

DECLARATION

We hereby declare that the project entitled “**Detecting Bots in Twitter**” has been carried out by us under the guidance of Dr. Nagegowda K S, Assistant Professor and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering of PES University, Bengaluru** during the academic semester January – May 2019. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

01FB15ECS069

Bhoomika R Rao

01FB15ECS214

Preetika

ACKNOWLEDGEMENT

I would like to thank Dr. Nagegowda K S for his continued support and guidance without which the completion of my project would not have been possible.

I would like to thank Mr. H B Mahesh, Mr. V R Badriprasad and Mr. Kiran P for their support and guidance without which the completion of my project would not have been possible.

I would like to thank Ms. Preet Kanwal and Ms. Sangeeta V I, Project Coordinators for the batch of 2015-2019, Computer Science for taking the pains to organize our internships and co-ordinate with a batch of over 350 students.

I would like to thank the Chairperson of the Department of Computer Science, Dr. Shylaja S Sharath, for giving students the opportunity to pursue an Internship during the final semester of college, as well as for the support over the four years of Engineering that enabled us to be fully functional Professionals.

I would like to thank the Dean of Faculty, Dr. B K Keshavan, for his determined work in ensuring students had the best possible faculty to prepare them for whatever activity they decide to pursue in the future.

I would like to thank the Vice Chancellor; Dr. K N B Murthy for his continued work towards ensuring a vibrant learning environment within PES University existed for students to excel.

I would like to thank CEO and Pro Chancellor, Prof. Jawahar Doreswamy for his work towards bettering PES University year upon year and ensuring the infrastructure for learning always existed.

I would like to thank the Chairman and Chancellor of PES University, Dr. M R Doreswamy, founder of PES Group of Institutions for his vital role in setting up this institution over 25 years ago, and ensuring it becomes a top college in India.

Finally, I would like to thank my parents, without whose continued support, I would not have been able to achieve any of the things I have so far.

ABSTRACT

One of the important problems in social media platforms like Twitter is the large number of social bots or Sybil accounts which are controlled by automated agents, generally used for malicious activities. These include directing more visitors to certain websites which can be considered as spam, influence a community on a specific topic, spread misinformation, recruit people to illegal organizations, manipulating people for stock market actions, and blackmailing people to spread their private information by the power of these accounts.

Consequently, social bot detection is of great importance to keep people safe from these harmful effects. In this study, we approach the social bot detection on Twitter as a supervised classification problem and use machine learning algorithms after extensive data preprocessing and feature extraction operations. Large numbers of features are extracted by analysis of Twitter user accounts for posted tweets, profile information and temporal behaviors. In order to obtain labeled data, we use accounts that are suspended by Twitter with the assumption that majority of these are social bot accounts. Our results demonstrate that our framework can distinguish between bot and normal accounts with reasonable accuracy.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	01
2.	PROBLEM DEFINITION	02
3.	LITERATURE SURVEY	03
	3.1 Provably Correct Peephole Optimizations with Alive	
	3.1.1 Introduction	
	3.1.2 Characteristics and Implementation	
	3.1.3 Features	
	3.1.4 Evaluation	
	3.2 Automatic Generation of Code Analysis Tools: The CastQL Approach	
	3.2.1 Introduction	
	3.2.2 Components	
4.	CUSTOMER REQUIREMENTS SPECIFICATION	
	4.1 Introduction	
	4.1.1 Scope	
	4.2 Product Perspective	
	4.2.1 User Characteristics	
	4.2.2 General Assumptions, Constraints and Dependencies	
	4.2.3 Risks	
	4.3 System Architecture	
	4.4 Requirements List	
	4.4.1 Module/Scenario 1	
	4.4.2 Module/Scenario 2	
	4.4.3 Module/Scenario 3	
	4.4.4 Module/Scenario 4	
	4.4.5 Module/Scenario 5	
	4.5 External Interface Requirements	
	4.5.1 Hardware Requirements	
	4.5.2. Software Requirements	
	4.5.3 Communication Interfaces	
	4.6 User Interfaces	
	4.7 Performance Requirements	

