Basic of Deep Learning

Basic of Deep Learning

Basic of Deep Learning

1) Define of Deep Learning

- Deep learning is a field of machine learning that trains data using artificial neural networks composed of deep layers.
- DL enables the achievement of specific objectives (such as classification, regression, etc.) by configuring artificial neural networks tailored to those objectives and training them on large datasets.

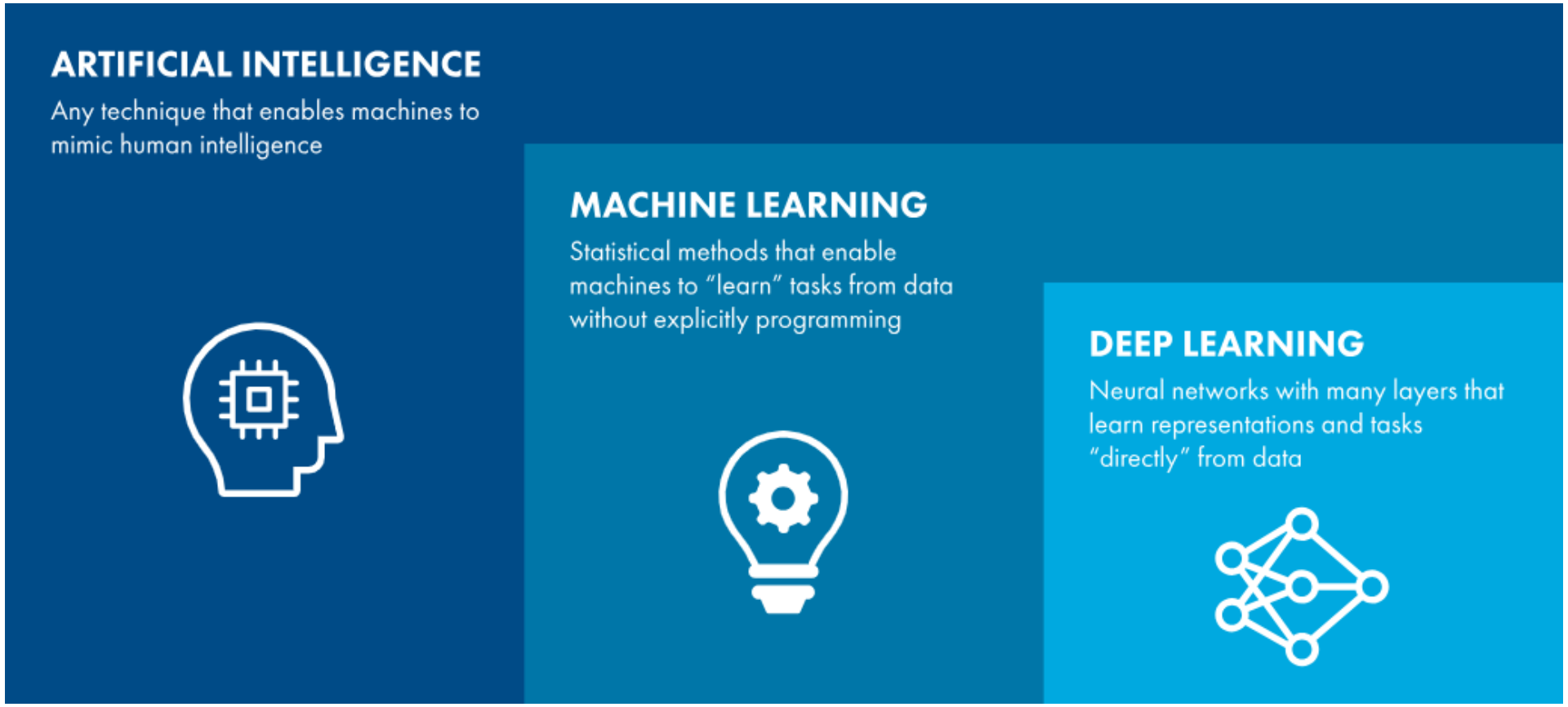


Fig1. concept of AI

2) Principle of Deep Learning

- Artificial neural networks were proposed based on human nerve cells, with neurons (b) as their basic unit.
- Neurons multiply their connections to neurons in the previous layer by their respective weights, sum these values, and then pass the result to the next layer's neurons using an activation function.

