2023.10.17

Design File

From:Group5

Content

circuit schematic

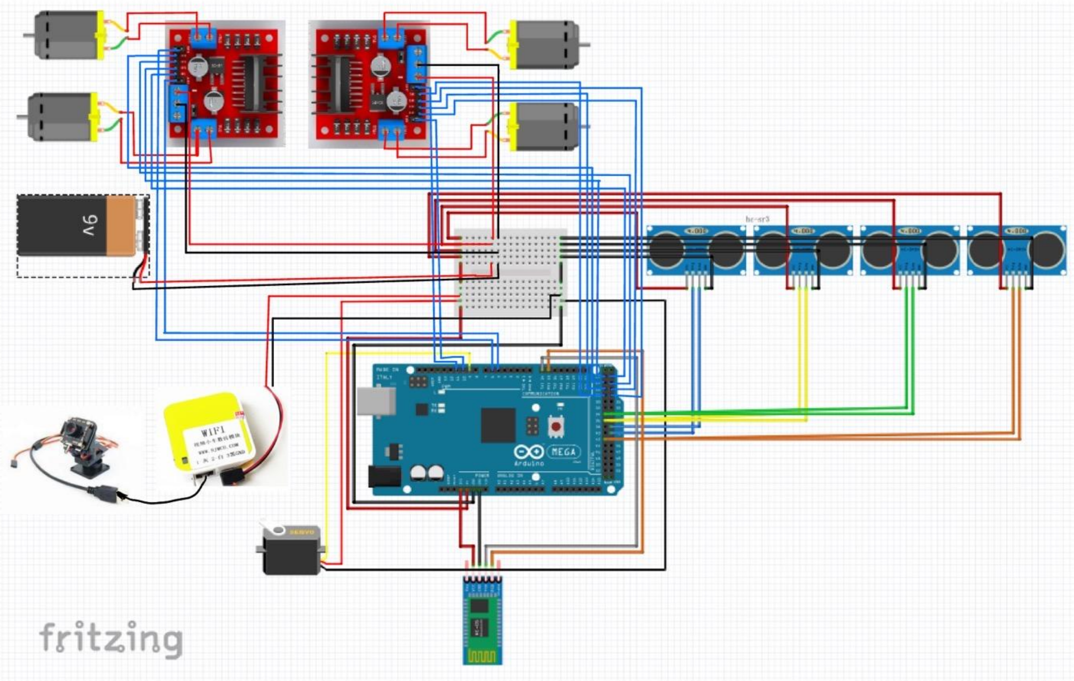


Fig1.Circuit

Flowcharts

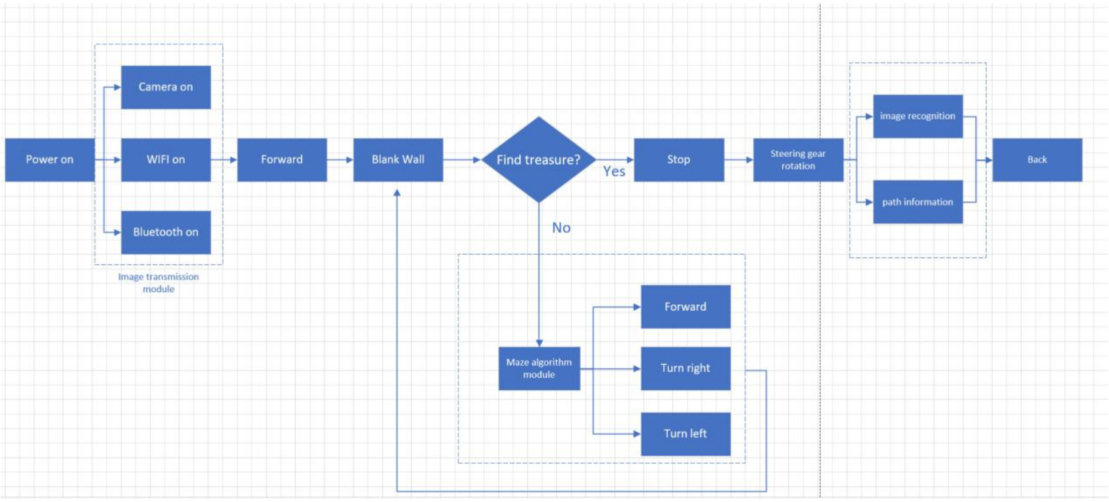


Fig2.Process of Detection

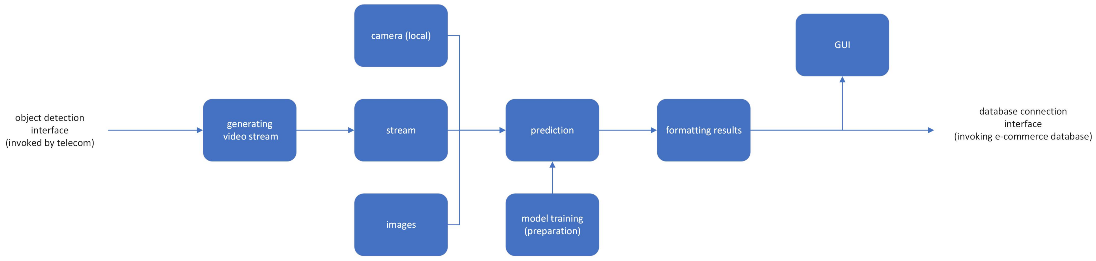
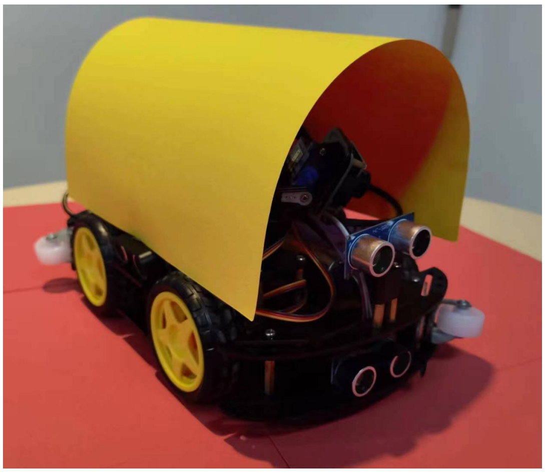

camera (local)
GUI
object detection interface (invoked by telecom)
generating video stream
stream
prediction
formatting results
database connection interface (invoking e-commerce database)
images
model training (preparation)

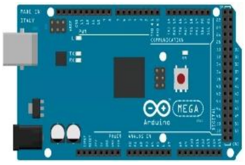
Fig3.Data Analysis

Hardware Specification

1. Appearance



2. Accessories

Arduino Mega 2560*1		The board to control the robot.
L298N motor diver*2		To drive four wheels
HC-SR04		To measure the distance around the robot.

Chassis with 4 motors*1		Carrier.
18560 battery*4		Energy source.
Bread board*4		As the hub of all VCCs and GEDs.
Dupont thread		To connect all the components.
HC-05 Bluetooth module*1		Transfer information

WIFI module*1		Transmit video signal
2 degrees of freedom gimbal and USB webcam*1		Take photos
power bank*1		Powering the wifi

3. General

3.1 Functional description.

