Visual Line Patrol (Deep Learning)

Visual Line Patrol (Deep Learning)
1. Program Function Description
2. Introduction to Deep Learning
3. Program reference path
4. Program startup

Note: This case needs to be run as root user, and administrator privileges are required to call the MIPI camera!

> **Switch to root user**

```
su root
```

Password: yahboom

1. Program Function Description

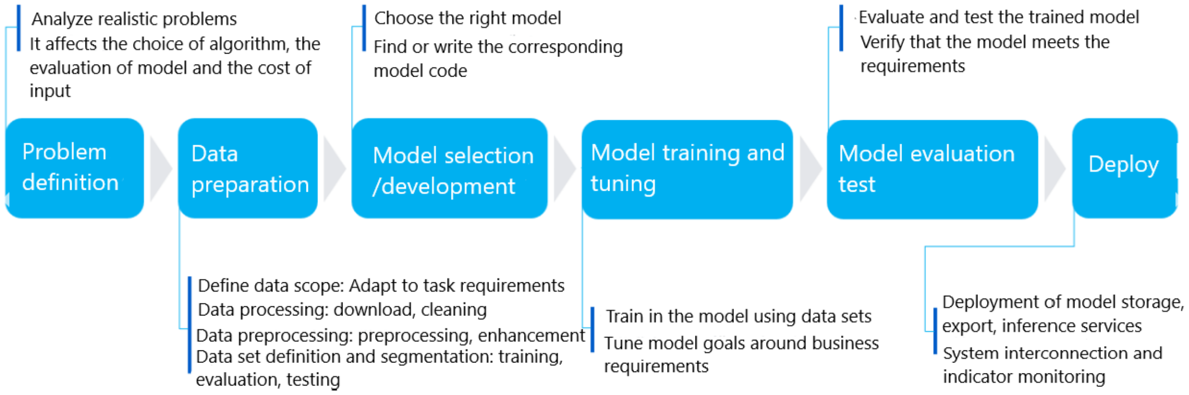
After the program is started, the car can use the deep learning model to automatically follow the [black] path line. Applicable to similar scenarios as shown in the figure below,

The [L1] button on the handle can lock/open the motion control of the car. When the motion control is turned on, the function will be locked; when the motion control is locked, the function can be turned on.

Note: This case needs to be switched to the root user to run, and administrator privileges are required to call the MIPI camera!

2. Introduction to Deep Learning

Compared with traditional image processing, deep learning can enable machine vision to adapt to more changes, thereby improving the accuracy in complex environments. The following figure is the basic process of deep learning.



Analyze realistic problems
It affects the choice of algorithm, the evaluation of model and the cost of input

Choose the right model
Find or write the corresponding model code

Evaluate and test the trained model
Verify that the model meets the requirements

Define data scope: Adapt to task requirements
Data processing: download, cleaning
Data preprocessing: preprocessing, enhancement
Data set definition and segmentation: training, evaluation, testing

Train in the model using data sets
Tune model goals around business requirements

Deployment of model storage, export, inference services
System interconnection and indicator monitoring

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

- **Problem definition**: What problem do we want to solve? For example, in the case of visual line patrol, we need to identify the position of the line in the image.
- **Data preparation**: Prepare data for the problem to be solved. For example, we need to prepare photos of various line patrol scenes for machine learning.
- **Model selection/development**: A model is a set of processes for processing data, which is what we often hear about: CNN convolutional neural network, GAN generative adversarial network, RNN recurrent neural network, etc.
- **Model training and tuning**: Put data into the model and train to get the optimal parameters, which can be understood as the process of machine learning.
- **Model evaluation test**: Just like a quiz, we give some data to the trained model to see how it works.
- **Deployment**: After everything is ready, you can put the trained model on the robot, that is, formally impart knowledge to a robot, and it can solve the problem raised before.

In the car code, a set of trained line-following models has been deployed, which means that you can skip the machine learning steps mentioned above and directly enter the deployment phase.

3. Program reference path

After SSH connects to the car, the source code of this function is located at,

```
/home/sunrise/yahboomcar_ws/src/yahboomcar_deeplearning/line_follower_perception/src/line_follower_perception.cpp
```

4. Program startup

Appropriately adjust the MIPI camera angle downward so that it can recognize the black path line 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.