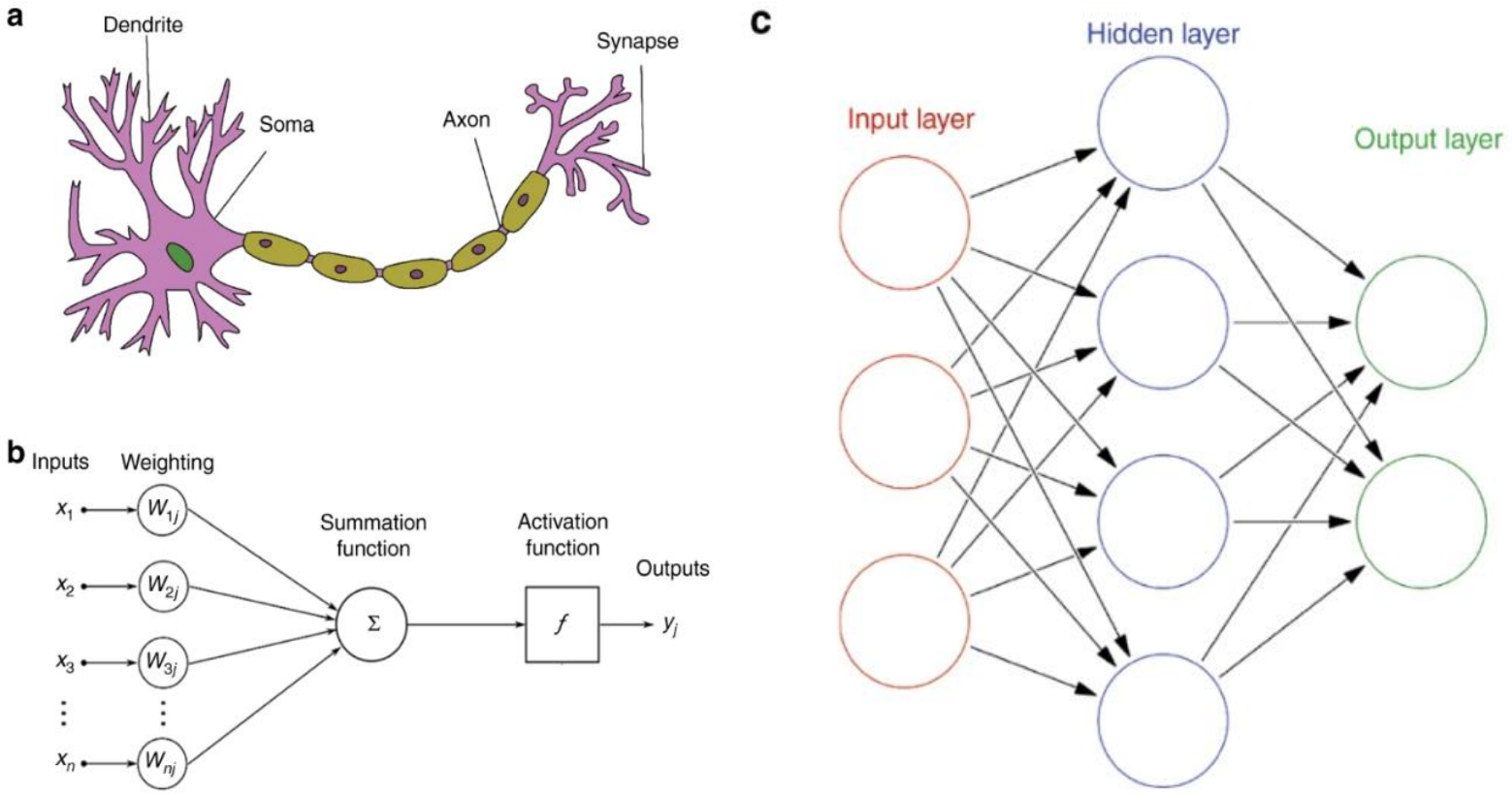


Fig2. Neural network components

2) Principle of Deep Learning

- The activation function enables the network to learn more complex and flexible features in a nonlinear manner.
- Without utilizing activation functions, only linear features are learned.
- If nonlinear functions are not used, only linear operations occur, making deep network structures inefficient.

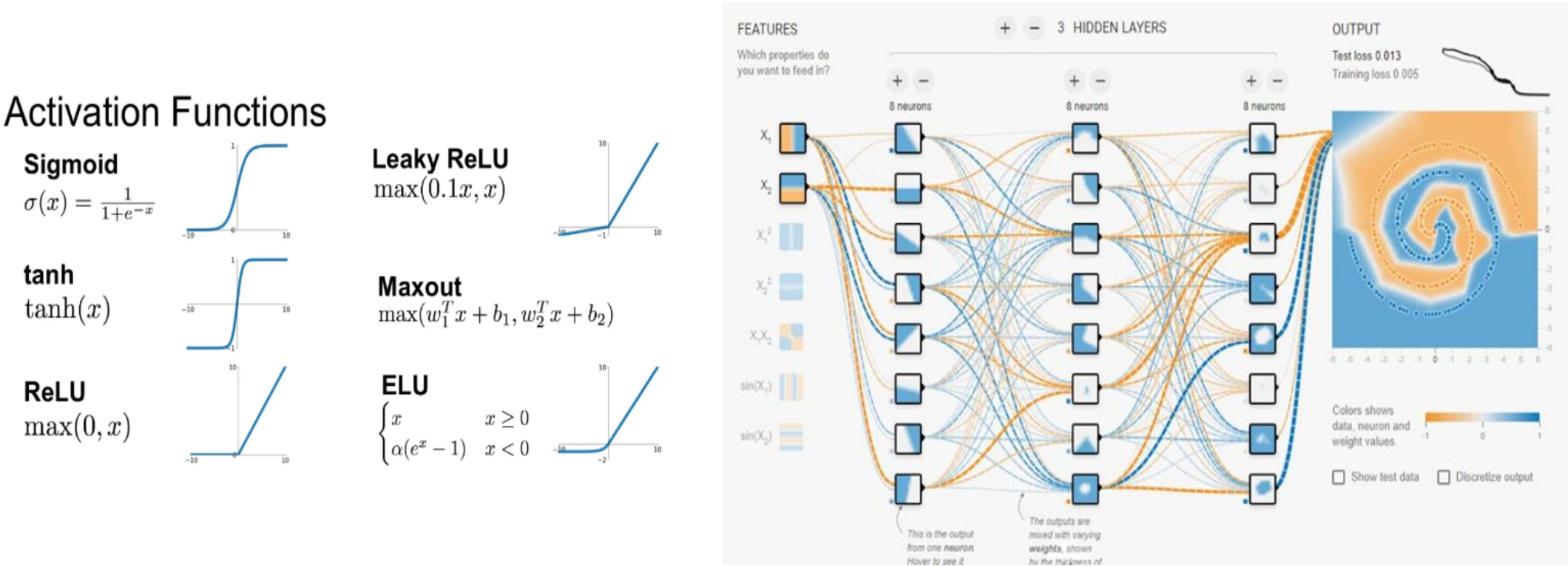


Fig3. Activation Function

2) Principle of Deep Learning

- Deep learning inference is performed through the operations of the network's input, the parameters (weights, biases) that compose the network, and the activation function.
- For a network to operate as intended, the values of its parameters are crucial. When optimal parameter values are set, stable inference becomes possible.
- To this end, deep learning networks undergo a process of finding optimal parameter values through learning.

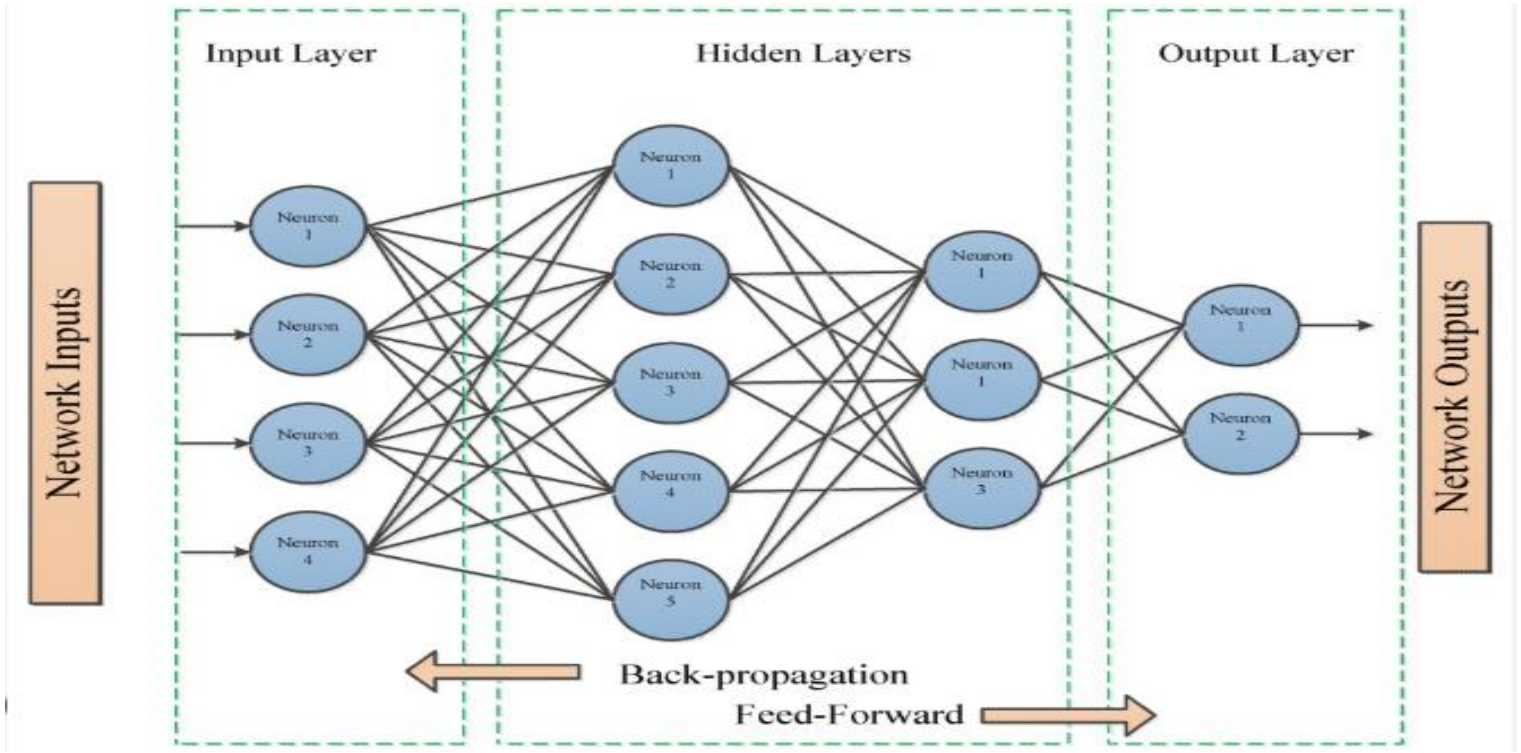


Fig4. Inference in Artificial Neural Networks

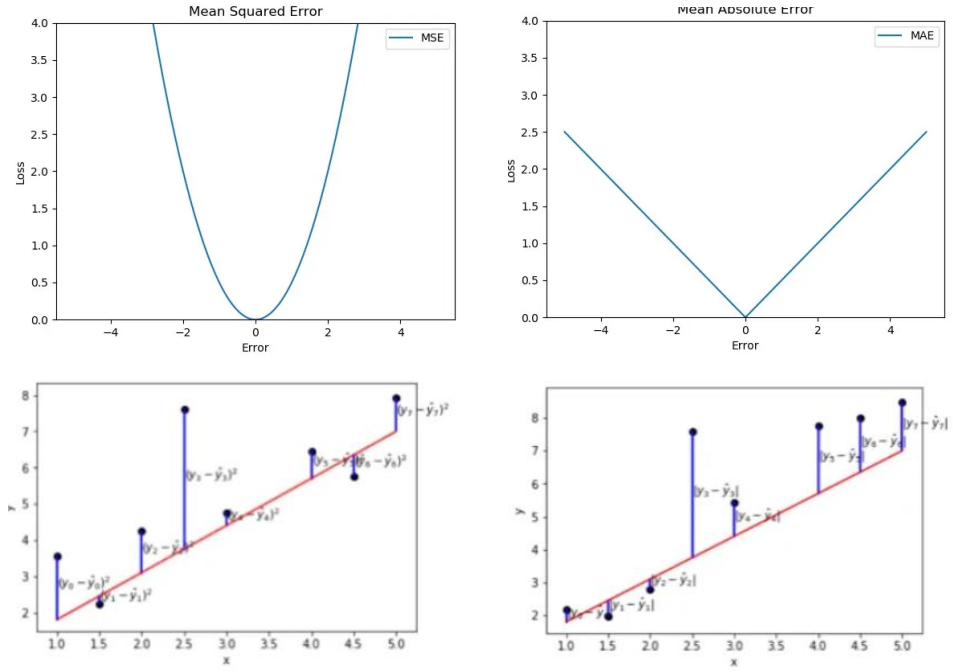
2) Principle of Deep Learning

- Take the partial derivative of the cost function with respect to the parameters that constitute the neural network, and update the values in the opposite direction of the derivative.
- Update the weights using gradient descent