The robot realizes obstacle avoidance and simple identification of treasure through four ultrasonic sensors and infrared sensors, transmits video information back and further processing through integrated cameras and Wi-Fi.

All logic is controlled through the Arduino MEGA 2560.

3.2 Instructions for use

1.connect Wi-Fi and Bluetooth.
2.put robot in the maze.
3.turn on the switch.
4.check the information and image received.

Model Training Specification

We use more than 3800 labeled pictures to train the model. Overall epochs are 150. We experience more than 40 adjustments to get 2 best models(shown in directories). We divided data set into 3 parts. 70% for training set, 20% for cross validation set and 10% for testing set. We try to make the model general and not be overfit or underfit. The final 2 models can detect treasures well.

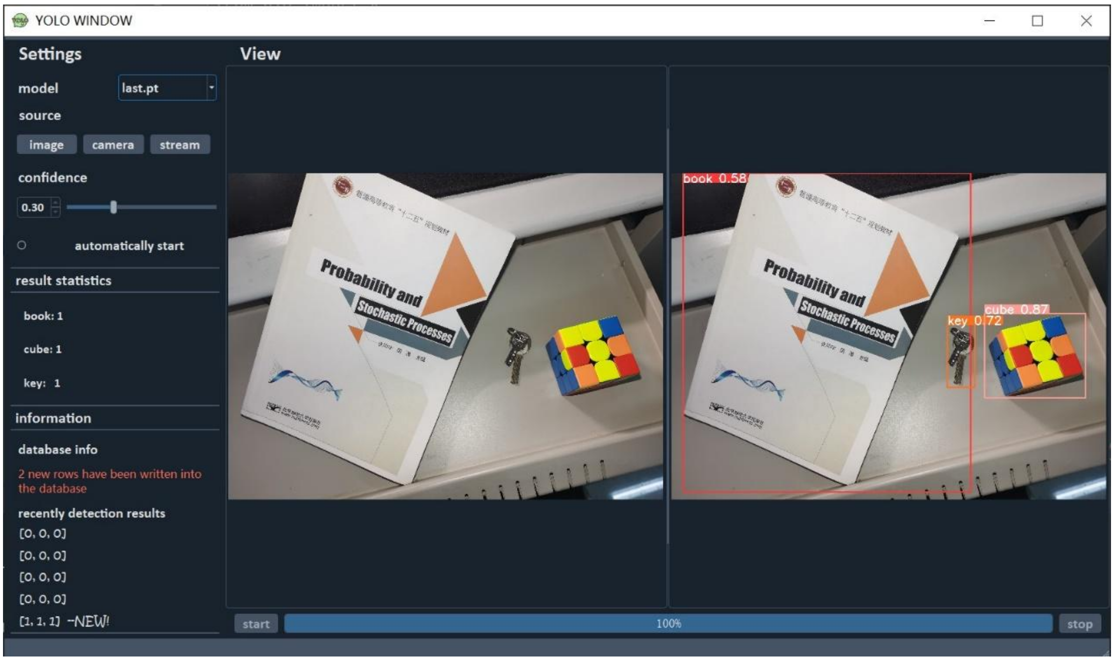
"pre1.py" is used to predict one single pictures. "train.py" is used to train the model. "video.py" is used to run the local camera prediction. "train" and "train2" are models and other results. "yolov8n.pt" is the model provided by official.

