**User Manual for Traffic Data Collection Program**

***Overview of the program:***

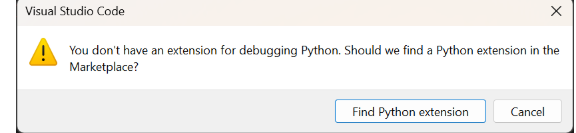
This program is designed to help you collect traffic observation data using your keyboard shortcuts. It allows you to tally vehicle movements (right turns, left turns, going through) and save this file along with other observations to a CSV file.

***Requirements:***

* Make sure that Python is installed on your computer.
* Also install Python’s keyboard library, which can be installed by using the following command in Visual Studio Code’s terminal or your command prompt on windows:

**pip install keyboard**

* If you are using Visual Studio Code, make sure you have installed a Python Extension, if you don’t have an extension installed, you will get the following error message:

****

* **To install Python Extension on Visual Studio Code, Click install in the following pop-up:**

**Graphical user interface, text

Description automatically generated**

***How to use the Program?***

* Step 1: Start the program
* Open Visual Studio Code on your Computer
* When Visual Studio Code is opened choose the folder where you have saved the python script and the csv file. Make sure that the python script file and the CSV file are saved on the same folder.
* The install/download a python extension

Run the program in your python environment by executing the following python script: ‘detector.py’.

* OR you can simply click on the run button on Visual Studio Code that can be found on the upper right corner of the Visual Studio Code’s Graphical User Interface.



* Step 2: Begin Tallying your Data

When you now click the run/play button the program asks you about the camera you want to start taking data from, now users can first select one of the four cameras in an intersection.

i.e. Press ‘1’ for ‘South Bound’

Press ‘2’ for East Bound’, Press ‘3’ for North Bound and Press ‘4’ for ‘West Bound’

Text

Description automatically generated

You will use the following keyboard shortcuts:

* Press ‘Q’ for U turns.
* Press ‘W’ for left turns.
* Press ‘E’ for vehicles going through.
* Press ‘R’ for right turns.
* Press ‘T’ for pedestrians.
* Press ‘D’ when done tallying for that time interval.

Now once you’re done recording data for South Bound or SB, press 2 for ‘East Bound/EB’ and so on.

Finally, when you’re done with the entire intersection press 5 to finish data collection and to move further for observations in the GUI.

* Step 3: Record Observations

Once you’re done tallying your data for the turns and you press ‘D’ when you’re done. A Graphical User Interface will pop up to collect other observations such as Weather, Time Interval, Detection Zones, etc.

Graphical user interface, text, application

Description automatically generated

* Please Enter the Time Interval in the following format: Hour/Minute/Second to Hour/Minute/Second
* Please Enter the Date in the following format: Month/Day/Year
* Weather (Sunny, Snowy, Cold, Raining, etc.)
* Detection Zone Stuck: Answer ‘Yes’ or ‘No’ to indicate if the detection zone is stuck.
* Detection Zones Changing Color: Answer ‘Yes’ or ‘No’ to indicate whether the detection zone changes color when a vehicle passes by.
* Step 4: Save Data

The program will automatically save your observations, including the time interval and vehicle counts to a csv file named ‘traffic\_intersection.csv’.

* Step 5: Continue or Exit

After collecting the observations data and clicking submit.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Users will see the following message which means their data has been saved and updated on the CSV file. So now, User will need to press ok and now they can access the data from the CSV file.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Step 6: Accessing your Data

All your recorded data will be saved in the ‘traffic\_intersection.csv’ file which is in the same directory as the python script.

* Now the updated version of the CSV/Excel file looks like this:

