



MUHAMMAD TEHAAM

01-134172-071

AHMED GHANI

01-134172-006

Criminal Threat Observing System (CTOS)

Bachelor of Science in Computer Science

Supervisor: Ma'am Maryam Bibi

Department of Computer Science
Bahria University, Islamabad

April 2021

Certificate

We accept the work contained in the report titled “Criminal Threat Observing System (CTOS)”, written by Mr. Muhammad Tehaam AND Mr. Ahmed Ghani as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Ms. Maryam Bibi ()

Internal Examiner: Name of the Internal Examiner ()

External Examiner: Name

Project Coordinator: Ms. Zubaria Inayat ()

Head of the Department: Dr. Muhammad Muzammal ()

April 30th, 2021

Abstract

Security in general has been improving with time, but time and again we have seen that our systems have failed. The recent motorway case was a hit for our entire nation where a helpless woman was harmed unimaginably, and the perpetrators are still on the loose and no one even has any legitimate evidence to find the culprits. This is one of the many cases that unfortunately have been happening on a daily basis and not only that but have been increasing in numbers. The project proposed, is designed to fill those holes in our currently used systems, that the criminals have been happily getting by to fulfil their aims.

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful. We would like to express our deepest thank and gratitude towards Ma'am Maryam Bibi who gave us the chance to work on this project, her supervision and her guidance proved very successful in making this project successful so far. She provided us with the reference material and many other project related materials that not only helped us making the project without any considerable trouble but also implementing it till so far.

MUHAMMAD TEHAAM, AHMED GHANI
Bahria University Islamabad, Pakistan

April 2021

Contents

Abstract	iii
1 Introduction	1
1.1 Project Background/Overview	1
1.2 Problem Description	2
1.3 Project Objectives	2
1.4 Project Scope	2
2 Literature Review	3
2.1 Image Processing based Gun Detection	3
2.2 Deep Learning based Gun Detection	5
2.2.1 Convolutional Neural Network	5
2.2.2 VGG16	6
2.2.3 Single Shot Detector	7
2.2.4 YOLO	7
2.2.5 Haar Cascade Classifier	8
2.2.6 Transfer Learning	8
2.3 Machine Learning vs Deep Learning	9
2.4 Common Practices	10
2.5 Comparison of the Discussed Techniques	11
3 Requirement Specification	15
3.1 Proposed System	15
3.2 Methodology	16
3.3 Non-Functional Requirements	17
3.4 Use Cases	18
3.4.1 Use Case 1: Start Desktop Application	18
3.4.2 Use Case 2: Select Images/Videos from Inventory	19
3.4.3 Use Case 3: Feed the Image/Video of Interest to the System	19
3.4.4 Use Case 4: Get the Results	20
4 Design	21
4.1 System Architecture	21
4.1.1 System Architecture Diagram:	22
4.1.2 Conceptual Diagram:	22
4.2 Design Constraints	23
4.2.1 Application Constraints	23
4.2.2 Software Requirements	23

4.3	Design Methodology	23
4.3.1	Process-wise Analysis	24
4.4	High Level Design	26
4.4.1	Component Diagram	26
4.4.2	System Interaction Diagram	26
4.4.3	Deployment Diagram	27
4.4.4	Modules	28
4.4.5	Security	28
4.5	Low Level Design	28
4.5.1	Data Acquisition	29
4.5.2	Image Processing	29
4.5.3	Train and Classification	29
4.6	GUI Design	30
5	System Implementation	31
5.1	System Summary	31
5.2	System Architecture	31
5.2.1	Algorithmic Workflow	31
5.3	System Components	32
5.3.1	Convert Video to Frames	33
5.3.2	Pre-processing Frames or Image	33
5.3.3	Displaying the Images or Videos	33
5.3.4	Passing Frames/Images to a Gun Detector Function for Parallel Check	34
5.3.5	Comparing the Results of both Model and the Function	34
5.3.6	Converting the Frames Back to Video (if video)	34
5.3.7	Displaying the Images or Video	35
5.4	Tools and Techniques	35
6	System Testing and Evaluation	37
6.1	Graphical User Interface Testing	37
6.1.1	Test Case	38
6.2	Usability Testing	39
6.3	Software Performance Testing	39
6.4	Exception Handling	40
6.5	Load Testing	40
6.6	Test Cases	40
6.6.1	Software Startup Test Case	40
6.6.2	Video/Image Load Test Case	41
6.6.3	Criminal Detection Test Case	41
6.6.4	Second Criminal Detection Test Case	42
6.7	Datasets	42
6.7.1	Custom Dataset	42
6.8	Detail Analysis and Results	43
6.8.1	Image Dataset	44
6.8.2	Video Dataset	47
6.9	Comparison	49