GUI -- YOLO WINDOW

This part is same as README.md in the GUI directory.

Introduction

YOLO WINDOW is a YOLOv8 GUI developed with PyQt5 and qdarkstyle(QSS).



YOLO WINDOW

Settings

model last.pt

source

confidence 0.30

○ automatically start

result statistics

book: 1
cube: 1
key: 1

information

database info

2 new rows have been written into the database

recently detection results

[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
[1, 1, 1] -NEW!

View

Fig4.recognize

YOLO WINDOW is a simple but powerful GUI. YOLO WINDOW has clean user interface and high-integrated functions. Every widget has their special functions and can be easily used. It is designed to integrate many useful functions, such as getting data stream, object detection and database connection. You can start your YOLO prediction easily with the help of YOLO WINDOW!

Install

Decompress the zip file, and then configure the environment.

```
pip install -r requirements.txt
```

CAUTION: Please do create a new virtual environment with conda because there are too many items in the requirement.txt and we cannot ensure they will not conflict with the packages you have installed before.

Start

Run the YOLO_WINDOW.py and everything will start.

Documentation

Settings

model

Pre-trained YOLOv8 models are provided to choose. The "last.pt" model is the default model which used more than 3800 labelled pictures to train. It can detect 3 treasures(book, cube, key) with high precision. There are also may other models you can choose, just click the box and chose you want.

You can also use self-trained models. Just place you models in the "model" directory, YOLO WINDOW can automatically detect it.

Source

image

Choose a single picture to do the prediction. The picture must in jpg or png format. It can also predict a directory, but only last picture will be shown on the UI. Results will be saved to "DnB3/ans" directory.

camera

Local camera can be shown on the UI as well as the video that is predicted. All of them are real time! Note that YOLO WINDOW can only invoke the default camera. Other camera need configuration. Notice that no video will be saved.

stream*

Stream button is used to connect the robot with WIFI module and get the video stream(will be saved to "images" directory). The default frame per second is 3(due to the restriction of hardware), but is can be changed by users. Results will be saved to "DnB3/ans" directory.

confidence

This is a parameter that used to do the prediction. You can adjust it with the slide or the double spin box(these 2 widgets can be changed at the same time). Note that please do not make the value too low, because it will do too many wrong predictions and will make the picture in a mess.

automatically start

With this button on, no clicking on the start button is needed.

result statistics

Show the number of treasures.

Information

Important information are shown which are database information & detection results saved.

View

The pictures before prediction and after prediction.

start

start button is used to start a prediction of a single image.

stop

Stop button is used to stop the camera stream.

Connection issues

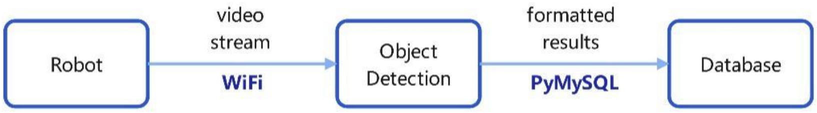


Fig5.Process of connection

getting video stream

-- from the robot developed by Telecom members.

We use WIFI module to get video stream. A Python program is written to get stream in a proper speed. The stream can be easily accessed by the given URL.

connecting to database

-- developed by E-commerce members.

We use PyMySQL module to connect to MySQL database. Use the IP address and all permitted privileges, we can access database developed by E-commerce members. MySQL queries are written to insert data or retrieve data.

All contents are produced by group 5(DnB2023)