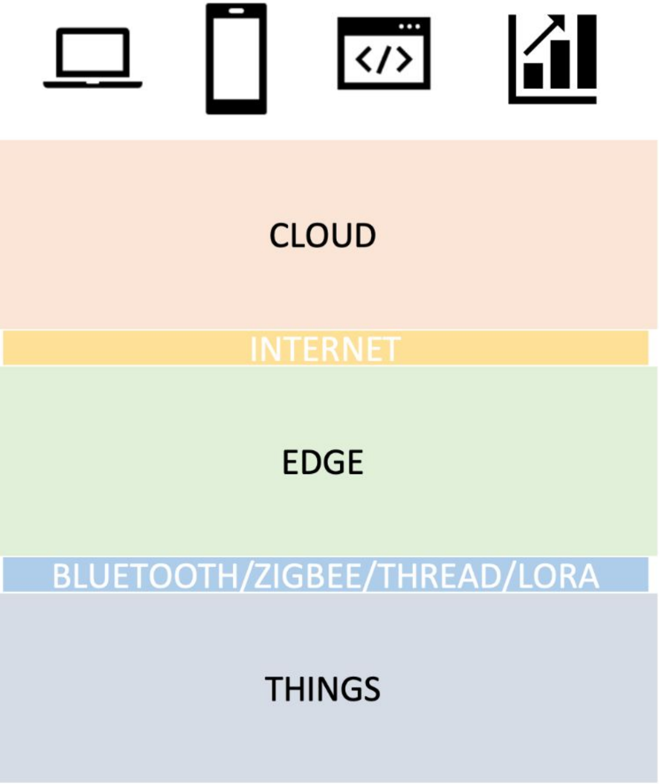
ML/DL: Edge AI


Processing
Trained Model
Dataset
Neural Network
Edge AI Processor

ML/DL: Edge AI Architecture



ML/DL: Edge AI Architecture



CLOUD

INTERNET

EDGE

BLUETOOTH/ZIGBEE/THREAD/LORA

THINGS

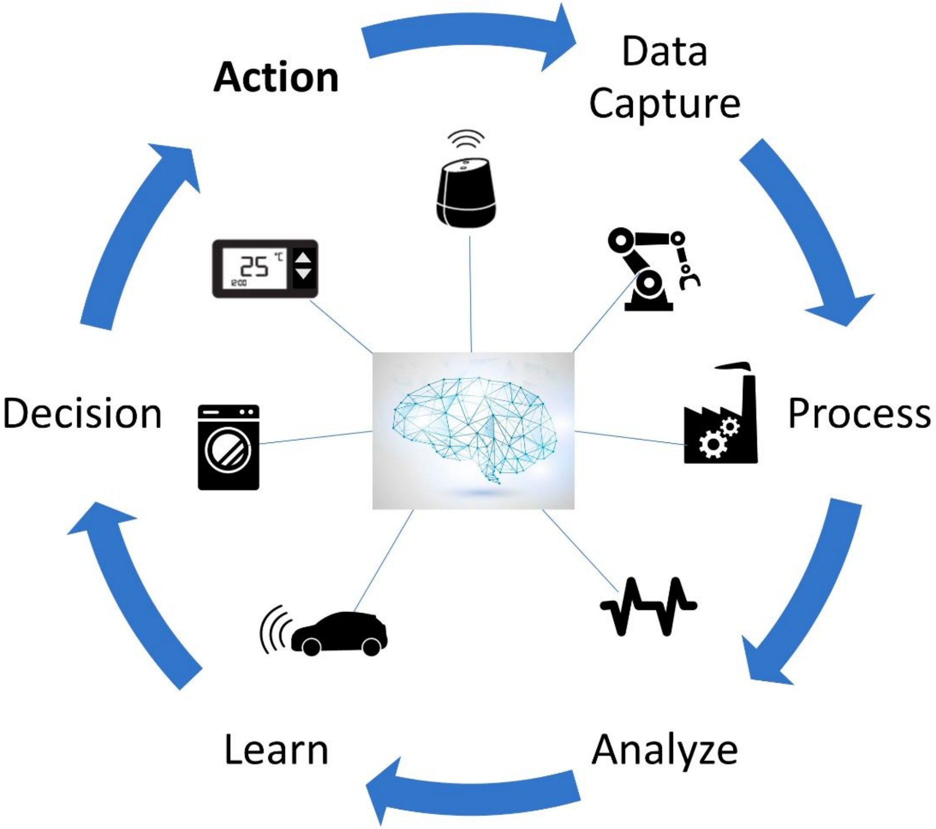
Backend Processing
Storage/Database
Web application/Platform