$$\theta := \theta - \alpha \nabla_\theta J(\theta)$$

$$MSE = \frac{1}{N}\sum_{i}^{N}(pred_i - target_i)^2$$

$$MAE = \frac{1}{N}\sum_{i}^{N}|(pred_i - target_i)|$$



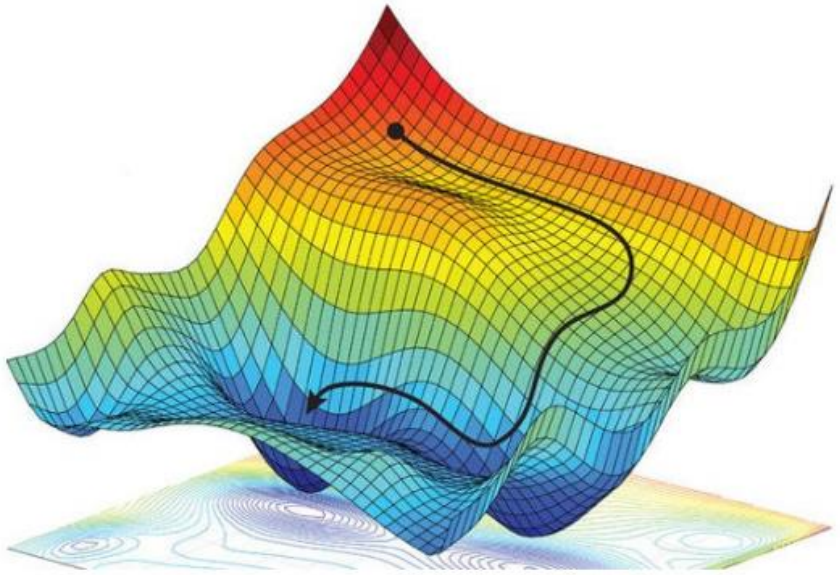
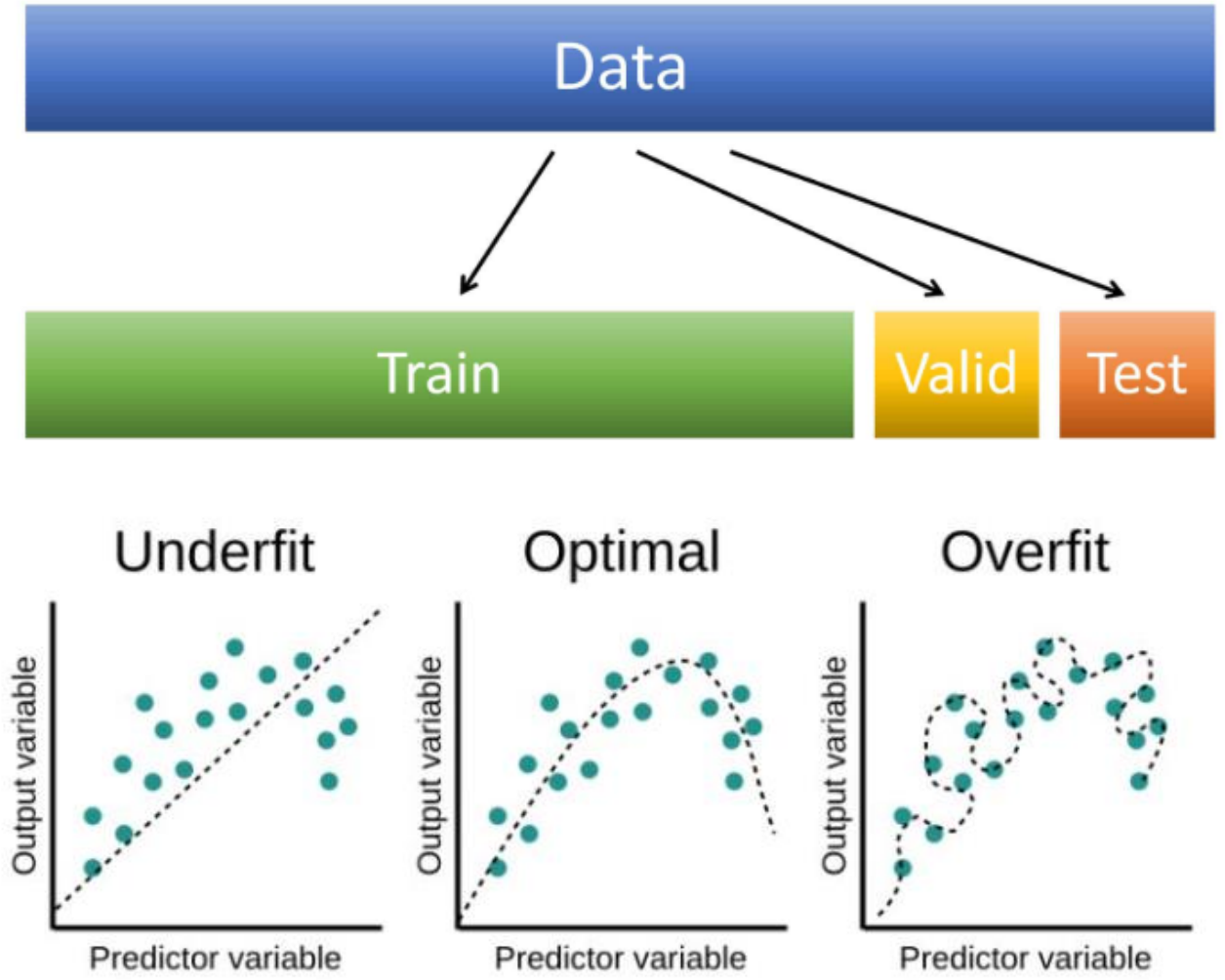


Fig5. Loss Function Training DL Network

2) Principle of Deep Learning



Object Detection

Basic of Object Detection

1) Object Detection Definition

- Providing classification and localization for multiple objects
- Classes represent the type of an object, and bounding boxes represent its position.

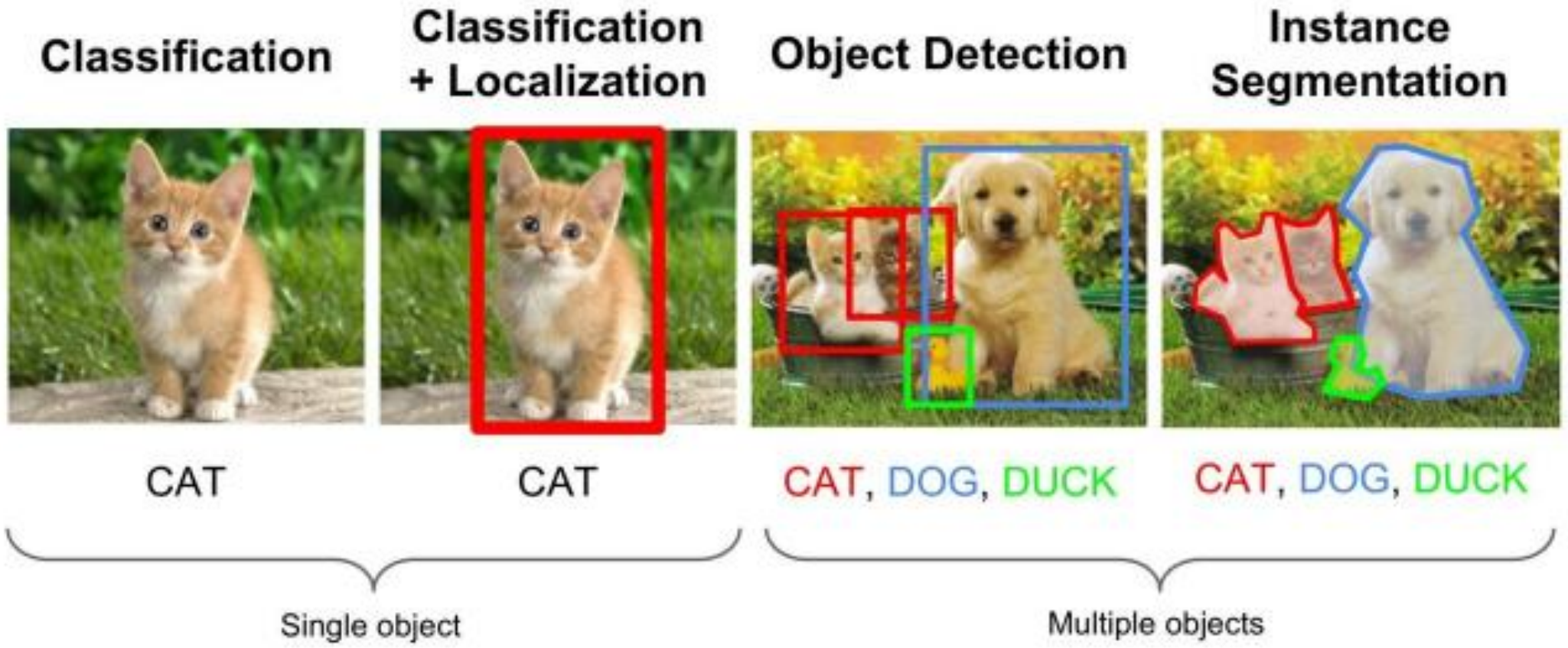


Fig7. concept of Object Detection Definition

1) Object Detection Definition

- For object detection, techniques are categorized into single-stage and two-stage approaches.

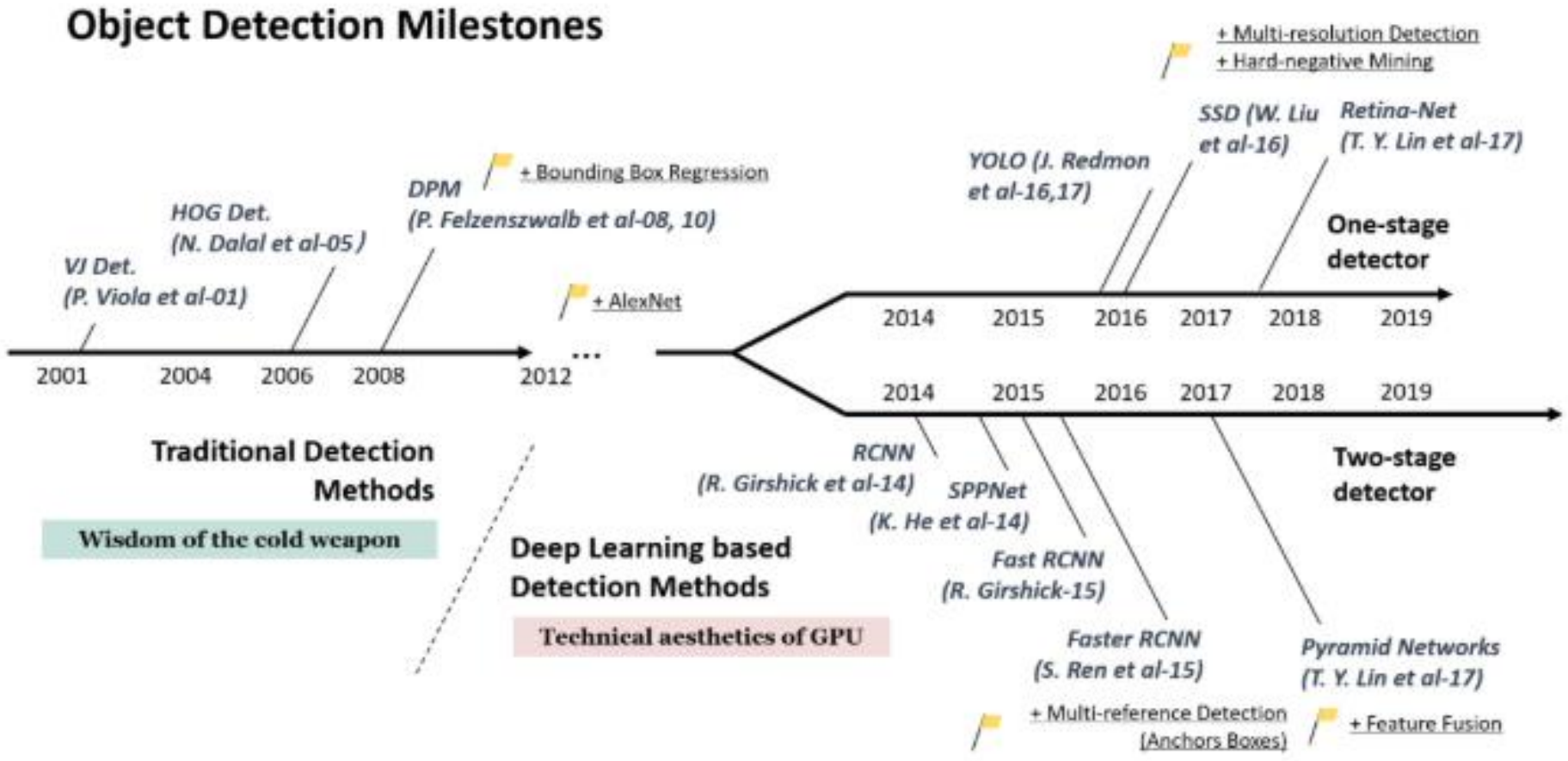


Fig8. concept of Object Detection Definition

2) 1-stage Object Detector

- Real-time object detection is possible
- Faster than a 2-stage system
- Lower detection performance compared to 2-stage
- Ex) RetinaNet, SSD, EfficientDet, YOLO

