

MiniMetroBot

Table of Contents

1. [Description](#)
2. [Project Scope and Targets](#)
 - [In Scope Targets](#)
 - [Out of Scope Targets](#)
3. [Project Structure](#)
4. [Dependencies](#)
5. [Installation of the Final Bot](#)
 - [Option 1: Using Git and Poetry](#)
 - [Option 2: Using Wheel File](#)
6. [Usage](#)
 - [Option 1: Using the Poetry Entry Point](#)
 - [Option 2: Running the Python Script Manually](#)
7. [YOLOv8 Model Training information](#)
 - [Model Information](#)
 - [Training Metrics](#)
 - [Model Training](#)
8. [Contributing](#)
9. [License](#)
10. [Show your support](#)

Description

MiniMetroBot is an object detection bot for the game MiniMetro. It uses YOLOv8 for object detection and OpenCV for image processing.

The recent status of the Bot is that it is capable to detect the main game object of the game Minimetro and it executes before the detection starts a basic automated mouse click and move.

Project Scope and Targets

Main scope of the project is to detect game object which includes the position of the relevant objects and to add the capability to automatically perform user inputs.

Out of scope are further steps to further train another model which actually can play the game.

In Scope Targets

- [x] Game object detection
- [x] Basic user input
- [x] Runs on a resolution which is available on a Macbook Pro M1 Max 2021

Out of Scope Targets (to be handled in a subsequent project)

- [] Fully automated gaming performed by the final bot

Project Structure

```
MiniMetroBot/                                # Root directory
|
|— minimetrobot/                             # Source code
|   |— core/                                # Core modules containing the main bot
|       functionality
|       |   |— __init__.py
|       |   |— ScreenDetector.py           # Class for detecting objects and to mock input
|       |   |— evidences/                  # Folder contains an evidence which shows that the
|       |   |   |— mock_activity_start.mov # Screenvideo activity mock -> startup Minimetro
|       |   |   |   |— detection_evidence.mov # Screenvideo from bot usage -> object detection
|       |   |   |       while gaming
|       |   |   |       |— models/          # Model files
|       |   |   |       |   |— __init__.py
|       |   |   |       |   |— best.pt      # Best model from training exercise
|       |   |   |       |   |— labels.yaml  # List of object labels which can be detected by
|       |   |   |       |       the model
|       |   |   |       |       |— yolo_tuning/ # Main folder which holds objects for model
|       |   |   |       |       |   training (not part of whl installation)
|       |   |   |       |       |   |— images_annotated/ # Annotated training data for YOLOv8 training
|       |   |   |       |       |   |— images_pre_selected/ # Updated version of the raw data images
|       |   |   |       |       |   |— images_raw/         # Raw images
|       |   |   |       |       |   |— model_source/
|       |   |   |       |       |   |   |— yolov8n.pt      # Base YOLOv8 model which is tuned based on the
|       |   |   |       |       |       training data
|       |   |   |       |       |       |— model_tuned
|       |   |   |       |       |       |— yolo_model_tuning.py # Script for model tuning (docstring available)
|       |   |   |       |       |   |— __init__.py
|       |   |   |       |       |   |— __main__.py          # Main program used for active object detection in
|       |   |   |       |       |       live game (docstring available)
|       |   |   |       |       |       |— docs/
|       |   |   |       |       |       |   |— README.md    # Documentation
|       |   |   |       |       |       |   |— README.pdf    # Project description as markdown
|       |   |   |       |       |       |       # Markdown converted to pdf
```

— LICENSE	# License information
— pyproject.toml	# Poetry configuration file

Dependencies

The project has the following dependencies:

- `python`: 3.10.9
- `comet-ml`: 3.33.6
- `ipython`: 8.14.0
- `numpy`: 1.24.2
- `opencv-contrib-python`: 4.7.0.72
- `opencv-python`: 4.7.0.72
- `pillow`: 10.0.0
- `pyyaml`: 6.0.1
- `screeninfo`: 0.8.1
- `torch`: 2.0.0
- `torchvision`: 0.15.1
- `ultralytics`: 8.0.162
- `pyautogui`: 0.9.54

To install the system and its dependencies follow the installation instructions below.

Installation of the Final Bot

Option 1: Using Git and Poetry

1. Clone the repository:

```
git clone https://github.com/TomGFFM/PraxisProjektSemester4_30313.git
```

2. Navigate to the project directory:

```
cd PraxisProjektSemester4_30313/MiniMetroBot
```

3. Install dependencies using [Poetry](#):

```
poetry install
```

Option 2: Using Wheel File

1. Download the latest wheel file, `minimetrobot-0.1.6-py3-none-any.whl`, from the repository.
2. Install the package using `pip`:

```
pip install path/to/minimetrobot-0.1.6-py3-none-any.whl
```

Usage

Option 1: Using the Poetry Entry Point

(notice different naming convention here. Just: "metrobot")

If poetry is installed you can use the following command to run the MiniMetroBot from the environment where the package is installed:

```
poetry run metrobot
```

Option 2: Running the Python Script Manually

Activate the virtual environment and execute the following command:

```
python -m minimetrobot
```

Either of these options will start the MiniMetroBot.

YOLOv8 Model Training information

Model Information

The project folder "yolo_tuning" contains a subfolder "model_tuned". Here are a number of model folders available which contain trained YOLOv8 object detection models.

The final model which was chosen and integrated into the recent version of the Bot is in the folder: "round_optimized_with_Adamax_True_bs_16_20230826_1342"

Training Metrics

The relevant training metrics can be reviewed in the public CometML report for this project.

Click here to visit the report page: [MiniMetroBot CometML Metrics Report](#).

Model Training

The YOLOv8 model training can be performed via the script `yolo_model_tuning.py`. The underlying base model is available in the folder "model_source".

This contains the basic YOLOv8 which is used as a foundation for the detection model training. Review contained docstring for further usage and information.

Contributing

Contributions, issues, and feature requests are welcome! Feel free to check [issues page](#).

License

This project is [BSD](#) licensed.

Show your support

Give a 🌟 if this project helped you!