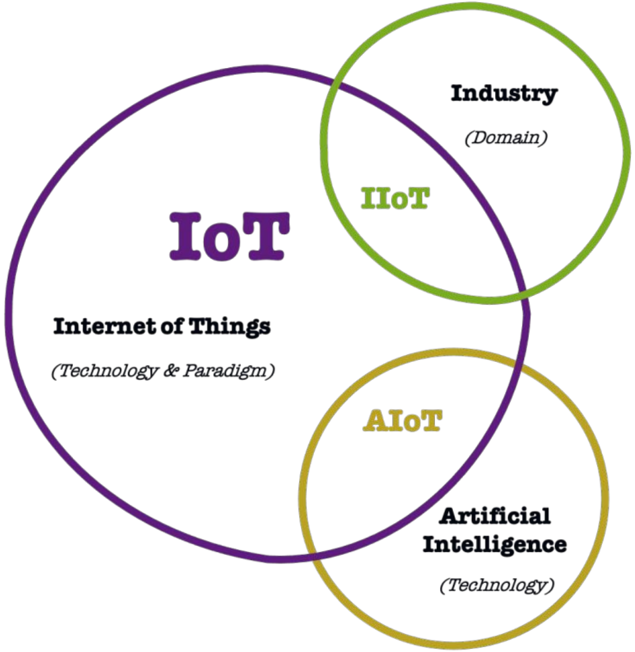
Gateways
Aggregation/Internet Connectivity
Computing close to application
Storage/Preprocessing

