**UNIVERSITY INSTITUE OF ENGINEERING AND TECHNOLOGY, KURUKSHETRA**

**(2019-2023)**

**PROJECT REPORT**

**ON**

**Smart Traffic Lights**

Submitted in the partial fulfillment

of the requirement for B.Tech 8th Semester

****

**Submitted To: Submitted By:**

Dr. Naresh Name – Akash

CSE Department Roll No. - 251902105

UIET, KUK CSE – B (8th sem)

**ACKNOWLEDGEMENT**

I want to express a deep sense of gratitude to all those people who have supported me to accomplish machine learning project. First and foremost. I would like to thank the management of **U.I.E.T. (Kurukshetra)** who has given me the opportunity to accomplish my training for partial fulfillment of the requirement for **B.Tech. 4th year to Kurukshetra University .**

**DECLARATION**

I, Akash (Roll Number: 251902105) of University Institute of Engineering and Technology, Kurukshetra hereby declare that the project entitled “**Smart Traffic Lights**” is proposed for partial fulfilment of Project for the session 2019-2023.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Content** | **Pg. no** |
| 1 | Abstract | 4 |
| 2 | Introduction | 5-6 |
| 3 | Flow Chart | 7 |
| 4 | Project Design and Description | 8 |
| 5 | Approach | 9-10 |
| 6 | Algorithms Used | 11 |
| 7 | Software and Hardware Requirements | 12 |
| 8 | Technologies and Libraries Used | 13 |
| 9 | File Structure | 14 |
| 10 | Code | 15-23 |
| 11 | Output | 24-26 |
| 12 | Conclusion and Future Scope | 27 |
| 13 | References | 28 |

**Abstract**

The main aim of the traffic light control system is to monitor and control the flow of vehicles

through the junction of many roads and make their smooth motion possible on the routes. However, the timers in these traffic light systems are hardcoded to a fixed duration resulting in large Red-light delays which in turn leads to congestion. Congestion of traffic is a serious problem these days, and the metro cities are the ones most affected by it. Conventional traffic light systems are not designed to handle variable flows of vehicles approaching the junctions. Therefore, the need to automate the traffic control system arises. In recent years, video monitoring systems have been applied to gather the data for the traffic at different points of the day, but these were only used to collect data and were not able to predict the traffic density. On the other hand, traffic density estimation can also be achieved using Image processing and fundamentals of Machine Learning. Therefore, we present STL (Smart Traffic Lights) which is an automated ML-based model, that uses Image processing to calculate the traffic density in live time. Through this, we aim to adjust traffic light timers according to live traffic conditions and also regulate the traffic as per the needs of emergency vehicles to manage critical medical situations effectively. We also plan to analyze Live Traffic Patterns on roads to increase the efficiency of Traffic Lights.

**INTRODUCTION**

**Project Overview**

The severity of traffic congestion has increased in India in recent decades. As per the current statistics by NGSIM data, we found that traffic congestion typically leads to an increase in fuel consumption of the order of 80%, and the traveling time has increased by a factor of up to 4.

This problem can be tackled in several different ways like building new ring roads, more emphasis on public transport, widening the existing roads, etc. But all these solutions require a very high capital investment and will take a lot of time. By means of this project we have tried to come up with a solution that will require less time for implementation as well as is not very costly. Let's first try and understand what the actual reasons behind traffic congestion are. One of the main reasons behind today's traffic problem is the techniques used for traffic management. It has no emphasis on live traffic scenarios, thus leading to inefficient traffic management systems and now coming to our solutions for this.

The main aim of the traffic light control system is to monitor and control the flow of vehicles through the junction of many roads and make their smooth motion possible on the routes. However, the timers in these traffic light systems are hard coded to a fixed duration resulting in large Red-light delays, which in turn leads to congestion. Conventional traffic light systems are not designed to handle variable flows of vehicles approaching junctions. Therefore, the need to automate the traffic control system arises. Our project, “STL” (Smart Traffic Lights) is an automated ML-based model that uses Image processing to calculate the traffic density in live time. Through this, we aim to adjust traffic light timers according to live traffic conditions and also regulate the traffic as per the needs of emergency vehicles to manage critical medical situations effectively. We also plan to analyze Live Traffic Patterns on roads to increase the efficiency of Traffic Lights.

**Need Analysis**

One of the main reasons behind today’s traffic problem is the techniques used for traffic management. It has no emphasis on live traffic scenarios, thus leading to inefficient traffic management systems. The most appropriate solution approach used by most solutions is to let the congested sides cross the intersection first. Our model is different than the others in the implementation part.

Contrary to this, we will implement our solution using Machine Learning and Image Processing. The model will use Variance Based Approach to calculate the time for each side of the intersection. Once the variance of the number of cars on the four sides is calculated, the program will know how spread out the numbers are and will allot the time for greenlight on each side depending upon the value of variance. This approach is cost-effective and takes lesser time compared to other techniques. Moreover, we will also be regulating the traffic as per the needs of emergency vehicles to manage critical medical situations effectively by detecting the sound of the emergency vehicle and automatically making the light green on that side of the road. This is a novelty and is not implemented in any other research paper.

**Objectives**

1. To make a large data set of traffic images collected from various countries.
2. To increase the efficiency of Traffic Lights and setting the timer by using machine learning algorithm as per the traffic congestion situation on any particular side of the road.
3. To regulate the traffic as per the needs of emergency vehicles to manage critical medical situations effectively by detecting the sound of emergency vehicle and automatically making the light green on that side of the road.
4. To simulate, test and validate our Machine Learning based model on the data set collected in order to check its effectiveness.

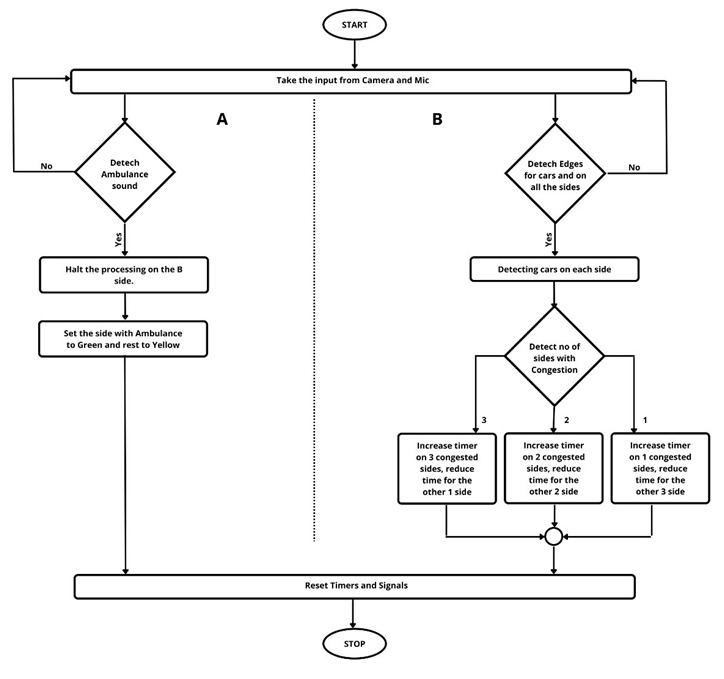
**Assumptions and Constraints**

1. Image noise, i.e., random variation of brightness or color information in images, is usually an aspect of electronic noise. It should be as small as possible to work our machine learning model better.
2. Cameras installed on the traffic signals can capture clear and comprehensible pictures of the present traffic scenarios.
3. We will be getting the image of an empty road as a reference.

**FLOW CHART**

**Methodology**

We begin by taking input from the camera and mic, which will be installed on the traffic lights. Images from the camera are fed into a machine learning model, which uses image processing to detect the number of vehicles on all the sides of an intersection and calculate the traffic density. We are using the OpenCV library of python for image processing, which is further used to calculate the number of cars. Then to calculate the traffic light time, we are using a variance-based algorithm. The algorithm works on the basic principle of first letting the congested sides cross the intersection. Whenever our model detects the sound of any emergency vehicle like an ambulance, every other process will be halted. The traffic light on the side of the ambulance will turn green to ensure that the ambulance is not delayed. During that time, all the other lights will be yellow.

**Flow Chart**

**PROJECT DESIGN AND DESCRIPTION**

**DESIGN OVERVIEW**

The project consists of majorly 2 parts:

1. Detection and counting of cars on all the side of the junction

2. Determining time for each side

**Detection and Counting of Cars**

We use Edge Detection Algorithm to detect the cars, Images are fed in the python model using OpenCV Library, they are then converted converts them into black and White for detection. We then remove the possible noise in the images (Gaussian Noise, Salt and Pepper Noise). The image is then dilated and the Percentage change with actual image is calculated to find the no of cars..

**Determination of Time**

Once the number of vehicles is detected on all the 4 sides, they are stored in a List. The number of vehicles is manipulated to clear the traffic congestion as soon as possible.

If the number of vehicles does not exceed the Normal Value, then the normal clockwise flow of vehicles is initiated, otherwise the Congestion algorithm comes to play. There are certain conditions to be followed:

1. Cycle is complete – Once a signal is green, it turns green again only after all the other 3 sides have turned green once.

2. Max Waiting Time – The Maximum wait time for any traffic light side is 180 second or 3 minute.

3. Few Cars Case – If a side has very less no of cars i.e., less than 3 then corresponding side will turn green instead of letting them wait for other side to move for max 180 second.

4. Single Sided Case – In a situation there is just one-sided traffic flow and all the other sides of the intersection are empty then complete preference is given to that side.

**APPROACH**

**Edge Detection: OpenCV**

A Camera is placed on each side of the road intersection and the images are processed to get the number of vehicles on each side of the road intersection.

