ABSTRACT

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advent of deep learning techniques, the accuracy for object detection has increased drastically. The project aims to incorporate state-of-the-art technique for object detection with the goal of achieving high accuracy with a real-time performance. A major challenge in many of the object detection systems is the dependency on other computer vision techniques for helping the deep learning based approach, which leads to slow and non-optimal performance. In this project, we use a completely deep learning based approach to solve the problem of object detection in an end-to-end fashion. The network is trained on the most challenging publicly available dataset (PASCAL VOC), on which object detection challenge is conducted annually. The resulting system is fast and accurate, thus aiding those applications which require object detection.

TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| Sl. No | Title | Page No |
|  | Abstract | i |
|  | List of Contents | ii |
|  | List of Figures | iii |
|  | Abbreviations | iv |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.1 Introduction | 1 |
|  | 1.2 Existing system | 2 |
|  | 1.3 Proposed system | 4 |
| **2.** | **REQUIREMENTS ANALYSIS** | **5** |
|  | 2.1 Requirement Analysis | **5** |
|  | 2.2 Requirements Specification  2.2.1 Functional Requirement  2.2.2 Non-Functional Requirement | 5  5  5 |
|  | 2.3 Computational resource requirements | 6 |
|  | 2.3.1 Hardware resources  2.3.2 Software resources | 6  6 |
| **3.** | **DESIGN** | **7** |
|  | 3.1 Design | 7 |
|  | 3.2 Architecture | 7 |
|  | 3.3 Use case Diagram | 8 |
|  | 3.4 Class Diagram | 9 |
|  | 3.5 Sequence Diagram | 10 |
|  | 3.6 State Chart Diagram | 11 |
| **4.** | **MODULES** | 12-13 |
|  | 4.1 Modules | 12 |
|  | 4.2 Bounding Box | 12 |
|  | 4.3 Classification | 12 |
|  | 4.4 Two Stage Method | 12 |
|  | 4.5 Unified Method | 13 |
| **5.** | **IMPLEMENTATION** | **14-24** |
|  | 4.1 Sample Code | 14 |
| **6.** | **SCREENSHOTS** | **25-30** |
|  | 6.1 Screenshots  6.1.1 Account Login Page  6.1.2 Registration Page  6.1.3 Login Page  6.1.4 Multy Object Detection Page  6.1.5 Uploaded Image  6.1.6 Tracking Image | **25**  25  26  27  28  29  30 |
| **7** | **TESTINGS** | **31** |
|  | 7.1 Overview of Testing | 31 |
|  | 7.2 Types of Testing  7.2.1 Unit Testing  7.2.2 Integration Testing  7.2.3 Functional Testing | 31  31  31  32 |
|  | 7.3 Unit Testing | 32 |
|  | 7.4 Integration Testing | 33 |
|  | 7.5 Acceptance Testing | 33 |
| **8.** | **CONCLUSION AND FUTURE SCOPE** | **34** |
|  | 8.1 Conclusion | 34 |
|  | 8.2 Scope for Future work | 34 |
| **9.** | **REFERENCES** | **35** |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Title** | **Page No.** |
| 3.1 | Use Case Diagram | 6 |
| 3.2 | Class Diagram | 7 |
| 3.3 | Sequence Diagram | 7 |
| 3.4 | State Chart Diagram | 8 |
| 6.1 | Account Login Page |  |
| 6.2 | Registration Page |  |
| 6.3 | Login Page |  |
| 6.4 | Multy Object Detection Page |  |
| 6.5 | Uploaded Image |  |
| 6.6 | Tracking Image |  |

ABBRIVATIONS

|  |  |
| --- | --- |
| GUI | Graphical User Interface |
| PY | Python |
| OPP | Object Oriented Programming |
| PIP | Package Installer for Python |