4.7.1 Static Requirements

4.7.2 Dynamic Requirements

4.8 Special Characteristics

4.9 Help

4.10 Other Requirements

4.10.1 Site Adaptation Requirements

4.10.2 Safety Requirements

4.11 Packaging

5. HIGH LEVEL DESIGN

5.1 Introduction

5.1.1 Overview

5.1.2 Purpose

5.1.3 Scope

5.2 Design Description

5.2.1 Master Class Diagram

5.2.2 Module 1 – Hardcoded Video

5.2.2.1 Description

5.2.2.2 Use Case Diagram

5.2.2.3 Class Diagram

5.2.2.3.1 Class Description

5.2.2.3.2 Sequence Description

5.2.3 Module 2 – News Video

5.2.3.1 Description

5.2.3.2 Use Case Diagram

5.2.3.3 Class Diagram

5.3 ER Diagrams

5.4 User Interface Diagrams

5.5 Help

5.6 Alternative Design Approach

6. LOW LEVEL DESIGN

6.1 Introduction

6.1.1 Overview

6.1.2 Purpose

- 6.1.3 Scope
- 6.2 Design Description
 - 6.2.1 Subtitle Extraction and Image Inpainting
 - 6.2.1.1 Frame Generator
 - 6.2.1.1.1 Description
 - 6.2.1.1.2 Data Members 1
 - 6.2.1.1.2.1 readVideo()
 - 6.2.1.1.2.2 generateFrames()
 - 6.2.1.1.2.3 writeFrames()
 - 6.2.1.2 Text Localizer and Detector
 - 6.2.1.2.1 Description
 - 6.2.1.2.2 Data Members 2
 - 6.2.1.2.2.1 checkFrames()
 - 6.2.1.2.2.2 readFrame()
 - 6.2.1.2.2.3 openSubtitle()
 - 6.2.1.2.2.4 imageEnhancement()
 - 6.2.1.2.2.5 detectTextBlocks()
 - 6.2.1.2.2.6 writeSubtitle()
 - 6.2.1.2.2.7 fillTextBlock()
 - 6.2.1.2.2.8 writeFrames()
 - 6.2.1.3 Inpainter
 - 6.2.1.3.1 Description
 - 6.2.1.3.2 Data Members 1
 - 6.2.1.3.2.1 checkFrames()
 - 6.2.1.3.2.2 readBinaryImage()
 - 6.2.1.3.2.3 readLocalizedImage()
 - 6.2.1.3.2.4 convertBinarytoGray()
 - 6.2.1.3.2.5 Inpaint()
 - 6.2.1.3.2.6 writeFrames()
 - 6.2.1.4 Video Generator
 - 6.2.1.4.1 Description
 - 6.2.1.4.2 Data Members 1
 - 6.2.1.4.2.1 readVideo()
 - 6.2.1.4.2.2
 - 6.2.1.4.2.3 checkFrames()

6.2.1.4.2.4 appendFrames()

6.2.1.4.2.5 writeVideo()

6.2.1.5 Add Audio

6.2.1.5.1 Description

6.2.1.5.2 Data Members1

6.2.1.5.2.1 extractAudio()

6.2.1.5.2.2 addAudio()

6.2.2 News Video Subtitle Extraction and Inpainting

6.2.2.1 News Video

6.2.2.1.1 Description

6.2.2.1.2 Data Members 1

6.2.2.1.2.1 readVideo()

6.2.2.1.2.2 readVideo2()

6.2.2.1.2.3 readVideo2()

6.2.2.1.2.4 openSubtitle()

6.2.2.1.2.5 generateFrames()

6.2.2.1.2.6 writeSubtitle()

6.2.2.1.2.7 resizeFrame()

6.2.2.1.2.8 overlayFrame()

6.2.2.1.2.9 stackHorizontal()

6.2.2.1.2.10 extractAudio()

6.2.2.1.2.11 addAudio()

7. **TEST STRATEGY**

7.1 Test Strategies

7.2 Performance Criteria

7.3 Test Environment

7.4 Risk Identification and Contingency Planning

7.5 Roles and Responsibilities

7.6 Test Schedule

7.7 Test Tools used

7.8 Acceptance Criteria

7.9 Test Case List

7.10 Test Data

8. IMPLEMENTATION AND PSEUDOCODE

8.1 RunParallelWindows ()

8.2 VideoToFrames ()

8.3 LocalizationDetection ()

8.4 Inpainting()

8.5 FramesToVideo()

8.6 AudioExtractorWindows()

8.6.1 AudioExtractorUbuntu ()

8.7 AudioAdderWindows ()

8.7.1 AudioAdderUbuntu ()

8.8 NewsVideo ()

8.9 GUI

9. TESTING

10. RESULTS AND DISCUSSION

11. SNAPSHOTS

12. CONCLUSIONS

13. FURTHER ENHANCEMENTS

REFERENCES/BIBLIOGRAPHY

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

APPENDIX B USER MANUAL (OPTIONAL)

LIST OF TABLES

Table No.	Title	Page No.
4.1	Processing times of videos	
4.2	Module 1	
4.3	Module 2	
4.4	Module 3	
4.5	Module 4	
4.6	Module 5	
4.7	Software Version and Source	
5.1	Use Case Item and Description	
5.2	ER Model	
6.1	Frame Generator	
6.2	Text Localizer and Detector	
6.3	Inpainter	
6.4	Video Generator	
6.5	Add Audio	
6.6	News Video	
7.1	Risk Identification and Contingency Plan	
7.2	Risk and Action	
7.3	Test Case List	

LIST OF FIGURES

Figure No.	Title	Page No.
1.1	Overlapping Subtitles – Use Case	
4.1	Architecture Diagram	
5.1	Master Class Diagram	
5.2	Use Case Diagram	
5.3	Class Diagram	
5.4	Sequence Diagram	
5.5	Use Case Diagram – Module 2	
5.6	Class Diagram – Module 2	
5.7	User Interface Diagram	
6.1	Frame Generator	
8.1	Parallel Execution of Code - Windows	
8.2	Parallel Execution of Code - Ubuntu	
8.3	Conversion of Video to Frames	
8.4	Text Localization and Detection	
8.5	Inpainting	
8.6	Conversion of Frames to Video	
8.7	Audio Extractor - Windows	
8.8	Audio Extractor - Ubuntu	
8.9	Audio-Adder - Windows	
8.10	Audio-Adder - Ubuntu	
8.11	News Video	
8.12	GUI	
11.1	Frames Generated from Video	
11.2	Individual Frame	
11.3	Localized Frames	
11.4	Individual Localized Frame	
11.5	Masked Frames	
11.6	Individual Masked Frame	
11.7	Subtitle.txt	
11.8	Inpainted Frames	

