

# 10. Visual Inspection (Deep learning)

---

## 10. Visual Inspection (Deep learning)

1. Program function description
2. Introduction to Deep learning
3. Program reference path
4. The program starts

## 1. Program function description

---

After the program is started, the car can use the deep learning model to automatically follow the path line of [black]. Applicable to similar scenarios as shown in the following figure,

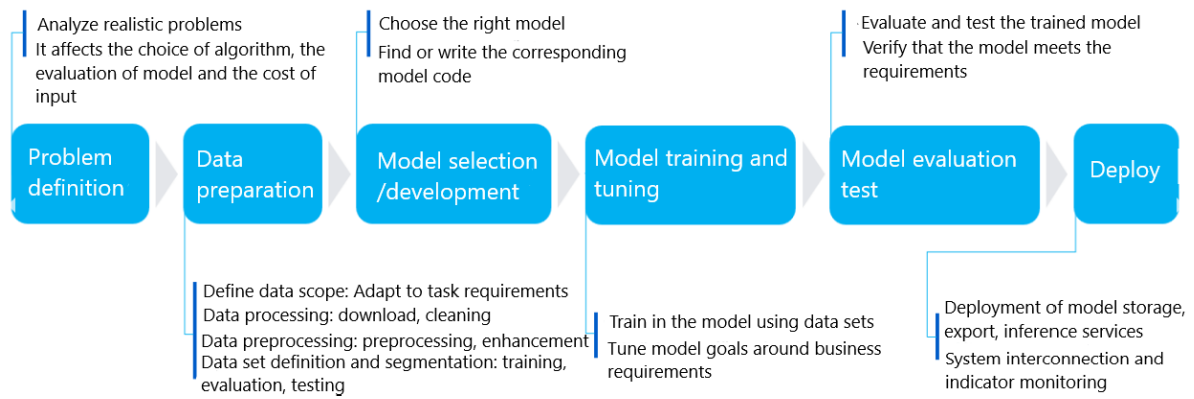


The [L1] button on the handle locks/turns on the car's motion control. When motion control is enabled, the function is locked; This function can be turned on when the motion control is locked.

## 2. Introduction to Deep learning

---

Deep learning allows machine vision to adapt to more changes than traditional image processing, improving accuracy in complex environments. The following is the basic flow of deep learning.



The core purpose of machine learning is to help us solve problems, which can be broken down into six main steps:

- **Problem Definition** : What is the problem we are trying to solve? For example, the visual line patrol here, it is necessary to identify the position of the line in the image.
- **Data preparation** : Start preparing data for the problem to be solved. For example, it is necessary to prepare photos of various line patrol scenes for machine learning.
- **Model selection/Development** : Model is a set of processes for processing data, that is, we often hear about CNN convolutional neural networks, GAN generative adversarial networks, RNN recurrent neural networks, and so on.
- **Model training and tuning** : Put the data into the model and train the optimized parameters, which can be understood as the process of machine learning.
- **Model Evaluation Test** : Just like a quiz, we give some data to a trained model to see how well it works.
- **Deployment**: After everything is ready, you can put the trained model on the robot, that is, formally impart the knowledge to a robot, and it can solve the problem raised before.

In the car side code, a trained line model has been deployed, that is, you can skip the machine learning steps mentioned above and go straight to deployment.

### 3. Program reference path

After SSH connection car, the function source code is located in,

```
/userdata/yahboomcar_ws/src/yahboomcar_deeplearning/line_follower_perception/src/line_follower_perception.cpp
```

### 4. The program starts

Adjust the MIPI camera Angle downward so that it can recognize black path lines and the background is not cluttered.

After SSH connects to the car, the terminal runs,

```
ros2 launch line_follower_perception linefollower_perception_launch.py
```

You can see that the car can follow the black path line.

