

ABSTRACT

In this paper author is describing concept to control or automate green traffic signal allotment time based on congestion available at road side using Canny Edge Detection Algorithm. To implement this technique we are uploading current traffic image to the application and application will extract edges from images and if there is more traffic then there will be more number of edges with white colour and if uploaded image contains less traffic then it will have less number of white colour edges. Empty edges will have black colour with value 0. By counting number of non-zeroes white pixels we will have complete idea of available traffic and based on that we will allocate time to green signal. If less traffic is there then green signal time will be less otherwise green signal allocation time will be more.

TABLE OF CONTENTS

Chapter. No	Title	Page No.
	Abstract	i
	Table of Contents	ii
	List of Figures	iv
	Abbreviation	v
1.	INTRODUCTION	1-3
	1.1 Introduction of the Project	1
	1.2 Project Overview	1
	1.3 Existing System	2
	1.4 Proposed System	2
	1.5 Advantages of proposed system	3
2.	REQUIREMENT ANALYSIS	4-5
	2.1 Requirement Analysis	4
	2.2 Requirement Specification	4
	2.2.1 Functional requirement	4
	2.2.2 Non Functional requirements	4
	2.3 Computational resource requirements	5
	2.3.1 Hardware requirements	5
	2.3.2 Software requirements	5
3	DESIGN	6-8
	3.1 Introduction	6
	3.2 Use case Diagram	6
	3.3 Class Diagram	7
	3.4 Sequence Diagram	8
	3.5 State Chart Diagram	9
4	MODULES	10

	4.1 Modules	10
	4.2 Upload image module	10
	4.3 Pre-process module	10
	4.4 White pixel count module	10
5	IMPLEMENTATION	11-16
	5.1 Sample Code	10
6	SCREENSHOTS	17-24
	6.1 Screenshots	17
	6.1.1 Image Folder	17
	6.1.2 Interface	18
	6.1.3 Image Selection	19
	6.1.4 Image Processing	20
	6.1.5 Reference Image	21
	6.1.6 Pixel Counts	22
	6.1.7 Green Signal Time	23
	6.1.8 Image A	24
7	TESTING	25-27
	7.1 Overview of Testing	25
	7.2 Types of Testing	25
	7.2.1 Unit Testing	25
	7.2.2 Integration Testing	25
	7.2.3 Functional Testing	26
	6.3 Unit Testing	26
	6.4 Integration Testing	27
	6.5 Acceptance Testing	27
8	CONCLUSION AND FUTURE WORK	28
	8.1 Conclusion	28
	8.2 Scope for future work	28
9	REFERENCES	29

LIST OF FIGURES

Sl No	Title	Page No.
3.1	Use Case Diagram	6
3.2	Class Diagram	7
3.3	Sequence Diagram	8
3.4	State Chart Diagram	9
6.1	Image Folder	17
6.2	Interface	18
6.3	Image Selection	19
6.4	Image Processing	20
6.5	Reference Image	21
6.6	Pixel Counts	22
6.7	Green Signal Time	23
6.8	Image A	24

ABBREVIATION

GUI	Graphical User Interface
PY	Python
OPP	Object Oriented Programming
DRY	Don't Repeat Yourself
PIP	Package Installer for Python
MRO	Method Resolution Order
PEP	Python Enhancement Proposals
BDFL	Benevolent Dictator For Life
REPL	Read-Eval-Print Loop