The Image processing is done using the following:

1. It is Read using the OpenCV Library of Python

2. It is resized according to need

3. Image then converted to Black and White image

4. Removal of the Gaussian Noise takes place

5. Removal of Salt and Pepper Noise takes place

6. Dilation

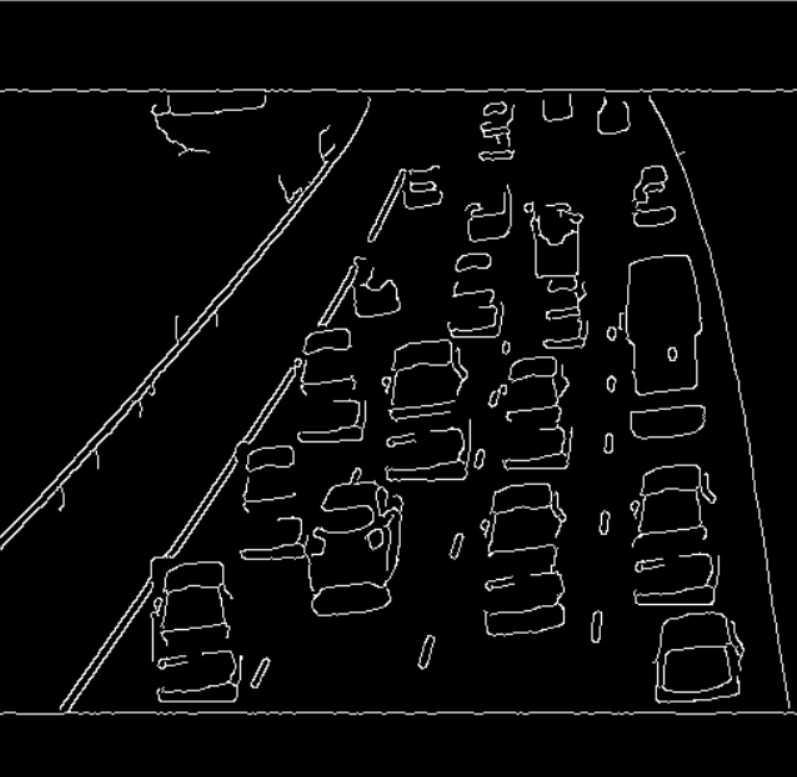
7. Take Difference from reference image

8. Percentage change is calculated

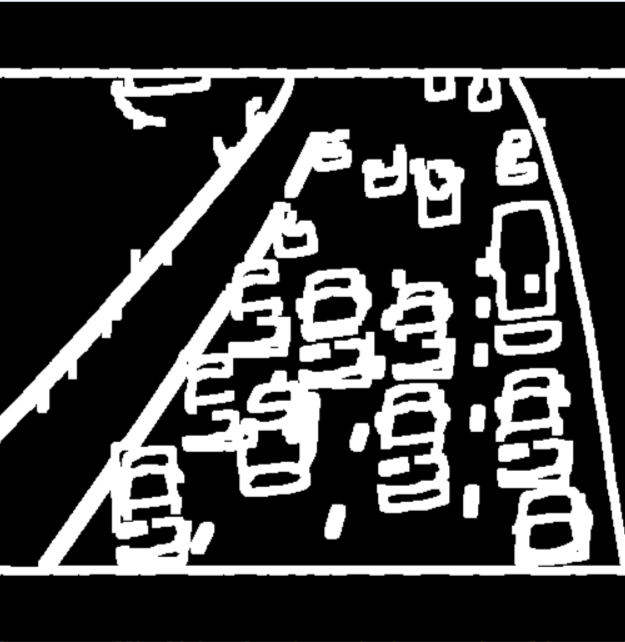
9. Number of vehicles is calculated



Black and White car Image



DilatedImage

****

**ALGORITHMS USED**

**Variance-Based Algorithm**

The variance in probability theory and statistics is a way to measure how far a set of numbers is spread out. Variance describes how much a random variable differs from its expected value. The variance is defined as the average of the squares of the differences between the individual (observed) and the expected value. Once Variance of the Number of the cars on the 4 sides is calculated , the program will know how spread out the numbers are and will allot the Time for Green light on each side depending upon the value of variance. Timers according to variance are used to make the data less spread out and it also tends to bring the maximum value close and in cases even less than second and third maximum number in the list thus making the List less complex and maintaining uniformity. Use of Variance for alloting the time for Green signal improves the efficiency as the algorithm becomes more dynamic and situation dependent instead of being static .

**For eg:**

- List = [40,50,45,43]

This will result in Low variance (13.25) and thus providing similar and comparitively less time for each side Green signal will be fair and performance wise relevant.

- List = [1,2,4,60]

This will result in High variance (624.6) and after taking in consideration the \_Few Car Case\_ performance wise relevant timimgs will be alloted.

- List = [5,20,30,45]

This will result in Moderate variance (212.5) and a timer of maximum 50 seconds will be allotted.

**Software and Hardware requirements**

**Software Requirements :**

1. Operating system - any user-friendly operating system
2. VS Code – IDE for development purposes

**Hardware Requirements :**

1. PC with 4 GB+ RAM
2. Intel i3+ processor

**TECHNOLOGIES AND LIBRARIES USED**

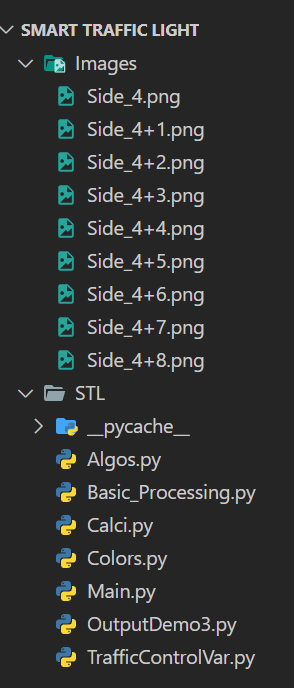
**Technology:**

1. Python
2. Python Turtle Graphics

**Libraries:**

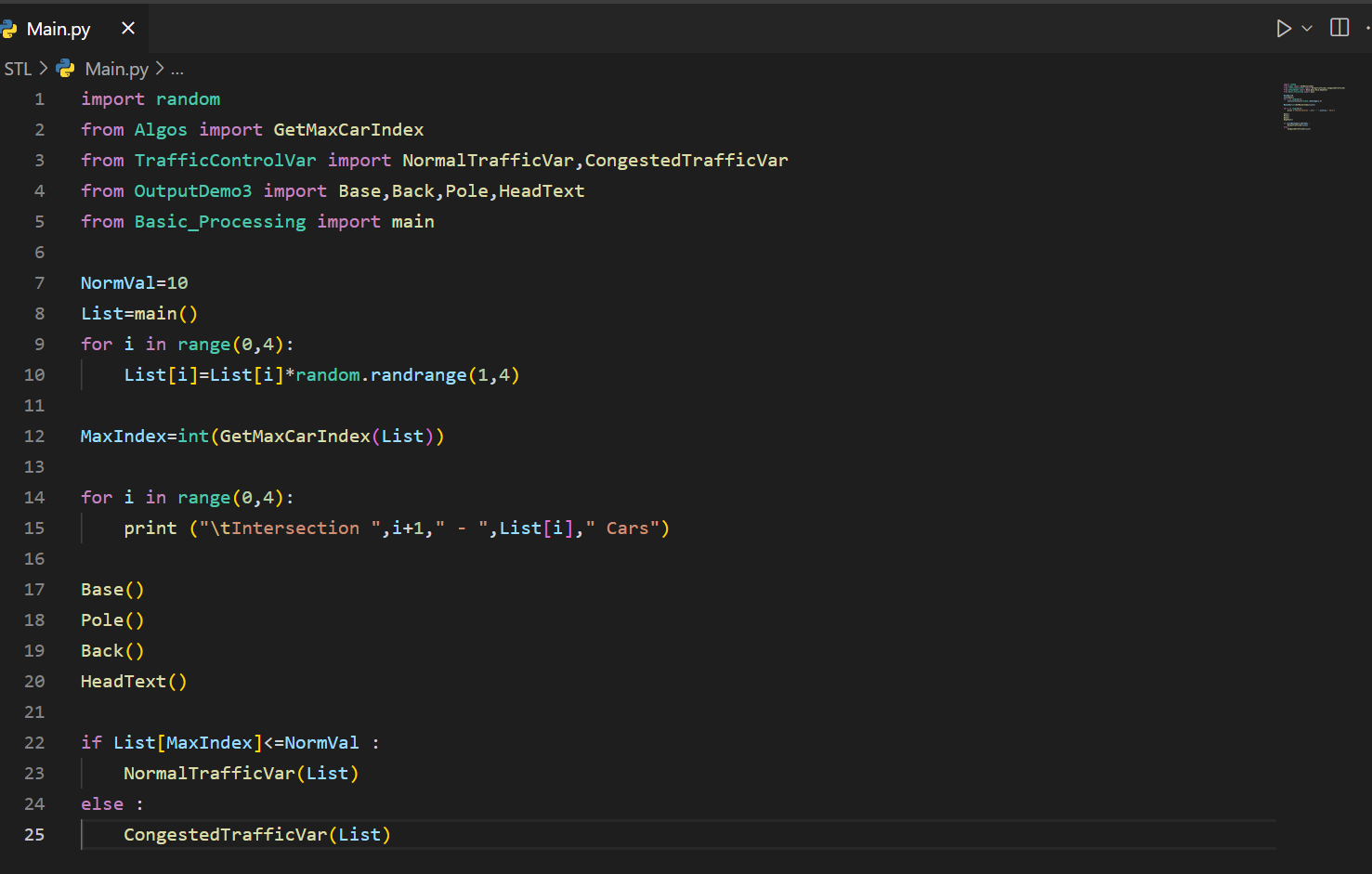
1. OpenCV
2. numpy
3. os
4. random
5. math
6. time
7. turtle

**FILE STRUCTURE**

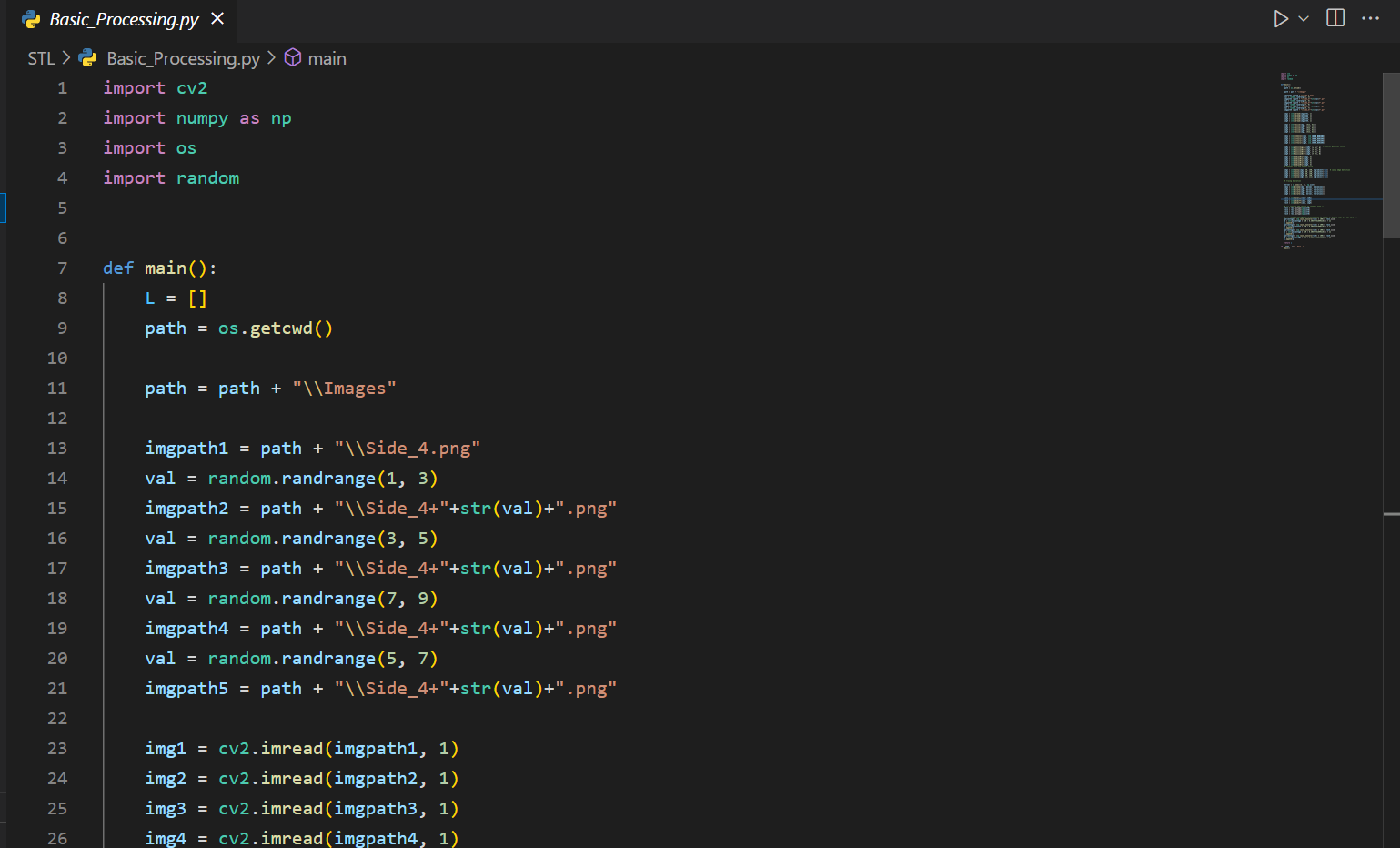


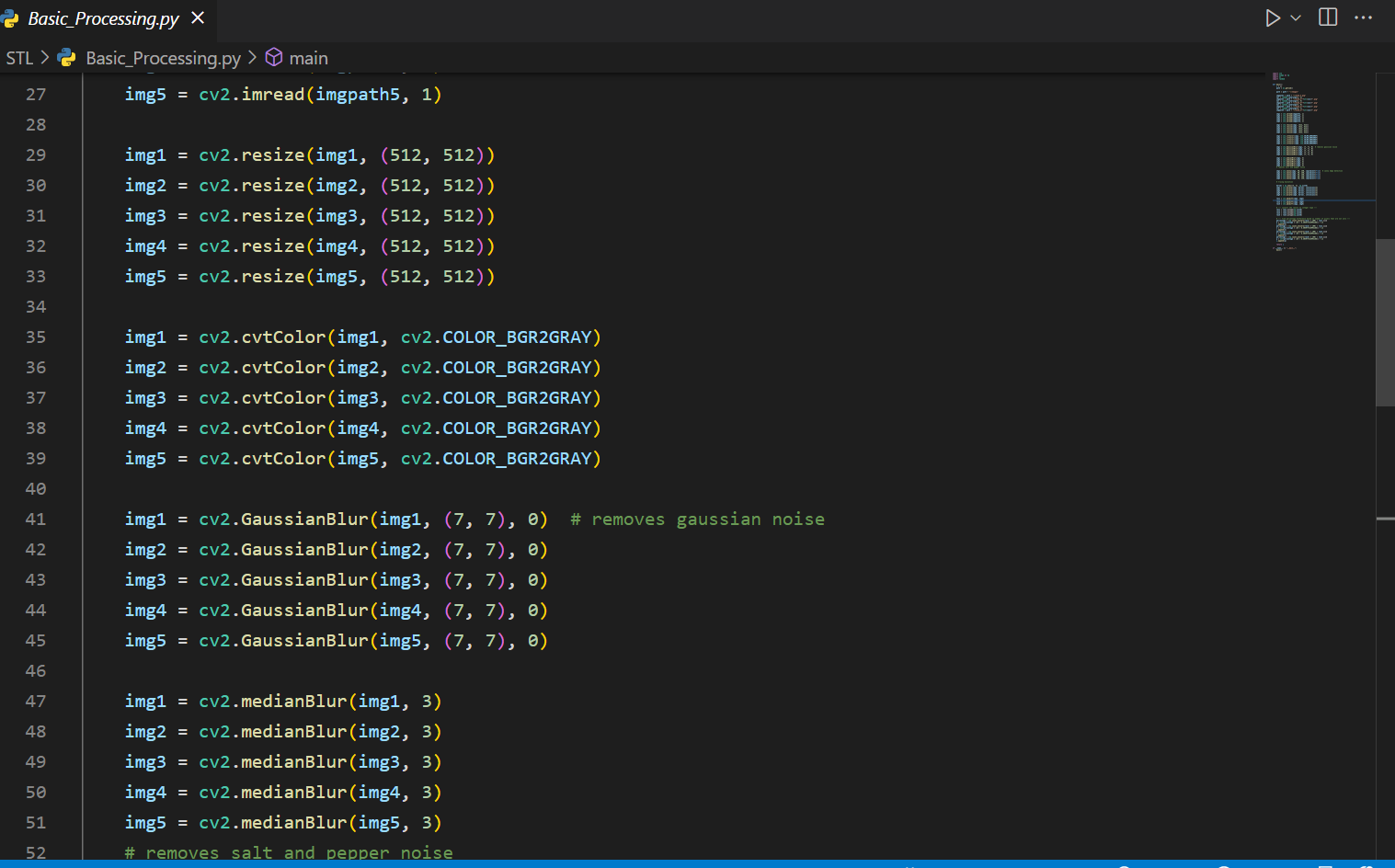
**CODE**

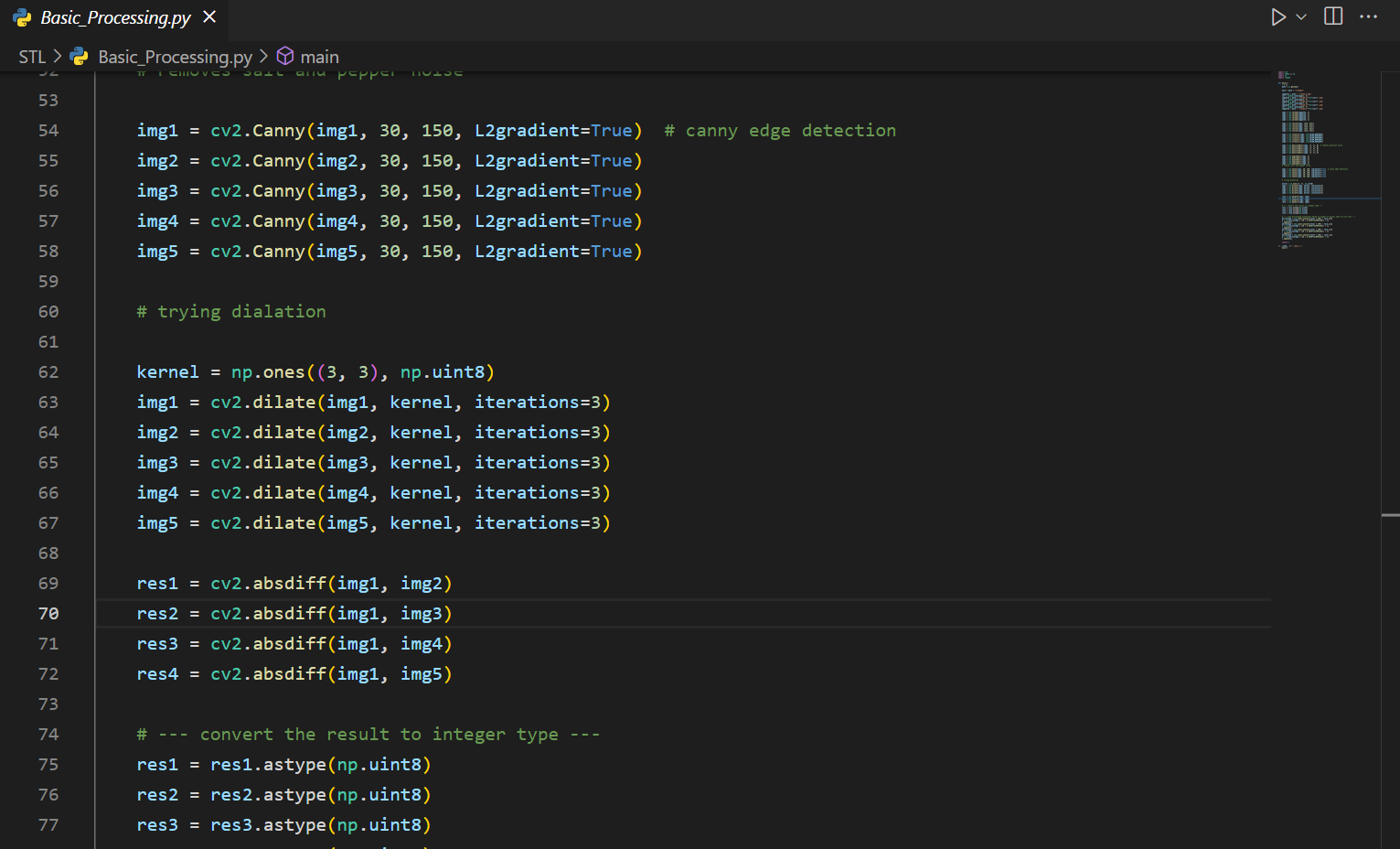
Main.py

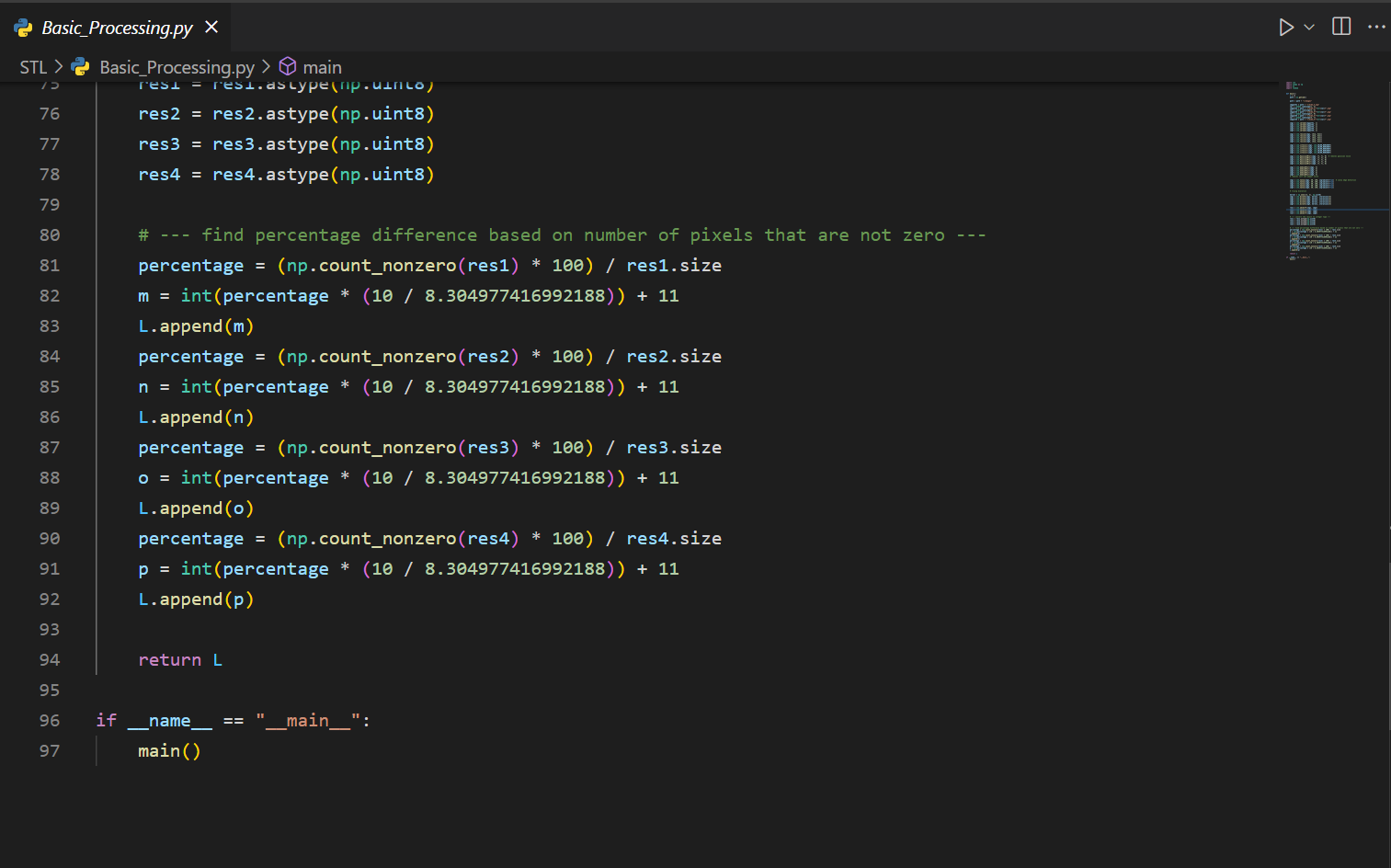


Basic\_Processing.py

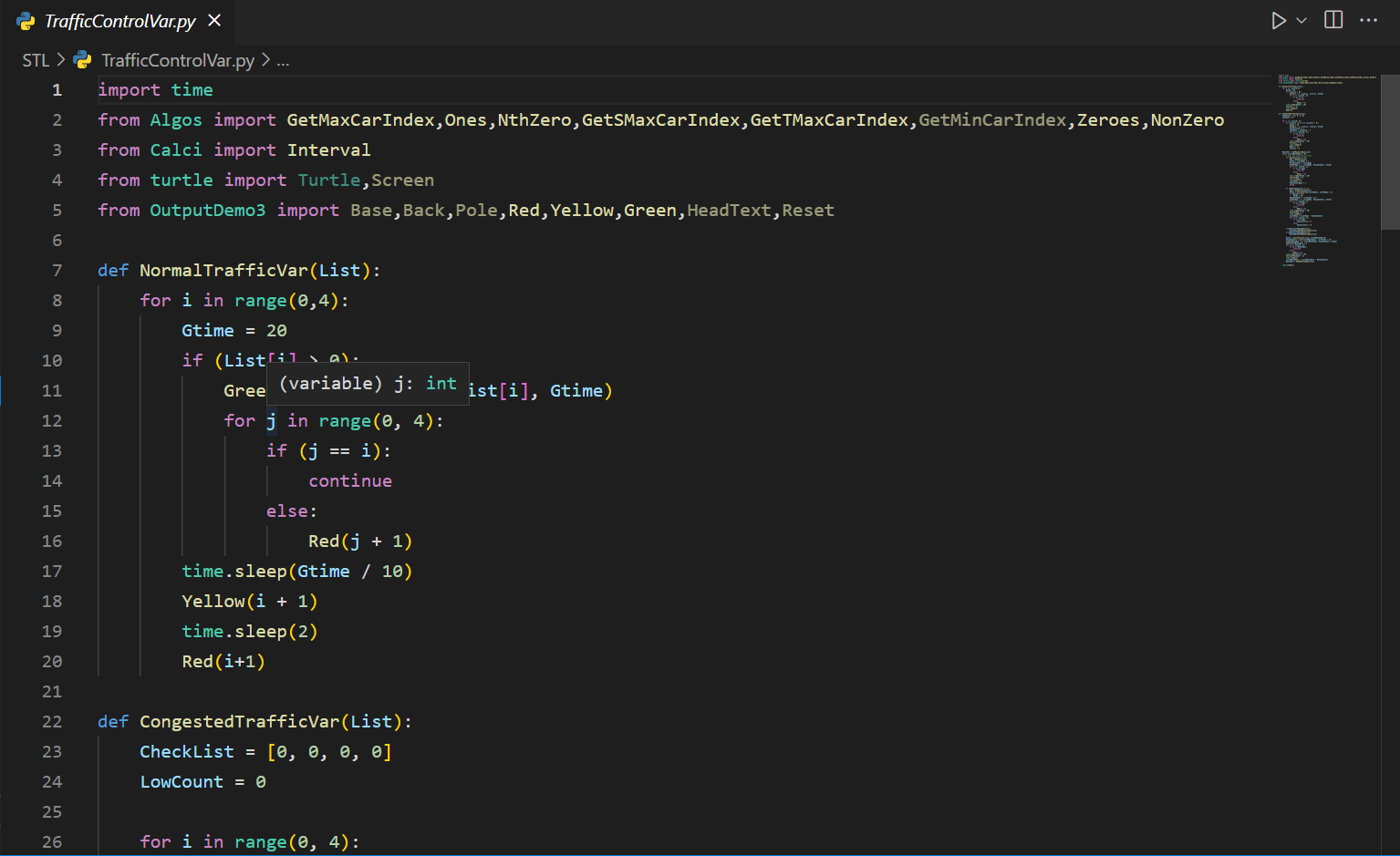


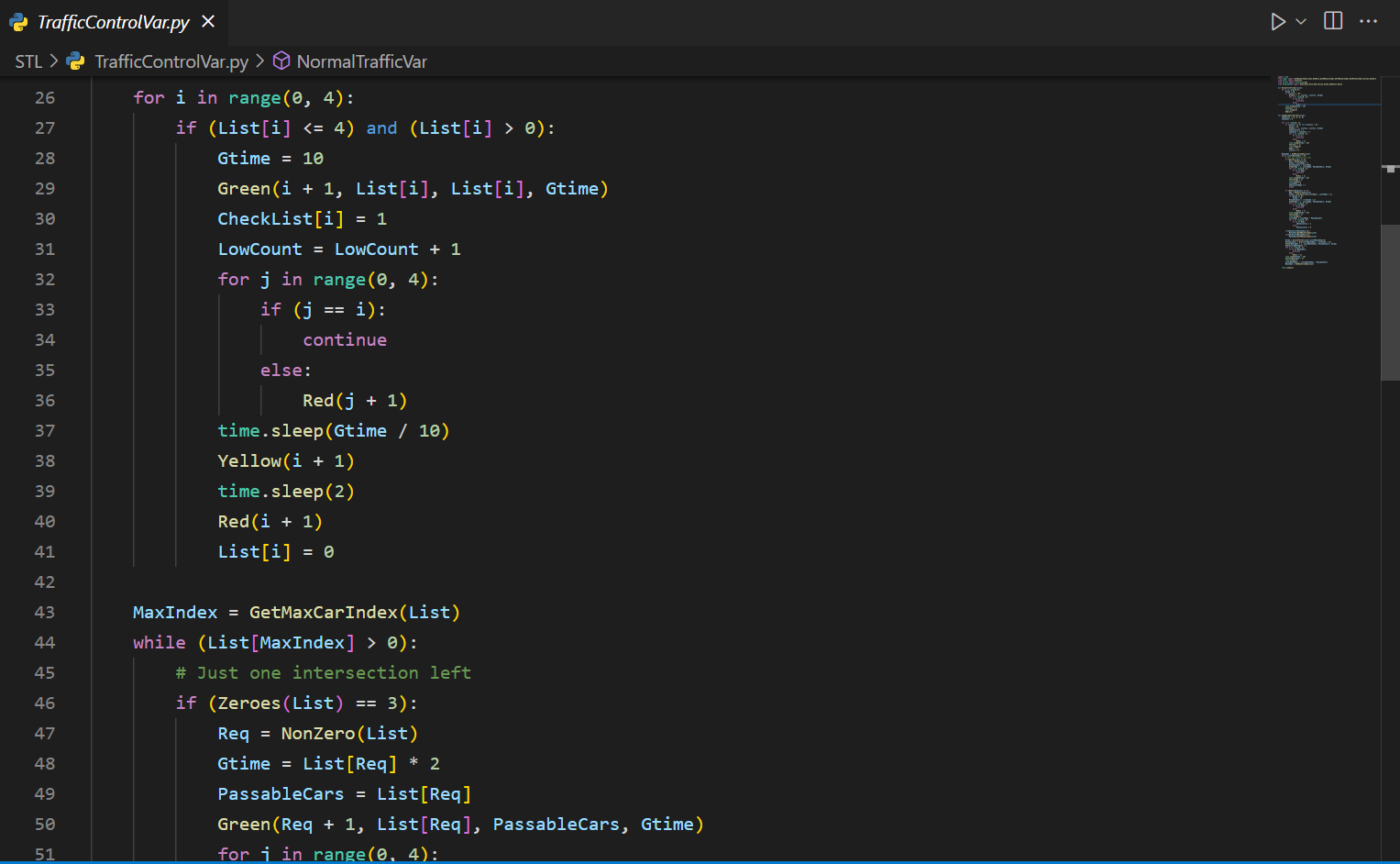




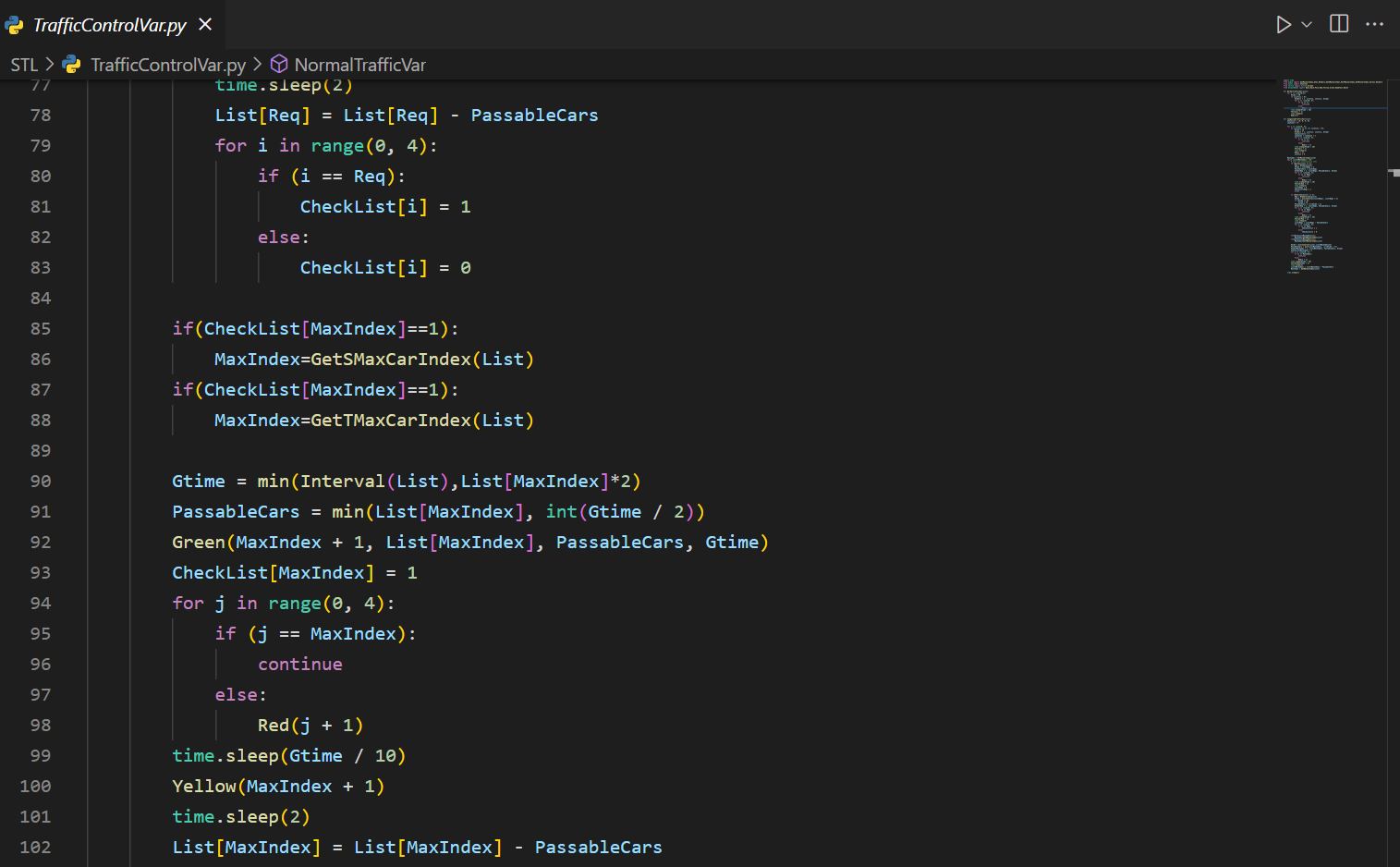


TrafficControlVar.py

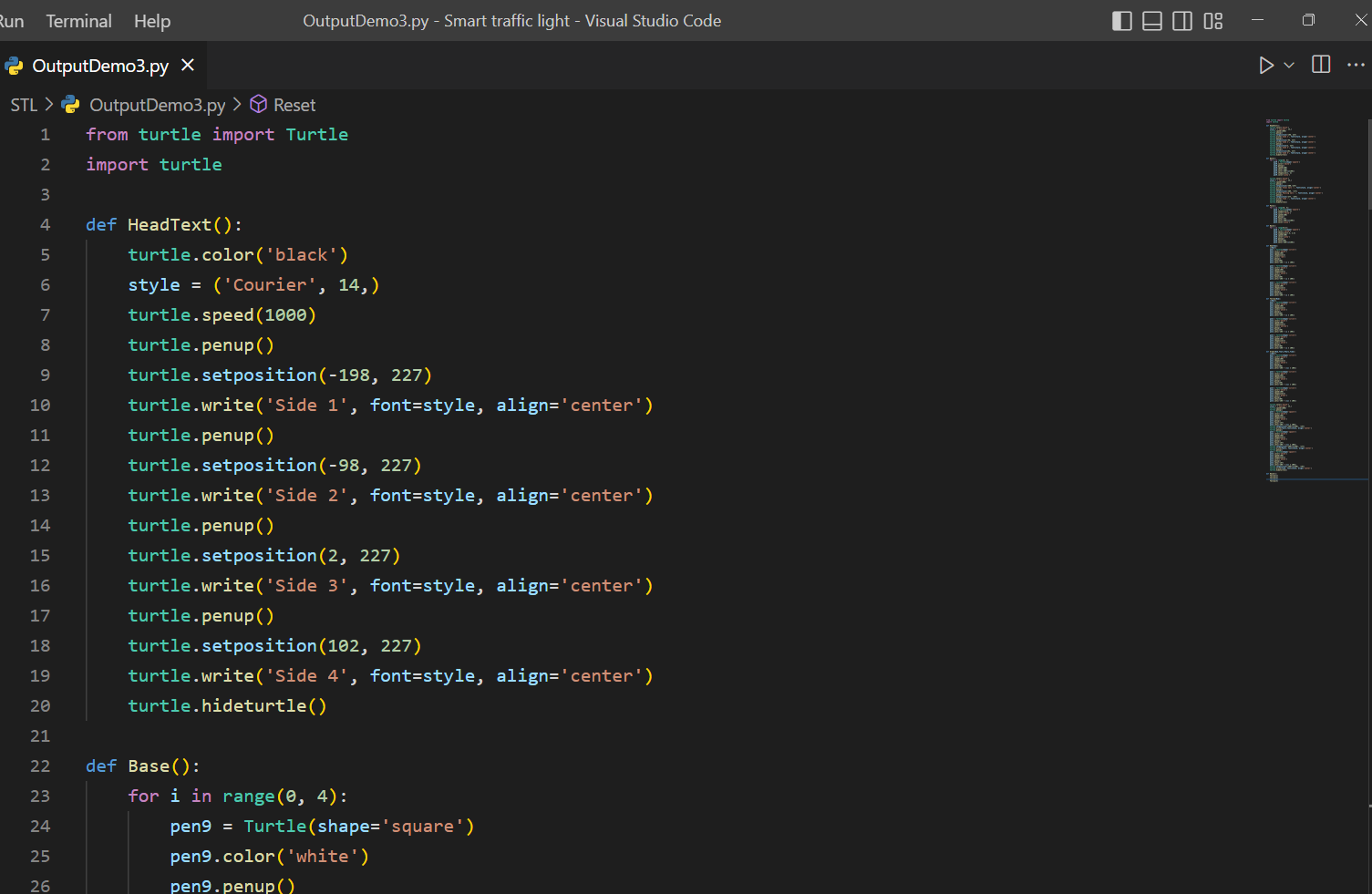


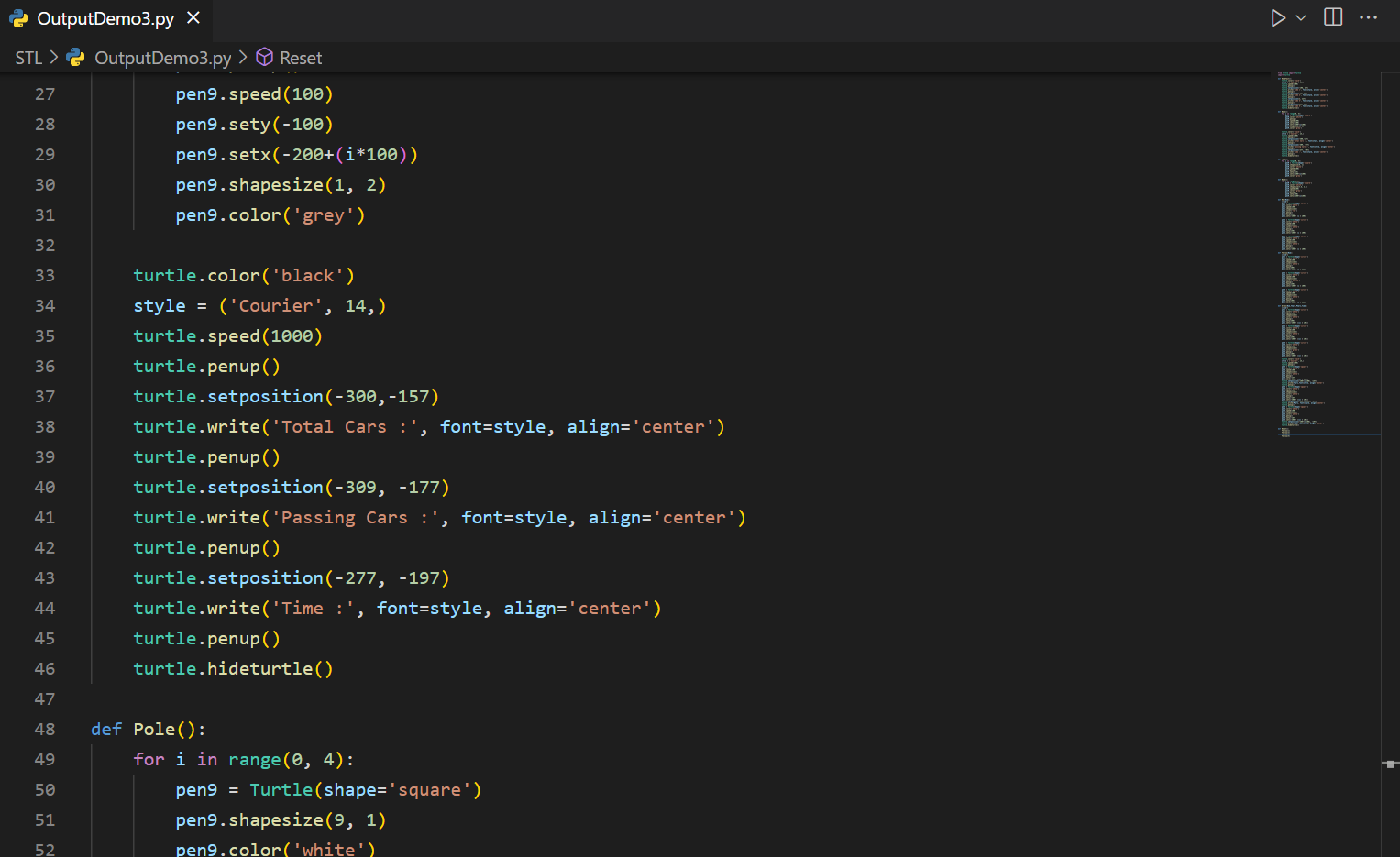
****

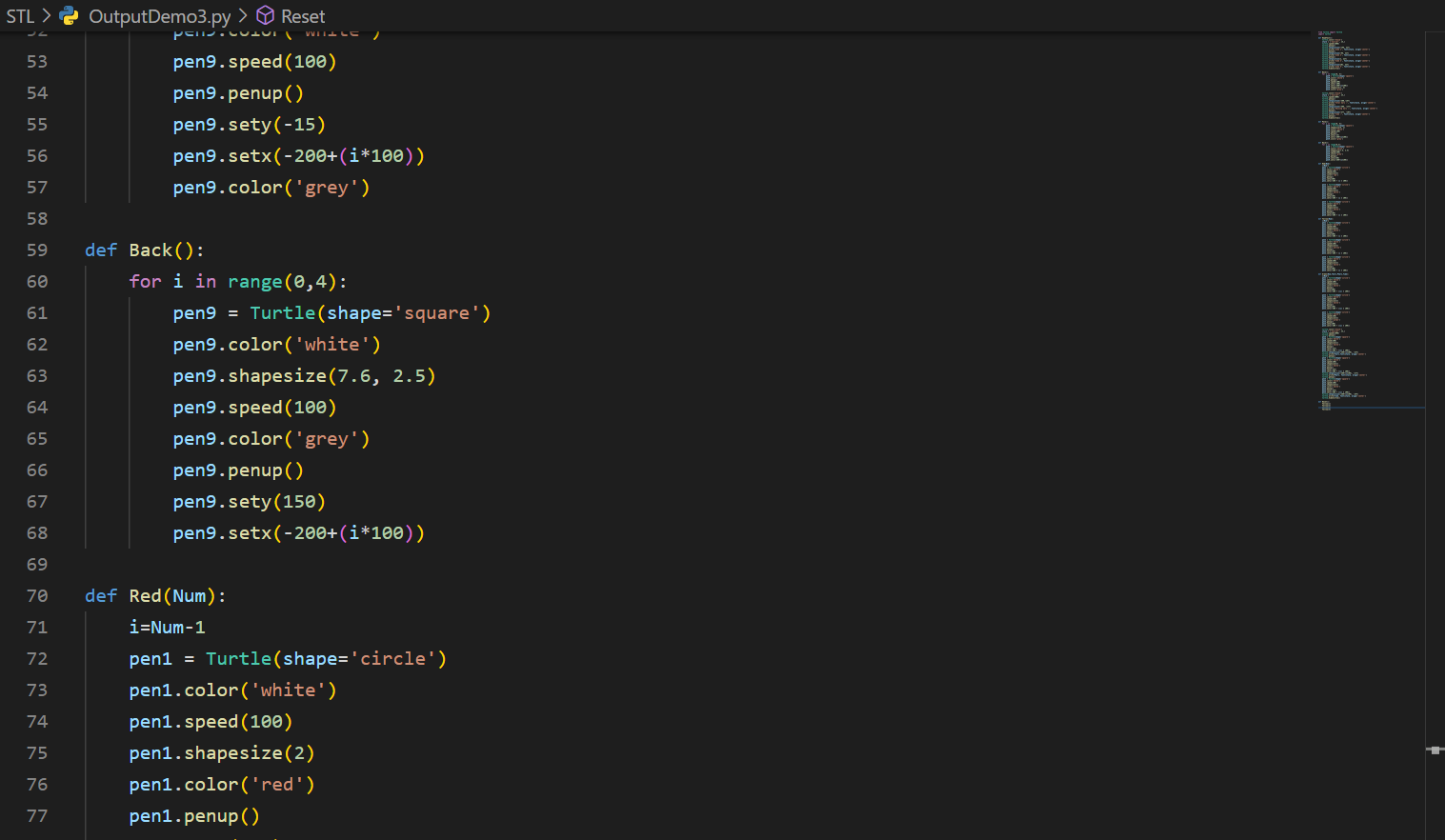
****

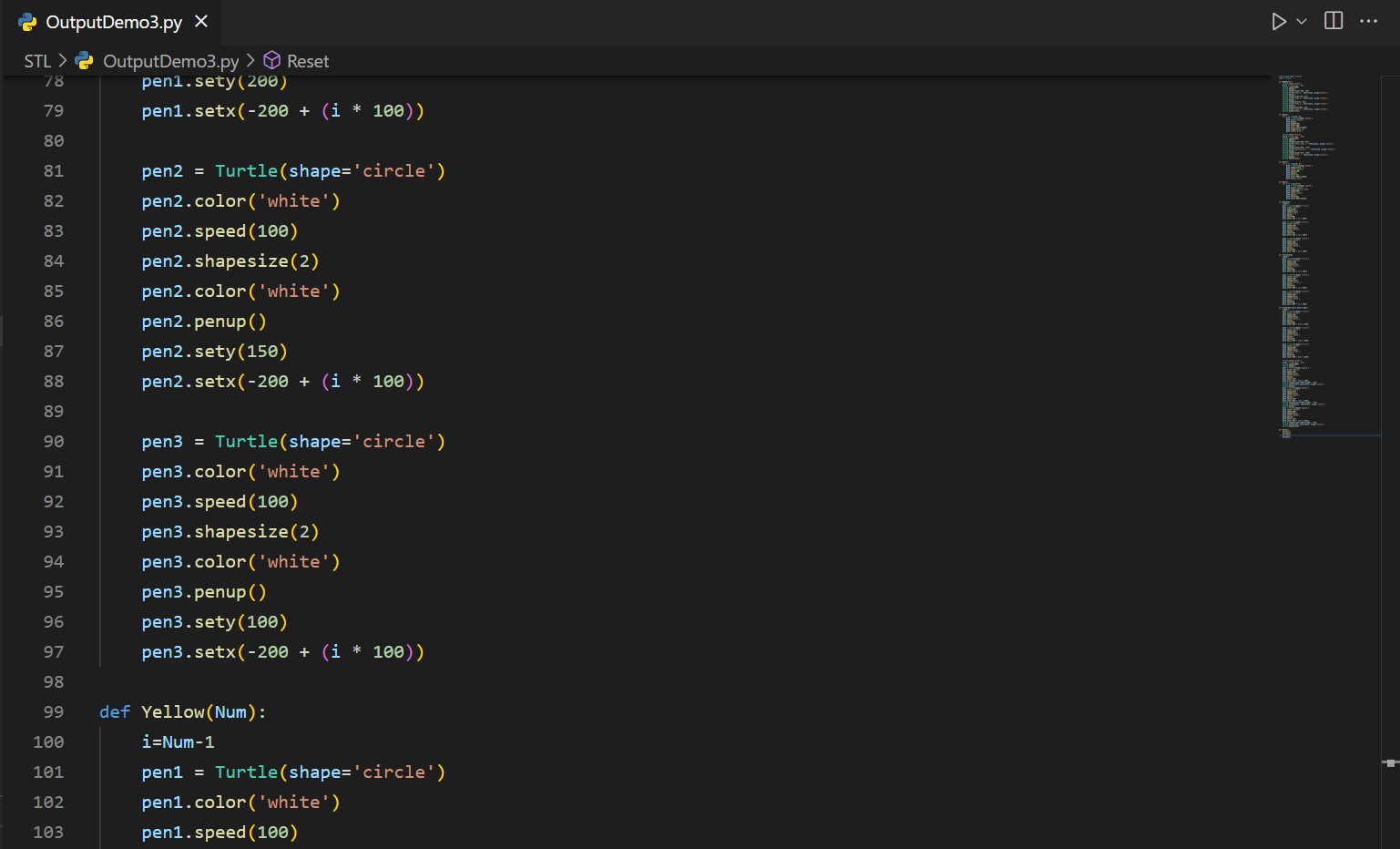
****

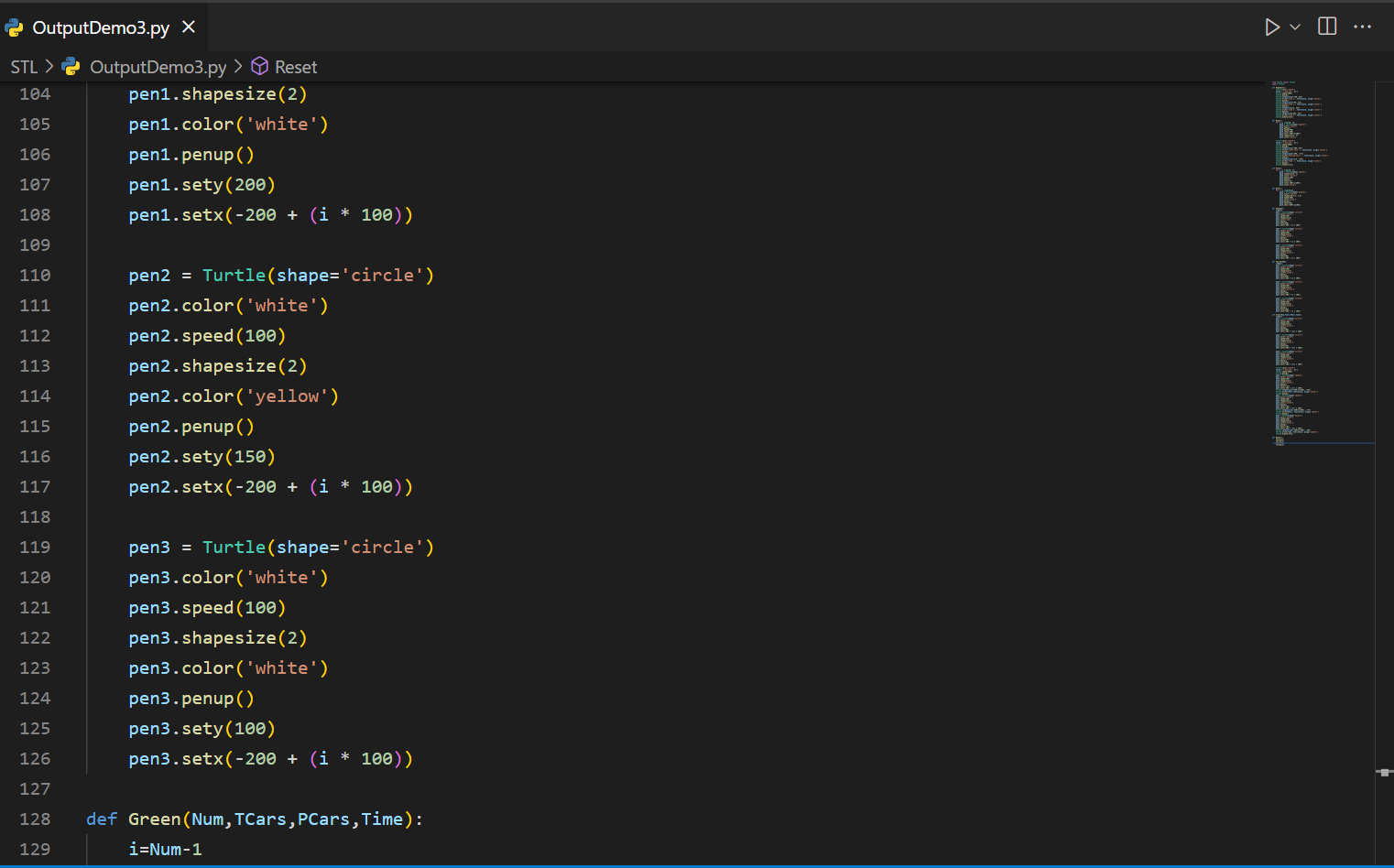
OutputDemo3.py

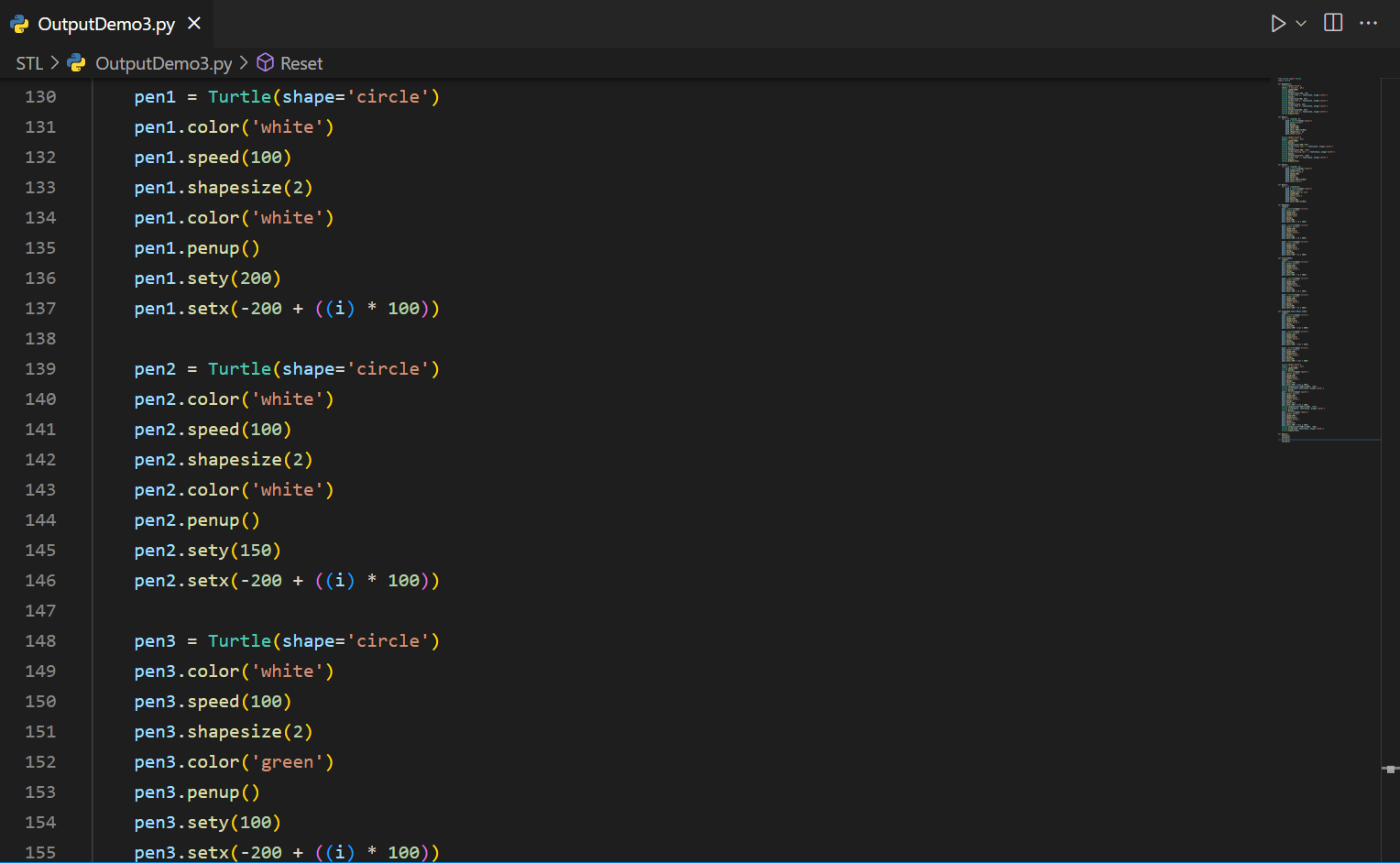


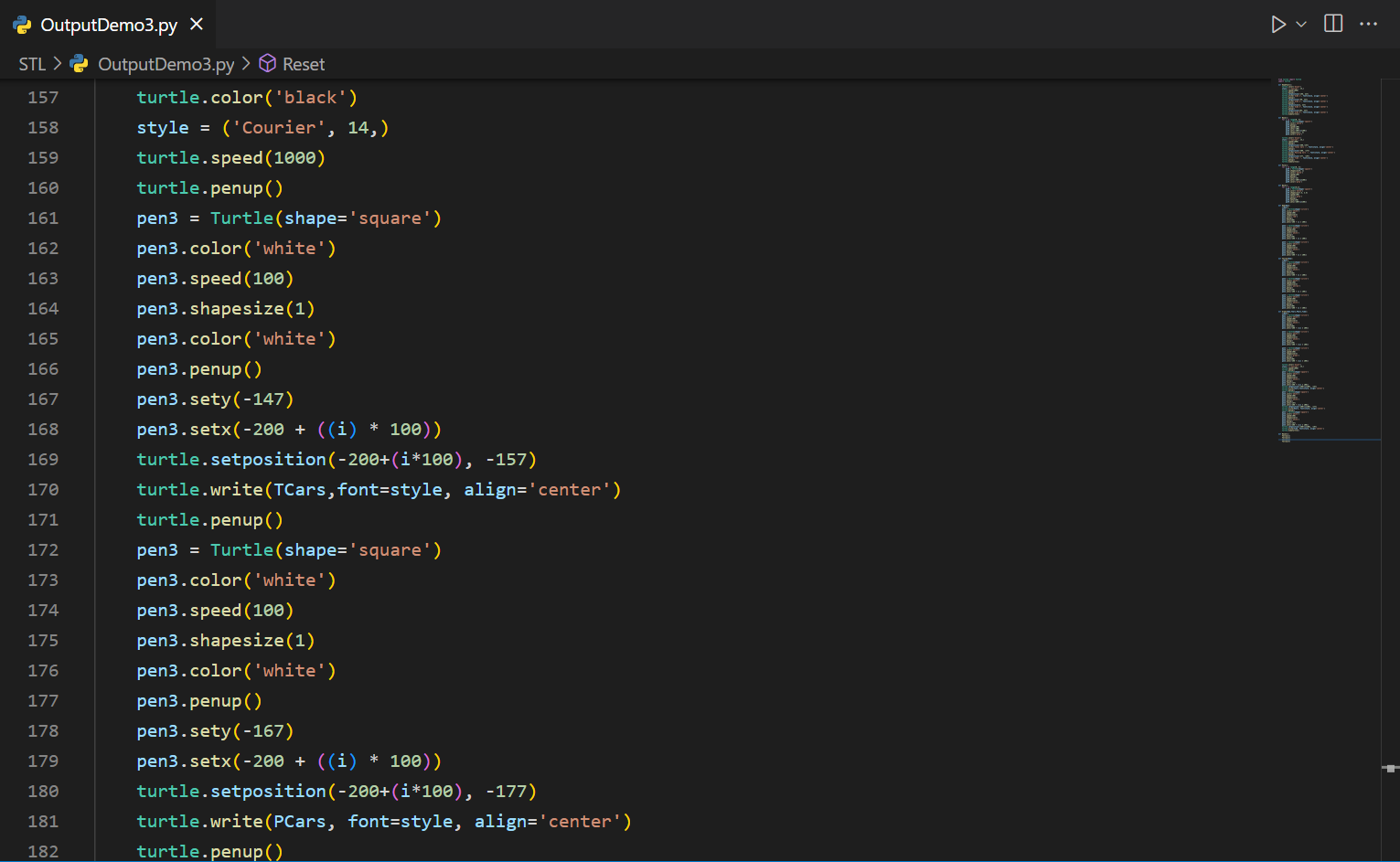






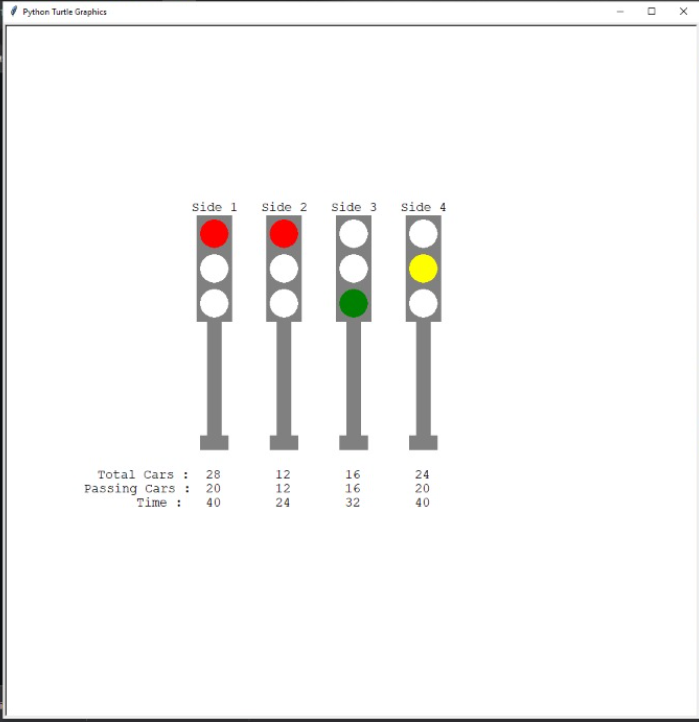