Sensors
Processing capability
Internet Connectivity

Image Source: Wikipedia , Creative commons

AIoT: AI of Things



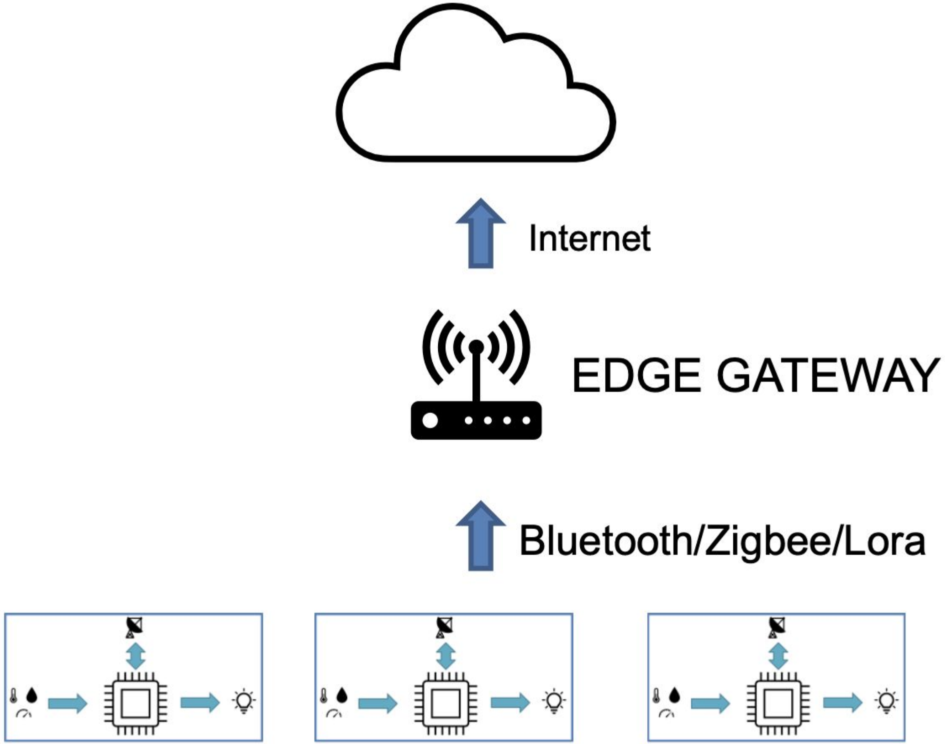


AIoT: Edge Computing

- IoT devices have a MPU/MCU and sensors
  - They can collect and transfer data, as well as perform simple tasks
  - ***AIoT:*** *the tasks are smart or intelligent*

- AIoT devices are used as
  - Smart home and fitness devices
  - Industrial settings

- Computations are done
  - RTOS: on device
  - Heavy DL models: on cloud servers
  - ***Edge computing:*** *running models closer to where the data is being generated*

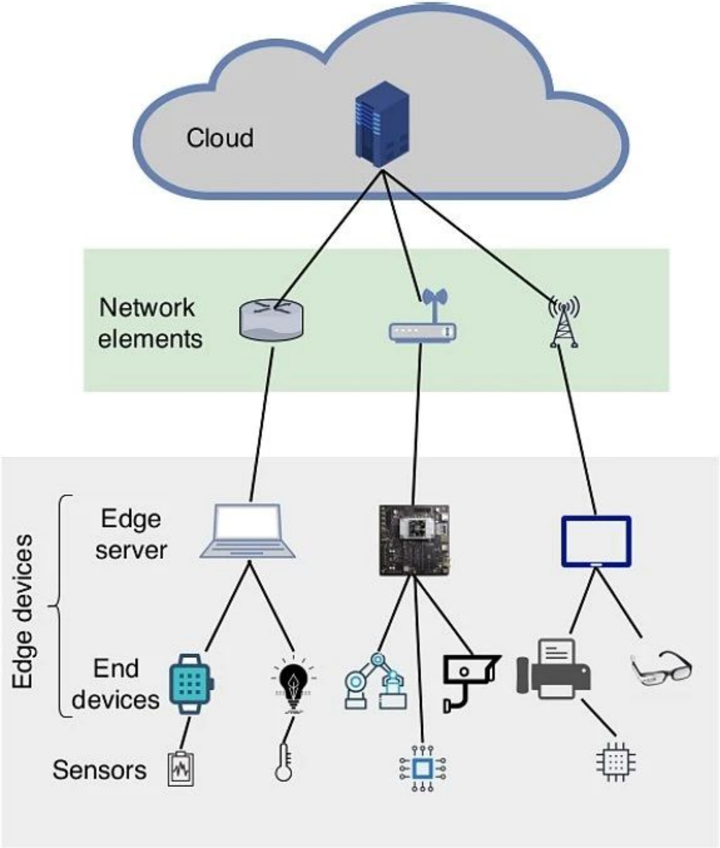


Edge Computing: Advantages

- Reduced latency and cost
- Reduced internet bandwidth and connectivity

- Increased security
  - Becoming increasingly important
  - Most defense and sensitive applications prefer edge AI and minimize data communication

- Reduction in dependence on cloud services
- Rapid growth on Edge AI hardware
  - MCUs / MPUs
  - AI boards



Edge Computing: MPUs vs MCUs

MPU VS MCU

MICROPROCESSOR UNIT

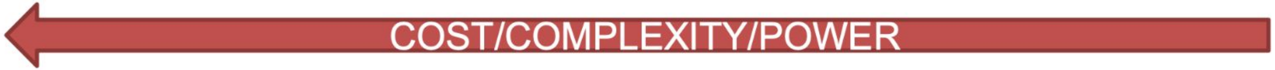
MICROCONTROLLER UNIT

1. Operating System Support
2. External memory and peripherals(PCB)
3. High Power Consumption
4. Memory in Megabytes to Gigabytes
5. Multicore/Multitasking
6. Not always Realtime
7. Optional GPU

1. Barebones/No Operating System/RTOS
2. Memory and Peripherals included
3. Low Power Consumption
4. Memory in Kilobytes to Megabytes
5. Single Core/Dual Core – limited multitasking
6. Realtime & DSP capabilities
7. Lower Cost

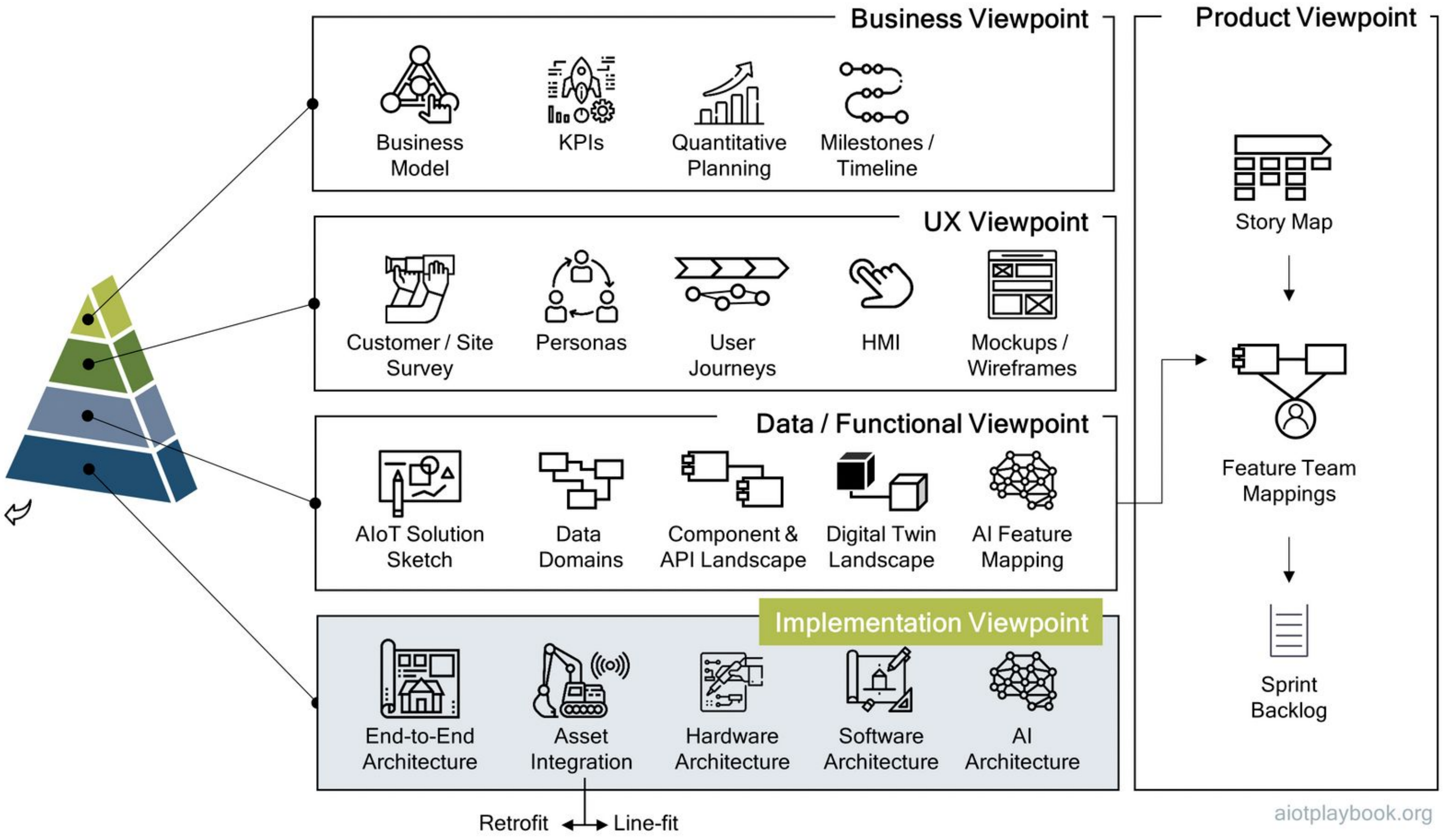
Raspberry Pi
(Single Board Computer)