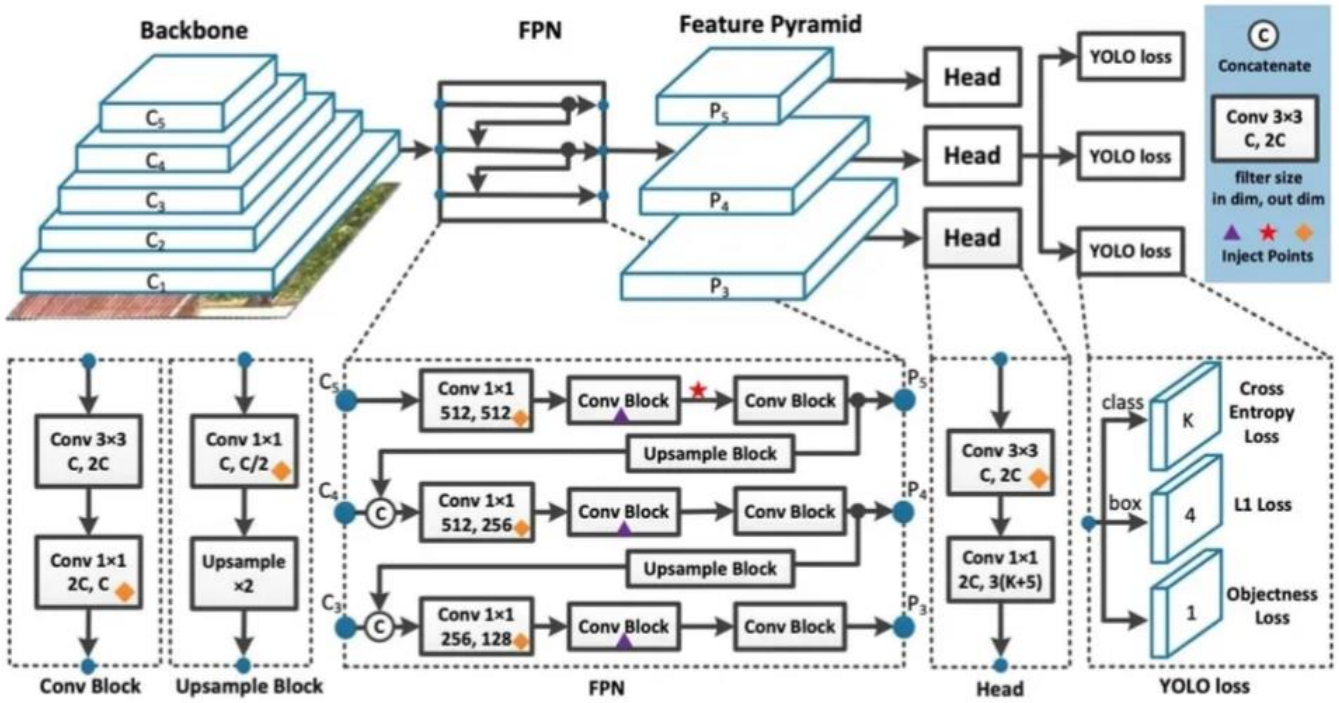


Fig9. 1-stage Object Detector

3) 2-stage Object Detector

- Higher detection performance compared to 1-stage
- Slower than 1-stage
- High accuracy but real-time object detection is difficult due to speed limitations.
- Ex) R-CNN / Fast R-CNN / Faster R-CNN