****

****

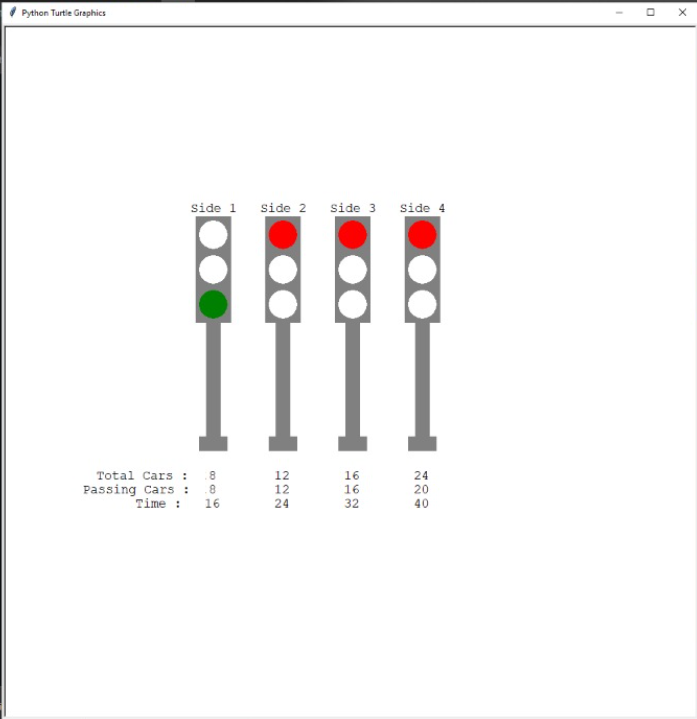
****

****

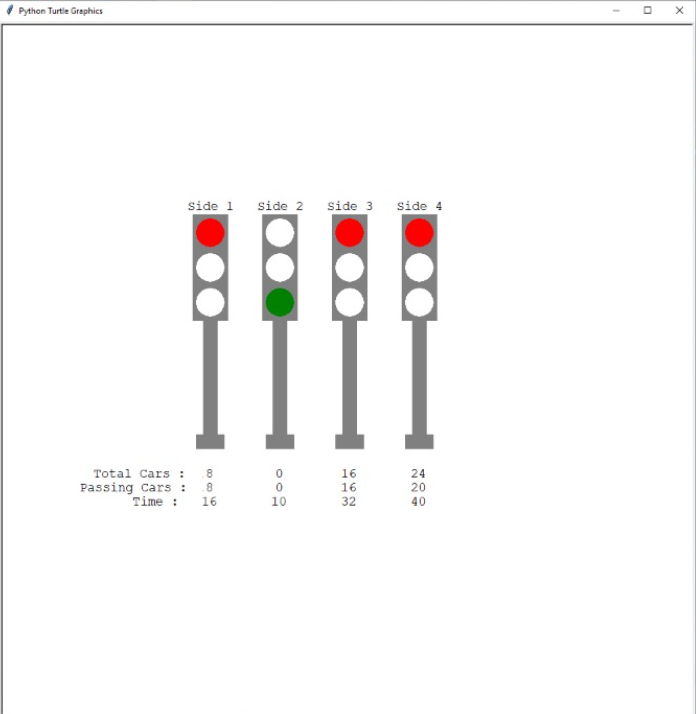
**Output**



Side 3 allowed to move



Side 1 allowed to move



Side 2 allowed to move

**CONCLUSION**

This Project is a prototype to Control the Traffic Congestion and control the flow of vehicles

through the junction of many roads and make their smooth motion possible on the routes. It also helps in the reduction of fuel consumption. Congestion of traffic is a serious problem these days, and the metro cities are the ones most affected by it. Conventional traffic light systems are not designed to handle variable flows of vehicles approaching the junctions. Therefore, the need to automate the traffic control system arises. Hence our model STL fits this situation by providing a dynamic approach to control traffic lights and congestion.

**FUTURE SCOPE**

1. Can be moulded to be used in situations like mass evacuation.
2. Linking of other traffic lights in sync to provide better congestion clearing.
3. Can be developed further to be used by emergency vehicles to provide Green route and decreasing accidents on intersections.

**REFERENCES**

* Python Documentation

<https://docs.python.org/3/>

* OpenCV

<https://docs.opencv.org/4.x/>

* Turtle

<https://docs.python.org/3/library/turtle.html>

* Image Processing

<https://www.geeksforgeeks.org/image-processing/>

* Variance

<https://statisticsbyjim.com/basics/variance/>