Arduino
(Embedded Platform)



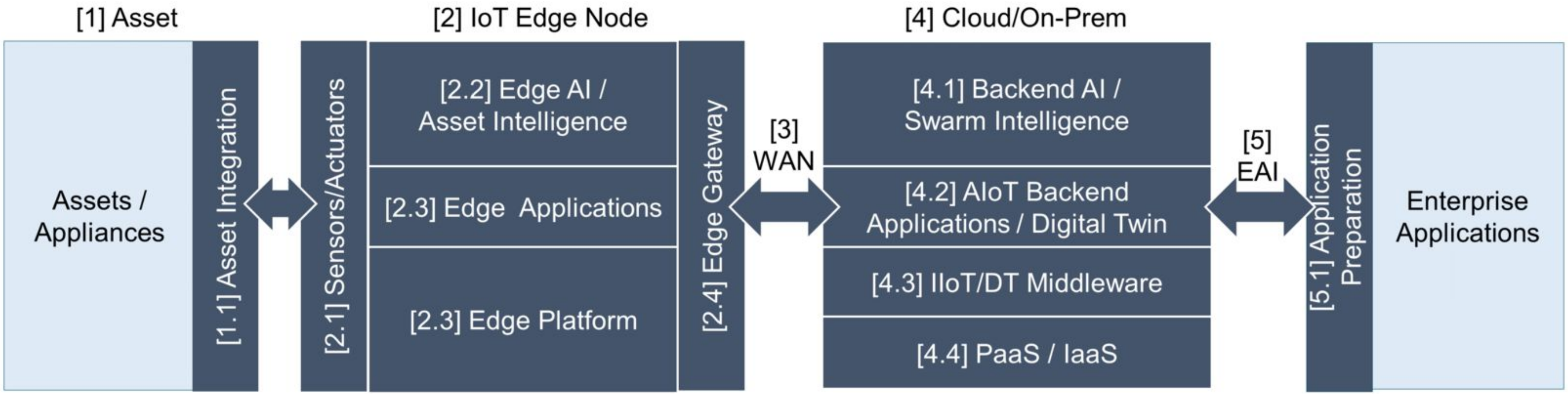
COST/COMPLEXITY/POWER

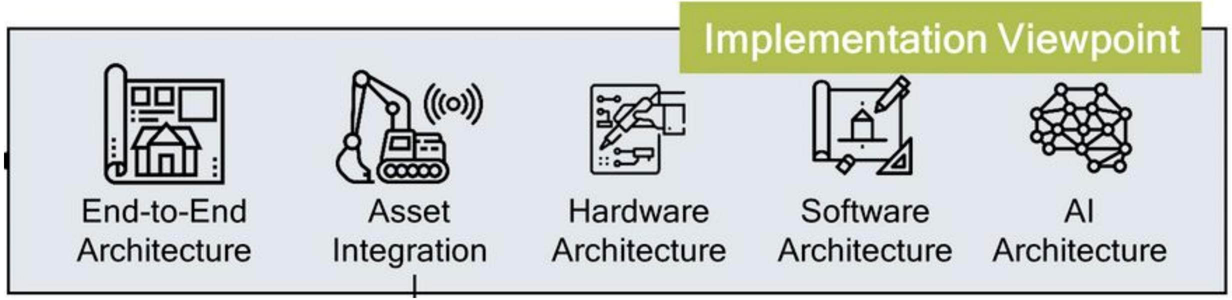
Edge AIoT: Different Viewpoints



aiotplaybook.org