7 Conclusion	51
7.1 Future Work	51
7.2 Recommendation	52
7.3 Learning Outcomes	52
A User Manual	53
A.1 Start Program	53
A.2 Pick a Video	54
A.3 Processing	55
A.4 Results	55
B Bibliography	57

List of Figures

2.1	Harris plus FREAK descriptor of gun [10]	4
2.2	(a) Referenced image; (b) Segmented image; (c) Extracted gun color part from referenced image [10]	4
2.3	Categorization of the DL methods and their representative works [12]	9
2.4	Block diagram of the proposed approach for classifying image pistol/gun or not [12]	10
2.5	The training process of the proposed system with negative and positive dataset images [12]	10
3.1	Images from Gun Dataset	16
3.2	Images from Not Gun Database	16
3.3	Methodology Diagram	17
3.4	Use Case Diagram	18
3.5	Use Case 1 Diagram	18
3.6	Use Case 2 Diagram	19
3.7	Use Case 3 Diagram	19
3.8	Use Case 4 Diagram	20
4.1	System Architecture	22
4.2	Package Diagram	22
4.3	Methodology Flowchart	24
4.4	Image/Frame Acquisition Flow Diagram	25
4.5	Image Pre-Processing Flow Diagram	25
4.6	Image/Frame Feature Extraction Flow Diagram	25
4.7	Classification Process Flow Diagram	26
4.8	Component Diagram	26
4.9	System Interaction Diagram	27
4.10	Deployment Diagram	28
4.11	Class Diagram for Numpy Module	29
4.12	Class Diagram for OpenCV Module	29
4.13	Class Diagram for Sklearn Module	30
4.14	Graphical User Interface (GUI)	30
5.1	System Activity Diagram	32
5.2	Video Frames	33
5.3	Image Resizing	33
6.1	Main Screen Layout	37
6.2	Browse Window for Videos/Images	38
6.3	Video Processing	38

6.4	Results	39
6.5	(a) Original Image, (b) Software Result	41
6.6	(a) Original Image, (b) Software Result	42
6.7	Gun Images	43
6.8	Not Gun Images	43
6.9	Videos	43
6.10	Model accuracy and loss graphs for Alex-Net	44
6.11	Model accuracy and loss graphs for CNN-6 Layers	45
6.12	Model accuracy and loss graphs for CNN-4 Layers	46
6.13	Model accuracy and loss graphs for CNN-6 Layers with tanh	47
6.14	Model accuracy and loss graphs for Alex-Net	48
6.15	Model accuracy and loss graphs for CNN-6 Layers	48
6.16	Model accuracy and loss graphs for CNN-4 Layers	49
A.1	Main Screen Layout	53
A.2	Browse Window for Images	54
A.3	Browse Window for Videos	54
A.4	Video Processing	55
A.5	Results for Videos	56
A.6	Results for Images	56

List of Tables

2.1	Technique Comparison Table	13
3.1	Use Case 1: Start Desktop Application Table	18
3.2	Use Case 2: Select Images/Videos from Inventory	19
3.3	Use Case 3: Feed the Image/Video of Interest to the System	19
3.4	. Use Case 4: Get the Results	20
6.1	Software Startup Test Case	40
6.2	Video/Image Load Test Case	41
6.3	Criminal Detected Test Case	41
6.4	Second Criminal Detected Test Case	42
6.5	Conventional Alex-Net Architecture with activation function Relu	44
6.6	Three Convolutional Layers and three Max-pooling Layers with activation function Relu	45
6.7	Two Convolutional Layers and two Max-pooling Layers with activation function Relu	46
6.8	Two Convolutional Layers and two Max-pooling Layers with activation function tanh	47
6.9	Conventional Alex-Net Architecture with activation function Relu	47
6.10	Three Convolutional Layers and three Max-pooling Layers with activation function Relu	48
6.11	Two Convolutional Layers and two Max-pooling Layers with activation function Relu	49
6.12	Comparison with Research Papers	49

Acronyms and Abbreviations

CNN	Convolutional Neural Network
R-CNN	Region based Convolutional Neural Network
YOLO	You Look Only Once
HOG	Histogram of Oriented Gradients
SPP-net	Spatial Pyramid Pooling-net
SSD	Single Shot Detector
GUI	Graphical User Interface
VGG	Visual Geometry Group