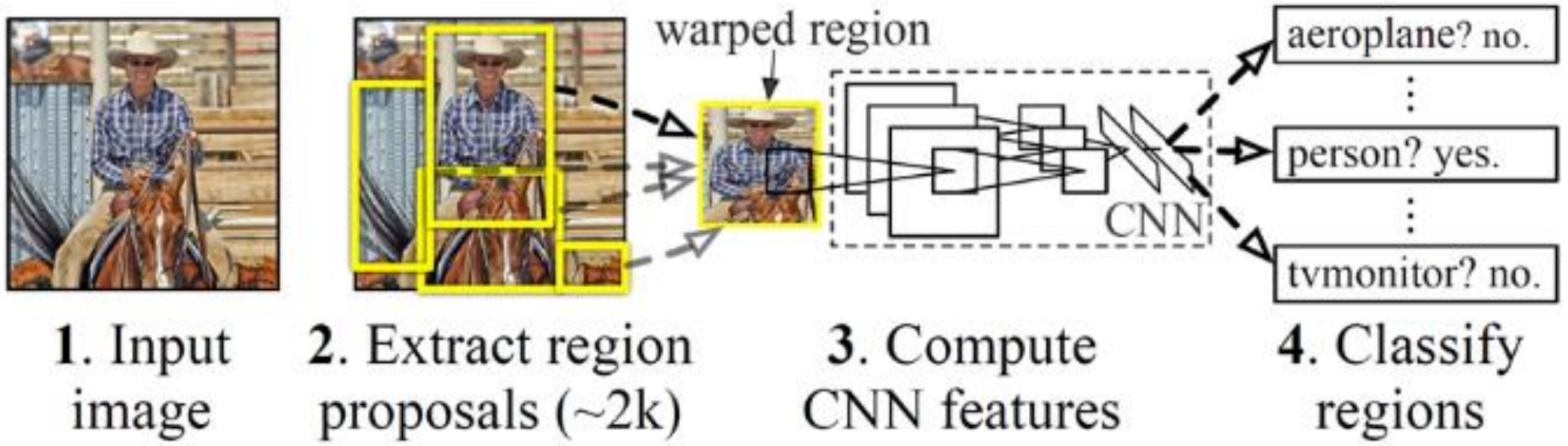


Fig10. concept of Object Detection Definition

4) Application of Object Detection

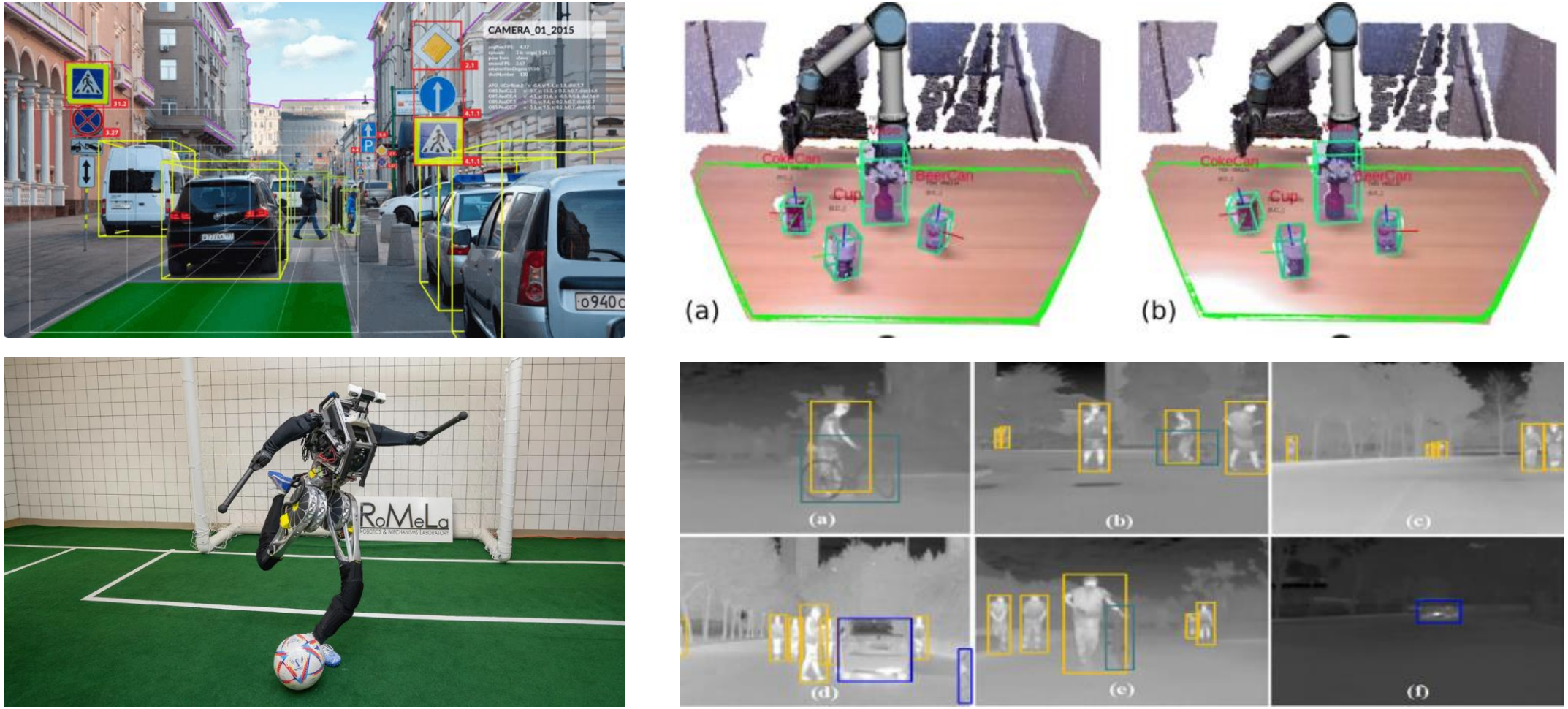


Fig10. concept of Object Detection Definition

Go2 Object Tracking

Go2 Object Tracking

ROS2 Humble GO2 Setting

1) go2 package

Git

```
$ git clone https://github.com/unitreerobotics/unitree_ros2
```

Dependencies

```
$ sudo apt install ros-humble-rmw-cyclonedds-cpp
$ sudo apt install ros-humble-rosidl-generator-dds-idl
$ sudo apt install libyaml-cpp-dev
```

Build

```
$ cd unitree_ros2/cyclonedds_ws/
$ source /opt/ros/humble/setup.bash
$ colcon build
```