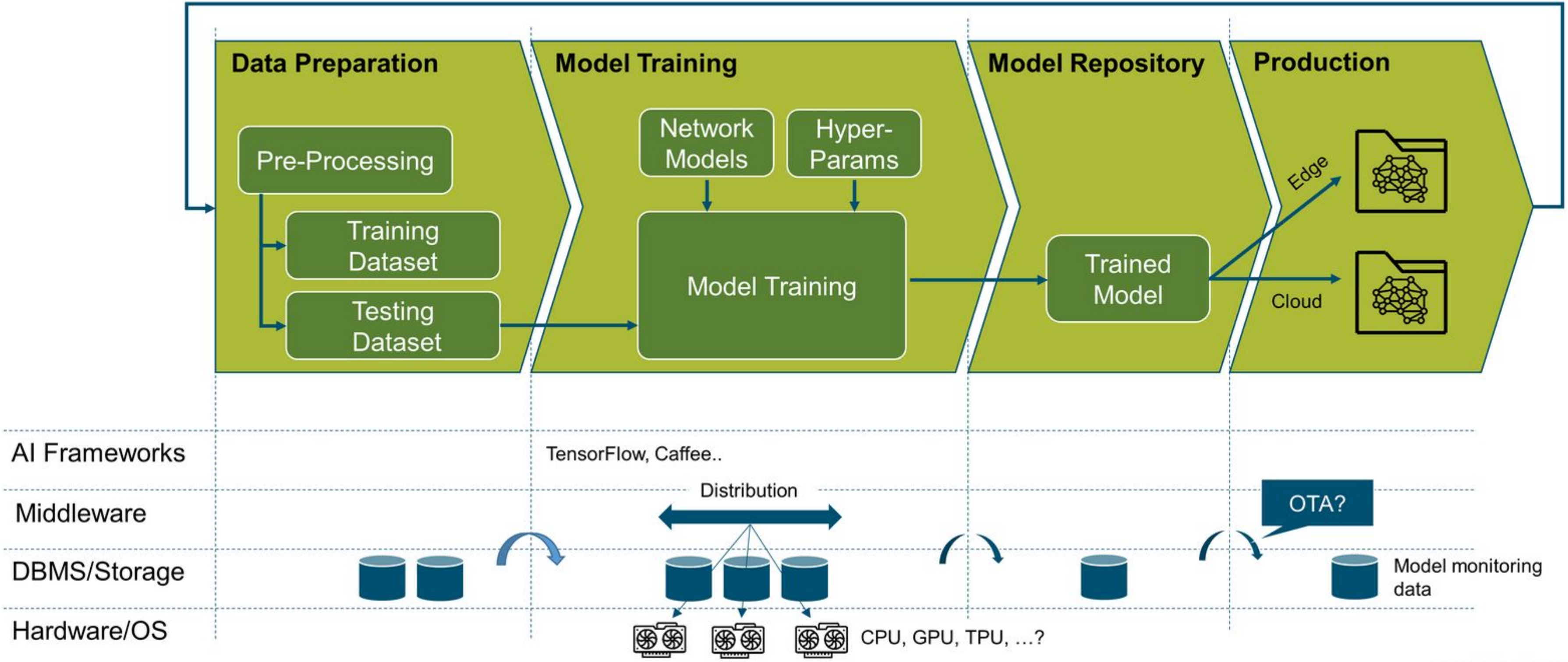
Edge AIoT: Implementation Viewpoint





https://www.aiotplaybook.org/index.php?title=AIoT_Implementation_Viewpoint

Edge AIoT: AI Pipeline



aiotplaybook.org