Rat Path Tracking

Meherab Mamun Ratul 1711786642[1] Nayeem Rahman 1712440642[1] S M Niaz Mahmud 1713079042[1]
Faculty: Nabeel Mohammed [1]

Abstract

Huge number of rats are used for the purpose of experiment throughout the year around the globe. More than 100 million mice and rats are killed in U.S. laboratories every year. They are abused in everything from toxicology tests (in which they are slowly poisoned to death) to painful burn experiments to psychological experiments that induce terror, anxiety, depression, and helplessness. They are deliberately electroshocked in pain studies, are mutilated in experimental surgeries, and have everything from cocaine to methamphetamine pumped into their bodies. They are given cancerous tumors and are injected with human cells in genetic-manipulation experiments. In the experiments the scientists not only observe their physical change but also the behavioral change and their movement is noticed. We are to track the rat's movement and extract graph upon that data.

I. Introduction

While rats are used in the experiments, they are observed with every detail they do. In our project we are to design a such plugin which will track the movement of the rat and produce graph. Here a video will be inserted in the root directory in .avi or .mp4 format. The plugin will process the video. After processing the video it will generate graph which will be beneficial for the scientists and the experiment.

II. Problem

In laboratory, huge number of rats die. The data are not tracked precisely. Human error occurs. This can be escaped. For precisely tracking the data there need to be a better solution. The solution is our Rat Path Tracking plugin.

III. Solution

The Rat Path Tracking plugin will process each and every single frames of the input video. For each frame the plugin will extract co-ordinates in cartesian co-ordinate system. After the successful integration of the data generated from the video, it will produce graph/s. This will highly help the scientists for their observation and so on.

IV. Architecture

We will be implementing **Plug-in Architecture** [2] . The plug-in architecture consists of two components: a core system and plug-in modules. We will be designing a plugin which will be integrated in the laboratory software system. The main key design here is to allow adding additional features as plugins to the core application, providing extensibility, flexibility, and isolation of application features and customs processing logic. The plug-ins are stand-alone, independent components that contain specialized processing, additional features, and custom code that is meant to enhance or extend the core system to produce additional capabilities.This plugin will maintain the **OCP - Open Closed Principle**[3] i.e., Open for Extension and Closed for Modification of SOLID – OOD. Besides we will be using OpenCV for tracking the path.

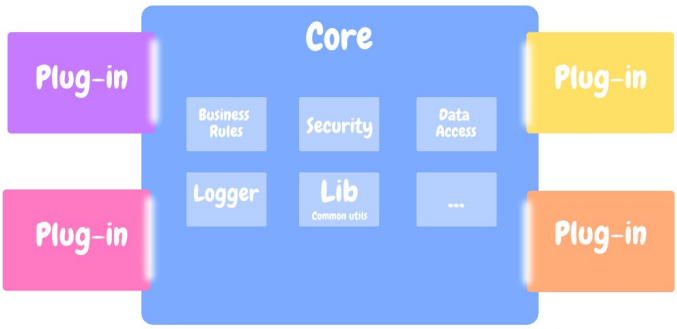

Fig. 1. Architecture.

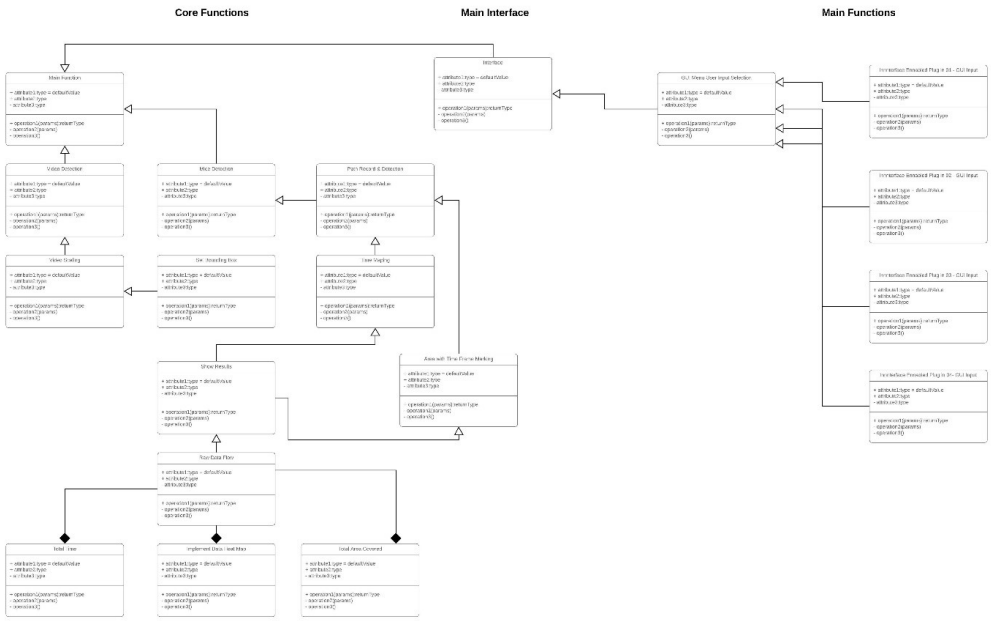

Fig. 2. UML Diagram.

V. UML Diagram

Here's the UML Diagram which we made during the offline class periods.

VI. Process

Here's the total process described below- Input sample video is input.avi placed in the root directory. Output video demonstration is placed on the link below. After processing the input video(s) by the script track.py you will get new directory ¡current date¿distance with CSV table and two sub directories:

- **timing** with processed video(s) as above
- **tracks** with images of animal(s) track(s) as on the left side of processed video

VII. Observation

We tried to make this plugin heart and soul. The plugin has some bugs which we couldn't solve completely. Besides, identification of the co-ordinates was also a tricky task which we couldn't done fully. Another problem is that when the rat was overlapping its path then there was a problem. We couldn't get back the previous path. We couldn't figure it out that why it was happening. But if we keep these bugs aside, the plugin gives us some data which will be useful to extract some information from the scientists' point of view.

VIII. Error/Obstacles

We faced several obstacles during this project completion. One of them was the Covid-19 pandemic. It actually ruined our total workflow. Besides, we had lack of resource. On the other hand, the plugin was giving some errors which was unexpectable.

IX. Conclusion

Our objective was to make a project where we can analyze and observed he behavior pattern of a rodent; the rodent's movement inside a test environment, where we can track its total movement though image analysis the video provided. In order to do that we analyzed the video through the python script of track.py, where for the analysis of the rodent movement the video can be cropped by the mouse pointer to define the 4 corners for the video input and to minimize noise in the image processing. First we implemented The tracking with OpenCV and NumPy in order to make the process simplified we used photo color scaling to make the script track the rodent more accurately. But we still prefer to use a video with black floor to track with more accuracy. Because it's hard to fit the floor of the box within camera frame in the laboratory, the script makes some prepossessing. The script takes a certain part of each frame of the videos. It takes a right square frame[:, w-h:w] of a frame with sides equal to height h of a frame. That's why the box with an animal have to be placed in the right area of camera field of view. For more accuracy we can change this behavior replacing frame[:, w-h:w] by frame [:, 0:w-h]. Input sample video is rat.avi placed in the root directory. After processing the input video(s) by the script track.py you will

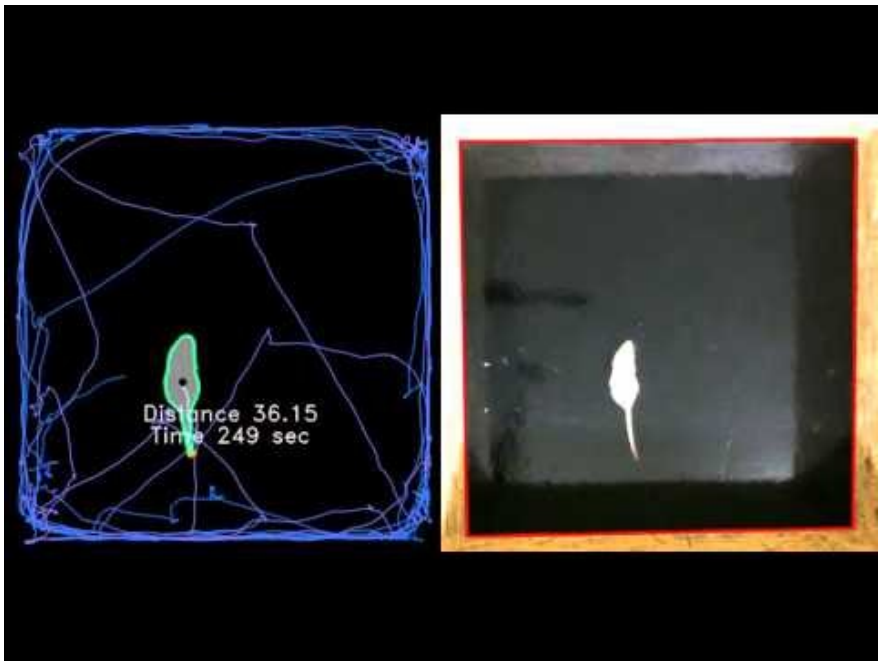

Fig. 3. Input-Output.

get new directory ¡current date¿distance with CSV table with column of Distance and Run Time recorded. But unfortunately due to some dependencies issues with numpy and after the new version integration the script shows error or needs to be freshly set up in the new environment or new pc with adjustment to analyze the video. Sometimes the script runs but the output isn't shown, then a fresh restart and re build of the script is advised. When making this project we learned a lot of things including how we can track objects using OpenCV . How we can use grayscale or video color optimization in order to track the object in a tricky environment for more accuracy when using opencv also. Derive the output from opencv and use the outputs of the analyzed video to customize it to our needs in the script. We also learned a lot about NumPy as we used many large collection of high-level mathematical functions to operate on the arrays we implemented in the script as well.

X. REFERENCES

- [2] Plug-in Architecture - OmarElgabry's Blog (https://medium.com/omarelgabrys-blog/plug-in-architecture-dec207291800)
- [3] SOLID Design Principles Explained: The Open/Closed Principle with Code Examples (https://stackify.com/solid-design-open-closed-principle/)