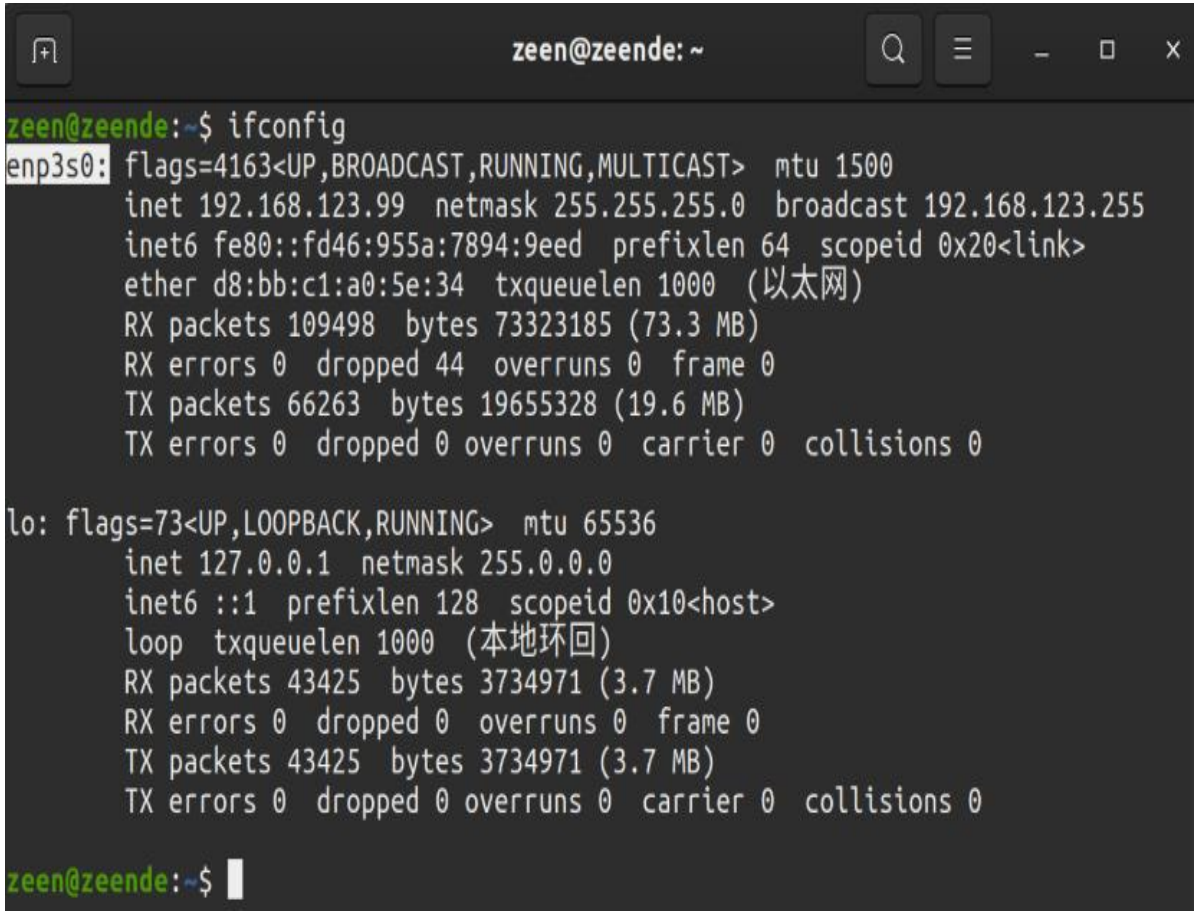
Intsall net-tools

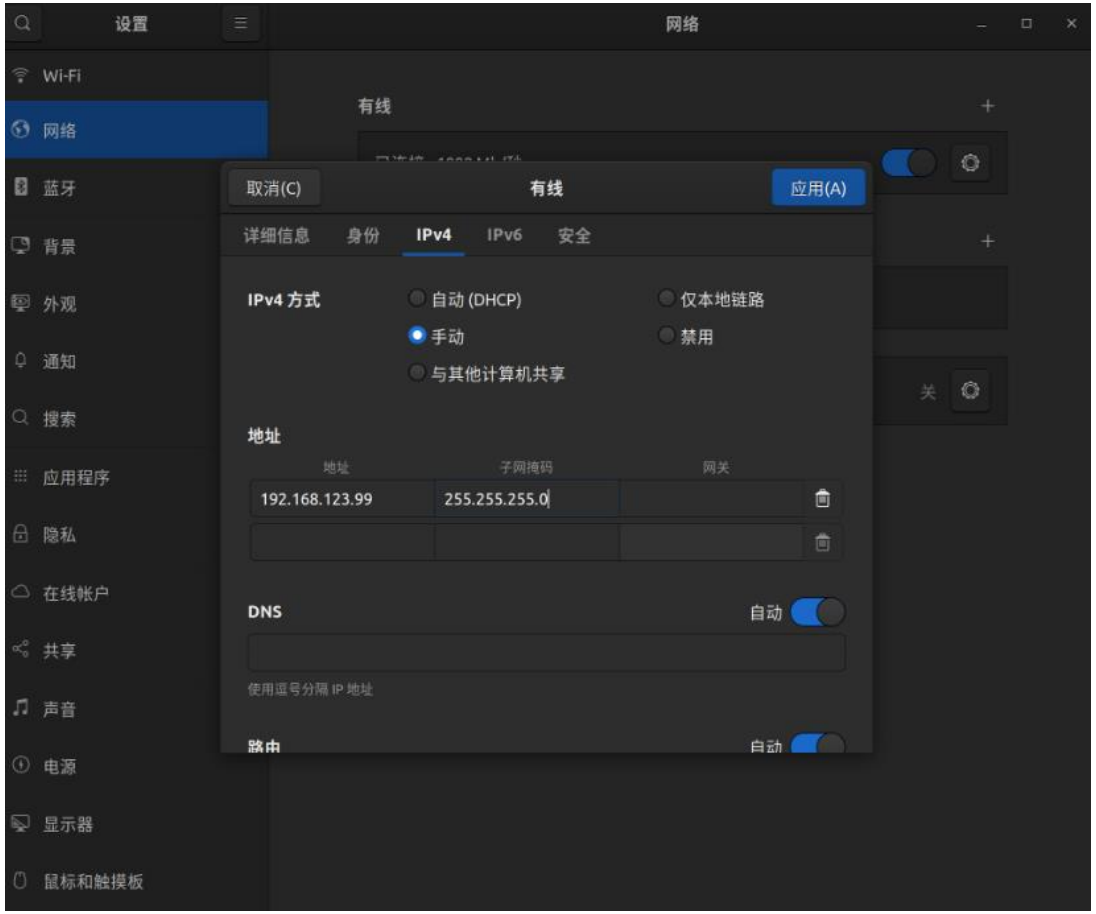
```
$ sudo apt install net-tools
```

ROS2 Humble GO2 Setting

1) go2 package

Network Setting





ROS2 Humble GO2 Setting

1) go2 package

Setup.sh Setting

```
$  gedit ~/unitree_ros2/setup.sh
```

```
#!/bin/bash
echo "Setup unitree ros2 environment"
source /opt/ros/foxy/setup.bash
source $HOME/unitree_ros2/cyclonedds_ws/install/setup.bash
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
export CYCLONEDDS_URI='<CycloneDDS><Domain><General><Interfaces>
                            <NetworkInterface name="enp3s0" priority="default" multicast="default" />
                        </Interfaces></General></Domain></CycloneDDS>'
```

run

```
$  source ~/unitree_ros2/setup.sh
```

ROS2 Humble GO2 Setting

2) go2 Object Tracking Package

Git

```
$  source /opt/ros/humble/setup.bash
$  cd ~/coss_ws/src
$  git clone https://github.com/RASLab-sjbyun/jetson_yolo_msg.git
$  git clone https://github.com/RASLab-sjbyun/second_week.git
$  cd ~/coss_ws
$  colcon build
$  source install setup.bash
```

Jetson SSH

```
$  ssh lab@192.168.123.111
$  ros2 launch realsense2_camera rs_launch.py
$  ------------------------------------------------------------------------
$  ssh lab@192.168.123.111
$  cd yolo_ws/
$  source install/setup.bash
$  ros2 run jetson_yolo cv_bridge_yolo_ros_ssh_node
```

ROS2 Humble GO2 Setting

2) go2 Object Tracking Package

Yolo topic sub

```
$ source /opt/ros/humble/setup.bash
$ cd ~/coss_ws/
$ source install setup.bash
$ ros2 run rqt_image_view rqt_image_view
$ ------------------------------------------------------------------------------
$ source /opt/ros/humble/setup.bash
$ cd ~/coss_ws/
$ source install setup.bash
$ ros2 topic echo /Jetson_Yolo_Detection
```

Tracking node

```
$ source /opt/ros/humble/setup.bash
$ source ~/unitree_ros2/setup.sh
$ cd ~/coss_ws/
$ source install setup.bash
$ ros2 run second_week yolo_control_node
```